FUNDAMENTAL THEORIES FOR THE MECHANICS OF POLYMER CHAINS AND NETWORKS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Michael Robert Buche

August 2021

Theoretical and Applied Mechanics
Sibley School of Mechanical and Aerospace Engineering
Cornell University
Ithaca, New York
United States of America

Fundamental Theories for the Mechanics of Polymer Chains and Networks

Michael Robert Buche
Cornell University 2021

Abstract

Elastomeric materials contain many individual polymer chains crosslinked together, forming a polymer network. The rubber-like elasticity of these networks is primarily due to the entropic elasticity of the long polymer chains and weak intermolecular interactions above the glass transition temperature. New, important elastomeric materials are actively researched today, such as those that incorporate bond breaking to enhance mechanical properties. These emerging materials challenge us to develop physically-founded constitutive models to gain predictive power and a more fundamental understanding. This dissertation presents the development of such models, starting from the statistical mechanics of a single chain and ending with the mechanics of the polymer network.

This dissertation begins with an additional contribution: a combined experimental and theoretical study of a metallopolymer and its mechanical properties. Neutral ligands are added to a polymer containing metal-coordination crosslinks, where the ligands bind at the metal centers, altering the crosslinking strength and allowing the macroscale mechanical properties to be tuned. Density functional theory is utilized to quantify the mechanics and thermodynamics of the crosslinks under this addition of ligands, in order to predict and understand observations from experimental mechanical tests. Accurate theoretical predictions are made when varying the number of added ligands, and qualitative conclusions are drawn in the case of varying the ligand type.

Next, the methodical development of a constitutive model for polymer networks without bond breaking is presented. A careful statistical mechanical treatment highlights the important connection between the single-chain mechanical response and the equilibrium distribution of chains in the network, as well as the correspondence between different thermodynamic ensembles. Using an example single-chain model, these effects are directly studied. This statistical theory is then brought into the continuum scale, where the stress is obtained in terms of the applied deformation and molecular parameters, and the effects of statistical correspondences are again studied. The resulting framework serves as a directly physically-linked constitutive model for elastomers and hyperelastic materials in general.

Following this, an asymptotic theory is presented for the statistical thermodynamics of classically-treated systems with strong interaction potentials. This development can be understood as the low-temperature analog of the well-established high-temperature perturbation theory. The asymptotic theory is applicable to approximating the mechanical response of single polymer chains, as well as to molecular modeling in general.

The primary contribution in this dissertation focuses on another meticulous approach applied to a polymer network with bonds that break, whether reversibly or irreversibly. Beginning with the fundamentals of nonequilibrium statistical mechanics, classical transition state theory is extended to account for a continuous distribution of polymer chain extensions. In the process, an important connection is established between mechanically-sensitive reaction rates, mechanical response, and equilibrium distribution. Moving to the macroscale, the second law of thermodynamics is shown to be arbitrarily satisfied and the relation for the stress is obtained, once again in terms of the applied deformation and molecular parameters. With the general framework complete, a single-

chain model is specified: the Morse-FJC model, a freely-jointed chain of Morse bonds, is introduced and asymptotically developed. The single-chain mechanical response and reaction rate functions are studied across a range in parameters. An original exact solution to the integro-partial differential equation governing the evolution of the network is presented. Two specialized versions of the general constitutive model – rate-independent irreversible breaking and transient breaking – are developed and applied to exemplary polymers from the literature. The general model is compared to these two specializations, a parameter study of the general model is performed across a wide range of molecular parameters, and finally the general model is applied to another polymer from the literature. The successes and shortcomings of the model are discussed considering all these results, and motivation for future theoretical development is provided.

Special Committee

Meredith N. Silberstein, Associate Professor of Mechanical and Aerospace Engineering.
Advisor and chair. Concentration: Mechanics of Materials.

Steven H. Strogatz, Jacob Gould Schurman Professor of Applied Mathematics.
Minor member. Concentration: Applied Mathematics.

Alan T. Zehnder, Professor of Mechanical and Aerospace Engineering.
Minor member. Concentration: Solid Mechanics.

Biographical Sketch

Michael R. Buche grew up in Crestwood, Illinois, raised by his parents Mark R. Buche and Laura M. Buche. He has an older brother, Matthew R. Buche, and two younger sisters, Jenna M. Buche and Krysta M. Buche. Michael enjoys rock music, drums, hockey, golf, video games, dogs, daydreaming, learning new things, and spending time with his family. He attended Incarnation Elementary School from kindergarten through eighth grade (2000–2009) before attending Saint Laurence High School (2009–2013). Michael then attended the University of Notre Dame, where he would receive a Bachelor of Science in Mechanical Engineering (2017). This is also where Michael would meet (2014) and later marry (2019) his best friend and beloved, Caitlin Reilly McGarry (Buche). While his choice of major was somewhat due to his interest and capabilities at the time, the choice was primarily motivated by the opportunities it would provide and the foresight of his mother Laura. Coincidentally, Michael developed a genuine passion and talent for engineering mechanics, mathematics, and related subjects. This occurred sometime after completing major coursework (thermodynamics, differential equations, solid/fluid mechanics) and engaging in undergraduate research (thermomechanics of soft matter) under Professors Tengfei Luo and Mihir Sen. Michael was also interested in both robotics and microelectronics while at Notre Dame, almost always working under the advisement of Professor Michael M. Stanisic (robotic football, senior design). Michael wanted to continue learning, challenge himself, and develop better skills for doing research, so he decided to take graduate courses in mechanics, mathematics, and physics. Especially impactful was a mathematics class taught by Professor Hsueh-Chia Chang and a continuum mechanics class taught by Professor Samuel Paolucci. Professor Paolucci received his Ph.D. in Theoretical and Applied Mechanics from Cornell, which is how Michael learned of the field and of one of the few institutions that still housed it. At that point, Michael knew what he wanted to do next and continued to work hard to get there.

Michael entered the field of Theoretical and Applied Mechanics in the fall of 2017, enrolling in the Sibley School of Mechanical and Aerospace Engineering at Cornell University. He quickly connected with his advisor Professor Meredith N. Silberstein, who offered him a great opportunity to develop physically-founded constitutive models for exciting polymer materials. Meredith provided Michael with outstanding advisement, training, and professional development, ensuring that Michael was successful while also allowing him to have a high degree of independence. The major research accomplishments of Michael, working with Professor Silberstein and collaborators, are contained in this dissertation. Michael acted as a teaching assistant for several semesters and won the the H.D. Block Teaching Prize twice from the Sibley School. He continued to take courses at Cornell, such as those covering topics in mechanics and mathematics taught by his committee members (Professors Meredith N. Silberstein, Steven H. Strogatz, and Alan T. Zehnder). A particularly vital course covered statistical mechanics and was taught by Professor Roger F. Loring; this

course greatly benefited Michael in his research and cemented his interest in the subject. Some of Michael's primary interests – continuum mechanics, constitutive theory, statistical mechanics, and mathematics – should be quite evident throughout this dissertation. Michael received a Master of Science from Cornell in August 2020 and will receive a Doctor of Philosophy in August 2021; his major concentration is Mechanics of Materials, his minor concentrations are Applied Mathematics and Solid Mechanics.

During their time in Ithaca, Michael and Caitlin started building their life together and made many great friends. In 2018, they brought home an adorable Chocolate Labrador Retriever named Charlie, who continues to brighten each and every day. On July 1st, 2021, Michael and Caitlin welcomed their firstborn son into the world, Maxwell James Buche, whom they love dearly. Max has brought his parents immense happiness and they are eternally grateful for his safe arrival. Michael and his family are planning to move to Albuquerque, New Mexico, where he is working as a postdoctoral appointee for Sandia National Laboratories. He originally became interested in Sandia back in 2019, after attending a recruitment presentation given by Dr. Scott J. Grutzik, another Theoretical and Applied Mechanics Ph.D. from Cornell. Michael now works with Dr. Grutzik and Dr. John M. Emery (also Cornell Ph.D.) in Materials and Failure Modeling at Sandia, managed by Dr. H. Eliot Fang. Michael is the second in his family to work for a Department of Energy National Laboratory. His grandfather, Maurice A. Broderick (grandmother, Jonella C. Broderick), worked at Argonne National Laboratory for many years. Michael is excited to continue doing mechanics research, and to continue learning what God's plan is for him and his family.

Acknowledgements

No man is an island, especially me. I have been quite fortunate to be surrounded by great family, friends, and teachers throughout my life. A special thank you to my parents, Mark and Laura, for loving me, teaching me to work hard, pushing me to do well, and supporting me in all my endeavors. Thank you to my brother Matthew for blazing the trail ahead of me for many years, and being a role model for Jenna, Krysta, and I. Thank you to my dog Charlie for making me happy each day, and for keeping me company over this past year as I worked from home. An extra special thank you goes to my wife and best friend, Caitlin, who has loved and supported me through my graduate career, even as I spent most days working. I love you to the moon and back, I treasure each day of our life together, and I look forward to raising a strong family with you. Thank you to my newborn son Maxwell for bringing us pure joy, and for providing me with plenty of motivation over the past few months. And most of all, I thank God for everything that He has given to me in life.

Though I started and completed this dissertation at Cornell, the instruction and support I received beforehand largely enabled my success. I thank all of the wonderful teachers I had at Incarnation Elementary School and Saint Laurence High School, especially: Thomas Lizik, Katharine Holich, Brian Vollan, Brother Timothy Smyth, William Gaichas, William McLean, Dennis Freyer, John Kroc, Patrick Henning, and Frank Loulousis. Thank you to all the professors that educated and inspired me at the University of Notre Dame, in particular: Hsueh-Chia Chang, David B. Go, J. William Goodwine, Tengfei Luo, Samuel Paolucci, Joseph M. Powers, Mihir Sen, and Michael M. Stanisic. I also thank my rector Rev. John E. Conley, C.S.C, and mentor Scott C. Malpass, as well as the Evans Scholars Foundation, the Chicago Blackhawk Alumni Association, and the William J. Cook Scholarship Fund. Additionally, I thank my friends for putting up with me while I endlessly engaged in schoolwork, especially Matthew T. Coyle, Thomas Salzburg Jr., and Zachary N. Zubic.

At Cornell, I have met many great people who have helped me along the way to completing this dissertation. I thank each member of Professor Silberstein's group, especially my friends Steven J. Yang and Xinyue (Joy) Zhang. Thank you to Dr. Yuval Vidavsky for working closely with me on what became my first published paper, and teaching me a great deal about chemistry in the process. I also want to thank our collaborators, Professor Robert A. DiStasio Jr. and Zachary M. Sparrow, who helped me learn how to perform quantum chemistry calculations. Thank you to Prathamesh D. Raiter, Professor Kyle M. Lancaster, and Professor Peter T. Wolczanski for help on the same work. I thank my Special Committee members – Professors Meredith N. Silberstein, Steven H. Strogatz, and Alan T. Zehnder – for providing me with essential guidance and feedback leading up to this point. And thank you to each professor that I have learned from and worked with at Cornell, especially Professors Roger F. Loring and Andy Ruina.

I want to express my gratitude to my advisor and committee chair, Professor Meredith N. Silberstein. Thank you to Meredith for providing me with outstanding advisement and training

in my research, and for improving my writing and presentation skills. Meredith permitted me to work through most things on my own but kept up with me at each step, challenging my ideas and results while helping me improve them. This has allowed me to become an independent researcher while ensuring that I am successful in completing a strong dissertation, and it prepared for my post-graduate career. I have learned quite a bit from Meredith, and I am truly grateful to have been one of her students.

I also want to thank the sources of funding and computational resources that helped make this dissertation possible. This research was funded in part by the National Science Foundation under Grant CAREER-1653059, and in part by the Office of Naval Research under Grant N00014-17-1-2989 under the administration of PO Dr. Armistead. This research made use of the Cornell Center for Materials Research Shared Facilities, which are supported through the NSF MRSEC program (DMR1719875). Calculations in this work used the Extreme Science and Engineering Discovery Environment (XSEDE) `Stampede2` cluster, which is supported by National Science Foundation Grant ACI-1548562. Calculations were also performed at the Department of Defense (DoD) Supercomputing Resources Centers (DSRCs) through the High Performance Computing Modernization Program (HPCMP). Lastly, I would like to acknowledge a few of my favorite textbooks: Mathematical Methods in Engineering by Joseph M. Powers and Mihir Sen, Continuum Mechanics and Thermodynamics of Matter by Samuel Paolucci, Statistical Mechanics by Donald A. McQuarrie, and Nonequilibrium Statistical Mechanics by Robert W. Zwanzig. These books have influenced me and were indispensable to me in my dissertation research.

To my wife, Caitlin, and our son, Maxwell; to my parents, Mark and Laura, and my siblings, Matthew, Jenna, and Krysta; to my friends and family; and to God, who has given me everything.

Ithaca, USA Michael Robert Buche

Contents

Title i

Copyright ii

Abstract iii

Biographical Sketch v

Acknowledgements vii

List of Figures xi

List of Tables xv

1 Introduction 1
- 1.1 Mechanics of polymer chains . . . 2
- 1.2 Mechanics of polymer networks . . . 2
- 1.3 Dissertation overview . . . 3

2 Tuning the Mechanical Properties of Metallopolymers via... 7
- 2.1 Introduction . . . 8
- 2.2 Results and Discussion . . . 9
- 2.3 Conclusion . . . 18

3 Statistical mechanical constitutive theory of polymer networks... 21
- 3.1 Introduction . . . 22
- 3.2 General theory . . . 24
- 3.3 Macroscopic Results . . . 34
- 3.4 Conclusion . . . 38

4 Low-temperature statistical thermodynamics by an asymptotic method 39
- 4.1 Introduction . . . 40
- 4.2 Theoretical development . . . 40
- 4.3 Applications . . . 44
- 4.4 Conclusions . . . 46

5 Chain breaking in the statistical mechanical constitutive theory... 47
5.1 Introduction . . . 48
5.2 General theory . . . 50
5.3 Single-chain model specification . . . 61
5.4 Macroscopic results . . . 69
5.5 Conclusion . . . 80

6 Conclusions and future outlook 83

A Supporting Information for Chapter 2 87
A.1 General . . . 87
A.2 Polymer synthesis . . . 87
A.3 Dog-bone mold preparation . . . 89
A.4 Tensile Test Methods . . . 89
A.5 Glass Transition Temperature Determination . . . 90
A.6 Theoretical backbone stiffness . . . 91
A.7 Cyclic stress-strain response compared to monotonic loading . . . 92
A.8 Computational details . . . 92
A.9 Dynamic mechanical analysis for Ni-nIm materials . . . 96
A.10 Infrared spectra and color of all materials . . . 97
A.11 Additional mechanical data . . . 98

B Supporting Information for Chapter 3 101
B.1 Solution for the network distribution . . . 101
B.2 Retrieving the stress . . . 102
B.3 Reduction to Neo-Hookean model . . . 105
B.4 Infinitesimal deformation . . . 105
B.5 `Python` package . . . 106

C Supporting Information for Chapter 5 115
C.1 Simplification under transition state theory . . . 115
C.2 Extended derivations for the macroscopic theory . . . 116
C.3 Extended details on implementing the uFJC model . . . 121
C.4 Solving for the distribution of intact chains . . . 126
C.5 Additional generalizations and extensions . . . 135
C.6 `Python` package . . . 139
C.7 Specialized `Python` package . . . 170

References 189

List of Figures

2.1 Depiction of the metallopolymer as ligands are introduced and bind to the crosslinks. 7

2.2 Effects of nickel crosslinking on the (a) IR spectrum and (b) uniaxial monotonic stress-strain response. 10

2.3 Effects of adding neutral imidazole ligands. (a) Theoretically optimized model systems for the nickel carboxylate crosslinking structure in the absence (**Model[Ni]**) and presence (**Model[Im$_2$]**) of two imidazole ligands (with dotted purple lines indicating hydrogen bonds). (b) Starting with the optimized structure, equal and opposite tensile forces were applied to the OAc groups in these model systems until one of the nickel-carboxylate bonds ruptures. (c) Theoretical mechanical responses of the **Model[Ni]** and **Model[Im$_2$]** systems using the EFEI approach. (d) Experimental monotonic stress-strain response in the absence (**Ni**) and presence (**Ni-2Im**) of 2 equiv of imidazole. (e) Experimental cyclic stress-strain response of the **Ni** and **Ni-2Im** materials. 11

2.4 Effects of varying the number of imidazole ligands. (a) Theoretically computed equilibrium population of the **Model[Ni]** and **Model[Im$_n$]**(for n = 1, 2, 3, 4) systems as a function of the added number of imidazole equivalents. (b) Theoretical mechanical responses of the **Model[Ni]** and **Model[Im$_n$]** systems using the EFEI approach. (c) Experimental monotonic stress-strain response of the **Ni**, **Ni-nIm**, and **Linear** materials. (d) Initial modulus and yield stress of these materials. (e) Experimental stress relaxation response of these materials. (f) Highlight of the experimental stress relaxation response of **Ni-2Im**, **Ni-3Im**, **Ni-4Im**, and **Linear** (zoomed in compared to (e)). 14

2.5 Theoretically optimized model systems for the nickel-carboxylate crosslinking structure in the presence of two methylimidazole (**Model[MeIm$_2$]**), pyridine (**Model[Py$_2$]**), piperidine (**Model[Pipe$_2$]**), and dimethylamine (**Model[DMA$_2$]**) ligands (with dotted purple lines indicating hydrogen bonds). 16

2.6 Effects of varying the types of ligands. (a) Theoretical mechanical responses of the **Model[Ni]**, **Model[Im$_2$]**, **Model[MeIm$_2$]**, **Model[Py$_2$]**, **Model[Pipe$_2$]**, and **Model[DMA$_2$]** systems using the EFEI approach. (b) Experimental monotonic stress-strain response of the **Ni**, **Ni-2Im**, **Ni-2MeIm**, **Ni-2Py**, **Ni-2Pipe**, **Ni-2DMA**, and **Linear** materials. (c) Initial modulus and yield stress of these materials. (d,e) Experimental cyclic stress-strain response of these materials. (f) Percentage of recovered energy per cycle in the cyclic stress-strain experiments. 17

3.1 Diagram describing the exact (Helmholtz) and approximate (Gibbs-Legendre) methods of arriving at the single chain Helmholtz free energy $\psi^*(\boldsymbol{\xi})$ of a chain with end-to-end vector $\boldsymbol{\xi}$ and equilibrium probability distribution $P^{\text{eq}}(\boldsymbol{\xi})$ of chains with that end-to-end vector. † The Gibbs-Legendre method is approximate since the necessary Legendre transformation is only valid in the thermodynamic limit of long chains. . 24

3.2 Non-dimensional free energy per link versus end-to-end stretch for $N_b = 3$, 5, 10, and 25, for (a) $\kappa = 50$ and (b) $\kappa = 5$. The free energy is plotted using both the Helmholtz and Gibbs-Legendre methods, as well as using the ideal model valid for $\gamma \ll 1$; the Gibbs-Legendre and ideal results are independent of N_b. 29

3.3 Non-dimensional equilibrium radial distribution function versus end-to-end stretch for $N_b = 3$, 5, 10, and 25, for (a) $\kappa = 50$ and (b) $\kappa = 5$. The distribution is plotted using the Helmholtz and Gibbs-Legendre methods as well as using the Gaussian distribution valid for $N_b \to \infty$. 30

3.4 Non-dimensional uniaxial stress-stretch results for the EFJC network with (a) $\kappa = 50$ and (b) $\kappa = 5$, for $N_b = 5$, 10, and 25. This mechanical response is plotted using the true method (Helmholtz) and two approximation methods (Gibbs-Legendre, Gibbs-Legendre-Gaussian). Shading indicates equal N_b value. The Neo-Hookean response is included as reference. 35

3.5 Non-dimensional (a) equibiaxial and (b) simple shear stress-stretch results for the EFJC network with $\kappa = 50$ for $N_b = 5$, 10, and 25. This mechanical response is plotted using the true method (Helmholtz) and two approximation methods (Gibbs-Legendre, Gibbs-Legendre-Gaussian). Shading indicates equal N_b value. The Neo-Hookean response is included as reference. 37

5.1 Illustration of an intact chain (A) with various links of instantaneous length ℓ_i that may act as reaction coordinates. In this illustration, the links may break irreversibly (see B_1) or reversibly (see B_2) when $\ell_i > \ell_i^{\ddagger}$. 51

5.2 Diagram describing the utilization of the general theory developed in Sec. 5.2. After specifying the single-chain model Hamiltonian $H(\Gamma)$, the temperature T, the deformation gradient $\mathbf{F}(t)$, and the total number density of chains n, all quantities of interest may be systematically computed, ultimately resulting in the stress $\boldsymbol{\sigma}(t)$. . 62

5.3 Single-chain mechanical response when using the Morse-FJC model. (a) The nondimensional force $\eta = \beta f \ell_b$ as a function of the nondimensional end-to-end length $\gamma = \xi/N_b\ell_b$ for $\beta u_b = 25$ and varying κ. (b) η as a function of γ for $\kappa = 200$ and varying βu_b. 64

5.4 The reaction rate coefficient function $k'(\gamma)$ scaled by its initial value $k'(0)$ as a function of the nondimensional end-to-end length $\gamma = \xi/N_b\ell_b$, using the Morse-FJC model for $\beta u_b = 25$, and varying κ. 66

5.5 Single-chain functions for the inhomogeneous Morse-FJC model with $N_b = 1$, $N_b^{\#} = 8$, $\kappa = 200$, $\beta u_b = 25$, and $\varsigma = 1$, while varying $\kappa^{\#}$. (a) The nondimensional force $\eta = \beta f \ell_b$ as a function of the nondimensional end-to-end length $\gamma = \xi/\ell_b(N_b + \varsigma N_b^{\#})$. (b) The reaction rate coefficient function $k'(\gamma)$ scaled by its initial value $k'(0)$ as a function of γ. 69

5.6 (a) Mechanical response of the triple ethyl acrylate network ($EA_{0.5}EAEA$) of Ducrot et al. [11] under cyclic uniaxial tension, and that predicted by the rate-independent irreversibly-breaking model. (b) Light emission from the sacrificial mechanoluminescent crosslinkers breaking within the first network in the same experiment, and that predicted by the model. 71

5.7 (a) Mechanical response of the PVA gel in Long et al. [167] under cyclic uniaxial tension at a rate of 0.03/s, with predictions provided by the transient network model and the model adjusted to use a relaxation function. (b) Mechanical response of the PVA gel of Mayumi et al. [46] under monotonic uniaxial tension at varying rates, with predictions provided again by both models. 73

5.8 (a) The applied deformation $F_{11}(t)$ as a function of nondimensional time $\dot{\epsilon}t$, where $\dot{\epsilon}$ is a constant and the stress is uniaxial tension. (b) The total probability that a chain is intact, $P_{\mathrm{A}}^{\mathrm{tot}}(t)$, as function of nondimensional time for the general model and the two special cases. (c) The applied nondimensional stress $\beta\sigma_{11}(t)/n$ as a function of nondimensional time for the same cases. (d) The distribution of intact chains $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)$ aligned with the loading direction (along the γ_1-axis) at the halfway point $t = 8/\dot{\epsilon}$. The nondimensional parameters are $N_b = 1$, $N_b^{\#} = 8$, $\kappa = 200$, $\kappa^{\#} = 500$, $\varsigma = 1$, $\beta u_b = 100$, $\beta\Delta\Psi_0 = 5$, and $k_0/\dot{\epsilon} = 1/100$, with $\gamma_c = \gamma_{\max} = 1.146$ for the special cases. 75

5.9 Parametric studies concerning rates and the unbreakable links, where we vary (a,b) $k_0/\dot{\epsilon}$, the nondimensional initial reaction rate, (c,d) $\kappa^{\#}$, the nondimensional stiffness of the unbreakable links, and (e,f) $N_b^{\#}$, the number of unbreakable links, while keeping the number of breakable links, $N_b = 1$, constant. For one cycle of uniaxial monotonic tension, the nondimensional stress, $\beta\sigma_{11}(t)/n$, and total probability that a chain is intact, $P_{\mathrm{A}}^{\mathrm{tot}}(t)$, are plotted as a function of the nondimensional time $\dot{\epsilon}t$. 77

5.10 Parametric studies concerning the breakable link(s), where we vary (a,b) βu_b, the nondimensional breakable link energy, (c,d) $\beta\Delta\Psi_0$, the net Helmholtz free energy change when breaking a chain, and (e,f) N_b, the number of breakable links, while keeping the total number of links, $N_b + N_b^{\#} = 9$, constant. For one cycle of uniaxial monotonic tension, the nondimensional stress, $\beta\sigma_{11}(t)/n$, and total probability that a chain is intact, $P_{\mathrm{A}}^{\mathrm{tot}}(t)$, are plotted as a function of the nondimensional time $\dot{\epsilon}t$. . 78

5.11 Stress as a function of stretch for one cycle of uniaxial monotonic tension, repeated for different maximum stretches. The experimental results for the metal-coordinated gel of Lin et al. [165] are shown with those predicted by the general model. 81

A.1 Polymerization setup; dog-bone silicone mold pressed between two laminated glass plates. 88

A.2 Dog-bone dimensions in mm. 89

A.3 Storage and loss modulus (left; a,c) and $\tan\delta$ (right; b,d) for **Linear** (top; a,b) and **Ni** (bottom; c,d). 90

A.4 Theoretically (MD) obtained mechanical response in the average length of backbone between crosslinks from which the initial stiffnesses is approximately 0.4 nN/Å and the maximum tangent stiffness is approximately 2.4 nN/Å. 91

A.5 Cyclic and monotonic response for (a) **Ni** and (b) **Ni-2Im**. 92

A.6 Illustration of the procedure used to calculate the theoretical binding free energies for the consecutive addition of $n = 1$, 2, 3, 4 imidazole ligands. 94
A.7 Storage and loss modulus (left; a,c,e,g) and $\tan\delta$ (right; b,d,f,h) for **Ni-1Im** (a,b), **Ni-2Im** (c,d), **Ni-3Im** (e,f), and **Ni-4Im** (g,h). 96
A.8 IR spectra of polymers with (a) different numbers of Im equivalents per Ni center and (b) different types of ligands with 2 equivalents per Ni center. 97
A.9 (a) Nickel carboxylate crosslinkers in solution with increasing amounts of imidazole, and (b) differences in color for solutions of Ni-2Pipe and Ni-2DMA (sp^3 hybridized nitrogen; group II) vs. Ni- 2Py, Ni-Im and Ni-2MeIm (sp^2 hybridized nitrogen; group I). All solutions were prepared according to the general procedure for neutral ligand attachment (see Sec. A.2.2). 97
A.10 Monotonic response at a rate of 0.023/s (top; a,b) and 0.57/s (bottom; c,d) when varying the number (left; a,c) and type (right; b,d) of ligands in the material. The **Ni-2DMA** material was not tested at these rates. 98
A.11 Stress relaxation response when varying the type of ligands in the material (a) and zoomed in (b). 99
A.12 Cyclic response for **Ni-2MeIm** and **Ni-2Py** over many cycles, showing remarkable similarity. 99
A.13 Cyclic response for the (a) **Linear**, (b) **Ni-1Im**, (c) **Ni-3Im**, and (d) **Ni-4Im** materials. 100

C.1 (a) The nondimensional force $\eta = \beta f \ell_b$ as a function of the leading order (solid) and first-order corrected (dashed) asymptotic approximations of the corresponding nondimensional end-to-end length $\gamma = \xi/N_b\ell_b$ using the Morse-FJC model for $\beta u_b = 25$ and varying κ. (b) The maximum relative error $\max[e(\eta)]$ for $\beta u_b = 25$ and varying κ. 123
C.2 (a) Engineering stress-strain results from the experiments and the theory. (b) Molecular release percentage as a function of the maximum compressive stress from the experiments and the theory. Plots from [199]. 139
C.3 (a) Molecular release percentage on a sample after first compression to 0.1 MPa, 90° rotation, and second compression to 0.1 MPa. (b) Theoretical molecular release percentage after first compression, 90° rotation, and second compression, for three different stress values. Plots from [199]. 140
C.4 Plots of single-chain functions generated by the `plot_single_chain` attribute of the `plotter` object. (a) the nondimensional single-chain mechanical response $\eta(\gamma)$, (b) the nondimensional equilibrium distribution of intact chains $\mathscr{P}_{\mathrm{A}}(\gamma)$, (c) the nondimensional equilibrium radial distribution of intact chains $4\pi\gamma^2\mathscr{P}_{\mathrm{A}}(\gamma)$, and (d) the net forward reaction rate coefficient function $k(\gamma)$, each as a function of the nondimensional end-to-end length γ. 163
C.5 The applied deformation F as a function of time t from `plotter.plot_deformation`. 164
C.6 Plots of the various results generated by the `plotter` object after solving the network model. (a) The nondimensional stress $\beta\sigma/n$ as a function of the deformation F and (b) of the time t, (c) the intact chain fraction $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ as a function of t, and (d) the total breaking (forward), reforming (reverse), and net (forward + reverse) rate of change of $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ as a function of t. 165

List of Tables

2.1 **Theoretically Computed Free Energy Differences (ΔG) and Barriers ($\Delta G^{\ddagger}$) between the Intact and Ruptured Configurations of the Model Systems** 12

A.1 **Theoretical binding free energies (in kcal/mol) for the consecutive addition of a given ligand to the nickel center.** 94

Chapter 1

Introduction

Elastomers are bulk materials consisting of many single polymer chains crosslinked together to form a polymer network. Due to the entropic elasticity of the long polymer chains, weak intermolecular interactions above the glass transition temperature, and the crosslinking of the network, elastomers tend to exhibit rubber-like elasticity [1]. Elastomers are characteristically compliant, elastic, and resilient; the attribute "elastomeric" is typically understood to be synonymous with "rubber-like." Elastomers occur naturally (for example, polyisoprene) and are made synthetically (for example, polybutadiene), finding everyday applications such as in shoes, tires, seals, and adhesives, as well as in cutting-edge technology like biomaterials and soft robotics.

Research on elastomers often focuses on improving mechanical properties and attaining new material behaviors using one or more polymer networks. Here, the focus will specifically be on polymer networks designed to benefit from bond breaking. Taking advantage of the energy dissipation that occurs when breaking bonds, materials may be strengthened, toughened, and made more stretchable through the incorporation of sacrificial networks [2–9]. This bond breaking can also be designed to provide additional functionality, for example mechanoluminescence [10–12] or chain-lengthening [13–16]. A range of reversible bonds may be utilized, allowing polymer chains to break and reform in the network. Reversible breaking permits similar properties as irreversible breaking, like high stretchability and toughness, while simultaneously permitting new properties like self-healing and reproccessability [17–19]. For example, alginate-based gels contain ionic crosslinks that break down as the polymer is deformed but reform later, allowing repeated cycles of high energy dissipation [20]. Metal-ligand interactions with inherent tunability [21] can provide a precise method to control bulk mechanical properties [22–28]. Control of material behavior can also be achieved using dynamic reversible bonds, which can be triggered by an external stimulus. Vitrimers behave as elastic solids at low temperatures but flow as viscous liquids after elevating the temperature, all while maintaining network integrity [29–32]. Some covalent adaptable networks use light or pH as the external stimulus, triggering the bond dynamics in order to control things like the material shape [33–38]. Combinations of reversible and irreversible bonding are often quite useful, such as the mixture of chemical and physical bonds [39–47]. In general, these emerging elastomeric materials are rate-dependent, mechanically-nonlinear, and chemically-sensitive, which then warrants a truly physically-based constitutive model in order to maintain predictive power and fundamental understanding across different materials systems. An approach using statistical mechanics can lead to such a model.

1.1 Mechanics of polymer chains

The mechanical response of a single polymer chain can be described by the end-to-end force as a function of the end-to-end length. At finite temperatures, a nonzero force is required, on average, to increase the end-to-end length of a polymer chain, even in the absence of bond stretching. This thermodynamic force results from the reduction in available configurational entropy as the chain is extended [48], which is quantified using Boltzmann's entropy formula. Equilibrium statistical mechanics, also known as statistical thermodynamics [49], provides the fundamental tools necessary to determine the mechanical response of polymer chains using single-chain models: idealized models that represent real polymers [50]. The most common of these single chain models is the freely-jointed chain (FJC) model [51], where rigid links are connected in series and rotate freely (no potential energy change) about their connecting hinges. In the case of an applied force, the mechanical response of the FJC is given by the Langevin function [1], and in the case of an applied extension, the mechanical response may be written using series expansions [52]. Analogous to the noninteracting gas of classical statistical thermodynamics, the results obtained using different thermodynamic ensembles will generally differ, but should shrink as the number of links in the chain grow [53]. As this number continues to grow, the chains begin to obey Gaussian statistics [54, 55], and for small extensions, the mechanical response of the chain becomes linear, i.e. the ideal chain model [56].

As the freely-jointed chain (or any other single-chain model with rigid links) is extended towards its contour length, the required thermodynamic force becomes infinite as the number of available configurations approaches unity. In reality, highly-extended polymer chains would begin to experience bond stretching at or before this point, so a more physically-realistic single-chain model is required. For example, the FJC model can be generalized to the extensible freely jointed chain (EFJC) model by replacing the rigid links with stiff harmonic springs [57]. Statistical thermodynamics still handles these cases, but now the canonical ensemble and Helmholtz free energy must be utilized in place of the microcanonical ensemble and Boltzmann entropy [49]. Overstretching results in chain breaking from forces becoming large enough rupture bonds, however, the principles of statistical thermodynamics have not yet been properly applied to these cases. While extension-dependent reaction rates have been proposed previously [58–60], a proper and general formulation has yet to be determined, leaving researchers to assume a form, typically inspired by Bell [61]. Upon specification of a single-chain model with breakable bonds, a proper formulation would yield complete information about the chemical kinetics of breaking, in addition to the single-chain mechanical response. Further, since most single-chain models are not analytically tractable like the FJC model, there is a need for a systematic approach to approximate single-chain mechanical responses. The current method [62] works well, but relies on a heuristic technique that is not entirely compatible with the principles of statistical thermodynamics, so a more rigorous method is desirable.

1.2 Mechanics of polymer networks

When constructing a model for the polymer network, the most important characteristic of elastomers to recall is that of weak intermolecular interactions [1]. Ignoring intermolecular interactions completely, single-chain models can be utilized in modeling the network. Next, some distribution of chains must be chosen to represent the arrangement of chains creating the network. For example, a discrete distribution of chains could act as a representative element, such as done with the 3-chain

[52, 63], 4-chain [64, 65], 8-chain [66, 67], and 21 chain [68, 69] models. A continuous distribution of chains provides a more realistic picture, where some models assume that all end-to-end lengths are initially the same and distributed angularly [70, 71], and others additionally consider a distribution of end-to-end lengths [59, 60, 72]. Critically, a clear connection between the distribution of chains creating the network and the underlying single-chain model has not yet been established, though obtaining equilibrium distributions using statistical thermodynamics is well understood [49]. In any case, polydispersity (varying contour lengths) may also be included within discrete [73–75] or continuous [76–78] chain arrangements.

After creating the network model using the single-chain model and an initial distribution, the evolution of the distribution under an applied deformation must be considered. An affine or non-affine deformation of the distribution can be prescribed, where in the latter case, non-affinity can be a fundamental aspect of the initial distribution [66] or based upon some physical constraint [68, 79, 80]. The evolution equations for the network are then completed after including the effects of bond breaking, typically done using transient [60] or irreversible damage [81] models. The Helmholtz free energy of the network is then formulated, and usually the Coleman-Noll procedure [82, 83] is utilized to verify that the second law of thermodynamics is satisfied and obtain a relation for the Cauchy stress, completing the development of the constitutive theory for the mechanics of the network. Many constitutive models for polymer networks have been developed along this mechanistic pathway, and together perform well for a range of materials. However, no modeling approaches have meticulously carried the statistical mechanics of a general single chain model through to the macroscale. Such an approach would inextricably connect (1) the single-chain mechanical response, (2) the initial distribution of chains in the network, and (3) the extension-dependent breaking rates to an arbitrary single-chain model. As a result of these connections, the obtained constitutive relations for the mechanics of the network would be almost entirely determined by the single-chain model. A deeply physical basis such as this provides both a fundamental understanding of the material and a predictive modeling approach.

1.3 Dissertation overview

Immediately following this introduction are four self-contained chapters containing fundamental theories for the mechanics of polymer chains and networks, which address many of the previously discussed gaps in the literature. The final chapter highlights these contributions and discusses relevant future research directions. Three extensive appendices appear afterward, which contain additional experimental data, extended derivations, source code, and example scripts. A list of references appears at the end of the dissertation.

In Chapter 2, a combined experimental and theoretical study of a metallopolymer and its mechanical properties is presented. The metallopolymer in this study is synthesized using nickel cations and carboxylate anion monomers to crosslink an acrylic copolymer. Neutral ligands are introduced to the material to bind at the nickel centers, offering a facile method to tune the strength of the crosslinking and thus the macroscale mechanical properties. As the number and type of ligands added to the material vary, density functional theory is utilized to quantify the mechanics and thermodynamics of the crosslinks. These theoretical findings are then used to predict and understand the mechanical properties of the material observed in experiment. Overall, density functional theory provides a nearly complete understanding when varying the number of added ligands, but provides

limited understanding when varying the ligand type. Important conclusions are drawn concerning the effects that ligand-environment interactions have on the macroscopic mechanical properties, providing a starting point for future theoretical investigation. In Appendix A, supplementary experimental data and theoretical results and details are provided, including unpublished mechanical testing data.

In Chapter 3, a fundamental constitutive theory is developed for the mechanics of polymer networks, beginning from the statistical mechanics of a single polymer chain. This constitutive theory stands out from previous theories, as it permits an arbitrary single-chain model to be utilized while still satisfying statistical correspondences that are discussed. Using the extensible freely-joined chain model as an example, the effects of these correspondences – those between the single-chain mechanical response, the equilibrium distribution of chains in the network, and the thermodynamic ensemble – are highlighted at both the single-chain and macroscopic level. Importantly, the efficacy of two approximations methods in the macroscale are studied, where it is showed that either method matches the exact method as the number of links becomes sufficiently large. In Appendix B, extended derivations for this manuscript are provided. This framework is an effective and highly physical three-parameter (one being the modulus) constitutive model, for elastomers and other hyperelastic materials. An intuitive, efficient, open-source computational implementation (`Python`) of the model is given at the end of Appendix B.

In Chapter 4, a theory is developed to asymptotically approximate the equilibrium statistical mechanics of systems with stiff degrees of freedom, i.e. strong potential energies of interaction. This classical theory is presented as the low-temperature analog of the high-temperature perturbation developed by Robert W. Zwanzig in 1954. Though the motivating application is modeling the stretching of polymer chains, this theory is expected to make a broader impact in molecular modeling.

In Chapter 5, a fundamental constitutive theory is developed for the mechanics of polymer networks that undergo bond breaking, whether reversible or irreversible. Similar to Chapter 3, this theoretical development begins with the statistical mechanics of a single polymer chain and ends with the mechanical response of the polymer network, and the same correspondences are highlighted, now including the mechanically-sensitive reaction rate of breaking chains. A second-law analysis is presented after bringing the statistical theory into the macroscale, where the Cauchy stress is retrieved and it is shown that the residual dissipation inequality is arbitrarily satisfied. With the general theory complete, the uFJC model is developed, which is the freely-jointed chain (FJC) model with stiff, but flexible links of potential energy u. The Morse potential is then implemented and utilized to study the single-chain mechanical response and reaction rate functions across many molecular parameters. Notably, an original exact solution to the integro-partial differential equation governing the evolution of the network is presented, with most of the derivation appearing in Appendix C. Equipped with a complete and exactly-solved framework, Chapter 5 turns to modeling the macroscale stress. After simplifying the model in the two special cases of rate-independent irreversible breaking and transient breaking, the model is applied to exemplary polymers from the literature. While finding the irreversible-breaking specialization quite effective, the results find the transiently-breaking specialization somewhat ineffective, and a supplement utilizing relaxation functions is suggested. Before applying the general, unspecialized version of the model to another material from the literature, it is compared to each specialization and a parameter study is performed, permitting the effects that various molecular parameters have on the mechanical response of the network to be directly understood. In concluding, the successes and shortcomings of this funda-

mental model are discussed, and improvements for future work are proposed. In Appendix C, many extended derivations for this manuscript are provided, including previously unpublished derivations in Appendix C.5. Notably, Appendix C.5.2 extends the general model to cases where chains remain intact after force-induced reaction, and this extended theory is applied to model the mechanics of small molecule release occurring in a recently-synthesized double network gel. Appendix C.6 presents an intuitive, efficient, open-source computational implementation (`Python`) of the general model, which includes a small library of single-chain functions and relaxation functions, options to consider the two special cases, and many example scripts. Appendix C.7 contains another version of this implementation specialized for the small molecule release application.

Chapter 2

Tuning the Mechanical Properties of Metallopolymers via Ligand Interactions: A Combined Experimental and Theoretical Study

Chapter 2 and Appendix A are adapted from: Y. Vidavsky, M. R. Buche, Z. M. Sparrow, X. Zhang, S. J. Yang, R. A. DiStasio Jr., and M. N. Silberstein. Tuning the mechanical properties of metallopolymers via ligand interactions: A combined experimental and theoretical study. Macromolecules **53**, 6 (2020). Y.V. and M.R.B. contributed equally to this work.

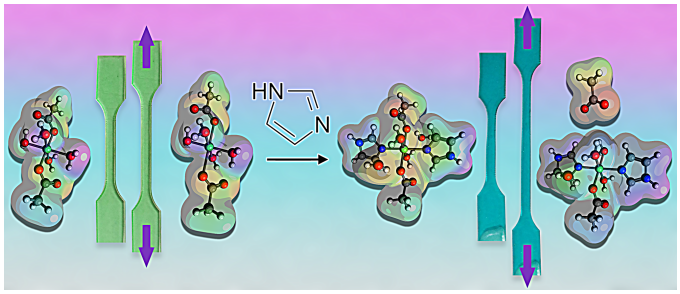


Figure 2.1: Depiction of the metallopolymer as ligands are introduced and bind to the crosslinks.

Metal-ligand interactions provide a means for modulating the mechanical properties of metallopolymers as well as an avenue toward understanding the connection between crosslink interaction strength and macroscale mechanical properties. In this work, we used nickel carboxylate as the tunable crosslinking interaction in a metallopolymer. Different numbers and types of neutral ligands that coordinate to the metal center are introduced as an easy approach to adjust the strength of the ionic interactions in the nickel carboxylate crosslinks, thus allowing macroscale mechanical properties to be tuned. Density functional theory (DFT) calculations, with the external forces explicitly included (EFEI) approach, were used to quantify how the number and type of ligands affect the stiffness, strength, and thermodynamic stability of the nickel carboxylate crosslinking interactions. Interpreting the bulk material properties in the context of these DFT results suggests that the stiffness and strength of the crosslinking interactions primarily control the initial stiffness and yield strength of the metallopolymer, while the mechanical behavior at higher strain is controlled by dynamical bond re-formation and interactions with the polymer environment. The physicochemical insight gained from this work can be used in the rational design of metallopolymers with a wide scope of targeted mechanical properties.

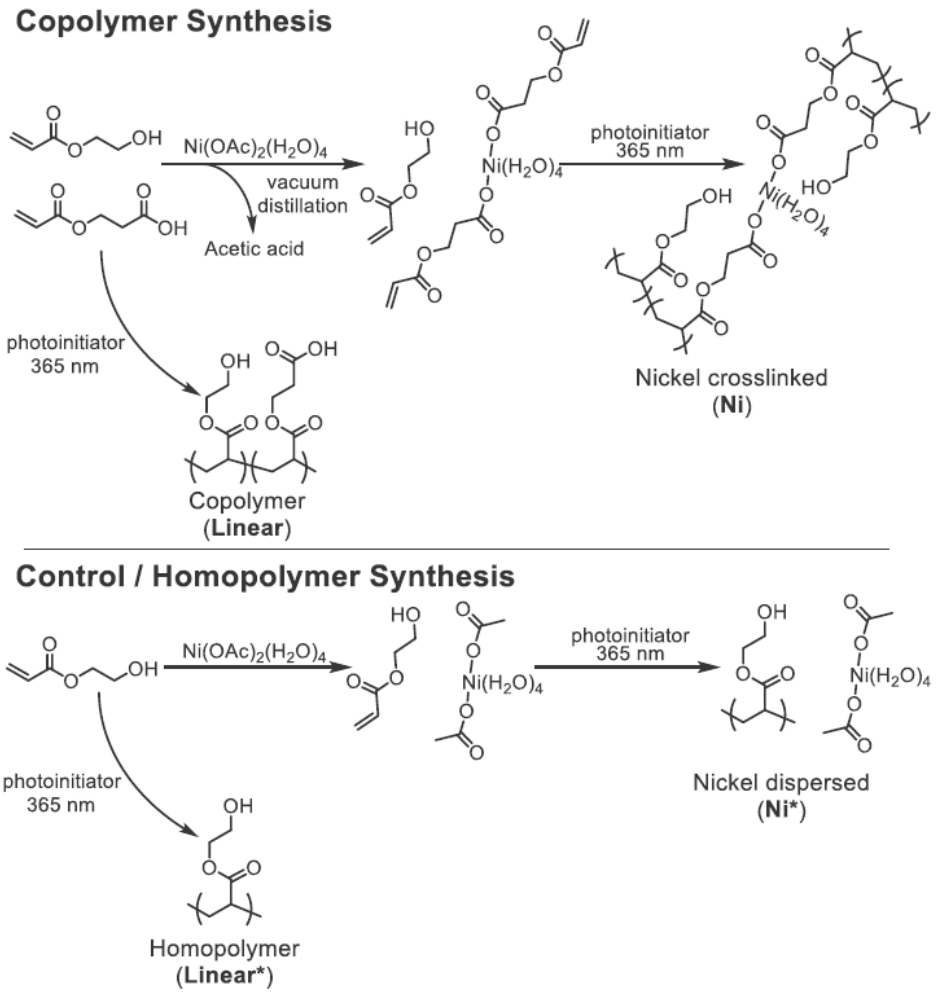
2.1 Introduction

The consolidation of metal cations into a polymer architecture has been demonstrated as an efficient method to modify the mechanical behavior of polymeric materials [23–26]. Organometallic complexes can function as network crosslinkers [24, 84], as part of a linear backbone [27, 85, 86], as grafting points [87], as the driving force for phase segregation [88, 89], and even as intrachain crosslinkers in organic nanoparticles [89–91]. The metal-ligand interactions that form these complexes can vary from weak, dynamic bonds to strong, covalent-like bonds that are effectively static [85]. This range of bond strengths makes the use of these complexes a flexible and robust approach for controlling the mechanical properties of polymers [92]. In addition, these complexes can impart desirable characteristics such as self-healing [18, 93, 94] and stimuli responsiveness [95–97] that are not usually achievable with covalent bonding alone. Several studies have shown that the stability and mechanical strength of these interactions in metallopolymers can be tuned by using either different metal cations or different binding ligands [22–24, 27]. However, tuning the mechanical properties of polymers with these methods often requires new synthetic routes and processing methods. As such, a facile method is needed for modulating the metal-ligand interaction and hence the mechanical properties of polymers.

In addition, an understanding of how the nature of metal-ligand interactions influence macroscale mechanical behavior will allow for more effective materials design to suit specific applications. Metal complexes fall within a broader class of force-sensitive bonded interactions [18, 22, 86, 91, 98–104]. Extensive computational and theoretical work has been conducted over the past 20 years to quantify the stiffness, strength, and force-dependent kinetics of mechanochemically sensitive bonds [14, 105–110]. In this regard, density functional theory (DFT)-based computational approaches such as COGEF (constrained geometry to simulate forces) [105, 111], AISMD (*ab initio* steered molecular dynamics) [112], and EFEI (external forces explicitly included) [107] have been invaluable in the design of materials that incorporate these bonds [113–115].

In this work, we present a simple technique for easily adjusting the strength of ionic crosslinking interactions between Ni^{2+} cations and the carboxylate anions on the backbone of an acrylic polymer. In particular, we study how the attachment of neutral ligands to the crosslinking metal center weakens the ionic interaction and alters the macroscale mechanical properties. By combining DFT calculations with bulk experiments, we provide a fundamental understanding of how changes in the number and type of ligands can be used to tune the mechanical properties of the bulk polymer. Theoretical analysis enables us to identify which features of the bulk mechanical behavior can be tuned by modifying the strength and stiffness of the crosslinks and which features can be tuned from changing the ligand-environment interactions. These findings have broad implications for the design of polymers with dynamic crosslinks and the possibility for secondary interactions to dominate aspects of the mechanical response. In doing so, we provide a facile and rational design procedure that uses ligands to create a versatile class of metallopolymers with highly tunable bulk mechanical properties.

Scheme 2.1: (top) Copolymer Synthesis with (Ni) and without (Linear) Nickel Crosslinking; (bottom) Control/Homopolymer Synthesis with (Ni*) and without (Linear*) Dispersed Nickel



2.2 Results and Discussion

2.2.1 Nickel Carboxylate Crosslinked Network Synthesis and Characterization

All polymers were synthesized by 365 nm UV-initiated free radical copolymerization of the acrylic formulation in a dog-bone-shaped silicone mold pressed between two laminated glass plates (Figs. S1 and S2). For the Ni^{2+} crosslinked network (**Ni**), the acrylic formulation was prepared by dissolving nickel(II) acetate tetrahydrate salt ($Ni(OAc)_2(H_2O)_4$) in a mixture of 2-hydroxyethyl acrylate (HEA) and 2-carboxyethyl acrylate (CEA); see Appendix A. HEA serves as a hydrophilic neutral monomer, and the CEA reacts with the Ni^{2+} cations to generate the ionic crosslinker after removal of the acetic acid under vacuum (Scheme 2.1). The quantity of $Ni(OAc)_2(H_2O)_4$ was chosen to provide a 1:2 ratio of Ni^{2+}:CEA. The combination of a fast curing process, a hydrophilic monomer composition that is compatible with the nickel cations, and a synthesized polymer with a glass transition temperature (T_g) above ambient temperature ensures homogeneous dispersion

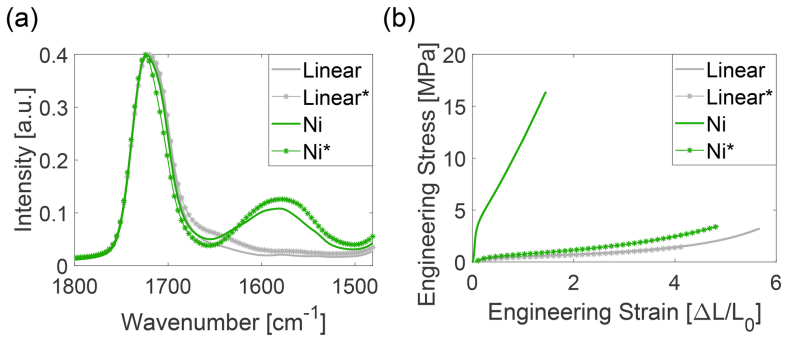


Figure 2.2: Effects of nickel crosslinking on the (a) IR spectrum and (b) uniaxial monotonic stress-strain response.

of the Ni^{2+} throughout the polymer bulk and suppresses potential phase segregation [116]. The linear copolymer (**Linear**) was synthesized under the same conditions without the addition of the nickel salt. In addition, homopolymers of HEA with and without $Ni(OAc)_2(H_2O)_4$ were synthesized as controls (**Ni*** and **Linear*** in Scheme 2.1), since they cannot generate nickel carboxylate crosslinkers.

The successful attachment of nickel cations to the carboxylate crosslinkers was determined by the peak shift of the CQO bond stretching vibration in the IR spectrum. The **Linear** and **Linear*** polymers show only one peak at ~1724 cm^{-1} for the CQO stretching vibration of the backbone ester groups and the free carboxylic acid (Fig. 2.2(a)). For the **Ni** and **Ni*** polymers, an additional peak is present at ~1581 cm^{-1}, corresponding to the nickel carboxylate bonds [117] in these systems.

Nickel crosslinking in the copolymer results in an increase in the T_g from 33.9 to 42.5 °C, as measured by dynamic mechanical analysis (DMA) (Fig. A.3). In contrast to this relatively small T_g change, nickel crosslinking has a dramatic effect on the mechanical properties of the copolymer. Fig. 2.2(b) shows the stress-strain curve of the **Ni** and **Linear** copolymers (as well as the **Ni*** and **Linear*** homopolymers) under uniaxial tensile stress at a strain rate of 0.114/s. Here, we find that the non-crosslinked **Linear** copolymer presents typical behavior for a lightly crosslinked or entangled elastomer. By introducing nickel crosslinking, we find that **Ni** is characterized by a 15-fold increase in stiffness (4.11 to 62.6 MPa) and yield stress (0.213 to 3.03 MPa) as well as a substantial reduction in elongation (from 596% to 156%) when compared to the **Linear** copolymer. This **Ni** behavior is reminiscent of a thermoplastic elastomer. Furthermore, the **Ni*** control homopolymer has only a slightly stiffer response than **Linear***, thereby corroborating our hypothesis that nickel carboxylate crosslinking is responsible for the dramatic changes in the mechanical properties.

2.2.2 Effects of Ligand Addition

We next sought to modulate the bulk mechanical properties of **Ni** by altering the stiffness and strength of the nickel carboxylate crosslinks. Our hypothesis here was that the introduction of ligands that are higher in the spectrochemical series than water – which originally stabilizes the octahedral coordination structure of the nickel – would bind more strongly to the metal centers, displacing some of the bound water molecules into the second coordination sphere. Doing so would increase the electron density on the metal centers and therefore weaken the ionic nickel carboxylate crosslinking interactions. As such, we chose imidazole (Im) to investigate first since it is higher in the spectrochemical series [118] than water and coordinates with Ni^{2+} cations [23, 84, 119].

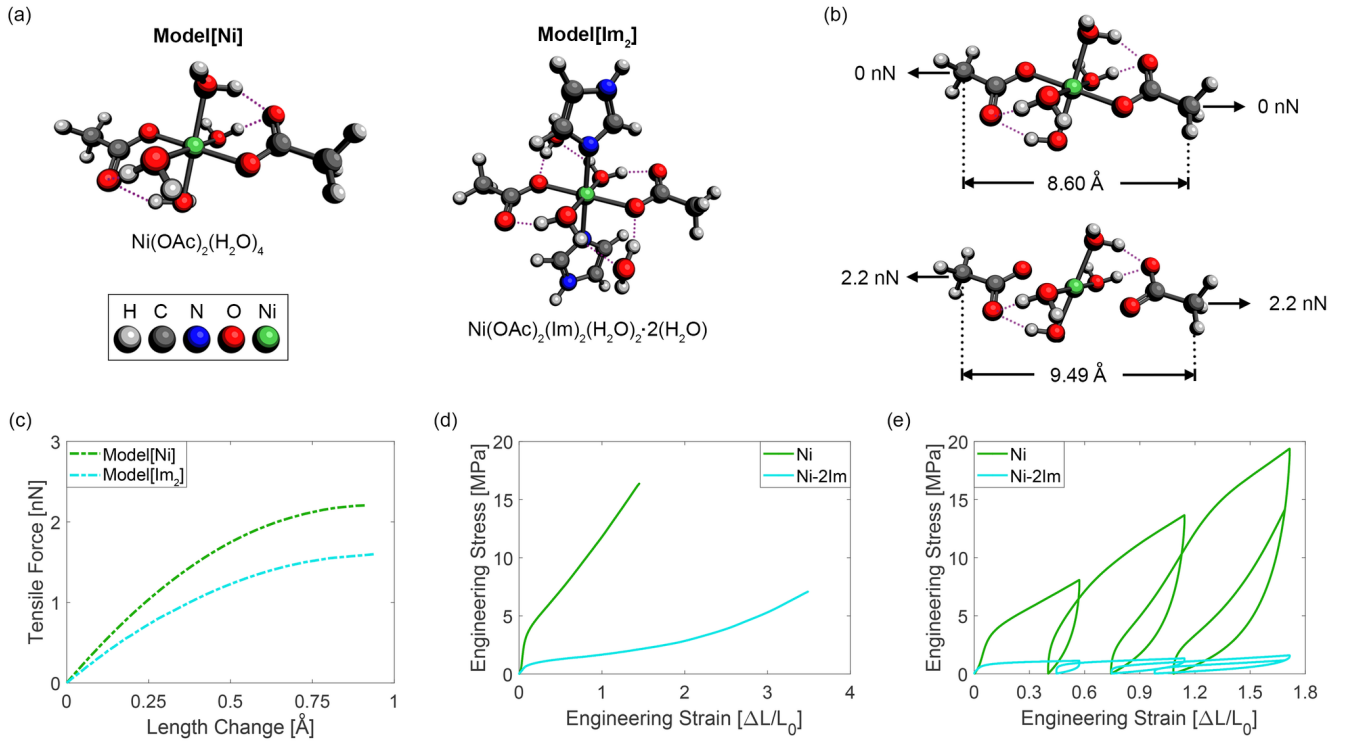


Figure 2.3: Effects of adding neutral imidazole ligands. (a) Theoretically optimized model systems for the nickel carboxylate crosslinking structure in the absence (**Model[Ni]**) and presence (**Model[Im$_2$]**) of two imidazole ligands (with dotted purple lines indicating hydrogen bonds). (b) Starting with the optimized structure, equal and opposite tensile forces were applied to the OAc groups in these model systems until one of the nickel-carboxylate bonds ruptures. (c) Theoretical mechanical responses of the **Model[Ni]** and **Model[Im$_2$]** systems using the EFEI approach. (d) Experimental monotonic stress-strain response in the absence (**Ni**) and presence (**Ni-2Im**) of 2 equiv of imidazole. (e) Experimental cyclic stress-strain response of the **Ni** and **Ni-2Im** materials.

To investigate how the introduction of imidazole ligands weakens the nickel carboxylate bonds, we utilized density functional theory (DFT) calculations (in conjunction with the external forces explicitly included (EFEI) procedure [107]) to study the structures and energetics associated with stretching the nickel carboxylate crosslinks in the presence of an external force. All EFEI calculations were performed on a series of model systems with molecular formula Ni-$(OAc)_2(H_2O)_{4-n}(Im)_n$, with decoordinated water ligands kept in the second coordination sphere (for $n > 0$). The lowest energy configurations for all of these model systems involved a triplet wave function and an octahedral geometry surrounding the Ni center, with monodentate (κ^1) and axial binding of the two acetate (OAc) ligands. Throughout this work, we will refer to these model systems as **Model[Ni]** (for $n = 0$) and **Model[Im$_n$]** (for $n = 1, 2, 3, 4$). Following geometry optimizations of the initial **Model[Ni]** and **Model[Im$_2$]** structures in the absence of any external forces (see Fig. 2.3(a)), equal and opposite tensile forces (F) of increasing magnitude were applied to the methyl carbon atoms on the OAc moieties, which were taken to represent the attachment points to the polymer (see Fig. 2.3(b)). In the presence of these external forces (which were incrementally increased by $\Delta F = 0.1$ nN), these structures were reoptimized for each increment in accordance with the EFEI procedure [107] to obtain the length change in the nickel carboxylate crosslinking site (taken as the distance between the methyl carbon atoms on the OAc groups). These applied forces were increased until one of the nickel-carboxylate bonds ruptured (see Fig. 2.3(b)). All DFT calcula-

Table 2.1: Theoretically Computed Free Energy Differences (ΔG) and Barriers ($\Delta G^{\ddagger}$) between the Intact and Ruptured Configurations of the Model Systems

System	ΔG (kcal/mol)	$\Delta G^{\ddagger}$ (kcal/mol)
Model[Ni]	5.06	18.13
Model[Im_2]	6.29	15.60
Model[$MeIm_2$]	4.68	15.36
Model[Py_2]	5.21	16.63
Model[$Pipe_2$]	5.95	16.33
Model[DMA_2]	6.51	16.66

tions were performed in `Q-Chem` (`v5.1`) [120] using the dispersion-inclusive range-separated hybrid ω`B97X-V` functional [121] and the `def2-TZVPP` basis set [122] (see Appendix A for more details).

As depicted in the theoretically determined mechanical response curves in Fig. 2.3(c), we find that **Model[Ni]** has a linear initial response with a stiffness of 4.5 nN/Å (taken as the initial slope) and a rupture force of 2.2 nN. With two imidazole ligands strongly bound to the metal center, the initial stiffness and rupture force of **Model[Im_2]** decrease to 3.1 nN/Å and 1.6 nN, respectively. To consider the possible effect that this change in the crosslink stiffness will induce in the bulk polymer, we performed a molecular dynamics (MD) simulation to estimate the mechanical response of the backbone segment between crosslinks (i.e., which is composed of seven repeat units of HEA monomers on average). This polymer segment was found to have an initial stiffness of 0.4 nN/Å and a maximum tangent stiffness of 2.4 nN/Å (see Fig. A.4). Given that the crosslink stiffness is within an order of magnitude of the backbone stiffness, the crosslink will not act as an effectively rigid connection until rupture; hence, we expect that changes in the crosslink stiffness will influence the bulk polymer stiffness. The reduction in the crosslink strength should reduce the bulk polymer yield strength and increase ductility since this crosslink rupture force is below that of a carbon-carbon bond along the backbone.

These model systems were also used to quantify the thermodynamics and kinetics associated with rupturing the nickel-carboxylate crosslinks. In particular, we computed the free energy differences (ΔG) and barriers ($\Delta G^{\ddagger}$) between the intact and ruptured configurations of the **Model[Ni]** and **Model[Im_2]** systems in the absence of external forces (see Table 2.1). To obtain the ruptured configuration for each model system, we performed an additional geometry optimization of the structure produced during the final step of the EFEI procedure (in which one of the nickel-carboxylate bonds ruptures). With force-free ΔG values of 5.06 kcal/mol (**Model[Ni]**) and 6.29 kcal/mol (**Model[Im_2]**), our calculations indicate that intact configurations of the nickel-carboxylate crosslinks are preferred over ruptured configurations at equilibrium (i.e., crosslink rupturing processes are not spontaneous). Furthermore, we also found that the forcefree barriers ($\Delta G^{\ddagger}$) to rupturing these model crosslinks were 18.13 and 15.60 kcal/mol for **Model[Ni]** and **Model[Im_2]**, respectively. This observed reduction in $\Delta G^{\ddagger}$ in **Model[Im_2]** is consistent with our hypothesis that the introduction of free imidazole ligands will weaken the effective nickel-carboxylate bond strength. We also note that these $\Delta G^{\ddagger}$ values are large enough such that crosslink rupturing processes will not occur at room temperature due to thermal fluctuations alone.

Inspired by these findings, we performed a series of mechanical tests on the nickel carboxylate crosslinked polymer (**Ni**) in the presence of 2 equiv of neutral imidazole ligands per cation (**Ni-2Im**). Following the expected trends from the DFT calculations, we observed decreases in the initial modulus (62.56 to 8.36 MPa) and yield stress (3.03 to 0.69 MPa) in going from **Ni** to **Ni-2Im** (Fig. 2.3(d)). Because load is transmitted through the polymer in an indirect manner, it is expected that these bulk property changes are not directly proportional to the crosslink-scale changes in initial stiffness and rupture strength; however, it is a bit surprising that an ∼30% decrease in crosslink stiffness leads to an ∼87% decrease in the bulk polymer modulus. This suggests that there is a mechanical coupling between the crosslink and the polymer backbone, wherein the backbone becomes effectively more compliant as the crosslink becomes less rigid. This coupling was also suggested by the slight decrease in T_g observed upon addition of free imidazole ligands. We interpret the yield stress as primarily resulting from the sudden rupture of a large percentage of the crosslinks and the post-yield hardening as largely resulting from the formation of new crosslinks. Following this interpretation, **Ni-2Im** requires less stress to break the crosslinks and more time to re-form them as compared to **Ni**. Under cyclic loading (Fig. 2.3(e)), both materials exhibit extensive viscoplastic deformation. There is also a substantial loss in the overall strength of **Ni-2Im**, which is visible in the reduced stress of the cyclic stress-strain response as compared to that of monotonic loading (Fig. A.5). This likely results from an overall loss in intact crosslinks and perhaps associated chain relaxation. The **Ni** cyclic stress response shows a dramatic reduction in yield strength upon each reload but then dramatically hardens to approach its monotonic response as strain is increased past the prior loading level. This difference between **Ni-2Im** and **Ni** indicates that the crosslinks re-form more slowly when imidazole ligands are bound to the metal center.

2.2.3 Effects of Varying the Number of Ligands

We next sought to obtain further control over these mechanical properties by tuning the number of ligands that were bound to the metal center. We again chose imidazole as the ligand and synthesized three additional polymers with 1 equiv (**Ni-1Im**), 3 equiv (**Ni-3Im**), and 4 equiv (**Ni-4Im**) of imidazole ligands per nickel. The existence of nickel carboxylate crosslinking interactions in all of the polymers with added imidazole ligands was confirmed by the presence of a CQO stretching vibrational peak at slightly smaller frequency compared to **Ni** [117] (∼1560 cm^{-1} in the IR spectra, Fig. A.8). In addition, changes in the color of the Ni^{2+} containing materials (Fig. A.9) further corroborate the presence of interactions between the imidazole ligands and the metal center. To study the effects of varying the number of imidazole ligands on the stiffness and strength of the nickel carboxylate crosslinks, we first needed to investigate how many imidazole ligands were attached to the metal centers in the **Ni-1Im**, **Ni-2Im**, **Ni-3Im**, and **Ni-4Im** cases.

To proceed, we used DFT to compute binding free energies ($\Delta G_{\text{bind}}[n]$) for the consecutive addition of n imidazole ligands to the **Model[Ni]** system (Fig. A.6; see Appendix A for additional details). Quite interestingly, this theoretical analysis showed that binding the first and second imidazole ligands was quite favorable, with $\Delta G_{\text{bind}}[1]$ and $\Delta G_{\text{bind}}[2]$ values of −4.60 kcal/mol and −8.34 kcal/mol, respectively. As indicated by the relative magnitudes $|\Delta G_{\text{bind}}[2]| > |\Delta G_{\text{bind}}[1]|$, binding a second imidazole is cooperative in nature. With $\Delta G_{\text{bind}}[3]$ and $\Delta G_{\text{bind}}[4]$ values of −0.80 kcal/mol and 0.53 kcal/mol, we find that it is significantly less favorable to bind a third or fourth imidazole. We attribute these findings to a both electronic and steric effects, as additional binding of imidazole ligands increases the electron density on the metal and crowds the complex.

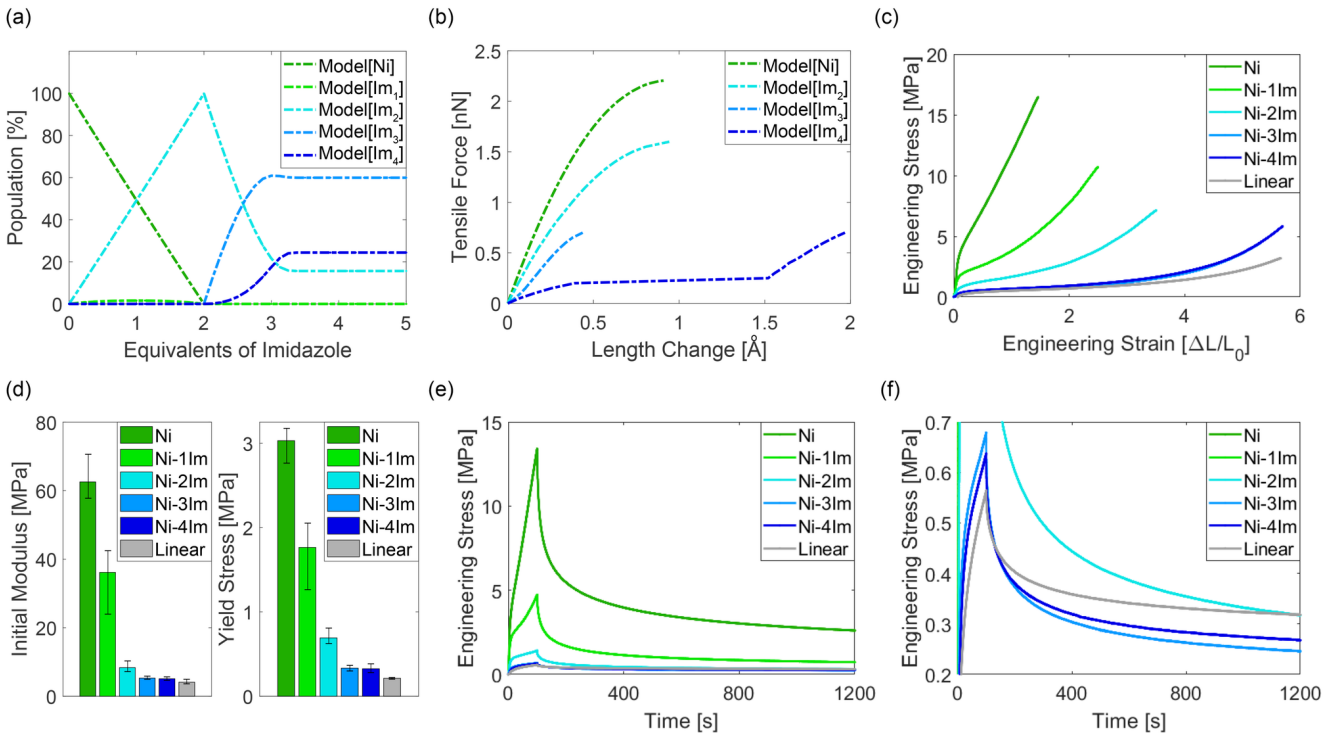


Figure 2.4: Effects of varying the number of imidazole ligands. (a) Theoretically computed equilibrium population of the **Model[Ni]** and **Model[Im$_n$]**(for n = 1, 2, 3, 4) systems as a function of the added number of imidazole equivalents. (b) Theoretical mechanical responses of the **Model[Ni]** and **Model[Im$_n$]** systems using the EFEI approach. (c) Experimental monotonic stress-strain response of the **Ni**, **Ni-nIm**, and **Linear** materials. (d) Initial modulus and yield stress of these materials. (e) Experimental stress relaxation response of these materials. (f) Highlight of the experimental stress relaxation response of **Ni-2Im**, **Ni-3Im**, **Ni-4Im**, and **Linear** (zoomed in compared to (e)).

With these ΔG_{bind} values in hand, we computed the equilibrium populations of **Model[Ni]** and **Model[Im$_n$]** (for n = 1, 2, 3, 4) as a function of the added number of imidazole equivalents (see Appendix A for additional details). As depicted in Fig. 2.4(a), we find that the equilibrium population of **Model[Ni]** rapidly decreases as imidazole is added to the system. This is accompanied by a rapid increase in the population of **Model[Im$_2$]**, while the population of **Model[Im$_1$]** remains at a small and constant (essentially steady-state) value, which is consistent with cooperative binding of the second ligand. Following the addition of 3 equiv, we find that the populations of **Model[Im$_2$]**, **Model[Im$_3$]**, and **Model[Im$_4$]** converge to ∼16%, ∼60%, and ∼24%, respectively, which is consistent with the slightly endergonic binding of the fourth imidazole.

As depicted in the theoretically determined mechanical response curves in Fig. 2.4(b) (obtained by using the EFEI procedure), we find that the initial stiffness and rupture force decrease from 3.1 nN/Å and 1.6 nN (**Model[Im$_2$]**) to 1.2 nN/ Å and 0.7 nN (**Model[Im$_3$]**) and 0.74 nN/Å and 0.7 nN (**Model[Im$_4$]**). We note in passing that the EFEI curve for **Model[Im$_1$]** was not computed since the population of this species was negligible (Fig. 2.4(a)). Interestingly, the EFEI curve for **Model[Im$_4$]** is relatively flat over a large length change (0.5–1.5 Å); we attribute this to a sudden extension of the carboxylate moieties, which were folded inward due to non-bonded interactions. At this point, the response transitions to a more direct pulling on the nickel-carboxylate bonds, wherein **Model[Im$_4$]** is now characterized by a stiffness of ∼1.2 nN/Å, and ultimately ruptures at

0.7 nN (which are the same values as **Model[Im$_3$]**). All these observations are consistent with our hypothesis that binding imidazole ligands increases the electron density on the metal and thereby weakens the ionic nickel carboxylate crosslinks. Consistent with our earlier findings regarding the free energies (ΔG) and barriers ($\Delta G^{\ddagger}$) between the intact and ruptured configurations of **Model[Ni]** and **Model[Im$_2$]**, we also find that intact configurations of **Model[Im$_3$]** and **Model[Im$_4$]** (with force-free ΔG values of 2.12 and 2.11 kcal/mol, respectively) are also preferred over ruptured configurations at room temperature. With a force-free $\Delta G^{\ddagger} = 11.57$ kcal/mol for **Model[Im$_3$]**, we also find that the barrier to crosslink rupturing remains inaccessible at room temperature. We note in passing that the corresponding $\Delta G^{\ddagger}$ for **Model[Im$_4$]** was not computed, as the force-free rupture also involves a multistep mechanism in which the carboxylate groups unfold.

With these theoretical findings in mind, we then performed an additional series of mechanical tests on the **Ni-1Im**, **Ni-3Im**, and **Ni-4Im** polymers. As depicted in Fig. 2.4(c), we find that simply varying the number of imidazole ligands allowed us to tune the monotonic stress-strain behavior between the **Ni** and **Linear** polymers. This tuning was accomplished without significantly altering the T_g of the materials (Fig. A.7). Here, we find that the initial modulus and yield stress both decrease as the number of imidazole equivalents increases from zero (**Ni**) to three (**Ni-3Im**) (Fig. 2.4(d)). With the addition of 4 equiv of imidazole (**Ni-4Im**), these quantities level off but remain slightly above that of the **Linear** polymer. These findings are consistent with the theoretical population and EFEI analyses (Fig. 2.4(a,b)), in that (i) the addition of imidazole ligands weakens the mechanical response of the crosslink structure and (ii) after the addition of 3 equiv these effects are still present but begin to level off (i.e., once the equilibrium populations converge). Experimental stress relaxation results (Fig. 2.4(e,f)) show that although **Ni-3Im** and **Ni-4Im** have a similar peak stress to **Linear**, their relaxation is more substantial as their curves cross over and continue to diverge from the **Linear** polymer. Although **Ni-2Im** starts from a much higher stress, this polymer also crosses over **Linear** during the 20 min experimental time frame. These results support the notion that the nickel carboxylate crosslinks are dynamic in nature, helping to facilitate stress relaxation by breaking and re-forming.

2.2.4 Effects of Varying the Types of Ligands

As an alternative approach to tune the mechanical properties of these polymers, we now investigate the effects of varying the types of ligands attached to the metal center. Here we consider four new nitrogen-based ligands: methylimidazole (MeIm), pyridine (Py), piperidine (Pipe), and dimethylamine (DMA). These ligands were chosen to span options in terms of nitrogen atom hybridization (sp^2 and sp^3), bulkiness, and potential for $\pi \cdots X$ interactions (with X = –H, cation, anion, or π) and hydrogen bonding with the surrounding polymer environment. In each case, 2 equiv of a given ligand was added to form the **Ni-2MeIm**, **Ni-2Py**, **Ni-2Pipe**, and **Ni-2DMA** polymers, and the nickel-carboxylate interactions were again verified by the position of the carboxylate CQO vibrational peak in the IR spectra (Fig. A.8). Binding energy calculations (see Appendix A for additional details) again confirm that the addition of 2 equiv of each ligand to the bulk material corresponds to an equilibrium population that is almost entirely composed of two ligands bound to each nickel center (Table A.1).

DFT calculations were again employed to quantify the mechanics and thermodynamics of the model crosslink structures depicted in Fig. 2.5. The EFEI results predict that adding any two of these nitrogen-based ligands will decrease the initial stiffness and rupture force with respect to

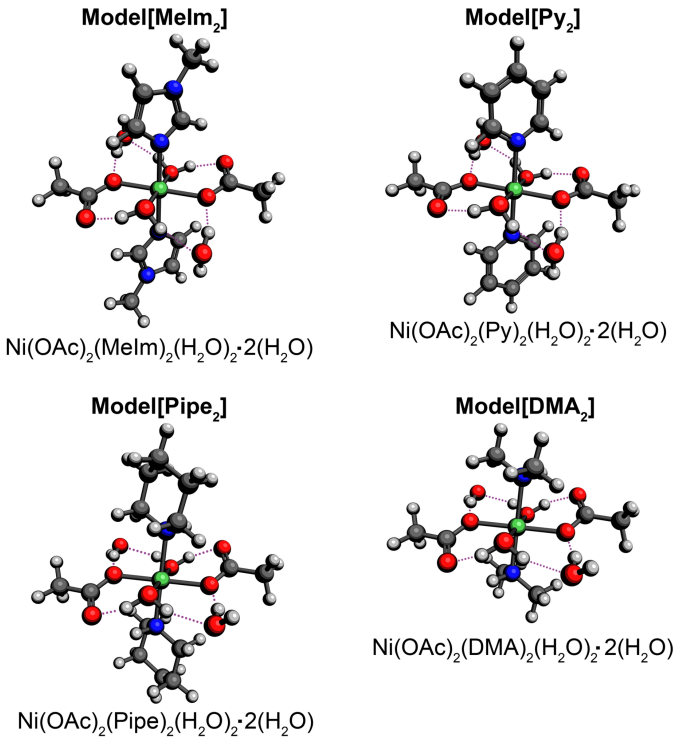


Figure 2.5: Theoretically optimized model systems for the nickel-carboxylate crosslinking structure in the presence of two methylimidazole (**Model[MeIm$_2$]**), pyridine (**Model[Py$_2$]**), piperidine (**Model[Pipe$_2$]**), and dimethylamine (**Model[DMA$_2$]**) ligands (with dotted purple lines indicating hydrogen bonds).

Model[Ni] (Fig. 2.6(a)). The theoretical results also predict that the mechanical properties of the crosslink for this set of ligands are very similar, with the largest observed differences being ∼0.25 nN/Å (initial stiffness) and ∼0.1 nN (rupture force). With ΔG and $\Delta G^{\ddagger}$ values ranging from 4.68 to 6.51 kcal/mol and 15.36 to 16.66 kcal/mol, we again find that intact configurations of these model systems are preferred over ruptured configurations at room temperature, and the barriers to crosslink rupturing always remain thermally inaccessible (Table 2.1). The experimentally obtained monotonic stress- strain responses are presented in Fig. 2.6(b). In the small-strain regime, the addition of any two ligand equivalents results in a dramatic decrease in both the initial modulus and yield stress when compared to **Ni**. However, the differences observed in these properties are rather small among the different ligand types (Fig. 2.6(c)), which is again consistent with the theoretical EFEI predictions on the model crosslinks. In the large strain regime, however, the monotonic stress-strain responses tend to be quite distinct. Since everything except the added ligands was held constant during the preparation and testing of these materials, and the EFEI calculations only considered single/isolated model crosslinks (and predicted little to no mechanical differences), we hypothesize that these large-strain differences are primarily governed by ligand-environment interactions rather than the mechanical response of the crosslinks.

In fact, one can categorize the stress-strain curves in Fig. 2.6(b) into the following two groups: (i) group I polymers (**Ni-2Im**, **Ni-2MeIm**, and **Ni-2Py**), which exhibit lower post-yield hardening and smaller instantaneous moduli (until a rapid increase near fracture), and (ii) group II polymers (**Ni-2Pipe** and **Ni-2DMA**), which exhibit higher post-yield hardening and retain near-constant instantaneous moduli until fracture. In this categorization, it is clear that the group I polymers

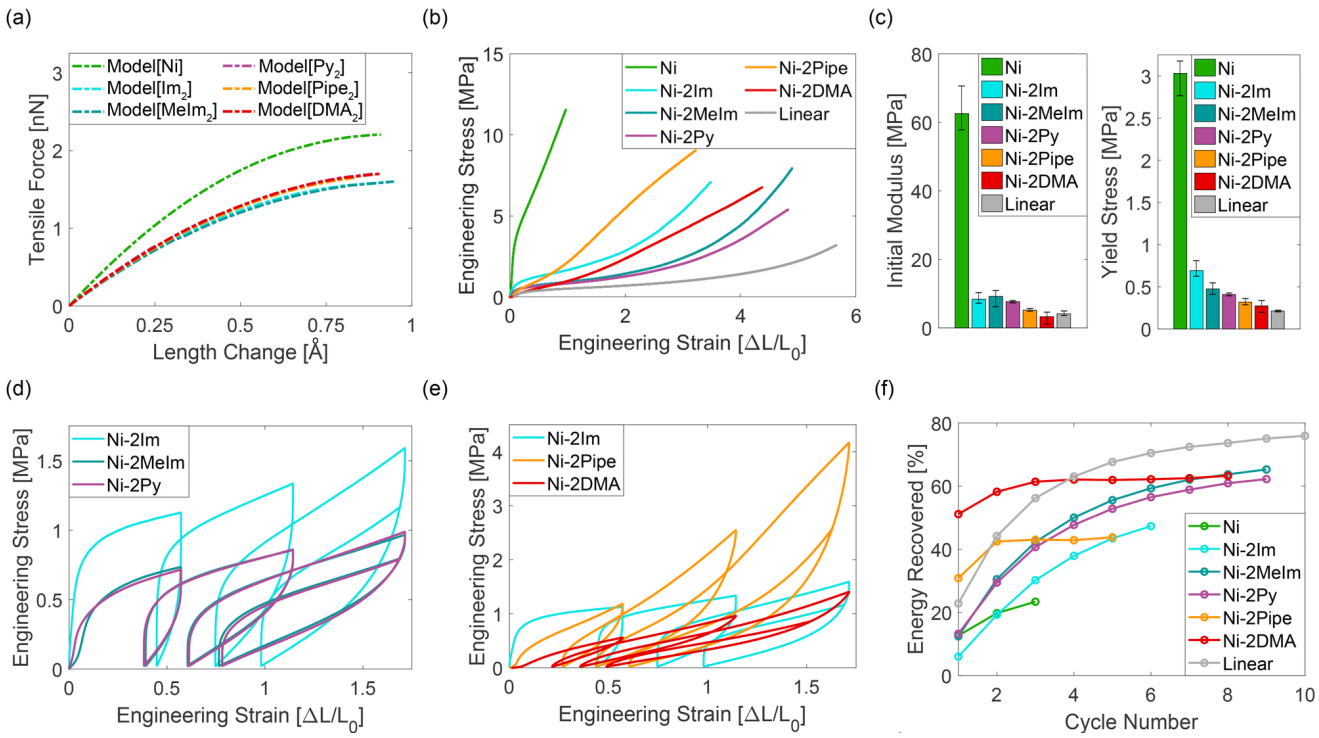


Figure 2.6: Effects of varying the types of ligands. (a) Theoretical mechanical responses of the **Model[Ni]**, **Model[Im$_2$]**, **Model[MeIm$_2$]**, **Model[Py$_2$]**, **Model[Pipe$_2$]**, and **Model[DMA$_2$]** systems using the EFEI approach. (b) Experimental monotonic stress-strain response of the **Ni**, **Ni-2Im**, **Ni-2MeIm**, **Ni-2Py**, **Ni-2Pipe**, **Ni-2DMA**, and **Linear** materials. (c) Initial modulus and yield stress of these materials. (d,e) Experimental cyclic stress-strain response of these materials. (f) Percentage of recovered energy per cycle in the cyclic stress-strain experiments.

contain aromatic ligands with sp^2-hybridized (metal-bound) nitrogen atoms, while the group II polymers contain nonaromatic ligands with sp^3-hybridized nitrogens. Moreover, differences in color among the materials made from these ligands (Fig. A.9) show a clear categorization into two distinct groups in accordance with the sp^2 vs sp^3 hybridization of the metal- bound nitrogen atom on each ligand. Because the theoretical EFEI predictions (Fig. 2.6(a)) show negligible differences in the crosslink mechanical response between ligand types, the hybridization of the (metal-bound) nitrogen does not seem to be directly responsible for the observed differences in the bulk mechanical behavior. However, we can correlate the observed experimental differences in Fig. 2.6(b) to the distinct nonbonded interactions that can occur between the ligands in each group and the surrounding polymer environment. In other words, the group I polymers contain aromatic ligands which are capable of forming favorable dispersion (or van der Waals) interactions, $\pi \cdots$ X interactions (with X = –H, cation, anion, or π), and hydrogen bonds (in the case of **Ni-2Im** only) with the environment, while the nonaromatic ligands in the group II polymers can only form dispersion interactions with the polymer environment. Unlike dispersion interactions (which are radial and nondirectional), the $\pi \cdots$ X and hydrogen bond interactions present in the group I polymers are directional and can constrain the diffusion of broken crosslinks throughout the polymer environment. As a result, the rate of dynamic crosslink re-formation in the group I polymers could be substantially lower than that in the group II polymers. This conjecture is consistent with the monotonic stress-strain curves in Fig. 2.6(b), in which the group I polymers (following the rupture of many crosslinks at yield) are

characterized by a lower post-yield hardening (due to the relatively slower crosslink re-formation). In contrast, the more rapid crosslink reformation in the group II polymers manifests as higher postyield hardening and larger instantaneous moduli until fracture.

This explanation for the large-strain monotonic features is also consistent with the observed bulk mechanical responses to cyclic loading. Here, we find that the group I polymers recover much less than the group II polymers (Fig. 2.6(d,e)), as the residual strain (i.e., at zero stress) after each cycle is much smaller for group II than group I. We further investigated this point by computing the percentage of recovered mechanical energy per cycle based on the cyclic stress-strain responses (Fig. 2.6(f)). From these results, we find that the group II polymers are characterized by a nearly constant energy recovery, which is consistent with breaking and re-forming a similar number of crosslinks during each cycle. On the other hand, we find that the group I polymers are characterized by an initially low recovery percentage that increases rapidly and begins to level off after $\sim$5 cycles. This is also consistent with a slower rate of crosslink re-formation among the group I polymers, in which the energy recovered during each cycle continues to change until most crosslinks are broken. These group I polymers essentially transition from a viscoplastic response to an elastomeric one as the crosslinks break, with polymer chains that are effectively linear dominating the response.

Ligand-environment interactions can also be used to rationalize the observed differences within each polymer group. For instance, the mechanical response of **Ni-2Im** is noticeably different from **Ni-2MeIm** and **Ni-2Py**. More specifically, **Ni-2Im** has a higher yield stress than **Ni-2MeIm** and **Ni-2Py**, and maintains this higher stress throughout the strain hardening regime. In this case, the imidazole ligand in **Ni-2Im** can form hydrogen bonds with the environment (via the non-metal-bound nitrogen atom) [123], while the methylimidazole and pyridine ligands in **Ni-2MeIm** and **Ni-2Py** cannot. Furthermore, the similarities between the **Ni-2MeIm** and **Ni-2Py** polymers are especially apparent in the cyclic responses, as their curves overlap to within experimental error. Considering the group II polymers, we find that **Ni-2Pipe** has a greater strain hardening slope than **Ni-2DMA**. We attribute this greater mechanical strength of **Ni-2Pipe** to the increased dispersion interactions between the larger (and significantly more polarizable) piperidine ligand and the surrounding polymer environment.

2.3 Conclusion

In this work, we have demonstrated a facile procedure to incorporate metal-ligand crosslinking in UV-curable acrylic polymers. By attaching a series of coordinating neutral ligands to the metal center, we were able to finely tune the mechanical properties of the polymer. The simplicity of this approach has enabled a systematic investigation of the effects of changing the number and type of coordinating ligands on the crosslink without otherwise modifying the polymer. DFT calculations were used to first determine the number of ligands that would attach as a function of the total ligands available per metal center and then to investigate the mechanical and thermodynamic characteristics of the equilibrated metal coordination structures. These calculations predicted that the strength and stiffness of the crosslink strongly depend on the number of nitrogen-based ligands, but not on the type. Experimentally, these predictions were borne out in the bulk mechanical behavior at small strain. However, in the strain hardening regime, large differences were observed among the materials with different ligands, suggesting that ligand-environment interactions must also be considered in the rational design process of these versatile and highly tunable polymers.

Acknowledgments

Y.V. and M.R.B. contributed equally to this work. This research was funded in part by the National Science Foundation under Grant CMMI-1653059 and in part by the Office of Naval Research under Grant N00014-17-1-2989 under the administration of PO Dr. Armistead. Z.M.S. and R.A.D. acknowledge start-up funding from Cornell University. This research made use of the Cornell Center for Materials Research Shared Facilities, which are supported through the NSF MRSEC program (DMR1719875). Calculations in this work used the Extreme Science and Engineering Discovery Environment (XSEDE) [124], which is supported by National Science Foundation Grant ACI-1548562. Calculations were also performed at the Department of Defense (DoD) Supercomputing Resources Centers (DSRCs) through the High Performance Computing Modernization Program (HPCMP). The authors thank Prathamesh D. Raiter for his help in running the MD simulations needed to compute the backbone stiffness. The authors also thank Kyle M. Lancaster and Peter T. Wolczanski for helpful scientific discussions.

Chapter 3

Statistical mechanical constitutive theory of polymer networks: The inextricable links between distribution, behavior, and ensemble

Chapter 3 and Appendix B are adapted from: M. R. Buche and M. N. Silberstein. Statistical mechanical constitutive theory of polymer networks: The inextricable links between distribution, behavior, and ensemble. Phys. Rev. E **102**, 012501 (2020). The `Python` package implementing the model [126] is available on `GitHub` and `PyPI` (`pip install Buche_Silberstein_model_2020`).

A fundamental theory is presented for the mechanical response of polymer networks undergoing large deformation which seamlessly integrates statistical mechanical principles with macroscopic thermodynamic constitutive theory. Our formulation permits the consideration of arbitrary polymer chain behaviors when interactions among chains may be neglected. This careful treatment highlights the naturally occurring correspondence between single-chain mechanical behavior and the equilibrium distribution of chains in the network, as well as the correspondences between different single-chain thermodynamic ensembles. We demonstrate these important distinctions with the extensible freely jointed chain model. This statistical mechanical theory is then extended to the continuum scale, where we utilize traditional macroscopic constitutive theory to ultimately retrieve the Cauchy stress in terms of the deformation and polymer network statistics. Once again using the extensible freely jointed chain model, we illustrate the importance of the naturally occurring statistical correspondences through their effects on the stress-stretch response of the network. We additionally show that these differences vanish when the number of links in the chain becomes sufficiently large enough, and discuss why certain methods perform better than others before this limit is reached.

3.1 Introduction

Understanding the mechanics of polymer networks is important for improving and predicting the mechanical behaviors of a wide range of polymeric materials, from physically-crosslinked rubbers to mechanochemically-responsive networks. Constitutive models that are grounded in statistical mechanics are especially useful because they allow the direct incorporation of molecular phenomena and thus a fundamental understanding of the material. In order to establish a model with such predictive power, one needs to proceed from the statistical mechanics of single polymer chains all the way to the macroscopic mechanical behavior of the entire material. The meticulous detail we offer here is required to retain generality throughout this process.

Polymer network constitutive models often utilize a single polymer chain statistical mechanical model. The most common of these single chain models is the freely jointed chain (FJC) model [51], where rigid links are connected in series and allowed to rotate about the connecting hinges without change in energy. The mechanical response of the FJC under end-to-end extension is determined by the reduction in configurational entropy, which is then directly connected to the equilibrium probability distribution of end-to-end lengths by Boltzmann's entropy formula [48]. For a large number of links or in the case of an applied force, the mechanical response and probability distribution of end-to-end lengths may be written analytically using the inverse Langevin function [1]. In the case of an applied extension, more sophisticated methods are necessary to obtain the mechanical response and distribution of end-to-end lengths, such as those using series expansions [52] or those that transform between thermodynamic ensembles [53]. When the number of links approaches infinity, the probability distribution of end-to-end lengths obeys Gaussian statistics [54, 55], and for end-to-end length much smaller than the contour length, the mechanics of the chain become that of the ideal, linear chain [56]. When the end-to-end length approaches the contour length, the FJC becomes infinitely stiff due to its inextensibility. The FJC model can be expanded to that of the extensible freely jointed chain (EFJC) by replacing the rigid links with stiff harmonic springs [57]. Now when the end-to-end length approaches the contour length, the EFJC begins to stretch the stiff harmonic links and subsequently achieves end-to-end lengths greater than the original contour length, hence it is extensible. Another popular set of models are the freely rotating chain (FRC) models, where the FJC model is adjusted by fixing all bond angles and only permitting torsional angles to freely rotate [50]. This model cannot be solved analytically and therefore requires careful numerical techniques [127]. For small bond angles, the FRC model becomes the Kratky-Porod (or discrete worm-like chain) model [128], and when the link length additionally becomes small compared to the contour length, the FRC model becomes the continuous worm-like chain (WLC) model. Both the discrete and continuous forms of the WLC model have been expanded to include stiff harmonic springs [129]. Recent single chain constitutive models include covalent bond rupture [62] and mechanochemically-activated bonds [130]. It is apparent from this vast literature that the correspondences between end-to-end length probability distribution, the mechanical behavior, and the applied boundary conditions (thermodynamic ensemble) are of vital significance.

Upon establishing a statistical description by way of single chain mechanics, the model derivation must then proceed to connect the macroscopic deformation of the polymer network to this single chain description. This is typically accomplished by way of constructing the Helmholtz free energy density, prescribing some aspect of the network evolution in terms of the macroscopic deformation, connecting the network evolution back to individual chains, and using 2nd law of thermodynamics analysis. Most often the 2nd law analysis results in a hyperelastic model, which means that the

stress is directly related to the derivative of the free energy density with respect to the deformation gradient. After choosing a single chain model, the construction of the free energy density for the network involves the choice of the distribution of initial chain lengths and orientations in the network. Several models have used discretely-oriented chains to represent the distribution of chains in the network, such as the 3-chain [52, 63], 4-chain [64, 65], 8-chain [66, 67], and 21 chain [68, 69] models. Other models utilize a continuous orientation distribution of chains to represent the network, where some assume that all end-to-end lengths are initially the same [70, 71] and others consider an initial distribution of end-to-end lengths [59, 60, 72]. Polydispersity, i.e. varying contour lengths, may be included in either the discrete [73, 74] or continuous [76, 77] distribution formulations. An affine or non-affine deformation of the distribution can be prescribed, where the non-affinity can be a fundamental aspect of the initial distribution [66] or based upon some physical constraint [68, 79, 80]. Unfortunately, the natural correspondence between the choice of single chain model and the chain length distribution within the network tends to be ignored in these models.

Despite decades of work, there remains a need for a methodical and general statistical mechanics derivation of polymer network mechanics such that the assumptions and their implications are apparent. The approach taken here begins from fundamental statistical mechanics, makes clear all assumptions and places emphasis on the correspondences between the network distribution, single chain mechanics, and the different thermodynamic ensembles. Most preceding constitutive models make no reference to such nuances, and as a result risk making considerable mistakes. Furthermore, there has been no study on the effects that these correspondences have on the macroscopic mechanical response of the network and when they may be ignored. Such an approach will also take great care in stitching this general statistical description into the macroscopic description, performing a detailed 2nd law analysis to retrieve the stress and making sure that all neglected terms are truly negligible. Many preceding constitutive models use considerable assumptions to construct the stress, lose generality by choosing a specific chain model during the 2nd law analysis, and/or neglect terms that could contribute to the stress without proof that they can be neglected.

In this manuscript, we present a constitutive model for polymer networks undergoing finite deformation that is constructed with great detail. We begin in Section 3.2.1 from fundamental statistical mechanics, ensuring that the correspondences between the distribution of chains in the network and the mechanics of single chains are understood and accounted for. We also account for the differences between thermodynamic ensembles, ensuring we utilize the correct ensemble and understand the correspondence relations that allow us to go from one to the other. Prescribing an affine deformation to the network distribution, we extend the statistical theory to the macroscale and perform a detailed 2nd law analysis in order to retrieve the stress in Section 3.2.2. In the process, we perform many mathematical manipulations in order to maintain the generality of the model. This includes the proof that a term produced when integrating by parts is indeed zero for relevant chain models, which has been previously taken for granted. With this detailed framework, in Section 3.3 we are able to study the macroscopic effects of the aforementioned statistical correspondences and show that they are considerable when chains are not sufficiently long. Throughout the manuscript, the EFJC model is used to demonstrate the statistical correspondences within the network and their effects on the macroscopic mechanical response of the network. This framework will prove useful in constructing future constitutive models for more complicated polymer networks.

3.2 General theory

3.2.1 Statistical mechanical description

Here we present a statistical mechanical description of an ensemble of noninteracting polymer chains. The statistical mechanical description naturally provides explicit relationships between the equilibrium distribution of polymer chain end-to-end vectors and the Helmholtz free energy of a polymer chain with a given end-to-end vector; we refer to these as the distribution-behavior correspondence relations. The original thermodynamic ensemble (Helmholtz) for a single chain is parameterized by an end-to-end vector, where a simple Laplace transformation allows parameterization by the end-to-end force in another ensemble (Gibbs). Since we desire results from the Helmholtz ensemble but are often only able to compute the partition function in the Gibbs ensemble, ensemble transformation relations between the two – both exact and in the thermodynamic limit – are provided. When obtaining the single chain free energy function and the equilibrium distribution of end-to-end vectors, we refer to the method that utilizes the exact transformation as the Helmholtz method, and that utilizing the transformation in the thermodynamic limit as the Gibbs-Legendre method; see Fig. 5.2 for a schematic. Next, we illustrate these features of our statistical description using the EFJC model as an example chain model. We complete the statistical theory by formulating the general evolution law for the polymer network distribution of end-to-end vectors.

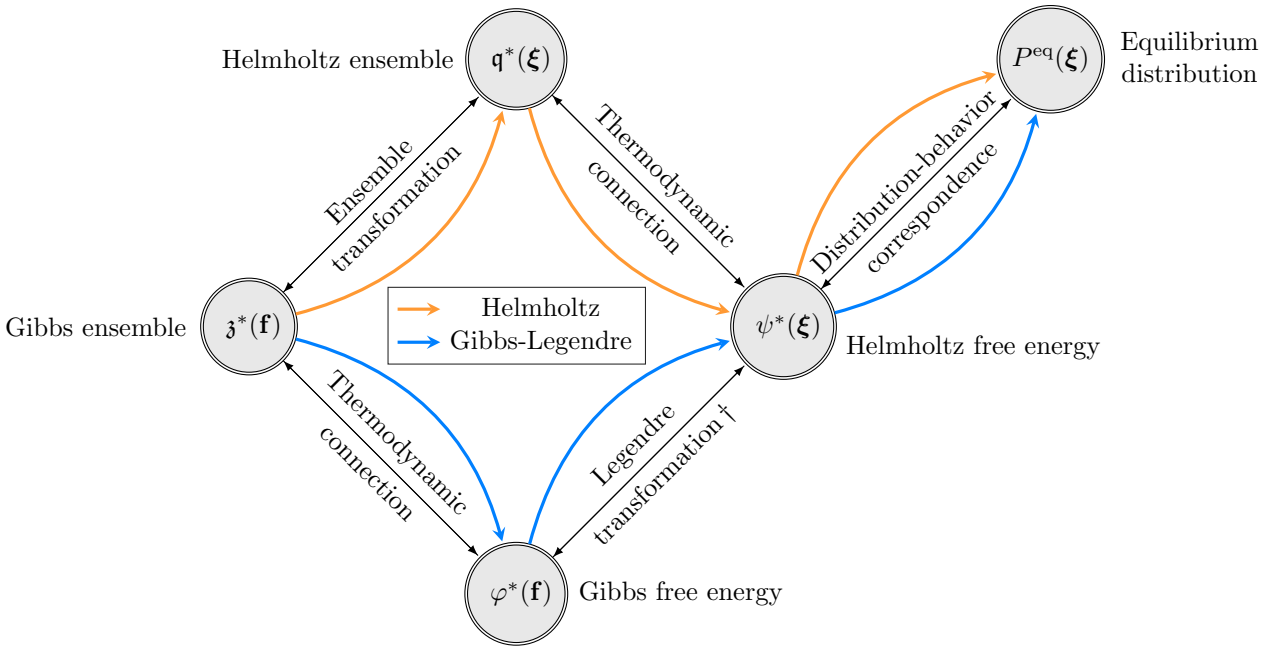


Figure 3.1: Diagram describing the exact (Helmholtz) and approximate (Gibbs-Legendre) methods of arriving at the single chain Helmholtz free energy $\psi^*(\boldsymbol{\xi})$ of a chain with end-to-end vector $\boldsymbol{\xi}$ and equilibrium probability distribution $P^{\mathrm{eq}}(\boldsymbol{\xi})$ of chains with that end-to-end vector. † The Gibbs-Legendre method is approximate since the necessary Legendre transformation is only valid in the thermodynamic limit of long chains.

Helmholtz ensemble

The polymer network is taken to be represented by an ensemble of N indistinguishable noninteracting polymer chains. The canonical partition function is then

$$Q = \frac{\mathfrak{q}^N}{N!}, \tag{3.1}$$

where the single chain partition function $\mathfrak{q}$ is given by a classical integration over the coordinates $\mathbf{q}_j$ and momenta $\mathbf{p}_j$ of each of the M atoms in the chain [49],

$$\mathfrak{q} = \frac{1}{h^{3M}} \int \cdots \int e^{-\beta H} \prod_{j=1}^{M} d^3\mathbf{p}_j \, d^3\mathbf{q}_j. \tag{3.2}$$

Here h is Planck's constant and $\beta = 1/\mathfrak{b}T$ is the inverse temperature, with Boltzmann's constant $\mathfrak{b}$ and temperature T. The Hamiltonian H of the chain is

$$H = U(\mathbf{q}_1, \ldots, \mathbf{q}_M) + \sum_{j=1}^{M} \frac{p_j^2}{2m_j}, \tag{3.3}$$

where U is the potential energy function describing interaction energies between atoms within the polymer chain, and m_j is the mass of j^{th} atom in the chain. The momentum integrations are completed to write a portion of the chain partition function:

$$\mathfrak{q}_{\text{mom}} = \prod_{j=1}^{M} \left(\frac{2\pi m_j \mathfrak{b}T}{h^2} \right)^{3/2}. \tag{3.4}$$

If we take the atomic coordinates relative to the first atom along the chain backbone, $\mathbf{r}_j = \mathbf{q}_j - \mathbf{q}_1$, we can complete the rigid body translation integration – where the chain is translated over the whole volume – and pick up a factor of V. We now have

$$\mathfrak{q} = \mathfrak{q}_{\text{con}} \mathfrak{q}_{\text{mom}} V, \tag{3.5}$$

where the chain configuration integral $\mathfrak{q}_{\text{con}}$ is then

$$\mathfrak{q}_{\text{con}} = \int \cdots \int e^{-\beta U} \prod_{j=2}^{M} d^3\mathbf{r}_j. \tag{3.6}$$

If the M^{th} atom is the last atom along the chain backbone, we seek to calculate the probability density distribution $P^{\text{eq}}(\boldsymbol{\xi})$ that a chain has the end-to-end vector $\mathbf{r}_M = \boldsymbol{\xi}$ at equilibrium. This means that the probability that a chain has the end-to-end vector within $d^3\boldsymbol{\xi}$ of $\boldsymbol{\xi}$ at equilibrium would be $P^{\text{eq}}(\boldsymbol{\xi}) \, d^3\boldsymbol{\xi}$. We then write $\mathfrak{q}^*$, the chain configuration integral corresponding to end-to-end vector $\boldsymbol{\xi}$, by integrating the Dirac delta function (δ)

$$\mathfrak{q}^*(\boldsymbol{\xi}) = \int \cdots \int e^{-\beta U(\mathbf{r}_2,\ldots,\mathbf{r}_M)} \delta^3\left(\mathbf{r}_M - \boldsymbol{\xi}\right) \prod_{j=2}^{M} d^3\mathbf{r}_j, \tag{3.7}$$

$$= \int \cdots \int e^{-\beta U(\mathbf{r}_2,\ldots,\boldsymbol{\xi})} \prod_{j=2}^{M-1} d^3\mathbf{r}_j. \tag{3.8}$$

According to Boltzmann statistics, the probability of a single chain configuration at thermodynamic equilibrium is $e^{-\beta U}/\mathfrak{q}_{\text{con}}$, so we integrate over all configurations that have the end-to-end vector $\boldsymbol{\xi}$ in order to retrieve $P^{\text{eq}}(\boldsymbol{\xi})$,

$$P^{\text{eq}}(\boldsymbol{\xi}) = \int \cdots \int \frac{e^{-\beta U}}{\mathfrak{q}_{\text{con}}} \delta^3\left(\mathbf{r}_M - \boldsymbol{\xi}\right) \prod_{j=2}^{M} d^3\mathbf{r}_j, \tag{3.9}$$

$$= \frac{\mathfrak{q}^*(\boldsymbol{\xi})}{\iiint \mathfrak{q}^*(\tilde{\boldsymbol{\xi}})\, d^3\tilde{\boldsymbol{\xi}}} = \frac{\mathfrak{q}^*(\boldsymbol{\xi})}{\mathfrak{q}_{\text{con}}}, \tag{3.10}$$

where $\tilde{\boldsymbol{\xi}}$ is a dummy variable of integration; the tilde will continue to denote dummy variables of integration. If this equilibrium distribution is rotationally symmetric (only varies with $\xi = \sqrt{\boldsymbol{\xi}\cdot\boldsymbol{\xi}}$), we can use the equilibrium radial distribution function

$$g^{\text{eq}}(\xi) = 4\pi\xi^2 P^{\text{eq}}(\boldsymbol{\xi}). \tag{3.11}$$

The chain Helmholtz free energy ψ^* associated with $\mathfrak{q}^*$ is, from the principal thermodynamic connection formula [49],

$$\psi^*(\boldsymbol{\xi}) = -\mathfrak{b}T \ln \mathfrak{q}^*(\boldsymbol{\xi}), \tag{3.12}$$

so we may finally write the equilibrium distribution as

$$P^{\text{eq}}(\boldsymbol{\xi}) = \frac{e^{-\beta\psi^*(\boldsymbol{\xi})}}{\iiint e^{-\beta\psi^*(\tilde{\boldsymbol{\xi}})}\, d^3\tilde{\boldsymbol{\xi}}}, \tag{3.13}$$

which, if $\psi^*(\boldsymbol{\xi}_{\text{ref}}) = \psi^*_{\text{ref}}$ is known for some $\boldsymbol{\xi}_{\text{ref}}$, we have

$$\psi^*(\boldsymbol{\xi}) = \psi^*_{\text{ref}} - \mathfrak{b}T \ln\left[\frac{P^{\text{eq}}(\boldsymbol{\xi})}{P^{\text{eq}}(\boldsymbol{\xi}_{\text{ref}})}\right]. \tag{3.14}$$

We refer to Eqs. (3.13) and (3.14) as the distribution-behavior correspondence relations, as they show a one-to-one correspondence between the free energy of a chain for a given end-to-end vector and the equilibrium distribution of such end-to-end vectors.

Gibbs ensemble

The Gibbs ensemble releases the end-to-end vector constraint of the Helmholtz ensemble and instead applies an end-to-end force. The Helmholtz ensemble coincides with the canonical ensemble (which

has the Helmholtz free energy as the principal thermodynamic potential), but the Gibbs ensemble does not exactly coincide with the isobaric-isothermal ensemble (which has the Gibbs free energy as the principal thermodynamic potential), despite an applied force seeming to be analogous to an applied pressure. The naming of the Gibbs ensemble is then perhaps a bit misleading, but we will continue to use it since it seems to have become standard. The Gibbs ensemble Hamiltonian is

$$H = U(\mathbf{q}_1, \ldots, \mathbf{q}_M) + \sum_{j=1}^{M} \frac{p_j^2}{2m_j} - \mathbf{f} \cdot (\mathbf{q}_M - \mathbf{q}_1), \tag{3.15}$$

where $\mathbf{f}$ is the force that acts equally and oppositely on atoms 1 and M at the ends of the polymer chain, and $\mathbf{q}_M - \mathbf{q}_1 = \mathbf{r}_M$ is the end-to-end vector of the chain. The system partition function and the momentum partition function take the same form as Eqs. (3.1) and (3.4), respectively, and we receive the same factor of V from the rigid body translation integration, but the chain configuration integral corresponding to force $\mathbf{f}$,

$$\mathfrak{z}^*(\mathbf{f}) = \int \cdots \int e^{-\beta U} e^{\beta \mathbf{f} \cdot \mathbf{r}_M} \prod_{j=2}^{M} d^3\mathbf{r}_j, \tag{3.16}$$

$$= \iiint \mathfrak{q}^*(\boldsymbol{\xi}) e^{\beta \mathbf{f} \cdot \boldsymbol{\xi}} \, d^3\boldsymbol{\xi}, \tag{3.17}$$

is now utilized. The chain configuration integral corresponding to the Gibbs ensemble is directly a Laplace transform of the Helmholtz ensemble chain configuration integral. The probability density distribution that a chain experiences the force $\mathbf{f}$ at equilibrium $P_{\mathfrak{z}}^{\mathrm{eq}}(\mathbf{f})$ is then given by the ratio of $\mathfrak{z}^*(\mathbf{f})$ to the integral of $\mathfrak{z}^*(\mathbf{f})$ over all end-to-end force vectors $\mathbf{f}$. The principal thermodynamic connection formula yields the Gibbs free energy φ^* associated with the force,

$$\varphi^*(\mathbf{f}) = -\mathfrak{b} T \ln \mathfrak{z}^*(\mathbf{f}), \tag{3.18}$$

from which we obtain the Gibbs ensemble distribution-behavior correspondence relations:

$$P_{\mathfrak{z}}^{\mathrm{eq}}(\mathbf{f}) = \frac{e^{-\beta \varphi^*(\mathbf{f})}}{\iiint e^{-\beta \varphi^*(\tilde{\mathbf{f}})} \, d^3\tilde{\mathbf{f}}}, \tag{3.19}$$

$$\varphi^*(\mathbf{f}) = \varphi^*_{\mathrm{ref}} - \mathfrak{b} T \ln \left[\frac{P_{\mathfrak{z}}^{\mathrm{eq}}(\mathbf{f})}{P_{\mathfrak{z}}^{\mathrm{eq}}(\mathbf{f}_{\mathrm{ref}})} \right]. \tag{3.20}$$

Ensemble transformations

It has been demonstrated that the mechanical response of a given polymer chain model can differ appreciably between the two ensembles if the thermodynamic limit (i.e. chains consisting of sufficiently many links) is not satisfied [53]. This is an issue because while traditional macroscopic constitutive theories require the Helmholtz free energy of the system, single polymer chain partition functions are often only solvable, if at all, in the Gibbs ensemble. It is for this reason we require general formulae to transform one ensemble into the other: Eq. (3.17) allows one to retrieve the

Gibbs ensemble from the Helmholtz ensemble; its inversion, from Manca, et. al. [53], allows one to retrieve the Helmholtz ensemble from the Gibbs ensemble:

$$\mathfrak{q}^*(\boldsymbol{\xi}) = \left(\frac{\beta}{2\pi}\right)^3 \iiint \mathfrak{z}^*(i\mathbf{f}) e^{-i\beta \mathbf{f}\cdot\boldsymbol{\xi}}\, d^3\mathbf{f}. \tag{3.21}$$

Eqs. (3.17) and (3.21) are the ensemble transformation relations. In the thermodynamic limit and under appreciable loads [131–133], fluctuations become negligible and the free energies of the two ensembles are related by the Legendre transformation

$$\varphi^*(\mathbf{f}) = \psi^*(\boldsymbol{\xi}) - \mathbf{f}\cdot\boldsymbol{\xi}. \tag{3.22}$$

Therefore, in the limit of long chains, the mechanical response of the chain can be obtained equivalently from either ensemble:

$$\mathbf{f} = \frac{\partial \psi^*}{\partial \boldsymbol{\xi}}, \qquad \boldsymbol{\xi} = -\frac{\partial \varphi^*}{\partial \mathbf{f}}, \tag{3.23}$$

and the two equilibrium distributions are related by

$$\frac{P^{\mathrm{eq}}_{\mathfrak{z}}(\mathbf{f})}{P^{\mathrm{eq}}_{\mathfrak{z}}(\mathbf{f}_{\mathrm{ref}})} = e^{-\beta(\psi_{\mathrm{ref}} - \varphi_{\mathrm{ref}} - \mathbf{f}\cdot\boldsymbol{\xi})} \left[\frac{P^{\mathrm{eq}}(\boldsymbol{\xi})}{P^{\mathrm{eq}}(\boldsymbol{\xi}_{\mathrm{ref}})}\right]. \tag{3.24}$$

Eqs. (3.22) and (3.24) are the ensemble transformation relations in the thermodynamic limit.

Example polymer chain model

In order to demonstrate the above sets of equations, we consider the EFJC model, where the polymer chain is represented by $M = N_b + 1$ atoms/hinges connected in series by N_b flexible links of rest length ℓ_b and harmonic potential stiffnesses k_b. Due to the nonzero potentials, the mechanical response of this model will be due to coupled contributions from both entropic and enthalpic effects. The EFJC model has a Gibbs ensemble partition function that can be evaluated analytically: one form is given by Fiasconaro and Falo [57] as

$$\mathfrak{z}^*(\eta) = \left\{ B_0 \frac{\sinh(\eta)}{\eta}\, e^{\eta^2/2\kappa} \left[1 + \frac{\eta}{\kappa}\coth(\eta)\right]\right\}^{N_b}, \tag{3.25}$$

where $\eta = \beta f \ell_b$ is the non-dimensional force, $\kappa = \beta k_b \ell_b^2$ is the non-dimensional link stiffness, and $B_0 = 2^{5/2}\pi^{3/2}\beta\ell_b^3\kappa^{-1/2}$. For $\kappa \to \infty$, we recover the freely joined chain (FJC) Gibbs ensemble partition function [50]. To retrieve the Helmholtz ensemble partition function we use Eq. (3.21), which in this case (spherically symmetric) as shown by Manca, et. al. [53] reduces to

$$\mathfrak{q}^*(\gamma) = \frac{1}{2\pi^2 N_b \ell_b^3}\frac{1}{\gamma}\int_0^\infty \mathfrak{z}^*(i\eta)\sin(N_b\eta\gamma)\eta\, d\eta, \tag{3.26}$$

where $\gamma = \xi/N_b\ell_b$ is the chain end-to-end length relative to the contour length $N_b\ell_b$. After using Eq. (3.26) to calculate $\mathfrak{q}^*$, we then use Eq. (3.12) in order to calculate ψ^* from $\mathfrak{q}^*$, and subsequently Eq. (3.13) to calculate P^{eq} from ψ^*. Although we have started from the Gibbs ensemble, we have calculated ψ^* and P^{eq} exactly, which we will refer to as the Helmholtz method (Fig. 5.2, top pathway).

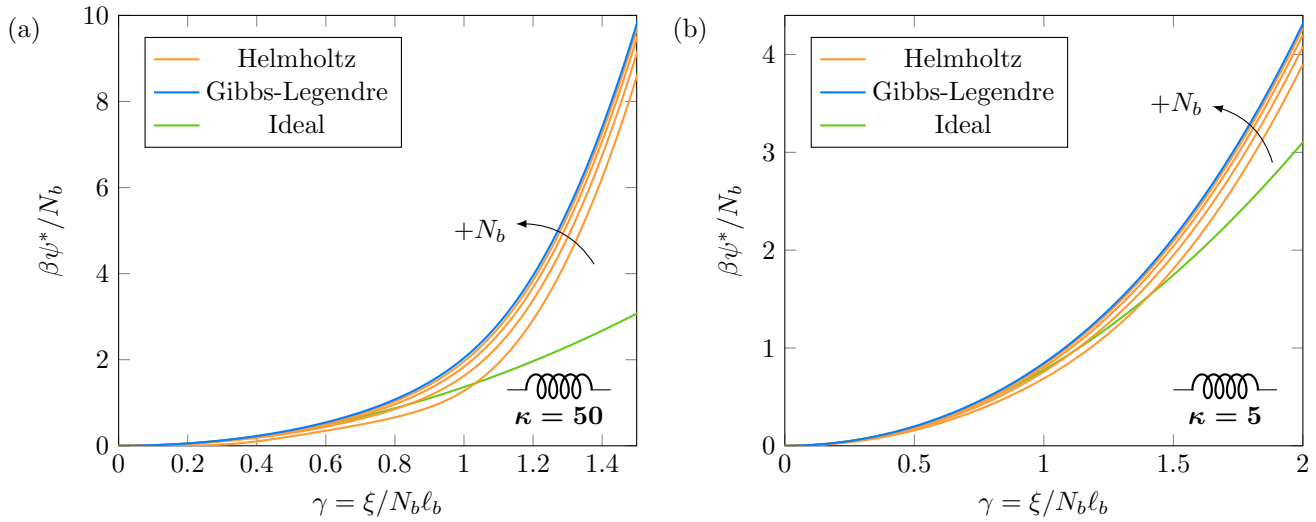


Figure 3.2: Non-dimensional free energy per link versus end-to-end stretch for $N_b = 3$, 5, 10, and 25, for (a) $\kappa = 50$ and (b) $\kappa = 5$. The free energy is plotted using both the Helmholtz and Gibbs-Legendre methods, as well as using the ideal model valid for $\gamma \ll 1$; the Gibbs-Legendre and ideal results are independent of N_b.

The Helmholtz method is often computationally challenging, so simpler approximate methods are typically used. When the thermodynamic limit $N_b \to \infty$ is satisfied, the Legendre transformation in Eq. (3.22) may be used to calculate the Helmholtz free energy ψ^* from the Gibbs free energy φ^*. We will refer to this method as the Gibbs-Legendre method (Fig. 5.2, bottom pathway). This method is expedient if φ^* is known exactly, which is true in the case of the EFJC model. The exact value of φ^* for the EFJC is calculated by plugging $\mathfrak{z}^*$ from Eq. (3.25) into Eq. (3.18). To obtain the mechanical response in order to perform the Legendre transformation, we use Eq. (3.23) to obtain the non-dimensional end-to-end length in the Gibbs ensemble

$$\gamma(\eta) = \mathcal{L}(\eta) + \frac{\eta}{\kappa}\left[1 + \frac{1 - \mathcal{L}(\eta)\coth(\eta)}{1 + (\eta/\kappa)\coth(\eta)}\right]. \tag{3.27}$$

Now we assume that the thermodynamic limit $N_b \to \infty$ is satisfied and use Eq. (3.22) to calculate the Helmholtz free energy for the EFJC to be

$$\psi^*(\gamma) = N_b \mathfrak{b}T\left\{\eta\mathcal{L}(\eta) + \ln\left[\frac{\eta}{B_0\sinh(\eta)}\right] - \ln\left[1 + \frac{\eta}{\kappa}\coth(\eta)\right] + \frac{\eta^2}{\kappa}\left[\frac{1}{2} + \frac{1 - \mathcal{L}(\eta)\coth(\eta)}{1 + (\eta/\kappa)\coth(\eta)}\right]\right\}, \tag{3.28}$$

where we solve for $\eta = \eta(\gamma)$ using Eq. (3.27) in order to get $\psi^* = \psi^*(\gamma)$. The equilibrium distribution in the thermodynamic limit is then calculated using Eq. (3.13),

$$P^{\text{eq}}(\gamma) = \frac{1}{\ell_b^3 C}\left(\frac{\sinh(\eta)\left[1 + (\eta/\kappa)\coth(\eta)\right]}{\eta\exp\left[\eta\mathcal{L}(\eta)\right]}\exp\left\{\frac{\eta^2}{\kappa}\left[\frac{1}{2} + \frac{1 - \mathcal{L}(\eta)\coth(\eta)}{1 + (\eta/\kappa)\coth(\eta)}\right]\right\}\right)^{N_b}, \tag{3.29}$$

where $C = C(N_b, \kappa)$ is such that the distribution is normalized. For $\kappa \to \infty$ in Eqs. (3.27) and (3.29), we recover the FJC mechanical response $\gamma = \mathcal{L}(\eta)$ and probability distribution in the thermodynamic limit [1].

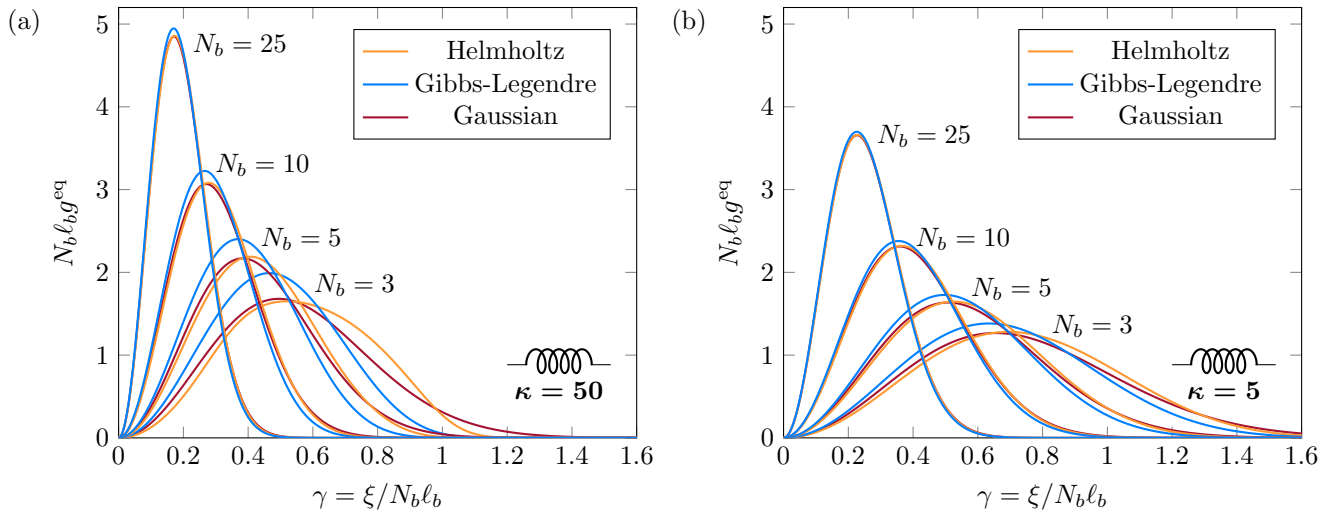


Figure 3.3: Non-dimensional equilibrium radial distribution function versus end-to-end stretch for $N_b = 3$, 5, 10, and 25, for (a) $\kappa = 50$ and (b) $\kappa = 5$. The distribution is plotted using the Helmholtz and Gibbs-Legendre methods as well as using the Gaussian distribution valid for $N_b \to \infty$.

An alternative approximation method is to assume a Gaussian distribution for the equilibrium distribution. This assumption is valid in the limit $N_b \to \infty$ due to the central limit theorem. In order to determine this Gaussian distribution for the EFJC model, we first approximate the mechanical response in Eq. (3.27) for small forces ($\eta \ll 1$) by the linear relation

$$\gamma(\eta) = \frac{\eta}{c_\kappa}, \qquad c_\kappa = \frac{\kappa(\kappa+1)}{\kappa^2 + 6\kappa + 3}, \tag{3.30}$$

and subsequently the free energy in Eq. (3.28) by a quadratic relation for $\eta \ll 1$. Combining these results yields the small stretch ($\gamma \ll 1$) free energy

$$\psi^*(\gamma) = \frac{3}{2} c_\kappa N_b \mathfrak{b} T \gamma^2. \tag{3.31}$$

We now make use of this small stretch approximation to construct the equilibrium distribution for $N_b \to \infty$ using Eq. (3.13), which is then

$$P^{\mathrm{eq}}(\gamma) = \left(\frac{3 c_\kappa}{2\pi N_b \ell_b^2} \right)^{3/2} \exp\left(-\frac{3}{2} c_\kappa N_b \gamma^2 \right). \tag{3.32}$$

We remark that this distribution is a valid approximation for any stretch as long as $N_b \to \infty$, so it is common to utilize this equilibrium distribution with the Gibbs-Legendre method free energy function in order to approximate the full Helmholtz method.

Since the Gibbs-Legendre method is often used to approximate the true Helmholtz free energy, we plot the EFJC non-dimensional free energy ($\beta\psi^*/N_b$) as a function of end-to-end chain stretch (γ) for $\kappa = 50$ and varying N_b in Fig. 5.1(a), obtained using both the Helmholtz and the Gibbs-Legendre methods, as well as the ideal chain free energy. See that for small values of N_b the difference in free energy between the Helmholtz and Gibbs-Legendre methods is quite considerable, such as for $N_b = 3$ where the relative difference is nearly constant at 60% for $\gamma \in (0.5, 1)$. As N_b increases, the difference between the two methods shrinks, becoming quite small when $N_b = 25$. We can also

observe that the ideal chain free energy, which matches the Gibbs-Legendre method free energy at small stretch, does not match that of the Helmholtz method until N_b becomes large. We repeat this analysis for the smaller EFJC link stiffness ($\kappa = 5$) in Fig. 5.1(b), where we observe the same trends but overall smaller differences among the methods. This can be understood by reconsidering the Gibbs ensemble partition function in Eq. (3.25) and the ensemble transformation relation in Eq. (3.26). First consider the case of $N_b \to \infty$, where we will receive $\mathfrak{q}^*(\gamma) \to \mathfrak{z}^*(\eta)e^{-N_b\eta\gamma}$ from Eq. (3.26), and where the Gibbs-Legendre method results will exactly match that of the Helmholtz method. This is because $\mathfrak{z}^*(i\eta)$ will decay rapidly as a function of η when N_b becomes large and effectively contain a Dirac delta function. Now for $\kappa \to 0$, we see that $\mathfrak{z}^*(i\eta)$ will also decay rapidly as a function of η, which will also act as a Dirac delta function via one definition,

$$\delta(\eta) = \lim_{\kappa \to 0^+} \frac{1}{\sqrt{2\pi\kappa}} e^{-\eta^2/2\kappa}, \tag{3.33}$$

which appears in $\mathfrak{z}^*(i\eta)$ after recalling that $B_0 \propto 1/\sqrt{\kappa}$ in Eq. (3.25). This is why we observe smaller differences between the Gibbs-Legendre and Helmholtz methods as κ decreases. It can also be understood intuitively as a decreasing correlation between the links: the link degrees of freedom in the Gibbs ensemble are completely independent, while that in the Helmholtz ensemble are because of the end-to-end length constraint. As κ decreases the link degrees of freedom in the Helmholtz ensemble become increasingly independent of each other, approaching the $\kappa = 0$ limit where they are completely independent.

We plot the EFJC non-dimensional radial distribution function ($N_b\ell_b g^{\text{eq}}$) as a function of end-to-end stretch for $\kappa = 50$ and varying N_b in Fig. 3.3(a). The radial distribution function is plotted using both the Helmholtz and the Gibbs-Legendre methods, as well as the $N_b \to \infty$ limiting Gaussian distribution. The Gibbs-Legendre distribution tends to be quite different from the Helmholtz distribution for small values of N_b, while the Gaussian distribution tends to be a bit closer. By $N_b = 25$, the Gaussian and Helmholtz distributions become nearly indistinguishable, and the Gibbs-Legendre distribution retains only a small difference from the other two. We repeat this analysis for a smaller EFJC link stiffness ($\kappa = 5$) in Fig. 3.3(b), where we observe the same trends but overall smaller differences among the methods. This difference is again explained by the more rapidly decaying $\mathfrak{z}^*(i\eta)$ in Eq. (3.26) as κ decreases, as previously discussed. Though the Gibbs-Legendre method free energy is immensely closer to the Helmholtz method free energy than the ideal chain free energy, we see here that the Gaussian distribution – obtained from the ideal chain free energy using the distribution-behavior correspondence in Eq. (3.13) – tends to be much closer to the Helmholtz method distribution than the Gibbs-Legendre method distribution. Looking back to Fig. 5.1, this is likely because the Gibbs-Legendre method overestimates the single chain free energy increase with stretch for smaller N_b, resulting in an underestimate of the probability of chains at larger stretch observed in Fig. 3.3 due to the distribution-behavior correspondence relations.

Distribution evolution

We introduce $P(\boldsymbol{\xi}, t)$ as the probability density distribution of chains with end-to-end vector $\boldsymbol{\xi}$ at time t, which we presume to initially be in the equilibrium distribution, $P(\boldsymbol{\xi}, 0) = P^{\text{eq}}(\boldsymbol{\xi})$. Liouville's equation [49] describes the evolution of a probability density P in the single chain phase space (atomic coordinates and momenta) as

$$\frac{\partial P}{\partial t} = -\sum_{j=1}^{M} \left(\frac{\partial P}{\partial \mathbf{q}_j} \cdot \dot{\mathbf{q}}_j + \frac{\partial P}{\partial \mathbf{p}_j} \cdot \dot{\mathbf{p}}_j \right). \tag{3.34}$$

When we apply Liouville's equation to $P(\boldsymbol{\xi}, t)$, the only nonzero derivative we retrieve is that relating to the chain end-to-end vector $\boldsymbol{\xi} = \mathbf{q}_M - \mathbf{q}_1$, and thus the evolution law for the distribution of end-to-end vectors is

$$\frac{\partial P}{\partial t} = -\frac{\partial P}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}}, \tag{3.35}$$

where $\dot{\boldsymbol{\xi}} = \dot{\boldsymbol{\xi}}(\boldsymbol{\xi}, t)$ is left to be prescribed.

3.2.2 Macroscopic constitutive theory

In order to extend our theory into the macroscale, we prescribe an affine deformation to an incompressible network and analytically solve for the distribution evolution. Equipped with this connection between the statistical and continuum mechanics of the polymer network, we use the Coleman-Noll procedure [82] to develop the macroscopic constitutive theory. We choose the deformation gradient $\mathbf{F}$ and the temperature T as the independent thermodynamic state variables. We presume these thermodynamic state variables to be complete, allowing us to consider time derivatives of constitutive functions to be implicit, i.e. we may expand them in terms of the time derivatives of the state variables. After some derivation – including the full treatment of a boundary integral term that, until now, has been either omitted or otherwise assumed to be zero – we ultimately retrieve a closed-form relation for the Cauchy stress in terms of the applied deformation, the chain free energy function, and the equilibrium distribution of end-to-end vectors.

Macroscopic connection

We assume that the evolution of end-to-end vectors is affine with the deformation, $\dot{\boldsymbol{\xi}} = \mathbf{L} \cdot \boldsymbol{\xi}$, where $\mathbf{L} = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1}$ is the velocity gradient. Eq. (3.35) then becomes

$$\frac{\partial P}{\partial t} = -\left(\frac{\partial P}{\partial \boldsymbol{\xi}} \right) \cdot \mathbf{L} \cdot \boldsymbol{\xi}. \tag{3.36}$$

This first order linear partial differential equation can be solved analytically using the method of characteristics (See Appendix B.1). Under the initial conditions $\mathbf{F}(0) = \mathbf{1}$ and $P(\boldsymbol{\xi}, 0) = P^{\text{eq}}(\boldsymbol{\xi})$, the solution is

$$P(\boldsymbol{\xi}, t) = P^{\text{eq}}\left[\mathbf{F}^{-1}(t) \cdot \boldsymbol{\xi} \right], \tag{3.37}$$

which simply states that the probability density of a chain having end-to-end vector $\boldsymbol{\xi}$ at time t is equal to the probability density of that end-to-end vector mapped backward to the corresponding end-to-end vector in the equilibrium distribution.

Second law analysis

Now that we are equipped with the probability distribution of polymer chains within the network as a function of the deformation, we write the current Helmholtz free energy density of the network (a) by integrating the probability-weighted free energy function over all end-to-end vectors,

$$a(t) = n \iiint P(\boldsymbol{\xi}, t)\psi^*(\boldsymbol{\xi})\, d^3\boldsymbol{\xi} - p(J-1), \tag{3.38}$$

where $n = N/V$ is the number density of chains and p is the pressure enforcing the incompressibility constraint that $J = \det(\mathbf{F}) = 1$. Note that we have only included contributions related to the chain configuration integral and left out those related to the chain momentum integral. This is because the latter terms will only introduce spherical terms to the stress (ideal gas law) and therefore can be lumped into the pressure without loss of generality. Thermodynamically admissible processes satisfy the Clausius-Duhem inequality [134],

$$\dot{a} + s\dot{T} - \boldsymbol{\sigma} : \mathbf{L} \leq 0, \tag{3.39}$$

where s is the entropy density and $\boldsymbol{\sigma}$ is the Cauchy stress tensor. This reduced form of the Clausius-Duhem inequality involves several classical assumptions that are standard in the Coleman-Noll procedure, such as the neglect of non-mechanical work, the constitutive relations for the entropy flux and entropy source, and in this case Fourier's law for the heat flux [135]. We expand the implicit time derivative of the Helmholtz free energy density using our complete set of state variables,

$$\dot{a} = \left(\frac{\partial a}{\partial t}\right)_T + \left(\frac{\partial a}{\partial t}\right)_{\mathbf{F}} \tag{3.40}$$

$$= \left(\frac{\partial a}{\partial \mathbf{F}}\right)_T : \dot{\mathbf{F}} + \left(\frac{\partial a}{\partial T}\right)_{\mathbf{F}} \dot{T}, \tag{3.41}$$

and substitute this result back into Eq. (3.39) for

$$\left[\left(\frac{\partial a}{\partial T}\right)_{\mathbf{F}} + s\right]\dot{T} + \left[\left(\frac{\partial a}{\partial \mathbf{F}}\right)_T \cdot \mathbf{F}^T - \boldsymbol{\sigma}\right] : \mathbf{L} \leq 0. \tag{3.42}$$

We now consider the set of processes where the deformation is held fixed, $\mathbf{L} = \mathbf{0}$, and the temperature is varied arbitrarily. Since $\dot{T}$ can be any real number, positive or negative, and this inequality must hold, we see that the term in the first set of brackets must always be zero and we receive the expected constitutive relation for the entropy density

$$s = -\left(\frac{\partial a}{\partial T}\right)_{\mathbf{F}}, \tag{3.43}$$

and after going back to our original derivative notation, we are left with the inequality

$$\left(\frac{\partial a}{\partial t}\right)_T - \boldsymbol{\sigma} : \mathbf{L} \leq 0. \tag{3.44}$$

Several steps are then taken in order to proceed from Eq. (3.44) and retrieve the stress. We first neglect dissipative stresses, thus taking the equality in Eq. (3.44) and receiving a hyperelastic

stress. Next, we take Eq. (3.38) and assume spherical symmetry in ψ^*, which causes the stress to be non-polar. We then require that ψ^* grows sufficiently fast as $\xi \to \infty$, in order to show that the boundary integral term resulting from integration by parts is zero. The full derivation is presented in Appendix B.2 and yields the stress to be

$$\boldsymbol{\sigma}(t) = n \iiint P^{\mathrm{eq}} \left[\mathbf{F}^{-1}(t) \cdot \boldsymbol{\xi}\right] \left(\frac{\partial \psi^*}{\partial \xi}\right) \left(\frac{\boldsymbol{\xi}\boldsymbol{\xi}}{\xi}\right) d^3\boldsymbol{\xi} - \left[p^{\mathrm{eq}} + \Delta p(t)\right] \mathbf{1}, \tag{3.45}$$

where $\mathbf{1}$ is the identity tensor, the differential pressure $\Delta p(t)$ enforces incompressibility, and the equilibrium pressure p^{eq} (from $\boldsymbol{\sigma}(0) = \mathbf{0}$) is

$$p^{\mathrm{eq}} = \frac{n}{3} \int g^{\mathrm{eq}}(\xi) \left(\frac{\partial \psi^*}{\partial \xi}\right) \xi \, d\xi. \tag{3.46}$$

The derivative of the chain Helmholtz free energy can be replaced with the force using Eq. (3.23), but one must be careful to ensure that the force is computed in the Helmholtz ensemble: the force from the Gibbs ensemble may only be used in the thermodynamic limit ($N_b \to \infty$). If we utilize the ideal chain free energy from Eq. (3.31), one can easily show that the Neo-Hookean model results, as expected (see Appendix B.3).

3.2.3 Implementation

To close this section, we would like to point out some important aspects of the model implementation. It happens that Eq. (3.26) is difficult to evaluate with the EFJC partition function in Eq. (3.25) for moderate to large N_b, which is due to the integrand oscillating rapidly and decaying slowly. While certain integration schemes may perform reasonably well for small or large N_b, it is most desirable to use an integration scheme that remains accurate for the full range of N_b being considered. To evaluate this integral with high precision, we used the double exponential quadrature scheme presented by Ooura and Mori [136] and their Fortran script `intdeo.f` that implements it, as well as the arbitrary precision `Fortran` package `MPFUN2015` provided by Bailey [137]. These calculations were carried out using the Extreme Science and Engineering Discovery Environment (XSEDE) `Stampede2` cluster [124]. When calculating the integrals in Eq. (3.45) in order to retrieve the stress, it is unwieldy to repeatedly call a function to exactly evaluate $\psi^*(\gamma)$ for the EFJC as $\gamma \to \infty$. See from Eqs. (3.27)–(3.28) that $\gamma \sim \eta/\kappa$ and $\beta\psi^* \sim N_b\eta^2/\kappa$ as $\eta \to \infty$, which combined shows that $\beta\psi^* \sim N_b\kappa\gamma^2$ as $\gamma \to \infty$. The neglected terms in this asymptotic relation for $\psi^*(\gamma)$ are quite small, so the relation is accurate even for γ only moderately above unity. Therefore, in order to greatly speed up the computation of the stress at negligible cost to accuracy, we fit a quadratic function to ψ^* for large γ and call this function instead when γ is above a certain value.

3.3 Macroscopic Results

Now that we have fully formulated the theory, we are able to explicitly examine the effects that changes in the statistical description have on the macroscopic mechanics. Traditionally in polymer network constitutive modeling, the Helmholtz ensemble has been approximated using the Legendre transformation from the Gibbs ensemble, so we will start by examining the difference in macroscopic mechanical response when using these Helmholtz and Gibbs-Legendre methods. Another common

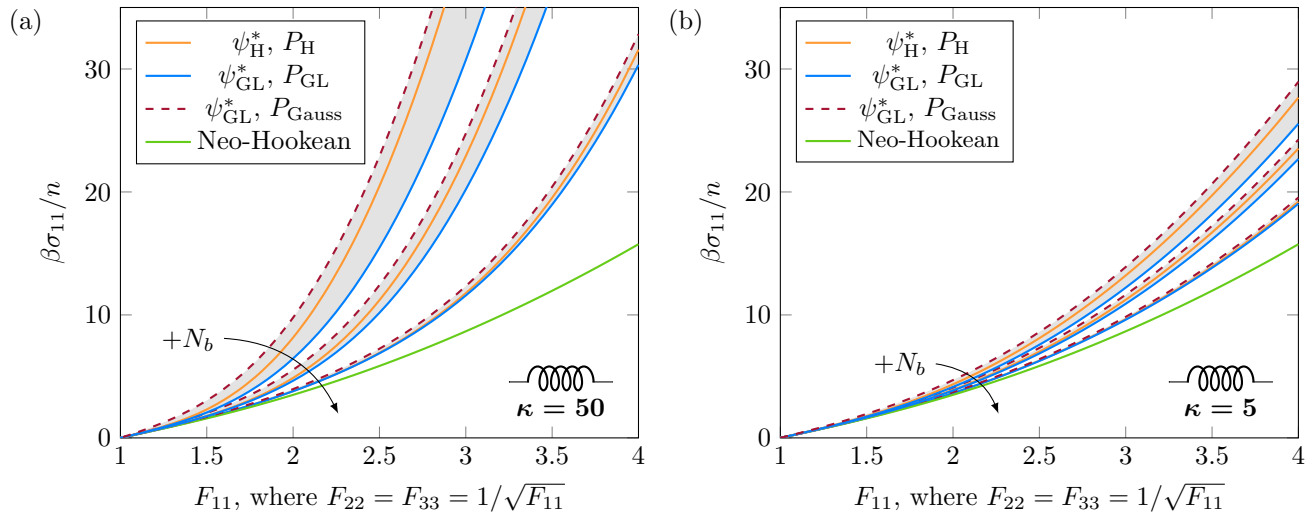


Figure 3.4: Non-dimensional uniaxial stress-stretch results for the EFJC network with (a) $\kappa = 50$ and (b) $\kappa = 5$, for $N_b = 5$, 10, and 25. This mechanical response is plotted using the true method (Helmholtz) and two approximation methods (Gibbs-Legendre, Gibbs-Legendre-Gaussian). Shading indicates equal N_b value. The Neo-Hookean response is included as reference.

choice in these constitutive models is to assume that the equilibrium distribution is Gaussian, so we will then examine how the true Helmholtz ensemble mechanical response differs from that assuming the Gaussian distribution and the Gibbs-Legendre free energy function; we will refer to this as the Gibbs-Legendre-Gaussian method. In both of these studies, we will see that a long enough polymer chain causes all approaches to result in the same mechanical response. This convergence occurs before the $N_b \to \infty$ limit represented by choosing the ideal chain free energy function and the Gaussian distribution, which is the Neo-Hookean model.

We apply the Helmholtz and Gibbs-Legendre methods to a polymer network modeled to consist of EFJCs. $\mathfrak{z}^*$ for the EFJC is given by Eq. (3.25). The Helmholtz method takes $\mathfrak{z}^*$ and uses Eq. (3.26) to compute $\mathfrak{q}^*$, then computes ψ^* using Eq. (3.12) and P^{eq} using Eq. (3.22). The Gibbs-Legendre method assumes $N_b \to \infty$ in order to use ψ^* in Eq. (3.28) and P^{eq} in Eq. (3.29). In both methods, we compute the stress under uniaxial tension in the 1-direction using Eq. (3.45), where the deformation gradient is diagonal with components due to symmetry and incompressibility, $F_{22} = F_{33} = 1/\sqrt{F_{11}}$. In Fig. 3.4(a) we plot the non-dimensional uniaxial stress $\beta\sigma_{11}/n$ versus the applied stretch F_{11} using the non-dimensional EFJC stiffness $\kappa = 50$ and an increasing numbers of links $N_b = 5$, 10, and 25. The Neo-Hookean stress-stretch response – retrieved through using Eq. (3.31) for ψ^* and Eq. (3.32) for P^{eq} – is included for reference. For small numbers of links such as $N_b = 5$, the Gibbs-Legendre method drastically underestimates the overall stiffness of the true stress response from the Helmholtz method. This difference shrinks as N_b increases, becoming only a tiny (but increasing with stretch) difference when $N_b = 25$, analogous to the differences between the free energy functions and the equilibrium distributions shrinking in Figs. 5.1 and 3.3. We see that the two methods seem to converge before the limit $N_b \to \infty$ is truly reached, where the Neo-Hookean mechanical response would be retrieved. We have repeated the same analysis for the lower stiffness ($\kappa = 5$) in Fig. 3.4(b), where we observe the same behavior as N_b increases, but in general less difference between the two methods compared to the $\kappa = 50$ case for any N_b. These features are independent of loading mode, as is evident by the above analyses implemented for equibiaxial tension and simple shear (Fig. 3.5).

For equibiaxial tension we apply $F_{11} = F_{22}$, where incompressibility requires $F_{33} = 1/F_{11}^2$, and for simple shear we apply F_{12}, where $F_{11} = F_{22} = F_{33} = 1$.

As previously mentioned, the equilibrium distribution P^{eq} tends towards Gaussian as $N_b \to \infty$. It is common for polymer network constitutive models to choose a non-ideal free energy function ψ^* but assume N_b is large enough to warrant the use of the Gaussian P^{eq}, rather than the P^{eq} from the distribution-behavior correspondence relation given by Eq. (3.14). For the case of the EFJC, the limit $N_b \to \infty$ results in the Gaussian P^{eq} given by Eq. (3.32). In attempting to approximate the true Helmholtz method in the limit as $N_b \to \infty$, one would then assume a Gaussian equilibrium distribution and either use the Helmholtz method or Gibbs-Legendre method for ψ^*. We will neglect the case of the Helmholtz method ψ^* combined with the Gaussian P^{eq}, since by distribution-behavior correspondence, one could simply find the true P^{eq} after knowing ψ^*. In either case, the macroscopic mechanical response will converge to that of the true Helmholtz method when N_b becomes sufficiently large.

We apply the Gibbs-Legendre-Gaussian method to a polymer network modeled to consist of EFJCs. The Gibbs-Legendre-Gaussian method uses ψ^* from Eq. (3.28) and the Gaussian P^{eq} from Eq. (3.32). Taking the non-dimensional EFJC stiffness $\kappa = 50$, in Fig. 3.4(a) we plot the non-dimensional stress $\beta\sigma_{11}/n$ versus the applied stretch F_{11} for increasing numbers of links $N_b = 5$, 10, and 25. The Gibbs-Legendre-Gaussian method does well matching the tangent stiffness and keeping the relative error small at larger stretches, but tends to do poorly at small to intermediate stretches ($F_{11} \leq 2$). We also see that this method seems to converge to the Helmholtz method stress-stretch response before the limit $N_b \to \infty$ is truly reached, where the Neo-Hookean mechanical response would be retrieved. We have repeated the same analysis for the lower stiffness $\kappa = 5$ in Fig. 3.4(b), where we observe the same trends but smaller relative errors, since as previously discussed and shown in Fig. 3.3, the distributions from different methods become more alike for smaller κ.

When comparing the three methods in Figs. 3.4 and 3.5, we see that the Gibbs-Legendre method tends to underestimate the true mechanical response given by the Helmholtz method, and the Gibbs-Legendre-Gaussian method tends to overestimate it, especially at small stretches. To understand this further, we can consider the initial moduli in each case by applying an infinitesimal deformation $\mathbf{F} = \mathbf{1} + \mathbf{E}$, where $\mathbf{E}$ is the infinitesimal strain tensor. Straightforward analysis shows (see Appendix B.4) that the stress from Eq. (3.45) then becomes

$$\boldsymbol{\sigma}(t) = 2\mu\mathbf{E}(t) - \Delta p(t)\mathbf{1}, \tag{3.47}$$

where method-specific shear modulus μ is given by

$$\mu = \frac{4\pi}{15}\, n\mathfrak{b}T \iiint \left(-\frac{\partial P^{\text{eq}}}{\partial \xi}\right)\left(\frac{\partial \beta\psi^*}{\partial \xi}\right)\xi^4\, d\xi. \tag{3.48}$$

We find Δp by enforcing incompressibility via $\text{tr}(\mathbf{E}) = 0$, from which we find the initial moduli to be 3μ, 6μ, and μ for uniaxial tension, equibiaxial tension, and simple shear, respectively. For the Neo-Hookean model we receive the nondimensional modulus $\beta\mu/n = 1$, as expected, whereas for the other methods we cannot analytically compute the integral but in general receive $\beta\mu/n \geq 1$. We can, however, compute the shear modulus in specific cases: for $N_b = 5$ and $\kappa = 50$, $\beta\mu/n = 1.1161$ for the Helmholtz method, $\beta\mu/n = 1.0957$ for the Gibbs-Legendre method, and $\beta\mu/n = 1.5108$ for the Gibbs-Legendre-Gaussian method. See that while the Gibbs-Legendre method underestimates the modulus, the Gibbs-Legendre-Gaussian method drastically overestimates it, which is why we observe

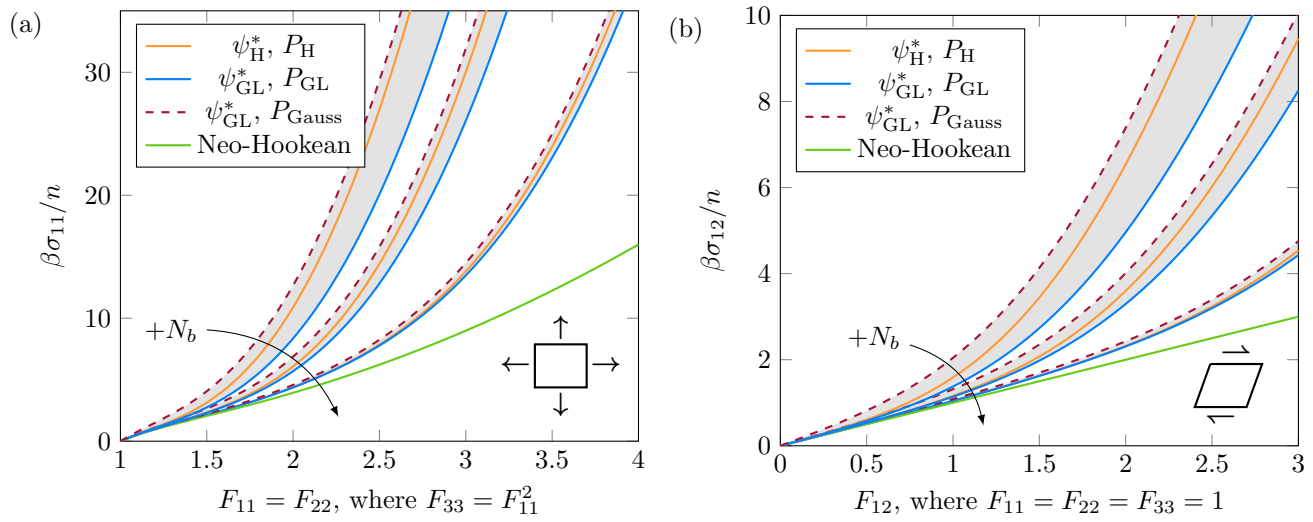


Figure 3.5: Non-dimensional (a) equibiaxial and (b) simple shear stress-stretch results for the EFJC network with $\kappa = 50$ for $N_b = 5$, 10, and 25. This mechanical response is plotted using the true method (Helmholtz) and two approximation methods (Gibbs-Legendre, Gibbs-Legendre-Gaussian). Shading indicates equal N_b value. The Neo-Hookean response is included as reference.

a poor performance of the Gibbs-Legendre-Gaussian method at small stretches in Figs. 3.4 and 3.5. We see this difference even persists when $N_b = 25$ and $\kappa = 50$ – where $\beta\mu/n = 1.0155$, 1.0153, and 1.0557 – and when $N_b = 5$ and $\kappa = 5$ – where $\beta\mu/n = 1.051$, 1.0402, and 1.1691 – for the Helmholtz, Gibbs-Legendre, and Gibbs-Legendre-Gaussian methods, respectively. These differences in modulus occur because the Gibbs-Legendre-Gaussian method ignores distribution-behavior correspondence. The Gibbs-Legendre method tends to overestimate the free energy (Fig. 5.1), which causes the Gibbs-Legendre method distribution (Fig. 3.3) to underestimate the equilibrium amount of chains at larger stretch. These two inaccuracies then naturally cancel to some extent when integrating for the modulus, but when a non-corresponding distribution is instead used – such as in the Gibbs-Legendre-Gaussian method – this cancellation does not occur. The Gaussian distribution (Fig. 3.3) predicts an increased equilibrium number of chains at larger stretch, which combined with the Gibbs-Legendre method overestimated free energy will produce a significantly overestimated modulus. This also explains the poor convergence of the Gibbs-Legendre-Gaussian method as N_b increases, which is especially evident from Fig. 3.5(b) where the Gibbs-Legendre method is nearly exact for $N_b = 25$ while the Gibbs-Legendre-Gaussian method is not. Therefore, it seems that it is better to obey distribution-behavior correspondence in using the Gibbs-Legendre method for small to intermediate stretches and/or larger number of links. For a large stretch and small number of links, it is seemingly better to instead use the method with the more accurate equilibrium distribution, which here is the Gibbs-Legendre-Gaussian method. For a truly large number of links, we can also be certain that either method will produce accurate approximations of the Helmholtz method mechanical response.

3.4 Conclusion

We have performed a fundamental statistical mechanical derivation in order to account for the naturally occurring correspondences between the mechanical behavior of a single polymer chain and the equilibrium distribution of a network of such chains. Correspondences between different single-chain thermodynamic ensembles – both exact and in the thermodynamic limit – as well as the Gaussian limit of the equilibrium distribution in either ensemble, were also accounted for and discussed in detail using the extensible freely jointed chain model as an example. This elaborate framework was then kept in-tact as we considered the macroscopic constitutive theory of the polymer network and derived the Cauchy stress in terms of the affine deformation of a general network of polymer chains. We used this constitutive relation for the stress to illustrate that important distinctions in the statistical description persist to play an important role in the observed macroscopic mechanical response, at least until the number of links in the polymer chains becomes large. Obeying the distribution-behavior correspondence relations allowed a more accurate approximation of the macroscopic stress at small to intermediate deformations and/or longer chain lengths, even though this corresponding equilibrium distribution was a worse match to the true distribution than the Gaussian approximation. However when chains are short and the deformation is large, we saw that it was better to utilize the Gaussian distribution, which can be attributed to the extensive evolution the initial distribution undergoes during large deformations. This meticulous treatment is vital for future constitutive model construction and is readily applicable to more complex polymer systems. This macroscopic framework is readily compatible with any chain models that allow the equilibrium probability distribution to be normalized and are infinitely extensible. Critically, this includes common biopolymer models such as the wormlike chain model as long as the extensible forms are used. This framework does not accommodate potentials that simulate bond breaking such as the Morse potential, but bond breaking could be captured by instead including a reaction pathway to broken chains.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant No. CAREER-1653059. Calculations in this work used the Extreme Science and Engineering Discovery Environment (XSEDE) `Stampede2` cluster [124], which is supported by National Science Foundation Grant ACI-1548562.

Chapter 4

Low-temperature statistical thermodynamics by an asymptotic method

Chapter 4 is adapted from an ongoing effort to develop a low-temperature analog of the high-temperature perturbation theory developed by Robert W. Zwanzig over 65 years ago [138]. Just as "high-temperature" is synonymous with weak potential energies of interaction, "low-temperature" is synonymous with strong potential energies of interaction. These strong interactions have been traditionally approximated as athermal rigid constraints or using the rigid-rotor-harmonic-oscillator approximation [49], but our theory now offers a systematic method to obtain better approximations. The main theory, which retrieves an asymptotic expansion for the canonical partition function, is complete (Sec. 4.2) and demonstrated for a simple case (Sec. 4.3.1). Ongoing work is concerned with applications to molecular stretching (Sec. 4.3.2) and broader molecular simulation. This work has been advised by and contributed to by both Roger F. Loring and Meredith N. Silberstein.

Interatomic interactions such as chemical bonds are often quite strong, becoming nearly rigid at sufficiently low temperatures. In classical statistical thermodynamics, these potential energies of interaction are traditionally approximated as athermal rigid constraints or using the rigid-rotor-harmonic-oscillator approximation. Though several methods of obtaining rotation-vibration coupled and anharmonic corrections to these traditional approximations have been demonstrated over the years, a general method has yet to be developed. Here a classical asymptotic theory is developed, where the thermodynamic properties of the full system are related to those of the system with rigid constraints and to the strong potentials near their minima. This is a general result and serves as the low-temperature analog of the high-temperature perturbation theory developed by Robert W. Zwanzig in his seminal paper from 1954. Using the simple example of the three-dimensional harmonic oscillator, the theory is shown to obtain the correct asymptotic expansion consisting of the rigid-rotor-harmonic-oscillator approximation and the correction for rotation-vibration coupling. Our theory is especially relevant to molecular stretching experiments, where considerable applied force prevents stiff bonds from being accurately approximated as rigid or even harmonic. Our theory is also applicable to molecular simulation in general, as it provides a systematic method to obtain corrections for molecular vibration spectra. In the future, we will consider each of these applications and provide more validations of our theory.

4.1 Introduction

In classical statistical thermodynamics (equilibrium statistical mechanics), solving systems of interacting particles is a challenging and often insurmountable task without using simulation approaches, such as molecular dynamics or Monte Carlo methods. As pointed out by McQuarrie [49], the configuration integral that appears when considering interacting particles was essentially responsible for most research in statistical thermodynamics. In 1954, Robert Zwanzig alleviated some of this burden by developing high-temperature perturbation theory [138]. For example, the theory allows a system of weakly interacting particles to be systematically approximated by the reference system of noninteracting particles and a power series. Zwanzig's theory led to several successful and popular applied perturbation theories, such as those for dense fluids [139, 140]. Decades later, researchers are essentially still equipped with only simulation and perturbation theory as tools when solving the vast majority of statistical mechanical systems [141].

In contrast to systems with weakly interacting particles, there is not yet a systematic approximation method for systems with strongly interacting particles. This method would be the low-temperature analog of high-temperature perturbation theory, where the reference system would consist of replacing the strong potentials with athermal rigid constraints. Traditionally, these systems are approximately solved using the rigid rotor and harmonic oscillator approximations [49]. Although anharmonic corrections can sometimes be determined for a system [49], a systematic approach for general potentials and systems is still unknown. Further, these sorts of approximations are not generally valid, such as in the case of molecular stretching, where sufficiently large applied force will cause the strong potentials to leave their harmonic regimes. This deviation is especially important when considering the mechanical response of a single polymer chain in the overstretching regime, which has been of recent interest in many models [67, 75, 78, 130, 142–144]. Currently, a heuristic method [62] is typically used and is somewhat successful, but a new approach that is explicitly compatible with the principles of statistical thermodynamics is more desirable.

Here, we present an asymptotically-correct method that approximates the statistical thermodynamics of systems containing strong potential energies of interaction. Our theory acts as the low-temperature analog of the high-temperature perturbation theory developed by Zwanzig, where the reference system here is a system with athermal rigid constraints. In Sec. 4.2 we develop the theory and obtain the asymptotic relation for the full system partition function and subsequently the Helmholtz free energy. In Sec. 4.3 we consider applications; in Sec. 4.3.1, we obtain the correct asymptotic expansion of the exact partition function for the three-dimensional harmonic oscillator. In Sec. 4.3.2, we discuss how the theory is applied in the case of molecular stretching, and we subsequently conclude.

4.2 Theoretical development

Consider the canonical partition function

$$\mathfrak{q} = \int d\Gamma_0 \int dX \, e^{-\beta H_0(\Gamma_0;X)} \, e^{-\beta U_1(X)}, \tag{4.1}$$

where H_0 is the Hamiltonian of the reference system, Γ_0 represents the degrees of freedom of the reference system, U_1 is the potential energy of interaction that is strong compared to thermal

energy $\mathfrak{b}T = 1/\beta$, and X represents the configurational degrees of freedom associated with the strong potential U_1. Let $\hat{X}$ represent the configuration where U_1 is minimized. As $\mathfrak{b}T$ becomes small (β becomes large) compared to the scale of U_1, the configurational degrees of freedom begin to become athermal rigid constraints with $X = \hat{X}$ and the full system behaves like the reference system. We will obtain an asymptotic approximation valid in this limit based on knowledge of the reference system and the strong potential.

4.2.1 Canonical partition function

Let U_1 be given by the sum over M independent degrees of freedom $X = \{x_1, \ldots, x_M\}$,

$$U_1(X) = \sum_{i=1}^{M} u_i(x_i), \tag{4.2}$$

and let the corresponding element of integration then be written as

$$dX = J(\mathbf{x}) \prod_{j=1}^{M} dx_j, \tag{4.3}$$

where $J(\mathbf{x})$ is the Jacobian that may be involved if the x_j are not Cartesian coordinates. The partition function in Eq. (4.1) can now be written as the nested set of integrals

$$\mathfrak{q} = \int d\Gamma_0 \int dx_1 \, e^{-\beta u_1(x_1)} \cdots \int dx_M \, e^{-\beta u_M(x_M)} J(\mathbf{x}) e^{-\beta H_0(\Gamma_0;\mathbf{x})}. \tag{4.4}$$

The more general case where $U_1(X)$ cannot be written in the form of Eq. (4.2) can also be treated by the following approach, but the mathematics are more complicated [145]. Conveniently, the form of $U_1(X)$ in Eq. (4.2) is valid for many physical systems of interest. Let $u_j(x_j) = \varepsilon_j \phi_j(x_j)$, where ε_j are the energy scales and $\phi_j(x_j)$ are the non-dimensional potential energy functions. Then $\mu_j = \beta\varepsilon_j$ are the non-dimensional energy scales and we rewrite Eq. (4.4) as

$$\mathfrak{q} = \int d\Gamma_0 \int dx_1 \, e^{-\mu_1 \phi_1(x_1)} \cdots \int dx_M \, e^{-\mu_M \phi_M(x_M)} J(\mathbf{x}) e^{-\beta H_0(\Gamma_0;\mathbf{x})}. \tag{4.5}$$

We now consider the limit in which the temperature is low compared to each energy scale, $\mu_j \to \infty$. We will denote derivatives of $\phi_j(x_j)$ using apostrophes, i.e. $\phi_j'(x_j) = d\phi_i'(x_j)/dx_j$. Let each $\phi_j(x_j)$ achieve a unique, hyperbolic minimum at $x_j = \hat{x}_j$, such that the Taylor series there is given by

$$\phi_j(x_j) = \phi_j(\hat{x}_j) + \frac{1}{2}\phi_j''(\hat{x}_j)\,(x_j - \hat{x}_j)^2 + \frac{1}{6}\phi_j'''(\hat{x}_j)\,(x_j - \hat{x}_j)^3 + \cdots, \tag{4.6}$$

and where the hyperbolic minimum requirement stipulates that $\phi_j''(\hat{x}_j) > 0$. Let $f(\Gamma_0;\mathbf{x}) \equiv J(\mathbf{x})e^{-\beta H_0(\Gamma_0;\mathbf{x})}$, and then consider the first integral within the nested set of integrals in Eq. (4.5),

$$I_M(\Gamma_0; x_1, \ldots, x_{M-1}) \equiv \int dx_M \, e^{-\mu_M \phi_M(x_M)} f(\Gamma_0;\mathbf{x}). \tag{4.7}$$

Laplace's method [146] for obtaining the asymptotic behavior of integrals shows that

$$I_M(\Gamma_0; x_1, \ldots, x_{M-1}) \sim \left\{ \sqrt{\frac{2\pi}{\mu_M \phi_M''(x_M)}}\, e^{-\mu_M \phi_M(x_M)} \left[f(\Gamma_0; \mathbf{x}) + \frac{g_M(\Gamma_0; \mathbf{x})}{\mu_M \phi_M''(x_M)} \right] \right\} \Bigg|_{x_M = \hat{x}_M} \tag{4.8}$$

as $\mu_M \to \infty$, where each function $g_j(\Gamma_0; \mathbf{x})$ is defined as

$$g_j(\Gamma_0; \mathbf{x}) \equiv \frac{1}{2} \frac{\partial^2 f(\Gamma_0; \mathbf{x})}{\partial x_j^2} - \frac{1}{2} \frac{\phi_j'''(x_j)}{\phi_j''(x_j)} \frac{\partial f(\Gamma_0; \mathbf{x})}{\partial x_j} - \frac{1}{8} \frac{\phi_j''''(x_j)}{\phi_j''(x_j)} f(\Gamma_0; \mathbf{x}) + \frac{5}{24} \left[\frac{\phi_j'''(x_j)}{\phi_j''(x_j)} \right]^2 f(\Gamma_0; \mathbf{x}). \tag{4.9}$$

We now repeat this process with the remaining $M-1$ integrals in Eq. (4.5) and disregard any terms that are $O(\mu_i^{-1}\mu_j^{-1})$, which allows us to obtain the asymptotic relation

$$\mathfrak{q} \sim \hat{\alpha}(\hat{\mathbf{x}}) \int d\Gamma_0 \left\{ f(\Gamma_0; \hat{\mathbf{x}}) + \sum_{j=1}^{M} \frac{g_j(\Gamma_0; \hat{\mathbf{x}})}{\mu_j \phi_j''(\hat{x}_j)} \right\} \quad \text{as } \mu_j \to \infty,\ j = 1, \ldots, M, \tag{4.10}$$

where the prefactor $\hat{\alpha}$ is defined as

$$\hat{\alpha}(\hat{\mathbf{x}}) \equiv \prod_{j=1}^{M} \sqrt{\frac{2\pi}{\mu_j \phi_j''(\hat{x}_j)}}\, e^{-\mu_j \phi_j(\hat{x}_j)}. \tag{4.11}$$

We will now complete the final integration, that over $d\Gamma_0$ in Eq. (4.10). The partition function of the reference system with rigid constraint coordinates $\mathbf{x}$ is given by

$$\mathfrak{q}_0(\hat{\mathbf{x}}) \equiv \int d\Gamma_0\, J(\hat{\mathbf{x}}) e^{-\beta H_0(\Gamma_0; \hat{\mathbf{x}})} \tag{4.12}$$

$$= \int d\Gamma_0\, f(\Gamma_0; \hat{\mathbf{x}}), \tag{4.13}$$

and each $g_j(\Gamma_0; \mathbf{x})$ in Eq. (4.9) is linear in $f(\Gamma_0; \mathbf{x})$ and its derivatives, allowing

$$\int d\Gamma_0\, g_j(\Gamma_0; \hat{\mathbf{x}}) = \frac{1}{2} \frac{\partial^2 \mathfrak{q}_0(\hat{\mathbf{x}})}{\partial x_j^2} - \frac{1}{2} \frac{\phi_j'''(\hat{x}_j)}{\phi_j''(\hat{x}_j)} \frac{\partial \mathfrak{q}_0(\hat{\mathbf{x}})}{\partial x_j} - \frac{1}{8} \frac{\phi_j''''(\hat{x}_j)}{\phi_j''(\hat{x}_j)} \mathfrak{q}_0(\hat{\mathbf{x}}) + \frac{5}{24} \left[\frac{\phi_j'''(\hat{x}_j)}{\phi_j''(\hat{x}_j)} \right]^2 \mathfrak{q}_0(\hat{\mathbf{x}}) \tag{4.14}$$

$$= \omega_j(\hat{\mathbf{x}}) \mathfrak{q}_0(\hat{\mathbf{x}}), \tag{4.15}$$

where we have defined

$$\omega_j(\hat{\mathbf{x}}) \equiv \frac{1}{2\mathfrak{q}_0(\hat{\mathbf{x}})} \left[\frac{\partial^2 \mathfrak{q}_0(\hat{\mathbf{x}})}{\partial x_j^2} - \frac{\phi_j'''(\hat{x}_j)}{\phi_j''(\hat{x}_j)} \frac{\partial \mathfrak{q}_0(\hat{\mathbf{x}})}{\partial x_j} \right] - \frac{1}{8} \frac{\phi_j''''(\hat{x}_j)}{\phi_j''(\hat{x}_j)} + \frac{5}{24} \left[\frac{\phi_j'''(\hat{x}_j)}{\phi_j''(\hat{x}_j)} \right]^2, \tag{4.16}$$

so we may now rewrite Eq. (4.10) as

$$\mathfrak{q} \sim \hat{\alpha}(\hat{\mathbf{x}}) \mathfrak{q}_0(\hat{\mathbf{x}}) \left[1 + \sum_{j=1}^{M} \frac{\omega_j(\hat{\mathbf{x}})}{\mu_j \phi_j''(\hat{x}_j)} \right]. \tag{4.17}$$

Returning to the original units, the results are summarized as follows: the full system partition function $\mathfrak{q}(\hat{\mathbf{x}})$ is asymptotically approximated by

$$\mathfrak{q} \sim \hat{\alpha}(\hat{\mathbf{x}})\mathfrak{q}_0(\hat{\mathbf{x}}) \left[1 + \sum_{j=1}^{M} \frac{\omega_j(\hat{\mathbf{x}})}{\beta u_j''(\hat{x}_j)}\right] \quad \text{as } \beta\varepsilon_j \to \infty,\ j = 1, \ldots, M, \tag{4.18}$$

where the $\hat{x}_j$ are the locations of the minima of the potential energies $u_j(\hat{x}_j)$. The full system partition function $\mathfrak{q}$ is a product of the prefactor $\hat{\alpha}$ given by

$$\hat{\alpha}(\hat{\mathbf{x}}) = \prod_{j=1}^{M} \sqrt{\frac{2\pi}{\beta u_j''(\hat{x}_j)}}\, e^{-\beta u_j(\hat{x}_j)}, \tag{4.19}$$

the reference system partition function $\mathfrak{q}_0(\hat{\mathbf{x}})$ given by Eq. (4.12), and the corrective term in square brackets, where each $\omega_j(\hat{\mathbf{x}})$ is given by

$$\omega_j(\hat{\mathbf{x}}) = \frac{1}{2\mathfrak{q}_0(\hat{\mathbf{x}})}\left[\frac{\partial^2 \mathfrak{q}_0(\hat{\mathbf{x}})}{\partial x_j^2} - \frac{u_j'''(\hat{x}_j)}{u_j''(\hat{x}_j)}\frac{\partial \mathfrak{q}_0(\hat{\mathbf{x}})}{\partial x_j}\right] - \frac{1}{8}\frac{u_j''''(\hat{x}_j)}{u_j''(\hat{x}_j)} + \frac{5}{24}\left[\frac{u_j'''(\hat{x}_j)}{u_j''(\hat{x}_j)}\right]^2. \tag{4.20}$$

We now have an asymptotic expansion for the full system partition function in terms of that of the reference system, and the strong potentials and their derivatives evaluated at their minima.

4.2.2 Helmholtz free energy

The Helmholtz free energy [49] is given by

$$\mathscr{A} = -\mathfrak{b}T \ln \mathfrak{q}, \tag{4.21}$$

so the Helmholtz free energy of the reference system is then $\mathscr{A}_0 = -\mathfrak{b}T \ln \mathfrak{q}_0$. In taking the logarithm of Eq. (4.18) we again utilize $\beta\varepsilon_j \to \infty$ in order to use the Mercator series to asymptotically approximate the logarithm of the term in the brackets:

$$\ln\left[1 + \sum_{j=1}^{M} \frac{\omega_j(\hat{\mathbf{x}})}{\beta u_j''(\hat{x}_j)}\right] \sim \sum_{j=1}^{M} \frac{\omega_j(\hat{\mathbf{x}})}{\beta u_j''(\hat{x}_j)}. \tag{4.22}$$

We may rewrite Eq. (4.20) in terms of $\mathscr{A}_0$ as

$$\omega_j(\hat{\mathbf{x}}) = \frac{1}{2}\left[\left(\frac{\partial \beta\mathscr{A}_0(\hat{\mathbf{x}})}{\partial x_j}\right)^2 - \frac{\partial^2 \beta\mathscr{A}_0(\hat{\mathbf{x}})}{\partial x_j^2} + \frac{u_j'''(\hat{x}_j)}{u_j''(\hat{x}_j)}\frac{\partial \beta\mathscr{A}_0(\hat{\mathbf{x}})}{\partial x_j}\right] - \frac{1}{8}\frac{u_j''''(\hat{x}_j)}{u_j''(\hat{x}_j)} + \frac{5}{24}\left[\frac{u_j'''(\hat{x}_j)}{u_j''(\hat{x}_j)}\right]^2, \tag{4.23}$$

and we recall $\sum_j u_j(\hat{x}_j) = U_1(\hat{\mathbf{x}})$. The resulting asymptotic relation for the Helmholtz free energy of the full system $\mathscr{A}$ is then

$$\mathscr{A} \sim \mathscr{A}_0(\hat{\mathbf{x}}) + U_1(\hat{\mathbf{x}}) + \mathfrak{b}T \sum_{j=1}^{M}\left[\frac{1}{2}\ln\left(\frac{\beta u_j''(\hat{x}_j)}{2\pi}\right) - \frac{\omega_j(\hat{\mathbf{x}})}{\beta u_j''(\hat{x}_j)}\right]. \tag{4.24}$$

Eq. (4.24) shows that the Helmholtz free energy of the full system can be asymptotically approximated by the sum of the Helmholtz free energy of the reference system, the minimized potential energy from the stiff degrees of freedom, a series of terms due to the logarithm of the prefactor, and a series of small corrections.

4.3 Applications

Here we consider applications of the asymptotic theory developed in Sec. 4.2. We first consider the three-dimensional harmonic oscillator and show that the theory is successful in obtaining asymptotic approximations of the exact result, which can be obtained analytically in this case. We then discuss how the theory is adjusted for the force-applied ensemble encountered when stretching molecules.

4.3.1 Three-dimensional harmonic oscillator

Consider two masses (m_1 and m_2) connected by a stiff harmonic spring with stiffness k and rest length ℓ_b. The classical Hamiltonian of this three-dimensional harmonic oscillator is given by

$$H(p,q) = \underbrace{\frac{p_1^2}{2m_1} + \frac{p_2^2}{2m_2}}_{H_0} + \underbrace{\frac{k_b}{2}\left(\|\mathbf{q}_1 - \mathbf{q}_2\|_2 - \ell_b\right)^2}_{U_1}. \tag{4.25}$$

As k_b becomes large, the spring begins to act as an athermal rigid constraint. This means that the kinetic energy is the reference system Hamiltonian H_0, the spring potential energy is the stiff potential U_1, the distance between the masses $\|\mathbf{q}_1 - \mathbf{q}_2\|_2$ is the stiff degree of freedom x, and the reference system has $\hat{x} = \ell_b$. The nondimensional Hamiltonian,

$$\beta H(p,q) = \frac{\beta p_1^2}{2m_1} + \frac{\beta p_2^2}{2m_2} + \frac{\kappa}{2}\left(\frac{\|\mathbf{q}_1 - \mathbf{q}_2\|_2}{\ell_b} - 1\right)^2, \tag{4.26}$$

makes it clear that we are considering when the nondimensional stiffness $\kappa \equiv \beta k_b \ell_b^2$ is large, i.e. when thermal energy is low compared to the energy scale of the spring $k\ell_b^2$. The partition function in Eq. (4.1) can be computed exactly in this case. The momentum partition function $\mathfrak{q}_{\text{mom}}$ is that of two free masses, and a factor of the volume V arises from rigid body translation [49]. The relative configuration integral $\mathfrak{q}_{\text{con}}$ involves the configurational degrees of freedom via rotation and vibration. The result is

$$\mathfrak{q} = \underbrace{4\pi\ell_b^3\left\{\frac{e^{-\kappa/2}}{\kappa} + \sqrt{\frac{\pi}{2\kappa}}\left(1 + \frac{1}{\kappa}\right)\left[2 - \operatorname{erfc}\left(\sqrt{\frac{\kappa}{2}}\right)\right]\right\}}_{\mathfrak{q}_{\text{con}}}\underbrace{\left(\frac{2\pi\sqrt{m_1 m_2}}{\beta}\right)^3}_{\mathfrak{q}_{\text{mom}}} V, \tag{4.27}$$

where $\operatorname{erfc}(z) = 1 - \operatorname{erf}(z)$ is the complementary error function. The low-temperature theory is now applied in order to retrieve an asymptotic approximation of this partition function. The reference system has $x = \hat{x} = \ell_b$ held fixed, resulting in the partition function

$$\mathfrak{q}_0(\ell_b) = 4\pi\ell_b^2 \mathfrak{q}_{\text{mom}} V. \tag{4.28}$$

Note that we have not constrained the momentum degrees of freedom in the fashion that would be consistent with the configurational constraint, as was done when deriving the general theory in Sec. 4.2. The prefactor $\hat{\alpha}(\ell_b)$ in Eq. (4.19), after using $u(\ell_b) = 0$ and $u'' = k$, is obtained:

$$\hat{\alpha}(\ell_b) = \sqrt{\frac{2\pi}{\beta k}} = \ell_b \sqrt{\frac{2\pi}{\kappa}}. \tag{4.29}$$

The correction factor $\omega(\ell_b)$ in Eq. (4.20) involves only a single term since the harmonic potential has only two nonzero derivatives:

$$\omega(\ell_b) = \frac{1}{2\mathfrak{q}_0(\ell_b)} \frac{\partial^2 \mathfrak{q}_0(\ell_b)}{\partial \ell_b^2} = \frac{1}{\ell_b^2}. \tag{4.30}$$

The asymptotic approximation of the full system partition function, given by Eq. (4.18) and obtained by combining Eqs. (4.28)–(4.30), is then

$$\mathfrak{q} \sim 4\pi \ell_b^3 \sqrt{\frac{2\pi}{\kappa}} \left(1 + \frac{1}{\kappa}\right) \mathfrak{q}_{\text{mom}} V \quad \text{for } \kappa \gg 1. \tag{4.31}$$

In order to compare this more directly to the exact result in Eq. (4.27), the asymptotic relation [147] for the complementary error function is necessary,

$$\text{erfc}\,(x) \sim \frac{e^{-x^2}}{x\sqrt{\pi}} \sum_{n=0}^{\infty} (-1)^n \frac{(2n-1)!!}{(2x^2)^n} \quad \text{for } x \gg 1. \tag{4.32}$$

Substitution of Eq. (4.32) into Eq. (4.27) yields the asymptotic relation

$$\mathfrak{q} \sim 4\pi \ell_b^3 \sqrt{\frac{2\pi}{\kappa}} \left[1 + \frac{1}{\kappa} + h(\kappa)\right] \mathfrak{q}_{\text{mom}} V \quad \text{for } \kappa \gg 1, \tag{4.33}$$

where the function $h(\kappa)$ is given by

$$h(\kappa) \equiv \frac{e^{-\kappa/2}}{\sqrt{2\pi\kappa}} \left[1 - \left(1 + \frac{1}{\kappa}\right) \sum_{n=0}^{\infty} (-1)^n \frac{(2n-1)!!}{\kappa^n}\right]. \tag{4.34}$$

Since $h(\kappa) < O(\kappa^{-1})$, the asymptotic theory has correctly obtained in Eq. (4.31) an asymptotic expansion of the exact result in Eq. (4.27). Since $h(\kappa)$ is transcendentally small [146], which means that $h(\kappa)$ shrinks faster than any power of κ^{-1}, the asymptotic theory will not obtain any additional corrections in this case. The leading order approximation in Eq. (4.31),

$$\mathfrak{q} \sim 4\pi \ell_b^3 \sqrt{\frac{2\pi}{\kappa}}\, \mathfrak{q}_{\text{mom}} V \quad \text{for } \kappa \gg 1, \tag{4.35}$$

is the partition function when applying the rigid-rotor-harmonic-oscillator approximation [49]. The correction to this approximation provided by Eq. (4.31) is then correcting for rotation-vibration coupling. If the spring were anharmonic, Eq. (4.31) would have additional terms correcting for anharmonicity as well. Finally, note that taking $\kappa \to \infty$ explicitly within either result, Eq. (4.27) or Eq. (4.31), obtains zero rather than the reference system result in Eq. (4.28). This illustrates the necessity of obtaining an asymptotic theory from the original partition function using Laplace's method, rather than building one up from the reference system partition function.

4.3.2 Molecular stretching

For isotensional molecular stretching experiments, we follow the same procedure in Sec. 4.2 but utilize the total potential energy $\mathit{\Pi} \equiv U - \mathbf{f} \cdot (\mathbf{q}_2 - \mathbf{q}_1)$ in place of U, where $\mathbf{f}$ is the external force applied equally and oppositely at the atoms labeled 1 and 2. The relevant partition function is now the Gibbs (isotensional) ensemble partition function [125]

$$\mathfrak{z} = \int d\Gamma_0 \int dX \, e^{-\beta H_0(\Gamma_0;X)} \, e^{-\beta U_1(X)} \, e^{\beta \mathbf{f}\cdot(\mathbf{q}_2-\mathbf{q}_1)}. \tag{4.36}$$

When the applied force is small compared to the stiff potentials, $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\mathbf{f})$ can be neglected and approximations becomes simpler, i.e. when the minimum of $\mathit{\Pi}$ and U approximately coincide. Generally, though, the applied force causes the $\hat{\boldsymbol{x}}$ to shift from their values minimizing U to those minimizing $\mathit{\Pi}$. The general theory in Sec. 4.2 can still be applied in this case after lumping $e^{\beta \mathbf{f}\cdot(\mathbf{q}_2-\mathbf{q}_1)}$ into $f(\Gamma_0; \mathbf{x})$, but $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\mathbf{f})$ causes many complicated terms to be encountered when calculating the mechanical response, $\partial(\mathfrak{b} T \ln \mathfrak{z})/\partial \mathbf{f}$. The isometric (Helmholtz) ensemble is much more complicated, as it involves computing derivatives of the Dirac delta function within the integrand, and the reference system is almost always analytically intractable [125].

4.4 Conclusions

We have developed an asymptotically-correct theory approximating the statistical thermodynamics of systems containing potential energies of interaction that are strong compared to available thermal energy. This theory is the low-temperature analog of the high-temperature perturbation theory developed by Zwanzig in 1954. We have demonstrated that the theory obtains the correct asymptotic expansion of exact results using a simple example. In the future, we will apply this theory to molecular stretching experiments, especially those concerning single polymer chains, and to general molecular simulation.

Chapter 5

Chain breaking in the statistical mechanical constitutive theory of polymer networks

Chapter 5 and Appendix C are adapted from: M. R. Buche and M. N. Silberstein. Chain breaking in the statistical mechanical constitutive theory of polymer networks. arXiv:2104.08866 [cond-mat.soft]. Accepted at J. Mech. Phys. Solids (2021). The `Python` implementation of the model [149] is available on GitHub and PyPI (`pip install chain_breaking_polymer_networks`).

Elastomers are used in a wide range of applications because of their large strain to failure, low density, and tailorable stiffness and toughness. The mechanical behavior of elastomers derives mainly from the entropic elasticity of the underlying network of polymer chains. Elastomers under large deformation experience bonds breaking within the backbone chains that constitute the polymer network. This breaking of chains damages the network, can lead to material failure, and can be utilized as an energy dissipation mechanism. In the case of reversible bonds, broken chains may reform and heal the damage in the network. If the reversible bonds are dynamic, chains constantly break and reform and create a transient network. A fundamental constitutive theory is developed to model the mechanics of these polymer networks. A statistical mechanical derivation is conducted to yield a framework that takes in an arbitrary single-chain model (a Hamiltonian) and outputs the following: the single-chain mechanical response, the breaking and reforming kinetics, the equilibrium distribution of chains in the network, and the partial differential equations governing the deformation-coupled network evolution. This statistical mechanical framework is then brought into the continuum scale by using macroscopic thermodynamic constitutive theory to obtain a constitutive relation for the Cauchy stress. The potential-supplemented freely jointed chain (uFJC) model is introduced, and a parametric study of its mechanical response and breaking kinetics is provided. This single-chain model is then implemented within the constitutive framework, which we specialize and apply in two exemplary cases: the mechanical response and irreversible breakdown of a multinetwork elastomer, and the mechanical response of a dual crosslink gel. After providing a parametric study of the general constitutive model, we apply it to a hydrogel with reversible metal-coordination crosslinks. In several cases, we find that the breakdown of the network causes secondary physical mechanisms to become important and inhibit the accuracy of our model. We then discuss these mechanisms and indicate how our existing framework can be adjusted to incorporate them in the future.

5.1 Introduction

Bulk elastomer materials often consist of many single polymer chains crosslinked together to form a network of chains. Elastomers tend to be soft, elastic, and highly stretchable, due to the entropic elasticity of the chains above the glass transition temperature [1]. From large tires and small seals to soft robotics, elastomers are utilized in a wide variety of applications due to their resilience. However, as the elastomer network is deformed more extensively, bonds begin to stretch and then chains begin to break. While chain breaking can result in material failure for simpler elastomers, many advanced elastomers are cleverly designed to take advantage of chain breaking. Firstly, elastomers may be strengthened, toughened, and made more stretchable through the incorporation of one or more sacrificial networks that begin to break down irreversibly when stretched, dissipating energy while secondary networks maintain the integrity of the material. The sacrificial network is often embrittled as a swollen gel [2], but could also be pre-stretched using the secondary networks [6] or even designed without need for pre-stretching [7]. The breaking in the sacrificial network may involve some additional functionality, such as mechanoluminescence [11] and recently, chain-lengthening [16]. Secondly, elastomers may utilize a range of reversible bonds in order to allow chains to reform after they have been broken. This reversible breaking allows similar properties as the irreversibly-breaking cases, such as high stretchability and toughness, while also allowing new properties such as self-healing [18]. Alginate-based gels contain ionic crosslinking that breaks reversibly as the polymer is deformed, increasing toughness while enabling both self-healing and shape-memory [20]. Metal-ligand interactions, which are inherently tunable [21], when used as crosslinks provide a precise method to control polymer mechanical properties via the simple addition of neutral ligands [28]. Dynamic reversible bonds may also be utilized: polymers with associative bond exchange reactions like vitrimers behave as an elastic solid at low temperatures while flowing more similarly to a viscous fluid at high temperatures, all the while maintaining the integrity of the network [29]. Some of these covalent adaptable networks use light as a stimulus in order to trigger the dynamic bonds to permanently alter the material shape [35]. Utilizing a combination of interactions is also useful, such as the combination of permanent covalent bonds and transient physical bonds in dual-crosslink gels [45]. Overall, the mechanical properties of these materials tend to be highly nonlinear, rate-dependent, and sensitive to changes in their chemistry. Therefore, a truly physical constitutive model that accounts for the complexities of chains breaking in a network is desirable to maximize both predictive power and fundamental understanding for the wide range of available chemistries and combinations.

There are a variety of physically-based constitutive models for polymer networks that incorporate chain breaking, frequently using the freely-jointed chain (FJC) single-chain statistical mechanical model [50]. A portion of these models are targeted towards the mechanical response of permanently-crosslinked elastomers, where chains or crosslinks are considered to break suddenly and irreversibly [62, 67]. This approach has been successfully applied when modeling the irreversible damage or fracture of polymer networks [75, 76, 78, 142, 143]. Additionally, irreversible breaking has been incorporated into many models for multinetwork elastomers and gels [74, 150–153], sometimes addressing a particular phenomenon such as necking instability [81, 154, 155]. Another portion of these physically-based constitutive models tends to be specialized for transient networks enabled by highly dynamic bonds. Transient network theory is typically attributed to Tanaka and Edwards [59, 156], which is built upon foundational work from the 1940s to the 1990s [58, 157–159]. Recent development has been driven by Vernerey et al. ([60, 160]), and has lead to successful application in

fracture scenarios [161, 162]. Other constitutive models for polymers with dynamic bonds combine physically-based insights with continuum-level constitutive laws, such those for the mechanics of dual-crosslink gels and of networks with temperature-sensitive dynamic covalent bonds [163–172].

Although these existing physically-based constitutive models perform well for a range of materials, any one of them lacks widespread applicability. In this manuscript we present a statistical mechanical derivation that can bridge these models, where these models are special cases of the general model. This derivation will yield, from an arbitrary single-chain model Hamiltonian, (1) the single-chain mechanical response, (2) the equilibrium distribution of chains in the network, and (3) the mechanically-dependent kinetics of chain breaking and reforming. While the first two connections have previously been established [125], the force-dependent kinetics have not yet been directly connected to the statistical mechanics of the single-chain model. With limited additional assumptions, this statistical mechanical foundation will then be used to formulate macroscopic constitutive relations entirely informed by an arbitrary single-chain model. This meticulous procedure carrying the underlying statistical mechanics through to the macroscale has many inherent benefits, such as consistency between the equilibrium configuration obtained by statistical thermodynamics and that obtained macroscopically, and the automatic satisfaction of the second law of thermodynamics.

This manuscript is organized as follows: In Sec. 5.2.1, beginning from the fundamentals of nonequilibrium statistical mechanics, we obtain evolution equations for the probability of finding an intact chain at a certain end-to-end vector within the network as a function of time. This derivation results in the single-chain mechanical behavior, equilibrium distributions of chains in the network, and chemical kinetics function of chain breaking/reforming all in terms of the single-chain partition functions. In Sec. 5.2.2, our statistical theory is brought into the macroscale through the formulation of the Helmholtz free energy of the incompressible network. After prescribing an affine deformation, a second-law analysis then results in the constitutive relation for the Cauchy stress entirely in terms of the intact chain distribution and single-chain mechanical response, where the residual inequality is shown to be arbitrarily satisfied. With the general theory complete, in Sec. 5.3 we introduce and implement the uFJC single-chain model: the freely jointed chain (FJC) model supplemented to have stiff, but flexible links with some potential energy u. We utilize the Morse potential for u and study various single-chain functions over a range of parameters. Additionally, we present an original exact solution for the evolving intact chain probability distribution. In Sec. 5.4, we consider several special cases from the limiting behavior of our model and apply them to exemplary polymers from the literature. We then study the general behavior of the model, drawing conclusions in comparison to these simpler special cases and examining the results over a range of single-chain parameters. Afterward we apply the general model to another polymers from the literature. Finally, we discuss the successes and shortcomings of our model and propose improvements for future work. We have implemented our model in a `Python` package and made it available on `GitHub` and `PyPI` to facilitate both adoption and adaption by interested readers [149].

5.2 General theory

5.2.1 Statistical mechanics

We consider a classical, canonical statistical mechanical ensemble of noninteracting polymer chains that may break/reform via multiple reaction pathways. Beginning from the general nonequilibrium formalism of Zwanzig [173], we derive a general evolution law for the probability distribution of intact chains at a certain end-to-end vector. We then apply the assumptions of transition state theory to obtain a simpler evolution law that does not require knowledge of the phase space distribution function. After making some assumptions about the behavior of broken chains, we obtain conservation requirements for all chains in the network, as well as an evolution law for the probability of each broken chain species.

Phase space principles

In classical statistical mechanics [49], the phase space distribution function $f(\Gamma;t)$ provides the probability density at time t that the system is in the state, denoted by Γ, with the atomic positions $\mathbf{q}$ and momenta $\mathbf{p}$. We may calculate the macroscopically observable value $\Phi(t)$ of some phase space function $\phi(\Gamma)$, which is the ensemble average of $\phi(\Gamma)$, or $\langle\phi\rangle$, as

$$\Phi(t) = \langle\phi\rangle \equiv \int \cdots \int f(\Gamma;t)\phi(\Gamma)\, d\Gamma. \tag{5.1}$$

In order to find $f(\Gamma;t)$, we integrate the evolution equation for $f(\Gamma;t)$, the Liouville equation

$$\frac{\partial f}{\partial t} = (-\mathscr{L})\, f = \left(\frac{\partial H}{\partial \mathbf{q}} \cdot \frac{\partial}{\partial \mathbf{p}} - \frac{\partial H}{\partial \mathbf{p}} \cdot \frac{\partial}{\partial \mathbf{q}}\right) f, \tag{5.2}$$

with $\mathscr{L}$ being the Liouville operator. Since $f(\Gamma;t)$ does not evolve at equilibrium, $\mathscr{L} f^{\mathrm{eq}} = 0$ for the equilibrium phase space distribution function $f^{\mathrm{eq}}(\Gamma)$. This equilibrium distribution is the Boltzmann distribution

$$f^{\mathrm{eq}}(\Gamma) = \frac{e^{-\beta H(\Gamma)}}{\mathfrak{q}}, \tag{5.3}$$

where $H(\Gamma)$ is the Hamiltonian of the system, $\beta = 1/\mathfrak{b}T$ is the inverse temperature, $\mathfrak{b}$ is Boltzmann's constant, T is the temperature, and

$$\mathfrak{q} = \int e^{-\beta H(\Gamma)}\, d\Gamma \tag{5.4}$$

is the canonical partition function for the system. Note that we neglect the factors of Planck's constant h that would nondimensionalize the partition functions, but this has no effect on our classically-obtained results. The equilibrium ensemble average of some phase space function $\phi(\Gamma)$ is the time-independent average

$$\Phi^{\mathrm{eq}} = \langle\phi\rangle^{\mathrm{eq}} \equiv \frac{1}{\mathfrak{q}} \int \cdots \int e^{-\beta H(\Gamma)} \phi(\Gamma)\, d\Gamma. \tag{5.5}$$

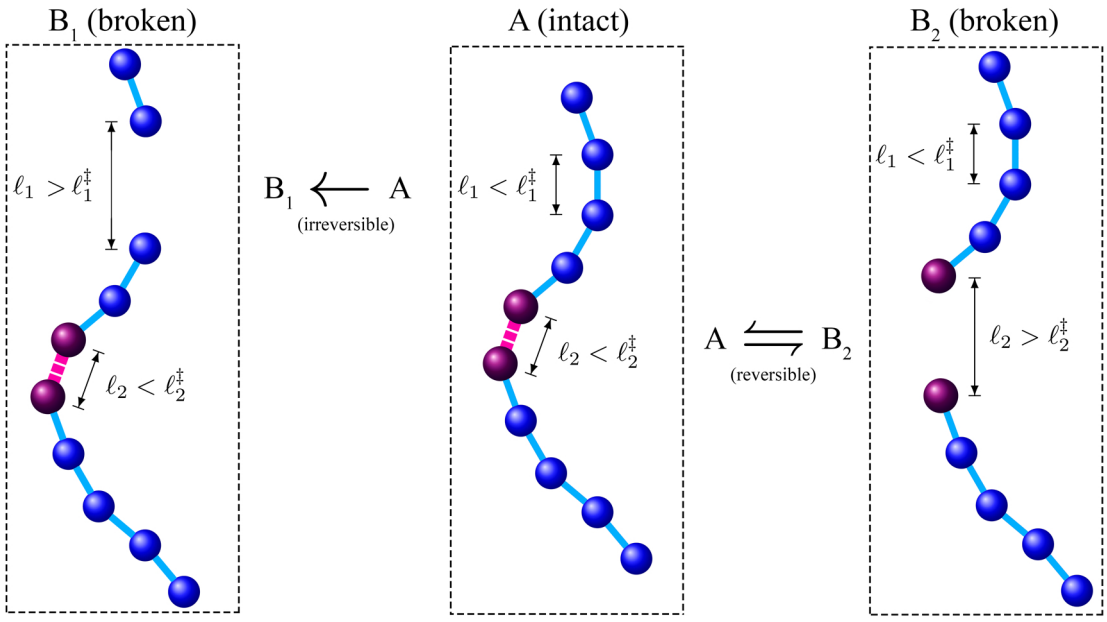


Figure 5.1: Illustration of an intact chain (A) with various links of instantaneous length ℓ_i that may act as reaction coordinates. In this illustration, the links may break irreversibly (see B_1) or reversibly (see B_2) when $\ell_i > \ell_i^\ddagger$.

While solving for $f(\Gamma;t)$ would give us full knowledge of the system, it is impractical for our purposes since Γ constitutes far too many state variables. Fortunately, the macroscopic observables of interest in our case only require knowledge of a subset of the probability distribution of the phase space variables. Specifically, we will only need the probability density distribution $P_A(\boldsymbol{\xi};t)$ of intact chains with end-to-end vector $\boldsymbol{\xi}$ at time t to calculate the macroscopic stress. In order to track $P_A(\boldsymbol{\xi};t)$, we will also need to consider the analogous distributions of broken chains $P_{B_j}(\boldsymbol{\xi};t)$, where j denotes that the chain has broken via the jth pathway; this is illustrated in Fig. 5.1. In the following section, we will write $P_A(\boldsymbol{\xi};t)$ in terms of $f(\Gamma;t)$, and subsequently utilize this relation and the evolution equation for $f(\Gamma;t)$ to obtain the evolution equation for $P_A(\boldsymbol{\xi};t)$.

Evolution of intact chains

The probability density distribution $P(\boldsymbol{\xi};t)$ of chains with end-to-end vector $\boldsymbol{\xi}$ at time t is given by the ensemble average

$$P(\boldsymbol{\xi};t) = \left\langle \delta^3 \left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right] \right\rangle, \tag{5.6}$$

where δ is the Dirac delta function and $\mathbf{R}(\Gamma)$ is the end-to-end vector of the chain as a function of phase space variables Γ. This ensemble average effectively results in an integration of $f(\Gamma;t)$ over the portion of the phase space where the end-to-end vector of the chain is $\boldsymbol{\xi}$. A subset of the phase space variables Γ are the reaction coordinates ℓ that determine whether a chain is intact or broken. We consider M of these reaction coordinates, where ℓ_i is then the ith reaction coordinate. A chain is intact (A) if all $\ell_i < \ell_i^\ddagger$, while a chain is broken (B) if any $\ell_i > \ell_i^\ddagger$. The reaction coordinates ℓ_i then create regions of the phase space where chains are intact ($\mathcal{A}$) or broken ($\mathcal{B}_i$). The boundaries separating these two regions are the transition states with $\ell_i = \ell_i^\ddagger$. Using Heaviside step functions

Θ, we may then write the probability density distribution $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ that a polymer chain is both intact and at end-to-end vector $\boldsymbol{\xi}$ at time t as

$$P_{\mathrm{A}}(\boldsymbol{\xi};t) = \left\langle \delta^3\left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right] \prod_{i=1}^{M} \Theta\left(\ell_i^{\ddagger} - \ell_i\right) \right\rangle . \tag{5.7}$$

We differentiate Eq. (5.7) with respect to time t in order to produce the evolution equation for $P_{\mathrm{A}}(\boldsymbol{\xi};t)$. Using Eq. (5.2) and the properties of the Liouville operator $\mathscr{L}$ noted by Zwanzig [173], we see that $\frac{d}{dt}\langle\phi\rangle = \langle\mathscr{L}\phi\rangle$ for some phase space function $\phi(\Gamma)$, so the evolution equation for $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ is

$$\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial t} = \left\langle \mathscr{L}\left\{ \delta^3\left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right] \prod_{i=1}^{M} \Theta\left(\ell_i^{\ddagger} - \ell_i\right) \right\} \right\rangle . \tag{5.8}$$

We will work from Eq. (5.8) to a readily usable form of the evolution equation for $P_{\mathrm{A}}(\boldsymbol{\xi};t)$, beginning with expansion using the product rule,

$$\begin{aligned} \frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial t} = &\left\langle \mathscr{L}\left\{\delta^3\left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right]\right\} \prod_{i=1}^{M} \Theta\left(\ell_i^{\ddagger} - \ell_i\right) \right\rangle \\ &+ \sum_{j=1}^{M} \left\langle \delta^3\left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right] \mathscr{L}\left\{\Theta\left(\ell_j^{\ddagger} - \ell_j\right)\right\} \prod_{\substack{i=1 \\ i\neq j}}^{M} \Theta\left(\ell_i^{\ddagger} - \ell_i\right) \right\rangle . \end{aligned} \tag{5.9}$$

We consider the first term in Eq. (5.9) where the Liouville operator $\mathscr{L}$ acts on the delta function. In effect, this term accounts for the evolution of $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ due to $\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t)$, the average rate of change of the end-to-end vector of an intact chain currently having end-to-end vector $\boldsymbol{\xi}$,

$$\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t) \equiv \left\langle \dot{\mathbf{R}}(\Gamma)\, \delta^3\left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right] \prod_{i=1}^{M} \Theta\left(\ell_i^{\ddagger} - \ell_i\right) \right\rangle . \tag{5.10}$$

As detailed in general by Zwanzig [174], the first term in Eq. (5.9) can then be written as

$$\left\langle \mathscr{L}\left\{\delta^3\left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right]\right\} \prod_{i=1}^{M} \Theta\left(\ell_i^{\ddagger} - \ell_i\right) \right\rangle = -\frac{\partial}{\partial \boldsymbol{\xi}} \cdot \left[\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t) P_{\mathrm{A}}(\boldsymbol{\xi};t)\right] . \tag{5.11}$$

We now consider the second set of terms in Eq. (5.9) where the Liouville operator $\mathscr{L}$ acts on each of the step functions $\Theta(\ell_j^{\ddagger} - \ell_j)$. The derivative of the step function is the delta function, and $\mathscr{L}$ acting on the coordinates ℓ_j produces the velocities p_j/m_j, such that

$$\mathscr{L}\left\{\Theta\left(\ell_j^{\ddagger} - \ell_j\right)\right\} = -\frac{p_j}{m_j}\, \delta\left(\ell_j^{\ddagger} - \ell_j\right) . \tag{5.12}$$

The summands of Eq. (5.9) are therefore the expected values of the velocity $-p_j/m_j$ along the jth reaction coordinate ℓ_j for a chain at end-to-end vector $\boldsymbol{\xi}$ in the jth transition state $\ell_j^{\ddagger}$. These summands are understood as the evolution of $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ due to flow across each transition state

boundary. If we use step functions to split these flows into the forward $\mathcal{A} \to \mathcal{B}_j$ and reverse $\mathcal{B}_j \to \mathcal{A}$ reactions, with respective rates

$$\mathcal{R}'_j(\boldsymbol{\xi};t) = \left\langle \frac{p_j}{m_j}\,\Theta(p_j)\delta^3\left[\mathbf{R}(\Gamma)-\boldsymbol{\xi}\right]\delta\left(\ell_j^\ddagger-\ell_j\right)\prod_{\substack{i=1\\i\neq j}}^{M}\Theta\left(\ell_i^\ddagger-\ell_i\right)\right\rangle, \tag{5.13}$$

$$\mathcal{R}''_j(\boldsymbol{\xi};t) = \left\langle -\frac{p_j}{m_j}\,\Theta(-p_j)\delta^3\left[\mathbf{R}(\Gamma)-\boldsymbol{\xi}\right]\delta\left(\ell_j^\ddagger-\ell_j\right)\prod_{\substack{i=1\\i\neq j}}^{M}\Theta\left(\ell_i^\ddagger-\ell_i\right)\right\rangle, \tag{5.14}$$

then the final evolution law for the distribution of intact chains is

$$\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial t} = \sum_{j=1}^{M}\mathcal{R}''_j(\boldsymbol{\xi};t) - \sum_{j=1}^{M}\mathcal{R}'_j(\boldsymbol{\xi};t) - \frac{\partial}{\partial\boldsymbol{\xi}}\cdot\left[\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t)P_{\mathrm{A}}(\boldsymbol{\xi};t)\right]. \tag{5.15}$$

We have now obtained the general evolution law for the probability distribution of intact chains with a certain end-to-end vector $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ in Eq. (5.15). It remains an issue, however, that we require knowledge of the phase space distribution function $f(\Gamma;t)$, which is necessary to compute the ensemble averages for the reaction rates Eqs. (5.13) and (5.14). Consequently, we now utilize transition state theory in order to avoid computing these ensemble averages and therefore eliminate $f(\Gamma;t)$ from the evolution equation entirely.

Transition state theory

Our derivation so far is general for noninteracting chains, but we now make our first approximation. Let the phase space function $f(\Gamma;t)$ maintain a local equilibrium in each species' region [173], such that we may take $f(\Gamma;t)$ in the phase space region $\mathcal{A}$ to be approximately

$$f(\Gamma;t) \approx \frac{P_{\mathrm{A}}(\boldsymbol{\xi};t)}{P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})}\,f^{\mathrm{eq}}(\Gamma) \quad \text{in region } \mathcal{A}, \tag{5.16}$$

where $P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})$ is the equilibrium distribution of the end-to-end vectors of intact chains, and $f^{\mathrm{eq}}(\Gamma) = e^{-\beta H(\Gamma)}/\mathfrak{q}$ from Eq. (5.3). This is representative of the fact that degrees of freedom not involved with breaking the chain (such as bond rotation) attain equilibrium much more quickly than those degrees of freedom associated with breaking the chain (i.e. bond breaking). Consequently, our transition state theory assumption now prevents us from considering cases where the timescales of bond breaking become close to those of intact chain dynamics.

The distribution of the end-to-end vectors of chains that have broken via the jth reaction pathway $P_{\mathrm{B}_j}(\boldsymbol{\xi};t)$ can be written by flipping the sign within the jth step function in Eq. (5.7),

$$P_{\mathrm{B}_j}(\boldsymbol{\xi};t) = \left\langle \delta^3\left[\mathbf{R}(\Gamma)-\boldsymbol{\xi}\right]\Theta\left(\ell_j-\ell_j^\ddagger\right)\prod_{\substack{i=1\\i\neq j}}^{M}\Theta\left(\ell_i^\ddagger-\ell_i\right)\right\rangle. \tag{5.17}$$

We may similarly approximate $f(\Gamma;t)$ in each broken chain phase space region as

$$f(\Gamma;t) \approx \frac{P_{\mathrm{B}_j}(\boldsymbol{\xi};t)}{P^{\mathrm{eq}}_{\mathrm{B}_j}(\boldsymbol{\xi})}\, f^{\mathrm{eq}}(\Gamma) \quad \text{in each region } \mathcal{B}_j, \tag{5.18}$$

where $P^{\mathrm{eq}}_{\mathrm{B}_j}(\boldsymbol{\xi})$ is the equilibrium distribution of the end-to-end vectors of chains that have broken via the jth reaction pathway. The equilibrium probabilities may be written using Eqs. (5.7) and (5.17) in the equilibrium system, or by using the ratio of partition functions [49, 125] as

$$P^{\mathrm{eq}}_{\mathrm{A}}(\boldsymbol{\xi}) = \left\langle \delta^3\left[\mathbf{R}(\Gamma)-\boldsymbol{\xi}\right] \prod_{j=1}^{M} \Theta\left(\ell_j^{\ddagger}-\ell_j\right) \right\rangle^{\mathrm{eq}} = \frac{\mathfrak{q}^*_{\mathrm{A}}(\boldsymbol{\xi})}{\mathfrak{q}}, \tag{5.19}$$

$$P^{\mathrm{eq}}_{\mathrm{B}_j}(\boldsymbol{\xi}) = \left\langle \delta^3\left[\mathbf{R}(\Gamma)-\boldsymbol{\xi}\right] \Theta\left(\ell_j-\ell_j^{\ddagger}\right) \prod_{\substack{i=1\\ i\neq j}}^{M} \Theta\left(\ell_i^{\ddagger}-\ell_i\right) \right\rangle^{\mathrm{eq}} = \frac{\mathfrak{q}^*_{\mathrm{B}_j}(\boldsymbol{\xi})}{\mathfrak{q}}, \tag{5.20}$$

where the partition functions of an intact chain at end-to-end vector $\boldsymbol{\xi}$ and a chain that has broken via the jth reaction pathway at $\boldsymbol{\xi}$ (the asterisk $*$ denotes the fixed $\boldsymbol{\xi}$) are given by

$$\mathfrak{q}^*_{\mathrm{A}}(\boldsymbol{\xi}) = \int\cdots\int e^{-\beta H(\Gamma)} \delta^3\left[\mathbf{R}(\Gamma)-\boldsymbol{\xi}\right] \prod_{j=1}^{M} \Theta\left(\ell_j^{\ddagger}-\ell_j\right)\, d\Gamma, \tag{5.21}$$

$$\mathfrak{q}^*_{\mathrm{B}_j}(\boldsymbol{\xi}) = \int\cdots\int e^{-\beta H(\Gamma)} \delta^3\left[\mathbf{R}(\Gamma)-\boldsymbol{\xi}\right] \Theta\left(\ell_j-\ell_j^{\ddagger}\right) \prod_{\substack{i=1\\ i\neq j}}^{M} \Theta\left(\ell_i^{\ddagger}-\ell_i\right)\, d\Gamma. \tag{5.22}$$

Similarly, the partition function of an intact chain at $\boldsymbol{\xi}$ in the jth transition state is

$$\mathfrak{q}^*_{\ddagger_j}(\boldsymbol{\xi}) = \int\cdots\int e^{-\beta H(\Gamma)} \delta\left(p_j\right) \delta^3\left[\mathbf{R}(\Gamma)-\boldsymbol{\xi}\right] \delta\left(\ell_j^{\ddagger}-\ell_j\right) \prod_{\substack{i=1\\ i\neq j}}^{M} \Theta\left(\ell_i^{\ddagger}-\ell_i\right)\, d\Gamma. \tag{5.23}$$

Now, after utilizing Eqs. (5.16)–(5.23) (see Appendix C.1), the rates in Eqs. (5.13)–(5.14) become

$$\mathcal{R}'_j(\boldsymbol{\xi};t) = k'_j(\boldsymbol{\xi}) P_{\mathrm{A}}(\boldsymbol{\xi};t), \tag{5.24}$$

$$\mathcal{R}''_j(\boldsymbol{\xi};t) = k''_j(\boldsymbol{\xi}) P_{\mathrm{B}_j}(\boldsymbol{\xi};t), \tag{5.25}$$

where the forward and reverse reaction rate coefficient functions are respectively given by

$$k'_j(\boldsymbol{\xi}) = \frac{1}{\beta}\frac{\mathfrak{q}^*_{\ddagger_j}(\boldsymbol{\xi})}{\mathfrak{q}^*_{\mathrm{A}}(\boldsymbol{\xi})}, \tag{5.26}$$

$$k''_j(\boldsymbol{\xi}) = \frac{1}{\beta}\frac{\mathfrak{q}^*_{\ddagger_j}(\boldsymbol{\xi})}{\mathfrak{q}^*_{\mathrm{B}_j}(\boldsymbol{\xi})}. \tag{5.27}$$

While extension-dependent rates have been previously considered for polymer networks [58, 59], exact relations have not yet been discovered, leaving models to assume they are constant [58, 60, 160, 161, 175], or assume some other form [144, 150, 156, 162, 164, 165, 169] typically inspired by or in some way similar to the model of Bell [61]. Eqs. (5.26) and (5.27) show that each forward and reverse reaction rate coefficient function is completely determined by the single-chain model via the partition functions. These partition functions similarly determine the single-chain mechanical response and the equilibrium distribution of chain end-to-end vectors in the network [125].

After relating the ratio of the reaction rate coefficient functions to the equilibrium probabilities,

$$\frac{k_j''(\boldsymbol{\xi})}{k_j'(\boldsymbol{\xi})} = \frac{\mathfrak{q}_{\mathrm{A}}^*(\boldsymbol{\xi})}{\mathfrak{q}_{\mathrm{B}_j}^*(\boldsymbol{\xi})} = \frac{P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})}{P_{\mathrm{B}_j}^{\mathrm{eq}}(\boldsymbol{\xi})}, \tag{5.28}$$

we finally rewrite Eq. (5.15), the evolution law for the probability distribution of intact chains as

$$\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial t} = -\sum_{j=1}^{M} k_j'(\boldsymbol{\xi}) \left[P_{\mathrm{A}}(\boldsymbol{\xi};t) - \frac{P_{\mathrm{B}_j}(\boldsymbol{\xi};t)}{P_{\mathrm{B}_j}^{\mathrm{eq}}(\boldsymbol{\xi})} P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi}) \right] - \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \left[\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t) P_{\mathrm{A}}(\boldsymbol{\xi};t) \right]. \tag{5.29}$$

This evolution equation depends only upon the independent variables $\boldsymbol{\xi}$ and t. We establish the evolution equation for the probability distribution of each species of broken chains $P_{\mathrm{B}_j}(\boldsymbol{\xi};t)$ in the following section. Later in Sec. 5.2.2 we constitutively prescribe $\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t)$ as a function of the deformation.

Evolution of broken chains

In this section, we obtain simplified evolution laws for the probability density distribution of the jth broken chains $P_{\mathrm{B}_j}(\boldsymbol{\xi};t)$, and in the process further simplify Eq. (5.29). To proceed, we first neglect the possibility that a chain may break via multiple pathways. This is reasonable when all breaking pathways remain approximately inaccessible without considerable force acting on the chain, since broken chains will not support the force required to break again. It is also reasonable when at most one breaking pathway is thermally accessible, such as the case with a chain with a single transient bond and many strong covalent bonds. This assumption inhibits our ability to model chains with many highly dynamic bonds, such as those within vitrimers. Second, we neglect the possibility that broken chains of different reaction pathways may cross-reform, which prevents us from considering cases where groups of chains break and reform together resulting in altered contour lengths. These two assumptions allow us to conclude that the total probability that a chain is either intact or is broken via a single reaction pathway is unity, yielding the conservation law

$$\iiint P_{\mathrm{A}}(\boldsymbol{\xi};t)\, d^3\boldsymbol{\xi} + \sum_{j=1}^{M} \iiint P_{\mathrm{B}_j}(\boldsymbol{\xi};t)\, d^3\boldsymbol{\xi} = 1. \tag{5.30}$$

We evaluate Eq. (5.30) at equilibrium and multiply it by the system partition function $\mathfrak{q}$, which with the equilibrium probabilities in Eqs. (5.19) and (5.20) then allows us to relate the partition functions as

$$\mathfrak{q} = \iiint \mathfrak{q}^*_{\mathrm{A}}(\boldsymbol{\xi})\, d^3\boldsymbol{\xi} + \sum_{j=1}^{M} \iiint \mathfrak{q}^*_{\mathrm{B}_j}(\boldsymbol{\xi})\, d^3\boldsymbol{\xi} \equiv \mathfrak{q}_{\mathrm{A}} + \sum_{j=1}^{M} \mathfrak{q}_{\mathrm{B}_j}. \tag{5.31}$$

Once we specify a chain model, we can calculate $\mathfrak{q}^*_{\mathrm{A}}(\boldsymbol{\xi})$ and each $\mathfrak{q}^*_{\mathrm{B}_j}(\boldsymbol{\xi})$, and with Eq. (5.31) we may then calculate the equilibrium distributions in Eqs. (5.19) and (5.20). Note that we have defined $\mathfrak{q}_{\mathrm{A}}$ and $\mathfrak{q}_{\mathrm{B}_j}$ in Eq. (5.31), which are the partition functions of unconstrained intact chains and broken chains, respectively. $\mathfrak{q}_{\mathrm{A}}$ and $\mathfrak{q}_{\mathrm{B}_j}$ are equivalently the integrals of $\mathfrak{q}^*_{\mathrm{A}}(\boldsymbol{\xi})$ and $\mathfrak{q}^*_{\mathrm{B}_j}(\boldsymbol{\xi})$ over all end-to-end vectors $\boldsymbol{\xi}$. By the principal thermodynamic connection formula [49] for the Helmholtz free energy $\psi = -\mathfrak{b}T \ln \mathfrak{q}$, we obtain

$$\Delta\Psi_{0_j} \equiv -\mathfrak{b}T \ln\left(\frac{\mathfrak{q}_{\mathrm{B}_j}}{\mathfrak{q}_{\mathrm{A}}}\right), \tag{5.32}$$

where $\Delta\Psi_{0_j}$ is then the net Helmholtz free energy change for the jth breaking reaction alone (only having to do with the reaction coordinate, not the rest of the chain). We now approximate the two ends of any broken chain as effectively noninteracting, which allows the partition function $\mathfrak{q}^*_{\mathrm{B}_j}(\boldsymbol{\xi})$ to be constant in $\boldsymbol{\xi}$,

$$\mathfrak{q}_{\mathrm{B}_j} \approx V_{\mathrm{B}_j} \mathfrak{q}^*_{\mathrm{B}_j}, \tag{5.33}$$

where each V_{B_j} has units of volume. Using Eq. (5.32), we can define V_{B_j} in terms of $\mathfrak{q}_{\mathrm{A}}$, $\mathfrak{q}^*_{\mathrm{B}_j}$, and $\beta\Delta\Psi_{0_j}$ as

$$V_{\mathrm{B}_j} \equiv (\mathfrak{q}_{\mathrm{A}}/\mathfrak{q}^*_{\mathrm{B}_j}) e^{-\beta\Delta\Psi_{0_j}}. \tag{5.34}$$

Though we have treated the broken chain mechanics as independent of $\boldsymbol{\xi}$, the reforming reaction rate coefficient function $k''_j(\boldsymbol{\xi})$ in Eq. (5.27) is still a strong function of $\boldsymbol{\xi}$ due to $\mathfrak{q}^*_{\ddagger_j}(\boldsymbol{\xi})$. Our approximation in Eq. (5.33) causes all broken chain end-to-end vectors to be equally probable and therefore allows us to equate the probability of broken chains at end-to-end vector $\boldsymbol{\xi}$ to the average of broken chains at any end-to-end vector,

$$P_{\mathrm{B}_j}(\boldsymbol{\xi}; t) \approx \frac{P^{\mathrm{tot}}_{\mathrm{B}_j}(t)}{V_{\mathrm{B}_j}}. \tag{5.35}$$

Since we do not track broken chains by end-to-end vector, we rewrite the conservation law from Eq. (5.30) as

$$\iiint P_{\mathrm{A}}(\boldsymbol{\xi}; t)\, d^3\boldsymbol{\xi} + \sum_{j=1}^{M} P^{\mathrm{tot}}_{\mathrm{B}_j}(t) = 1, \tag{5.36}$$

and, after taking the time derivative,

$$\iiint \frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi}; t)}{\partial t}\, d^3\boldsymbol{\xi} + \sum_{j=1}^{M} \frac{\partial P^{\mathrm{tot}}_{\mathrm{B}_j}(t)}{\partial t} = 0. \tag{5.37}$$

Applying the conservation requirement given by Eq. (5.31) and the relation between the partition functions given by Eq. (5.33), we may now rewrite the equilibrium probabilities from Eqs. (5.19) and (5.20) as

$$P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi}) = \frac{1}{1+\sum_{\ell=1}^{M} e^{-\beta\Delta\Psi_{0_\ell}}} \left(\frac{\mathfrak{q}_{\mathrm{A}}^{*}(\boldsymbol{\xi})}{\iiint \mathfrak{q}_{\mathrm{A}}^{*}(\tilde{\boldsymbol{\xi}})\, d^3\tilde{\boldsymbol{\xi}}} \right), \tag{5.38}$$

$$P_{\mathrm{B}_j}^{\mathrm{tot,eq}} = \frac{e^{-\beta\Delta\Psi_{0_j}}}{1+\sum_{\ell=1}^{M} e^{-\beta\Delta\Psi_{0_\ell}}}, \tag{5.39}$$

and the evolution of the intact chains from Eq. (5.29) as

$$\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial t} = -\sum_{j=1}^{M} k_j'(\boldsymbol{\xi}) \left[P_{\mathrm{A}}(\boldsymbol{\xi};t) - \frac{P_{\mathrm{B}_j}^{\mathrm{tot}}(t)}{P_{\mathrm{B}_j}^{\mathrm{tot,eq}}} P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi}) \right] - \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \left[\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t) P_{\mathrm{A}}(\boldsymbol{\xi};t) \right]. \tag{5.40}$$

After writing the analogous evolution law for $P_{\mathrm{B}_j}(\boldsymbol{\xi};t)$, integrating for $P_{\mathrm{B}_j}^{\mathrm{tot}}(t)$, and using Eq. (5.35), the evolution equation for $P_{\mathrm{B}_j}^{\mathrm{tot}}(t)$ is given by

$$\frac{\partial P_{\mathrm{B}_j}^{\mathrm{tot}}(t)}{\partial t} = \iiint k_j'(\boldsymbol{\xi}) \left[P_{\mathrm{A}}(\boldsymbol{\xi};t) - \frac{P_{\mathrm{B}_j}^{\mathrm{tot}}(t)}{P_{\mathrm{B}_j}^{\mathrm{tot,eq}}} P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi}) \right] d^3\boldsymbol{\xi} - \frac{P_{\mathrm{B}_j}^{\mathrm{tot}}(t)}{V_{\mathrm{B}_j}} \iiint \left[\frac{\partial}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}}_{\mathrm{B}}(\boldsymbol{\xi};t) \right] d^3\boldsymbol{\xi}. \tag{5.41}$$

The last term in Eq. (5.41) is found equal to zero as follows: Eq. (5.40) is integrated over all $\boldsymbol{\xi}$ and the conservation requirement in Eq. (5.37) is applied in substituting in for $P_{\mathrm{B}_j}^{\mathrm{tot}}(t)$, where all the reaction-related terms then cancel. We then apply the divergence theorem to this integral, which produces a balance law that we satisfy by requiring that the integrand is zero for all $\boldsymbol{\xi}$, which is

$$\left[\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t) P_{\mathrm{A}}(\boldsymbol{\xi};t) + \sum_{j=1}^{M} \dot{\boldsymbol{\xi}}_{\mathrm{B}}(\boldsymbol{\xi};t) \frac{P_{\mathrm{B}_j}^{\mathrm{tot}}(t)}{V_{\mathrm{B}_j}} \right]_{\partial_{\boldsymbol{\xi}}\mathcal{A}} = 0. \tag{5.42}$$

Here $\partial_{\boldsymbol{\xi}}\mathcal{A}$ is the boundary of the region $\mathcal{A}$ with outward-pointing unit normal vector $\hat{\mathbf{n}}_{\partial_{\boldsymbol{\xi}}\mathcal{A}}$, a surface beyond which no intact chain may exist. Eq. (5.42) is a balance law for intact chains that are instantaneously broken via $\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t)$ carrying them across $\partial_{\boldsymbol{\xi}}\mathcal{A}$. We will now assume that a negligible amount of intact chains become extended to this intact limit without first breaking via the chemical reaction, which is $P_{\mathrm{A}}(\boldsymbol{\xi};t)|_{\partial_{\boldsymbol{\xi}}\mathcal{A}} \approx 0$. We then have $\dot{\boldsymbol{\xi}}_{\mathrm{B}}(\boldsymbol{\xi};t)|_{\partial_{\boldsymbol{\xi}}\mathcal{A}} \approx 0$ via Eq. (5.42), which causes the last term in Eq. (5.41) to become zero after again using the divergence theorem. Eq. (5.41) is now

$$\frac{\partial P_{\mathrm{B}_j}^{\mathrm{tot}}(t)}{\partial t} = \iiint k_j'(\boldsymbol{\xi}) \left[P_{\mathrm{A}}(\boldsymbol{\xi};t) - \frac{P_{\mathrm{B}_j}^{\mathrm{tot}}(t)}{P_{\mathrm{B}_j}^{\mathrm{tot,eq}}} P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi}) \right] d^3\boldsymbol{\xi}. \tag{5.43}$$

Eqs. (5.40) and (5.43) together with a prescription for $\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t)$ create a set of evolution equations that govern the polymer network, are reasonable to solve, and have a firm foundation in the principles of statistical mechanics. Equipped with this framework to evaluate the relevant probabilities of chains in the network, we now use statistical thermodynamics to formulate the Helmholtz free energy and then use macroscopic constitutive theory to obtain the constitutive relation for the Cauchy stress.

5.2.2 Macroscopic theory

With our statistical mechanical framework complete, we turn now to the macroscopic description of the network. We begin by formulating the Helmholtz free energy of the network using statistical thermodynamics, which preserves our statistical mechanical framework as we move into the continuum scale. Knowledge of the Helmholtz free energy allows us to utilize the Coleman-Noll procedure [82, 83] to obtain constitutive relations for the entropy density and Cauchy stress. We do so after assuming that the temperature (T), deformation gradient ($\mathbf{F}$), probability density distribution of attached chains (P_{A}), and probability of each broken chain type ($P_{\mathrm{B}_j}^{\mathrm{tot}}$) form a complete set of thermodynamic state variables. We additionally assume that, on average, the deformation gradient acts affinely on the intact chain end-to-end vectors. Lastly, we show that the residual inequality – solely dissipation due to the breaking/reforming of chains – is already arbitrarily satisfied for the evolution laws we have derived in Sec. 5.2.1.

Network Helmholtz free energy

The Helmholtz free energy $\mathscr{A}(t)$ of the network is analogous to that of a system of noninteracting particles of different chemical species [49]

$$\mathscr{A}(t) = N_{\mathrm{A}}(t)\mu_{\mathrm{A}}(t) + \sum_{j=1}^{M} N_{\mathrm{B}_j}(t)\mu_{\mathrm{B}_j}(t) - N\mathfrak{b}T(t), \tag{5.44}$$

with the chemical potentials $\mu_i(t)$ given by

$$\mu_i(t) = -\mathfrak{b}T \ln\left[\frac{\mathfrak{q}_i}{N_i(t)}\right], \quad i = \mathrm{A}, \mathrm{B}_1, \ldots, \mathrm{B}_M, \tag{5.45}$$

where $N_i(t)$ is the number of either intact or broken chains, and N is the constant total number of chains. We use Gibbs' postulate [49] to write $\mu_{\mathrm{A}}(t)$ as the time-dependent average

$$\mu_{\mathrm{A}}(t) = \frac{1}{P_{\mathrm{A}}^{\mathrm{tot}}(t)} \iiint P_{\mathrm{A}}(\boldsymbol{\xi};t)\mu_{\mathrm{A}}^*(\boldsymbol{\xi};t)\, d^3\boldsymbol{\xi}, \tag{5.46}$$

where (using $N_i(t) = P_i^{\mathrm{tot}}(t)N$) the chemical potential of an intact chain at end-to-end vector $\boldsymbol{\xi}$ is

$$\mu_{\mathrm{A}}^*(\boldsymbol{\xi};t) = -\mathfrak{b}T \ln\left[\frac{\mathfrak{q}_{\mathrm{A}}^*(\boldsymbol{\xi})}{NP_{\mathrm{A}}(\boldsymbol{\xi};t)}\right]. \tag{5.47}$$

The broken chains have been assumed to be insensitive to extension, so we similarly utilize their chemical potentials as independent of the end-to-end vector $\boldsymbol{\xi}$,

$$\mu_{\mathrm{B}_j}(t) = -\mathfrak{b}T \ln\left[\frac{V_{\mathrm{B}_j}\mathfrak{q}_{\mathrm{B}_j}^*}{NP_{\mathrm{B}_j}^{\mathrm{tot}}(t)}\right]. \tag{5.48}$$

We may now write the Helmholtz free energy density $a(t) = \mathscr{A}(t)/V$ of our incompressible network of noninteracting polymer chains as

$$a(t) = n\iiint P_{\mathrm{A}}(\boldsymbol{\xi};t)\mu_{\mathrm{A}}^*(\boldsymbol{\xi};t)\, d^3\boldsymbol{\xi} + n\sum_{j=1}^{M} P_{\mathrm{B}_j}^{\mathrm{tot}}(t)\mu_{\mathrm{B}_j}(t) - n\mathfrak{b}T(t) - p[J(t)-1], \tag{5.49}$$

where $n = N/V$ is the constant total number density of chains, and p is the pressure acting as a Lagrange multiplier enforcing the incompressibility constraint that $J = \det(\mathbf{F}) = 1$. This specific formulation of $a(t)$ in Eq. (5.49) is essential to our approach: in Appendix C.2.1, we show that it allows the equilibrium probabilities obtained from minimizing $a(t)$ with respect to each probability to exactly match those obtained beforehand from the statistical mechanical derivation (Eqs. (5.38) and (5.39)).

Constitutive relations

The time derivative of the Helmholtz free energy density is

$$\dot{a}(t) = n \iiint \dot{P}_{\mathrm{A}}(\boldsymbol{\xi};t)\mu^*_{\mathrm{A}}(\boldsymbol{\xi};t)\, d^3\boldsymbol{\xi} + n\sum_{j=1}^{M} \dot{P}^{\mathrm{tot}}_{\mathrm{B}_j}(t)\mu_{\mathrm{B}_j}(t) - nk\dot{T}(t) - p(t)\left[\mathbf{1} : \mathbf{L}(t)\right], \tag{5.50}$$

where $\mathbf{L} = \dot{\mathbf{F}} \cdot \mathbf{F}^{-1}$ is the velocity gradient. We have factored out the time derivatives of the chemical potentials after utilizing the conservation requirement from Eq. (5.37). Thermodynamically admissible processes must satisfy the second law of thermodynamics regarding irreversible entropy production, which is embodied in the Clausius-Duhem inequality

$$\dot{a} + s\dot{T} - \boldsymbol{\sigma} : \mathbf{L} \leq 0, \tag{5.51}$$

where $s(t)$ is the entropy density and $\boldsymbol{\sigma}(t)$ is the Cauchy stress tensor [134]. Note that this simplified form of the inequality already assumes incompressibility, neglects nonmechanical work, and assumes the classical constitutive relations for the entropy flux, the entropy source, and the heat flux [135]. Further, we will not impose hyperbolicity requirements that would guarantee finite speeds of propagation [176].

We will assume that $T(t)$, $\mathbf{F}(t)$, $P_{\mathrm{A}}(\boldsymbol{\xi};t)$, and $P^{\mathrm{tot}}_{\mathrm{B}_j}(t)$ together create a complete set of thermodynamic state variables. This allows us to treat time derivatives of constitutive variables like $a(t)$ as fully implicit, where we may expand those time derivatives as partial derivatives with respect to the state variables. The three independent thermodynamic processes accounted for through our state variables are temperature change, deformation, and the chain breaking/reforming chemical reactions (denoted as rxn). The evolution of $a(t)$ is then expanded as

$$\dot{a} = \left(\frac{\partial a}{\partial T}\right)_{\mathbf{F},\mathrm{rxn}} \dot{T} + \left(\frac{\partial a}{\partial \mathbf{F}}\right)_{T,\mathrm{rxn}} : \dot{\mathbf{F}} - \mathcal{D}_{\mathrm{rxn}}, \tag{5.52}$$

where $\mathcal{D}_{\mathrm{rxn}}$ is the chemical dissipation per unit volume, calculated from the rate of change of Helmholtz free energy density over all breaking/reforming reactions

$$\mathcal{D}_{\mathrm{rxn}} \equiv -\left(\frac{\partial a}{\partial t}\right)_{\mathbf{F},T}. \tag{5.53}$$

Substitution of our expansion into the Clausius-Duhem inequality in Eq. (5.51) yields

$$\left[\left(\frac{\partial a}{\partial T}\right)_{\mathbf{F},\mathrm{rxn}} + s\right]\dot{T} - \mathcal{D}_{\mathrm{rxn}} + \left[\left(\frac{\partial a}{\partial \mathbf{F}}\right)_{T,\mathrm{rxn}} \cdot \mathbf{F}^T - \boldsymbol{\sigma}\right] : \mathbf{L} \leq 0. \tag{5.54}$$

We first consider the set of processes where the temperature varies arbitrarily, the deformation is held fixed, and the reactions do not proceed. In order to arbitrarily satisfy the inequality, we must then have

$$s = -\left(\frac{\partial a}{\partial T}\right)_{\mathbf{F},\text{rxn}}, \tag{5.55}$$

which is the expected constitutive relation for the entropy density. The inequality in Eq. (5.54) then becomes

$$\left[\left(\frac{\partial a}{\partial \mathbf{F}}\right)_{T,\text{rxn}} \cdot \mathbf{F}^T - \boldsymbol{\sigma}\right] : \mathbf{L} - \mathcal{D}_{\text{rxn}} \leq 0. \tag{5.56}$$

We next consider processes where the motion varies arbitrarily via $\mathbf{L}(t)$ and the reactions do not proceed. Since we have assumed to have a complete set of thermodynamic state variables, and that set does not include time derivatives of the deformation gradient, we have already ruled out dissipative stresses [135]. In order to arbitrarily satisfy the inequality in Eq. (5.56), we must have

$$\boldsymbol{\sigma} = \left(\frac{\partial a}{\partial \mathbf{F}}\right)_{T,\text{rxn}} \cdot \mathbf{F}^T, \tag{5.57}$$

which is the form of the stress for a hyperelastic material [134]. We will now assume that, on average, the end-to-end vectors $\boldsymbol{\xi}$ are affinely deformed by the deformation gradient $\mathbf{F}(t)$, which can be expanded as $\dot{\boldsymbol{\xi}}_{\text{A}}(\boldsymbol{\xi};t) = \mathbf{L}(t)\cdot\boldsymbol{\xi}$. After applying this assumption and simplifying (see Appendix C.2.2), the evolution equation for intact chains Eq. (5.40) becomes

$$\frac{\partial P_{\text{A}}(\boldsymbol{\xi};t)}{\partial t} = -\left[\frac{\partial P_{\text{A}}(\boldsymbol{\xi};t)}{\partial \boldsymbol{\xi}}\,\boldsymbol{\xi}\right] : \mathbf{L}(t) - \sum_{j=1}^{M} k'_j(\boldsymbol{\xi})\left[P_{\text{A}}(\boldsymbol{\xi};t) - \frac{P^{\text{tot}}_{\text{B}_j}(t)}{P^{\text{tot,eq}}_{\text{B}_j}}\,P^{\text{eq}}_{\text{A}}(\boldsymbol{\xi})\right], \tag{5.58}$$

and the stress in Eq. (5.57) becomes

$$\boldsymbol{\sigma}(t) = n\iiint P_{\text{A}}(\boldsymbol{\xi};t)\,\frac{\partial \psi^*_{\text{A}}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\,\boldsymbol{\xi}\,d^3\boldsymbol{\xi} - p(t)\mathbf{1}. \tag{5.59}$$

This general form of the stress has been obtained previously [125], but the evolution of $P_{\text{A}}(\boldsymbol{\xi};t)$ is now more complicated here due to the breaking and reforming of chains.

Residual inequality

Now that we have established each constitutive relation, we are left with the residual portion of the Clausius-Duhem inequality due to the dissipation $\mathcal{D}_{\text{rxn}}(t)$. Showing that $\mathcal{D}_{\text{rxn}}(t) \geq 0$ is similar to the procedure for a reacting system of a finite number of discrete chemical species [177, 178], but here we have the additional complication of having the reactions (chains breaking and reforming) occurring over the continuous variable $\boldsymbol{\xi}$ (see Appendix C.2.3 for details). We find that the dissipation may be written succinctly as

$$\mathcal{D}_{\text{rxn}}(t) = \sum_{j=1}^{M}\iiint \mathcal{D}^*_j(\boldsymbol{\xi};t)\,d^3\boldsymbol{\xi}, \tag{5.60}$$

where the dissipation density for the jth reaction occurring at the end-to-end vector $\boldsymbol{\xi}$ is defined as

$$\mathcal{D}_j^*(\boldsymbol{\xi};t) \equiv n\mathfrak{b}T\left[\mathcal{R}_j'(\boldsymbol{\xi};t) - \mathcal{R}_j''(\boldsymbol{\xi};t)\right]\ln\left[\frac{\mathcal{R}_j'(\boldsymbol{\xi};t)}{\mathcal{R}_j''(\boldsymbol{\xi};t)}\right]. \tag{5.61}$$

Since $[\mathcal{R}_j'(\boldsymbol{\xi};t) - \mathcal{R}_j''(\boldsymbol{\xi};t)]\ln[\mathcal{R}_j'(\boldsymbol{\xi};t)/\mathcal{R}_j''(\boldsymbol{\xi};t)] \geq 0$ and $n\mathfrak{b}T > 0$, we are able to conclude

$$\mathcal{D}_j^*(\boldsymbol{\xi};t) \geq 0 \quad \text{for all } j, \text{ and therefore} \quad \mathcal{D}_{\text{rxn}}(t) \geq 0. \tag{5.62}$$

This result means that not only is the residual inequality satisfied, but each chain breaking/reforming reaction at every end-to-end vector has a positive semi-definite dissipation. Further, this is found without any additional restrictions on the thermodynamic state variables or the obtained constitutive relations, which can be directly attributed to the strong statistical mechanical foundation we have incorporated.

5.2.3 General theory summary

Our general theory is now complete and can be utilized as illustrated by Fig. 5.2. Two inputs are needed to constitutively define the polymer – the single-chain model and the total number density of chains n. Two external conditions are also prescribed as inputs – the temperature T and the deformation gradient $\mathbf{F}(t)$. A single-chain model is chosen through specification of a Hamiltonian $H(\Gamma)$, which contains M transition state coordinates ℓ_j and locations $\ell_j^\ddagger$. Next, we calculate the partition function at the end-to-end vector $\boldsymbol{\xi}$ of the intact chains $\mathfrak{q}_{\text{A}}^*(\boldsymbol{\xi})$ using Eq. (5.21), of each broken chain species $\mathfrak{q}_{\text{B}_j}^*$ (at some large $\boldsymbol{\xi}$ where the two broken ends do not interact) using Eq. (5.22), and of each transition state $\mathfrak{q}_{\ddagger_j}^*(\boldsymbol{\xi})$ using Eq. (5.23). We also compute the net free energy changes $\Delta\Psi_{0_j}$ using $H(\Gamma)$ or otherwise specify them as parameters. Equipped with these quantities, we are able to compute each reaction rate coefficient function $k_j'(\boldsymbol{\xi})$ using Eq. (5.26), the equilibrium probability density distribution of intact chains $P_{\text{A}}^{\text{eq}}(\boldsymbol{\xi})$ using Eq. (5.38), and the equilibrium probability of each broken chain species $P_{\text{B}_j}^{\text{tot,eq}}$ using Eq. (5.39). We also compute $\psi_{\text{A}}^*(\boldsymbol{\xi})$, the Helmholtz free energy of an intact chain at the end-to-end vector $\boldsymbol{\xi}$, using Eq. (C.18). With a prescribed incompressible deformation history and assuming initial conditions, we have all the necessary information to formulate the evolution law for the probability density distribution of intact chains $P_{\text{A}}(\boldsymbol{\xi};t)$ using Eq. (5.58) and those for the probability of each broken chain species $P_{\text{B}_j}^{\text{tot}}(t)$ using Eq. (5.43). The stress $\boldsymbol{\sigma}(t)$ is then computed using Eq. (5.59), where the pressure $p(t)$ is solved for using the traction boundary conditions.

5.3 Single-chain model specification

We are now ready to specify a single-chain model and push it through our general framework as shown Fig. 5.2, computing each quantity of interest to ultimately obtain the stress as a function of deformation. For the materials we will model, we require a single-chain model that incorporates force-sensitive reversible bond breaking. Force-sensitive irreversible and force-insensitive (transient) reversible bonds are special cases of this broader class. We propose the uFJC for our single-chain model: a freely jointed chain of flexible links, each with a potential u that depends on the difference between the link length and its rest-length. The Morse potential [179] is used for each link in order

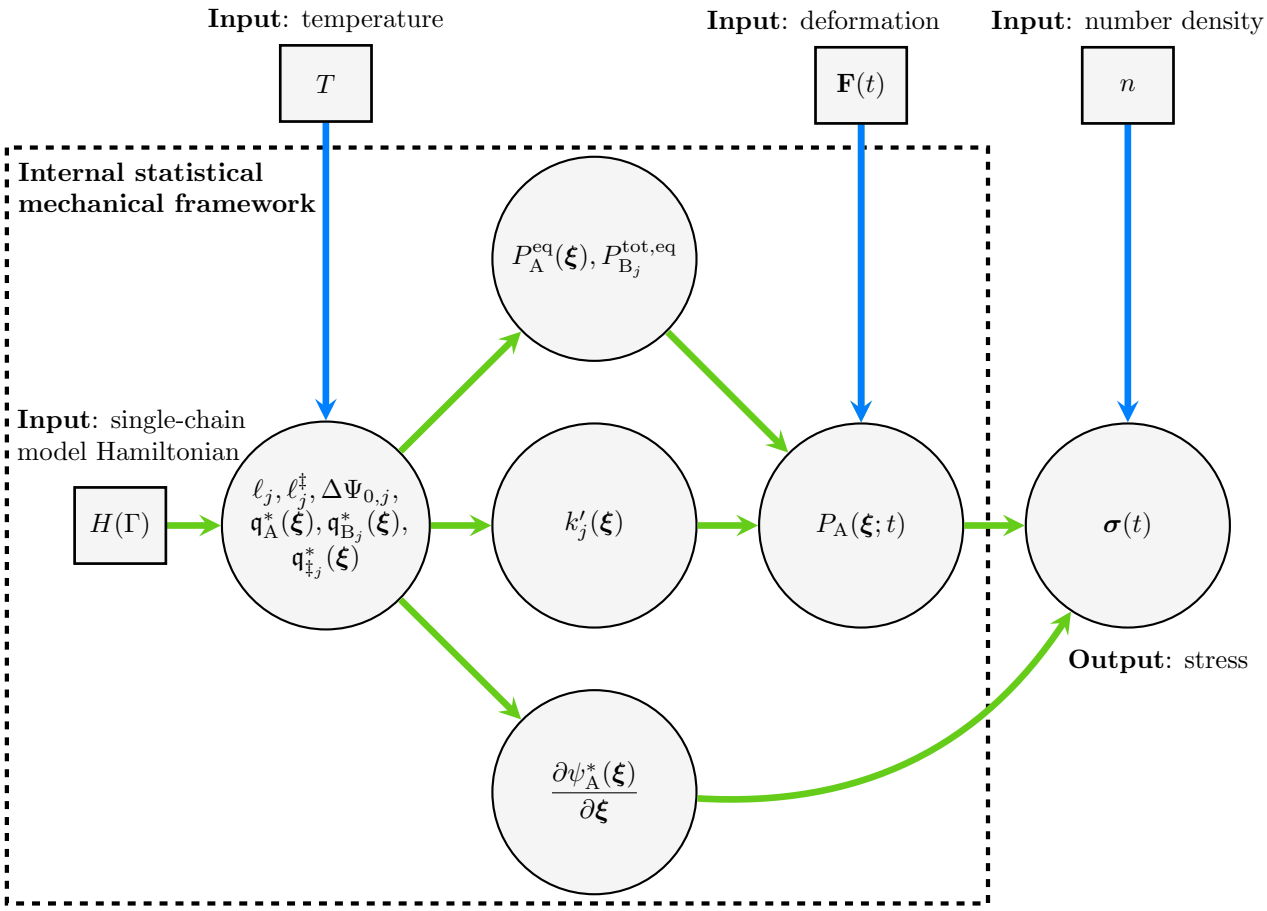


Figure 5.2: Diagram describing the utilization of the general theory developed in Sec. 5.2. After specifying the single-chain model Hamiltonian $H(\Gamma)$, the temperature T, the deformation gradient $\mathbf{F}(t)$, and the total number density of chains n, all quantities of interest may be systematically computed, ultimately resulting in the stress $\boldsymbol{\sigma}(t)$.

to allow the links to break and reform. In the following section, we compute the functions related to the mechanical behavior, equilibrium distribution, and kinetics of breaking/reforming for the uFJC single-chain model and provide results with the Morse potential. We then exactly solve the evolution equation for the distribution of intact chains in the network for all single-chain models with reaction pathways that are all equivalent. Then we present how small adjustments can be made to the framework to account for when some links in the uFJC are weaker than the rest.

5.3.1 The uFJC model

The uFJC model is a freely jointed chain of N_b flexible links, each with potential $u(\ell)$ that depends on the difference between the link length ℓ and its rest-length ℓ_b. This is similar to the freely jointed chain or FJC model [1, 50], but with the rigid links replaced by these flexible ones. If the potential is strictly harmonic, we retrieve the extensible freely jointed chain or EFJC model [57]. The links of the uFJC are considered broken if $\ell > \ell_{\ddagger}$ and intact otherwise. Each of the $N_b + 1$ hinges are considered to be point masses with a mass of m, while the links are massless. The single-chain Hamiltonian of this model is

$$H(\Gamma) = \sum_{i=1}^{N_b+1} \frac{p_i^2}{2m} + \sum_{i=1}^{N_b} u\left(\ell_i\right). \tag{5.63}$$

We substitute this Hamiltonian into Eq. (5.21) to compute $\mathfrak{q}_{\mathrm{A}}^*(\boldsymbol{\xi}) = \mathfrak{q}_{\mathrm{A,mom}}^*\mathfrak{q}_{\mathrm{A,con}}^*(\boldsymbol{\xi})$, the partition function of an intact chain with end-to-end vector $\boldsymbol{\xi}$, where "mom" and "con" denote the momentum and configuration contributions. Since the link potential only gives information about the intact state of the links, we must treat the net Helmholtz free energy change when breaking a link, $\Delta\Psi_0$, as an independent parameter. While we may compute $\mathfrak{q}_{\mathrm{A,mom}}^*$ exactly without trouble [49], $\mathfrak{q}_{\mathrm{A,con}}^*(\boldsymbol{\xi})$ cannot be computed analytically in general and is difficult to evaluate numerically. This issue is typically resolved through computation of the Gibbs (isotensional) ensemble partition function and transforming back to the Helmholtz (isometric; canonical) partition function we desire. This transformation is accomplished using an inverse Fourier transform, which is often approximated using a Legendre transformation when chains are sufficiently long [53, 180], the so-called Gibbs-Legendre method [125] of obtaining $\mathfrak{q}_{\mathrm{A,con}}^*(\boldsymbol{\xi})$. After asymptotically approximating the mechanical response of the uFJC in the Gibbs ensemble, we will integrate it and use the Gibbs-Legendre method to obtain the Helmholtz free energy and then compute the desired partition function.

We obtain an asymptotic approximation for the single-chain mechanical response as

$$\gamma(\eta) \sim \mathcal{L}(\eta) + \lambda(\eta) - 1 \quad \text{for } \kappa \gg 1, \tag{5.64}$$

where $\gamma \equiv \xi/N_b\ell_b$ is the nondimensional end-to-end length, $\mathcal{L}(\eta) = \coth(\eta) - 1/\eta$ is the Langevin function, $\lambda(\eta)$ is the link stretch ℓ/ℓ_b under the nondimensional force $\eta \equiv \beta f \ell_b$, and

$$\kappa \equiv \beta\ell_b^2 \left.\frac{\partial^2 u(\ell)}{\partial \ell^2}\right|_{\ell=\ell_b} \tag{5.65}$$

is the nondimensional link stiffness. The full derivation of Eq. (5.64) is in Appendix C.3.1, where it is shown that the entropically-based mechanical response of the FJC (given by the Langevin function) may be approximated as decoupled from the link stretching for sufficiently stiff links. There has been recent interest in similarly supplementing entropic polymer chain models with potential energy contributions [67, 75, 78, 130, 142–144] by a method where a chain free energy is minimized with respect to the potential degrees of freedom [62]. Although this current method performs well, we recommend that the asymptotic approach be used instead for both practical and physical reasons. Practically, asymptotically-correct formulas such as those we provide here for the uFJC are easier to use than the current method, which requires implicitly solving nonlinear algebraic equations during the minimization process. Physically, the minimization of thermodynamic free energies may only involve macroscopic thermodynamic state variables, not phase space degrees of freedom such as link length, which would approach their potential energy minima as thermal energy becomes scarce. The apparent success of the current method can be attributed to the dominance of potential energy over the free energy minimization in the same limit, which in effect produces similar results.

We select the Morse potential [179] as the specific form of the uFJC link potential,

$$u(\ell) = u_b\left[1 - e^{-\sqrt{k_b/2u_b}(\ell-\ell_b)}\right]^2, \tag{5.66}$$

where u_b is the depth of the Morse potential energy well and k_b is the curvature near the bottom of the well (the stiffness). The nondimensional parameters here are the nondimensional energy

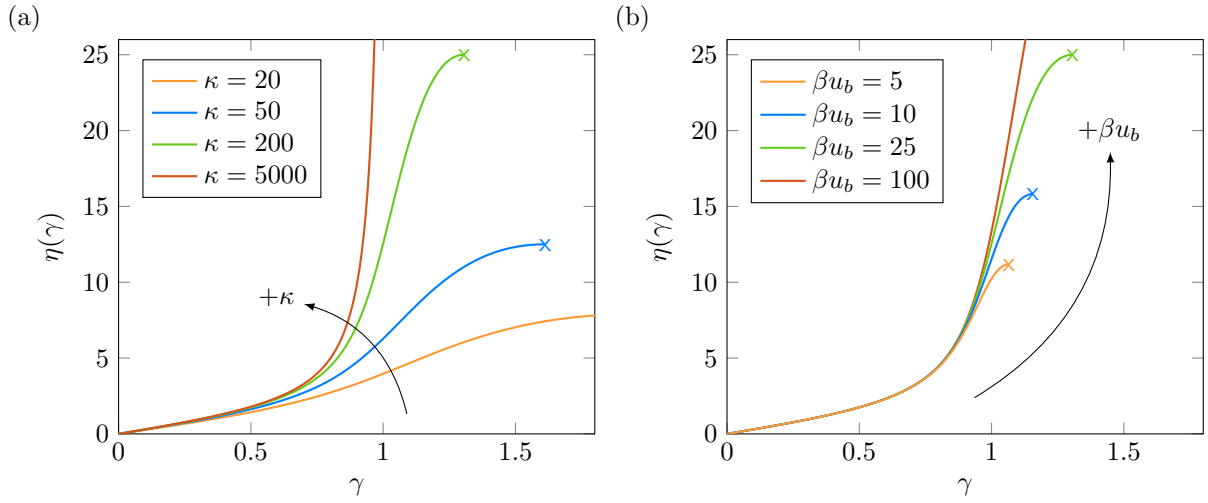


Figure 5.3: Single-chain mechanical response when using the Morse-FJC model. (a) The nondimensional force $\eta = \beta f \ell_b$ as a function of the nondimensional end-to-end length $\gamma = \xi/N_b\ell_b$ for $\beta u_b = 25$ and varying κ. (b) η as a function of γ for $\kappa = 200$ and varying βu_b.

βu_b, the nondimensional stiffness $\kappa \equiv \beta k_b \ell_b^2$, and the link stretch $\lambda \equiv \ell/\ell_b$. The stretch $\lambda_\ddagger \equiv 1 + \ln(2)\sqrt{2\beta u_b/\kappa}$ is the transition state stretch since this is where the nondimensional force η will reach its maximum possible value of $\eta_{\max} = \sqrt{\kappa\beta u_b/8}$. The derivative of the Morse potential gives the force as a function of link length, which is then inverted and rescaled to obtain the stretch of a single link under force,

$$\lambda(\eta) = 1 + \sqrt{\frac{2\beta u_b}{\kappa}}\,\ln\left[\frac{2}{1+\sqrt{1-\eta/\eta_{\max}}}\right] \quad \text{for } \eta \le \eta_{\max} = \sqrt{\frac{\kappa\beta u_b}{8}}. \tag{5.67}$$

Utilizing Eq. (5.67) with Eq. (5.64), we plot the mechanical response of the Morse-FJC in Fig. 5.3, varying κ in Fig. 5.3(a) and varying βu_b in Fig. 5.3(b). As κ increases, we observe a more dramatic transition near $\gamma = 1$ as the increasingly-stiff links begin to be stretched. Increasing κ directly increases the maximum force $\eta_{\max}$ and causes it to be reached at lower stretches, thereby causing the maximum nondimensional end-to-end length to decrease with increasing κ ($\gamma_{\max} \sim \lambda_\ddagger$). As we increase βu_b, we see an increase in both $\eta_{\max}$ and $\gamma_{\max}$; the overall mechanical response away from $\eta_{\max}$ is unchanged. When varying either κ or βu_b, the mechanical response at low γ is unchanged since this regime is dominated by the initially-compliant entropic behavior of $\mathcal{L}(\eta)$.

We now compute the other thermodynamic functions of interest: the equilibrium distribution $P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})$ and the reaction rate coefficient function $k'(\boldsymbol{\xi})$. For this single-chain model it is most practical to work in terms of the nondimensional chain end-to-end vector $\boldsymbol{\gamma} = \boldsymbol{\xi}/N_b\ell_b$. Further, our single-chain functions and equilibrium distributions depend only on $\gamma = \sqrt{\boldsymbol{\gamma}\cdot\boldsymbol{\gamma}}$. We therefore introduce the nondimensional configurational Helmholtz free energy per link

$$\vartheta^*_{\mathrm{A,con}}(\gamma) \equiv \frac{\beta\psi^*_{\mathrm{A,con}}(\gamma)}{N_b}. \tag{5.68}$$

We will assume that the number of links in our chain N_b remains high enough to utilize the Gibbs-Legendre method [125] to approximate $\psi^*_{\mathrm{A,con}}(\gamma)$, which in this case causes $\vartheta^*_{\mathrm{A,con}}(\gamma)$ to be indepen-

dent of N_b. While the Gibbs-Legendre method is invalid under sufficiently small forces [131], the regime of end-to-end lengths where this matters essentially vanishes with increasing N_b [53, 180]. Further, these tiny forces contribute little when integrating over all end-to-end lengths in Eq. (5.59) for the stress, causing the Gibbs-Legendre method to become correct as N_b increases [125]. As detailed in Appendix C.3.2, we obtain

$$\vartheta^*_{\mathrm{A,con}}(\gamma) = \ln\left\{\frac{\eta\exp[\eta\mathcal{L}(\eta)]}{\sinh(\eta)}\right\} + \beta u(\eta), \tag{5.69}$$

where $\eta = \eta(\gamma)$ is implied, which allows us to compute the nondimensional equilibrium distribution

$$\mathscr{P}^{\mathrm{eq}}_{\mathrm{A}}(\gamma) = \frac{1}{1+N_b e^{-\beta\Delta\Psi_0}}\left(\frac{e^{-N_b\vartheta^*_{\mathrm{A,con}}(\gamma)}}{\iiint e^{-N_b\vartheta^*_{\mathrm{A,con}}(\tilde{\gamma})}\,d^3\tilde{\boldsymbol{\gamma}}}\right). \tag{5.70}$$

In Appendix C.3.2 we also obtain the forward reaction rate coefficient function

$$k'(\gamma) = \frac{\omega_\ddagger}{2\pi}\,e^{-\beta\Delta\Psi^*_\ddagger(\gamma)}. \tag{5.71}$$

Here $\omega_\ddagger \equiv \sqrt{2\kappa/m\beta\ell_b^2}$ is the attempt frequency, and $\Delta\Psi^*_\ddagger(\gamma)$ is the Helmholtz free energy barrier of a single link to its transition state as a function of chain extension,

$$\Delta\Psi^*_\ddagger(\gamma) \equiv u(\ell_\ddagger) - u(\eta) - \mathfrak{b}T\ln\left\{\frac{\lambda_\ddagger\sinh(\lambda_\ddagger\eta)\exp[\eta\mathcal{L}(\eta)]}{\sinh(\eta)\exp[\lambda_\ddagger\eta\mathcal{L}(\lambda_\ddagger\eta)]}\right\}. \tag{5.72}$$

Note that we often use the initial rate $k'(0)$ in place of the attempt frequency $\omega_\ddagger$ or mass m as a more convenient but equivalent parameterization. The Helmholtz free energy barrier $\Delta\Psi^*_\ddagger(\gamma)$ consists of a positive contribution from the potential energy difference and a negative contribution from the entropy difference. The initial nondimensional barrier is $\beta\Delta\Psi^*_\ddagger(0) = \beta u_b/4 - 2\ln\lambda_\ddagger$, where recall $\lambda_\ddagger = 1+\ln(2)\sqrt{2\beta u_b/\kappa}$. Due to our requirement $\kappa \gg 1$, the nondimensional potential energy barrier βu_b will tend to dominate the scale of $\beta\Delta\Psi^*_\ddagger(\gamma)$ and therefore $k'(\gamma)$. If we assume $\lambda_\ddagger \approx 1$, we ignore the entropic term and take $u(\eta) \approx \eta\lambda(\eta)/2$, which then causes $k'(\gamma)$ in Eq. (5.71) to resemble the short-distance force-modified-barrier transition state theories [61, 181] that have been applied to polymer chains [69, 76, 156, 163, 182, 183]. The general case behavior of $k'(\gamma)$ here in Eq. (5.71) is more similar to the model of Dudko et al. [184] than these short-distance approximated models, especially since it accounts for both entropic and potential energy effects. Dudko's model is based on Kramers' theory of diffusive barrier crossing [173, 185] and has proved useful both in polymer chain AFM experiments [186] and polymer network constitutive models [130]. Our formulation for $k'(\gamma)$ has an advantage over Dudko's model: our $k'(\gamma)$ is directly connected to the statistical mechanics of the single-chain model, which provides guarantees such as dissipation inequality satisfaction and solution existence.

The reaction rate coefficient $k'(\gamma)$ is plotted as a function of the nondimensional end-to-end length γ in Fig. 5.4 for $\beta u_b = 25$ and varying κ. We find that $k'(\gamma)$ decreases slightly from $k'(0)$ as the chain is extended, which is due to the increasing entropy of the links. After a critical chain extension just above unity, we find that $k'(\gamma)$ increases dramatically due to the potential energy barrier to the transition state being rapidly reduced. As κ is increased, this trend becomes even more dramatic since the potential energy barrier is proportional to κ, and the observed critical extension approaches unity. Varying βu_b provides little change to the shape of $k'(\gamma)/k'(0)$ and

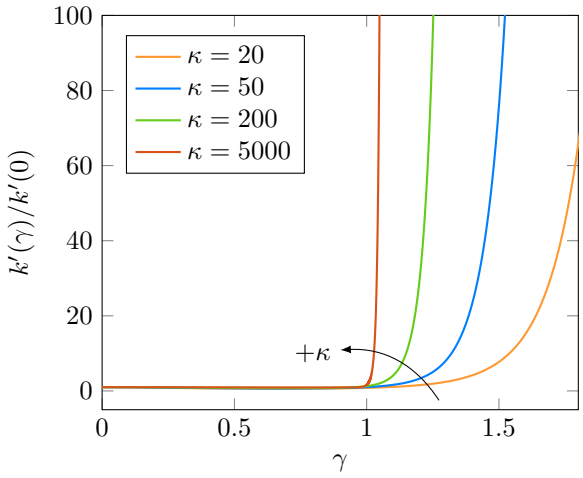


Figure 5.4: The reaction rate coefficient function $k'(\gamma)$ scaled by its initial value $k'(0)$ as a function of the nondimensional end-to-end length $\gamma = \xi/N_b\ell_b$, using the Morse-FJC model for $\beta u_b = 25$, and varying κ.

effectively varies the maximum allowable extension (not shown). Eqs. (5.71)–(5.72) show that $k'(0)$ decays exponentially fast as βu_b becomes large, so varying βu_b changes the scale of $k'(\gamma)$ but not its shape. Having both $\kappa \gg 1$ and $\beta u_b \gg 1$ simultaneously tends to result in reaction rate coefficient function $k'(\gamma)$ that is essentially constant before rapidly becoming effectively infinite at and above some critical extension $\gamma_c \gtrsim 1$, similar to the $\kappa = 5000$ case in Fig. 5.4.

5.3.2 Distribution evolution and solution

Now that the single-chain mechanics and kinetics for the uFJC model have been formulated, we next determine the evolution equation for the distribution of intact chains in the network and calculate the stress using Eq. (5.59). We will continue writing our results in terms of the nondimensional chain end-to-end vector $\boldsymbol{\gamma} = \boldsymbol{\xi}/N_b\ell_b$, and we correspondingly utilize the nondimensional distribution $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t) \equiv (N_b\ell_b)^3 P_{\mathrm{A}}(\boldsymbol{\xi};t)$. It is equally probable that any given link will break, since each behaves the same under the Gibbs-Legendre method, so when we begin at equilibrium, we will have the same $P_{\mathrm{B}_j}^{\mathrm{tot}}(t) = P_{\mathrm{B}}^{\mathrm{tot}}(t)/M$ for each j of the $M = N_b$ paths, and therefore we may apply conservation and simplify Eq. (5.58) using

$$\rho(t) \equiv \frac{P_{\mathrm{B}_j}^{\mathrm{tot}}(t)}{P_{\mathrm{B}_j}^{\mathrm{tot,eq}}} = \frac{P_{\mathrm{B}}^{\mathrm{tot}}(t)}{P_{\mathrm{B}}^{\mathrm{tot,eq}}} = \frac{1 - \iiint P_{\mathrm{A}}(\boldsymbol{\xi};t)\, d^3\boldsymbol{\xi}}{1 - \iiint P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})\, d^3\boldsymbol{\xi}} = \frac{1 - \iiint \mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)\, d^3\boldsymbol{\gamma}}{1 - \iiint \mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})\, d^3\boldsymbol{\gamma}}. \tag{5.73}$$

Eqs. (5.58) and (5.73) allows us to retrieve a linear first order integro-partial differential equation for $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)$,

$$\frac{\partial \mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)}{\partial t} = -\left[\frac{\partial \mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)}{\partial \boldsymbol{\gamma}}\boldsymbol{\gamma}\right] : \mathbf{L}(t) - k(\boldsymbol{\gamma})\left\{\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t) - \frac{\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})}{P_{\mathrm{B}}^{\mathrm{tot,eq}}}\left[1 - \iiint \mathscr{P}_{\mathrm{A}}(\tilde{\boldsymbol{\gamma}};t)\, d^3\tilde{\boldsymbol{\gamma}}\right]\right\}, \tag{5.74}$$

where $k(\boldsymbol{\gamma}) \equiv N_b k'(\boldsymbol{\gamma})$ is the net reaction rate coefficient function. The proportionality of the net rate of breaking $k(\boldsymbol{\gamma})$ to the number of links N_b is the effect predicted by Lake and Thomas [187]

and has been used by recent models [150]. This Lake-Thomas effect is also observed in the manner of our Gibbs-Legendre approximation, which causes an equal force to be experienced across all links and the total energy to scale with N_b. Eq. (5.74) appears similar to several from the literature [59, 60, 160, 161, 175], but there are two fundamental differences. First, two evolution equations are typically written – one for the normalized probability $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)/P_{\mathrm{A}}^{\mathrm{tot}}(t)$ and another for the concentration $nP_{\mathrm{A}}^{\mathrm{tot}}(t)$ – rather than the single evolution equation for $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)$ in Eq. (5.74). Second, it is often assumed that one may prescribe both forward and reverse reaction rate coefficients independently, where the forward rate would be the same $k(\boldsymbol{\gamma})$ here, but the reverse rate would be $k_a(\boldsymbol{\gamma})\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})/P_{\mathrm{B}}^{\mathrm{tot,eq}}$. Not only would any $k_a(\boldsymbol{\gamma}) \neq k(\boldsymbol{\gamma})$ violate the statistical mechanical derivation we have outlined in Sec. 5.2.1, namely Eq. (5.28), but it would also cause the equilibrium distribution of the evolution equation $[k_a(\boldsymbol{\gamma})/k(\boldsymbol{\gamma})]\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})$ to differ from the equilibrium distribution from statistical mechanics $\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})$. Separately specifying the forward and reverse reaction rate coefficient functions then causes a thermodynamic inconsistency regardless of the single-chain model used. Conceptually, the kinetic rate(s) at which chemical systems approach equilibrium should not change the equilibrium configuration, since statistical thermodynamics allows equilibrium configurations to be obtained independently of the kinetics.

We present the path to obtain the exact solution to Eq. (5.74) in Appendix C.4. For $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t \leq 0) = \mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})$ and $\mathbf{F}(t \leq 0) = \mathbf{1}$, the solution can be written as

$$\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t) = \int_{-\infty}^{t} \mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\gamma}\right] k\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\gamma}\right] \exp\left\{-\int_{\tau}^{t} k\left[{}_{(t)}\mathbf{F}(s)\cdot\boldsymbol{\gamma}\right]\, ds\right\} \rho(\tau)\, d\tau, \tag{5.75}$$

where the relative deformation [135] is defined as ${}_{(t)}\mathbf{F}(\tau) \equiv \mathbf{F}(\tau)\cdot\mathbf{F}^{-1}(t)$, and where the solution for $\rho(t)$, consistent with its definition in Eq. (5.73) and the solution for $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)$ in Eq. (5.75), is given in Appendix C.4. The stress from Eq. (5.59), which in nondimensional form is

$$\frac{\boldsymbol{\sigma}(t)+p(t)\mathbf{1}}{n/\beta} = N_b \iiint \mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)\,\eta(\gamma)\left(\frac{\boldsymbol{\gamma}\boldsymbol{\gamma}}{\gamma}\right) d^3\boldsymbol{\gamma}, \tag{5.76}$$

can now be evaluated at any time t. As shown in Appendix C.4.2, the relation for the stress obtained when substituting into Eq. (5.75) is objective. For $k(\boldsymbol{\gamma}) \propto k'(\boldsymbol{\gamma})$ and $\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})$ derived from a single-chain model, such as the Morse-FJC, our solution is guaranteed to converge. If one wishes to instead prescribe positive semidefinite functions $k'(\boldsymbol{\gamma})$ and $\mathscr{P}^{\mathrm{eq}}(\boldsymbol{\gamma})$ independently of a chain model, our solution still holds as long as

$$\iiint k'(\boldsymbol{\gamma})\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})\, d^3\boldsymbol{\gamma} < \infty. \tag{5.77}$$

If this condition is not met, not only does our solution not hold, but there is no solution in general, which means numerical methods will also fail. This condition in Eq. (5.77) also keeps the dissipation $\mathcal{D}_{\mathrm{rxn}}(t)$ in Eq. (5.60) finite. Thus, given admissible $k'(\boldsymbol{\gamma})$, $\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})$, and $\mathbf{F}(t)$, we may evaluate $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)$ in Eq. (5.75) when integrating for the stress in Eq. (5.76). Our results here in Sec. 5.3.2 can be applied to other chains that have all identical reaction pathways and are long enough to utilize the Gibbs-Legendre method, or to chains of any length that instead have only a single reaction pathway (i.e. a chain with a single weak link).

5.3.3 Adjustments for inhomogeneous chains

Polymer networks are often synthesized to contain sacrificial bonds that are designed to break or activate before the rest of the bonds in the network [11, 16, 188]. We can adjust our previous relations to accommodate these cases – we consider the same uFJC model, but now with N_b breakable links and $N_b^{\#}$ unbreakable links. The unbreakable links are assumed to remain in the harmonic region (effectively EFJC), have rest-length $\ell_b^{\#}$, and nondimensional stiffness $\kappa^{\#}$. Due to the nature of the Gibbs-Legendre method, we may simply add these links onto our asymptotic approximations for the mechanical response and Helmholtz free energy. Beginning with the mechanical response in Eq. (5.64), for an inhomogeneous chain we now have

$$\gamma(\eta) \sim \frac{N_b}{N_b + \varsigma N_b^{\#}} \left[\mathcal{L}(\eta) + \lambda(\eta) - 1\right] + \frac{\varsigma N_b^{\#}}{N_b + \varsigma N_b^{\#}} \left[\mathcal{L}(\varsigma\eta) + \frac{\varsigma\eta}{\kappa^{\#}}\right], \tag{5.78}$$

where $\varsigma \equiv \ell_b^{\#}/\ell_b$ is the ratio of rest-lengths of the two link types. Note that the contour length of the chain that scales ξ for γ is now $N_b\ell_b + N_b^{\#}\ell_b^{\#} = N_b\ell_b(1 + \varsigma N_b^{\#}/N_b)$. Similarly, we adjust Eq. (5.69) for the nondimensional configurational Helmholtz free energy

$$\left(N_b + N_b^{\#}\right)\vartheta^*_{\mathrm{A,con}}(\gamma) \sim N_b\left[\ln\left\{\frac{\eta\exp[\eta\mathcal{L}(\eta)]}{\sinh(\eta)}\right\} + \beta u(\eta)\right] + N_b^{\#}\left[\ln\left\{\frac{\varsigma\eta\exp[\varsigma\eta\mathcal{L}(\varsigma\eta)]}{\sinh(\varsigma\eta)}\right\} + \frac{(\varsigma\eta)^2}{2\kappa^{\#}}\right], \tag{5.79}$$

The equilibrium distribution $\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\gamma)$ is still given by Eq. (5.70) after taking $N_b\vartheta^*_{\mathrm{A,con}}(\gamma) \mapsto (N_b + N_b^{\#})\vartheta^*_{\mathrm{A,con}}(\gamma)$ and using Eq. (5.79). The forward rate coefficient function $k'(\gamma)$ is still given by Eqs. (5.71)–(5.72), and the net reaction rate coefficient function is still $k(\gamma) = N_b k'(\gamma)$. The nondimensional stress from Eq. (5.76) is now

$$\frac{\boldsymbol{\sigma}(t) + p(t)\mathbf{1}}{n/\beta} = \left(N_b + \varsigma N_b^{\#}\right)\iiint \mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)\,\eta(\gamma)\left(\frac{\boldsymbol{\gamma}\boldsymbol{\gamma}}{\gamma}\right)d^3\boldsymbol{\gamma}. \tag{5.80}$$

The inhomogeneous single-chain mechanical response and reaction rate coefficient function is studied in Fig. 5.5 for varying $\kappa^{\#}$. For these results we use $N_b = 1$, $N_b^{\#} = 8$, $\kappa = 200$, $\beta u_b = 25$, and $\varsigma = 1$. For $\kappa^{\#} < \kappa$, the mechanical response of the chain is dominated by the stretching of the unbreakable links, and the stiffer breakable link is stretched slowly due to the smaller forces reached per overall extension. This in turn causes the reaction rate coefficient function to more gradually increase with extension since the breakable link will require larger chain extensions to experience the forces necessary for it to begin breaking. For $\kappa^{\#} > \kappa$, the mechanical response of the chain is dominated by the stretching of the breakable link, where for $\gamma > 1$ the chain extension becomes localized almost entirely in stretching the breakable link. Due to this stretch localization, the reaction rate coefficient function begins to spike almost instantaneously at $\gamma = 1$. We also observe that for $\kappa^{\#} \gg \kappa$ we may instead take $\kappa^{\#} \to \infty$ to obtain an accurate FJC-based approximation, which is equivalent (after switching the Morse potential to the relevant potential) to many recent models [62, 67, 75]. Interestingly, combining $\kappa^{\#} \to \infty$ with $\varsigma \to \infty$ results in the FJC model that fails instantaneously for some $\gamma_c \lesssim 1$, which is a simplified form of the approach of Vernerey et al. [81].

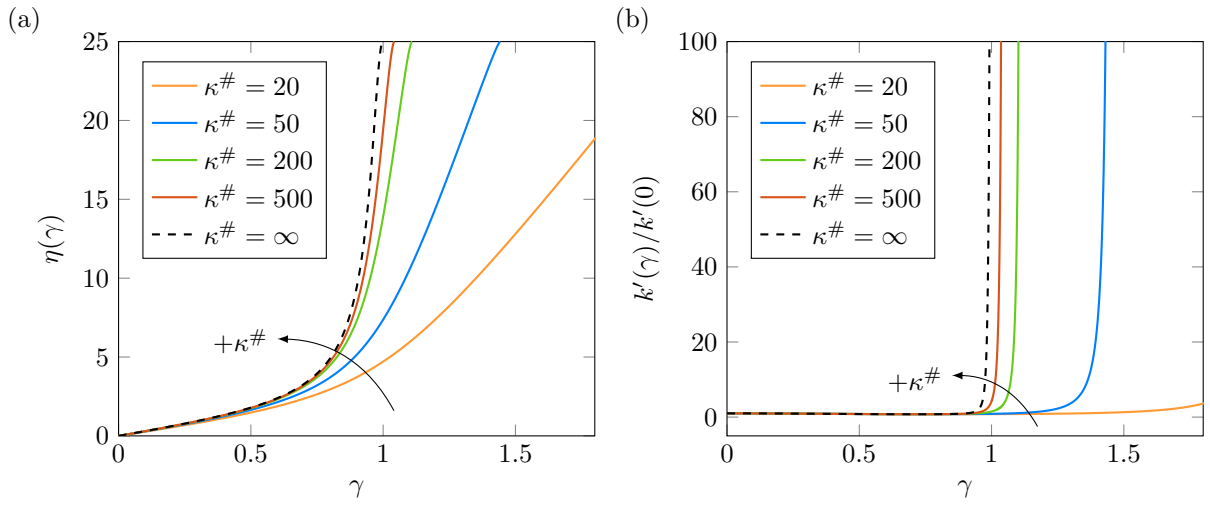


Figure 5.5: Single-chain functions for the inhomogeneous Morse-FJC model with $N_b = 1$, $N_b^\# = 8$, $\kappa = 200$, $\beta u_b = 25$, and $\varsigma = 1$, while varying $\kappa^\#$. (a) The nondimensional force $\eta = \beta f \ell_b$ as a function of the nondimensional end-to-end length $\gamma = \xi/\ell_b(N_b + \varsigma N_b^\#)$. (b) The reaction rate coefficient function $k'(\gamma)$ scaled by its initial value $k'(0)$ as a function of γ.

5.4 Macroscopic results

Now that we have implemented the uFJC single-chain model within our general theory, the resulting constitutive model is applied to predict the mechanics of polymer networks with bond breaking. First, we consider two special cases (1) where chains break rate-independently and irreversibly, and (2) where chains dynamically break and reform in a transient manner. In either case, we compare the theoretical results from the specialized models to experimental results from exemplary polymers in the literature. Next, we demonstrate the features of the general model and examine the results in several parametric studies. We then apply the general model to another polymer that is considered to have force-sensitive reversible chain breaking.

5.4.1 Rate-independent irreversible breaking

The reaction rate coefficient function $k'(\gamma)$ often behaves as being nearly constant at its initial value $k'(0)$ before suddenly becoming effectively infinite beyond some critical extension γ_c, as observed in Figs. 5.4–5.5. Physically, this corresponds to stiff but breakable links remaining unstretched until the entire chain is extended past the contour length, where the high and rapidly increasing force required for further extension almost immediately breaks the chain. When thermal energy alone is insufficient to break the links, we neglect the force-free breaking by letting $k'(0) \to 0$, which then also implicitly neglects reforming. Mathematically, the capability to neglect $k'(0)$ results from the nondimensional link energy βu_b being sufficiently large in order to cause $k'(0) \propto e^{-\beta u_b}$ to become negligible compared to the slowest rate $1/\mathcal{T}$, where $\mathcal{T}$ is the total time of testing. As shown in Appendix C.4.4, the solution for the distribution of intact chains $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma}; t)$ in this special case of rate independent irreversible breaking simplifies to

$$\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma}; t) = \mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}^{-1}(t) \cdot \boldsymbol{\gamma}\right] \Theta(\boldsymbol{\gamma}; t, 0), \tag{5.81}$$

where the yield function $\Theta(\boldsymbol{\gamma};t,\tau)$ is defined as

$$\Theta(\boldsymbol{\gamma};t,\tau) \equiv \begin{cases} 1, & \left\|{}_{(t)}\mathbf{F}(s)\cdot\boldsymbol{\gamma}\right\|_2 \leq \gamma_c \quad \forall s\in[\tau,t], \\ 0, & \text{otherwise.} \end{cases} \tag{5.82}$$

Similar forms of this special case have been considered previously, sometimes accounting for variability in the value of γ_c [81]. Here, if we take $\gamma_c \to \infty$, chains never break and we retrieve $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t) = \mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\gamma}]$, the expected solution for a network of non-breaking chains [125].

This irreversible breaking is especially relevant when considering the sacrificial networks designed to break down within toughened elastomers. These polymers experience noticeable hysteresis under cyclic deformation, exhibiting the Mullins effect [189] and dissipating considerable amounts of energy [190]. Here, we consider the triple ethyl acrylate network ($\mathrm{EA}_{0.5}$EAEA) of Ducrot et al. [11]. The first network (denoted $\mathrm{EA}_{0.5}$) is synthesized using ethyl acrylate monomers and mechanoluminescent crosslinkers that are specifically weaker than the EA links, allowing the damage in the first network to be visualized. The second network and third networks (each denoted EA) are synthesized by swelling the existing network in ethyl acrylate monomers and sparsely crosslinking them. The resulting $\mathrm{EA}_{0.5}$EAEA material, at a temperature of $T = 20°\mathrm{C}$, was loaded in cyclic uniaxial tension while the stress and light emission were measured. Repeat cycles showed negligible light emission and no change in stress, supporting the essential argument that chains in the network effectively break both irreversibly and rate-independently. We model the first network as isotropically-swollen, with the volumetric swelling ratio $J = 15.625$ known from the experiment. The theory presented here may be quickly adjusted to account for this swelling: the equilibrium distribution under the swelling transforms as $\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\gamma) \mapsto \mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(J^{-1/3}\gamma)/J$ due to the isotropic swelling deformation $J^{1/3}\mathbf{1}$, where the factor of J^{-1} preserves the total probability. The number density of chains n transforms under swelling as $n \mapsto n/J$, so if n is known in the pre-swollen configuration, the nondimensional stress from Eq. (5.80) under swelling is

$$\frac{\boldsymbol{\sigma}(t)+p(t)\mathbf{1}}{n/\beta} = \left(N_b + \varsigma N_b^{\#}\right) \iiint J^{-2}\,\mathscr{P}_{\mathrm{A}}(J^{-1/3}\boldsymbol{\gamma};t)\,\eta(\gamma)\left(\frac{\boldsymbol{\gamma}\boldsymbol{\gamma}}{\gamma}\right) d^3\boldsymbol{\gamma}. \tag{5.83}$$

The nondimensional modulus can be shown to be $3J^{-1/3}$ for long chains, which is expected for the isotropic swelling of a network [151]. The first network is modeled as a network of chains of $N_b = 1$ irreversibly-breaking links and $N_b^{\#}$ unbreakable links. The $\mathrm{EA}_{0.5}$ material was reported to have a modulus of 0.6 MPa, which corresponds to $n/\beta = 0.2$ MPa for the first network. The second and third networks were over 100 times more sparsely crosslinked than the first, so we treat them as one effective filler network represented by the Neo-Hookean model, valid when chains are sufficiently long [125]. Since the $\mathrm{EA}_{0.5}$EAEA material was reported to have a modulus of 1.5 MPa, we obtain $n/\beta = 0.3$ MPa for the filler network. The average number of monomers between mechanoluminescent crosslinkers in the first network was approximately 34, and the crosslinker itself offers some additional effective monomers, so we use $N_b^{\#} = 38$. We take $\beta u_b = 61.57$ corresponding to 150 kJ/mol, the zero-force bond energy of the mechanoluminescent crosslinker [188]. We find a good fit to the overall mechanical response for $\kappa = 9000$, $\kappa^{\#} = 6000$, and $\varsigma = 4$, where $\varsigma > 1$ represents that the bond breaking within the mechanoluminescent crosslinker is short compared to the monomer backbone length. These large stiffness values and the small effective Kuhn length resulting from a link representing a single monomer are reasonable, as similar parameters have been used to fit the EFJC model to AFM experimental results for other acrylate chains [191,

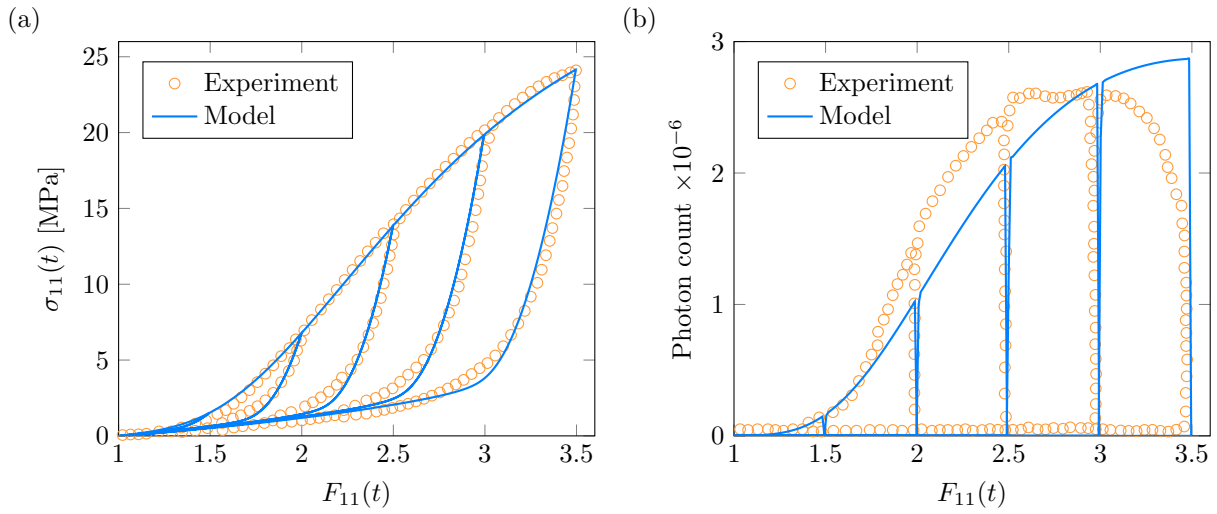


Figure 5.6: (a) Mechanical response of the triple ethyl acrylate network ($EA_{0.5}EAEA$) of Ducrot et al. [11] under cyclic uniaxial tension, and that predicted by the rate-independent irreversibly-breaking model. (b) Light emission from the sacrificial mechanoluminescent crosslinkers breaking within the first network in the same experiment, and that predicted by the model.

192]. The critical chain extension $\gamma_c = 1.17$ results from the intact limit of the chain at $\eta_{\max} = 263.18$. The predicted mechanical response of the material under cyclic uniaxial tension is shown in Fig. 5.6(a) along with the experimental results. We find a good overall agreement between the prediction of our theory and the mechanical response of the material but find some difficulty precisely capturing the unloading curve shape, a somewhat common issue when modeling this material [74, 78, 81, 151]. In addition to the mechanical response, we use our theory to predict the light intensity measured experimentally as the mechanoluminescent crosslinkers break by assuming that the intensity is proportional to the rate at which chains break in the first network [74]. The intensity would then be proportional to $-\frac{d}{dt}P_{\mathrm{A}}^{\mathrm{tot}}(t)$, and we find a good fit for a proportionality constant of 10^9 photon·seconds. The theoretical prediction is shown in Fig. 5.6(b) along with the experimental results, where we observe reasonable agreement but an important shape difference. Specifically, our theory predicts a more gradual breakdown of the first network than is observed in the experiment. There are several possible explanations for this discrepancy that we cannot distinguish among within the current framework. First, it could be due to the polydispersity that is present within the first network, which has been modeled here as effectively monodisperse. Second, the assumption that the distribution of chains in the first network swell isotropically as the filler network is introduced could be invalid for this large degree of swelling. Third, the breaking of chains in the first network could induce a more complicated damage mechanism that involves the filler networks [155]. Fourth, the discrepancy could be related to transfer reactions that create additional crosslinks during synthesis [11].

5.4.2 Transient breaking

When the initial value of the net reaction rate coefficient function $k_0 \equiv k(0)$ is appreciable and the critical extension γ_c is large enough to be neglected, we obtain a constant reaction rate coefficient $k(\gamma) = k_0$ over all chain extensions. This idea is traditionally referred to as the transient network

model [59, 60]. As shown in Appendix C.4.4, the solution for the distribution of intact chains $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)$ in this special case simplifies to

$$\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)=\int_{-\infty}^{t}\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\gamma}\right]k_0e^{-k_0(t-\tau)}\,d\tau. \tag{5.84}$$

The transient network model has been considered previously in several forms, such as for a Gaussian distribution of freely-jointed chains [60] or the Neo-Hookean representation retrieved when using ideal chains [193]. The solution in Eq. (5.84), however, is more general since it is independent of single-chain model. We would like to emphasize several vital features of this special transient case. First, we are now limited to the utilization of single-chain models that are infinitely extensible since we have neglected γ_c. Here we will simply replace the uFJC model with the EFJC model, which is infinitely extensible. Second, note that we have taken $\rho(t)=1$, meaning that the total fraction of intact chains $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ remains constant at $P_{\mathrm{A}}^{\mathrm{tot,eq}}$ for all time, which is true here (see Appendix C.4.4). The decoupling of single-chain kinetics from chain extension and the constant fraction of intact chains allows us to write the stress as

$$\boldsymbol{\sigma}(t)=\int_{-\infty}^{t}{}_{(t)}\boldsymbol{\sigma}(\tau)\,k_0e^{-k_0(t-\tau)}\,d\tau, \tag{5.85}$$

where ${}_{(t)}\boldsymbol{\sigma}(\tau)$ is the stress that the network, if permanent, experiences under the relative deformation ${}_{(\tau)}\mathbf{F}(t)$. Using Eq. (5.76), we write this stress in nondimensional form as

$$\frac{{}_{(t)}\boldsymbol{\sigma}(\tau)+{}_{(t)}p(\tau)\mathbf{1}}{n/\beta}=N_b\iiint\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\gamma}\right]\,\eta(\gamma)\left(\frac{\boldsymbol{\gamma}\boldsymbol{\gamma}}{\gamma}\right)\,d^3\boldsymbol{\gamma}. \tag{5.86}$$

This simplification allows us to eliminate considerable computational expense: for a given model, we may interpolate from tabulated values of ${}_{(t)}\mathbf{F}(0)$ and ${}_{(t)}\boldsymbol{\sigma}(0)$ in order to rapidly perform the integration over the history in Eq. (5.85). In the ideal chain limit ($N_b\to\infty$) we obtain the Neo-Hookean model [125], where the right-hand side of Eq. (5.86) becomes ${}_{(\tau)}\mathbf{F}(t)\cdot{}_{(\tau)}\mathbf{F}^T(t)$.

It is common to consider a network that consists of both permanent and transiently-bonded chains [166]. In this case, with the fraction of permanent chains $0\le x_p\le 1$, Eq. (5.85) becomes

$$\boldsymbol{\sigma}(t)=x_p\,{}_{(t)}\boldsymbol{\sigma}(0)+(1-x_p)\int_{-\infty}^{t}{}_{(t)}\boldsymbol{\sigma}(\tau)\,k_0e^{-k_0(t-\tau)}\,d\tau \tag{5.87}$$

We apply Eq. (5.87) under the ideal chain limit to the polyvinyl alcohol (PVA) gel with both permanent and transient crosslinks in Long et al. [167] using $k_0=0.37/\mathrm{s}$, $x_p=16\%$, and $n/\beta=24.15$ kPa. Two cycles of uniaxial tension at a rate of $0.03/\mathrm{s}$ are applied to the material with $\mathcal{T}_{\mathrm{wait}}=30$ min between cycles. Since $k_0\mathcal{T}_{\mathrm{wait}}\gg 1$, our transient model will fully relax between cycles and exactly repeat the mechanical response of the first cycle, as observed in experiment. The results in Fig. 5.7(a) indicate show reasonable overall agreement, but there are discrepancies near the beginning of the loading and unloading potion of the cycles. These results seem to indicate that the single-timescale approach of the transient network model is effective at short and long times, but that many timescales are required to capture the full material behavior. In order to generalize for many timescales, we can adjust Eq. (5.87) to

$$\boldsymbol{\sigma}(t)=x_p\,{}_{(t)}\boldsymbol{\sigma}(0)+(1-x_p)\int_{-\infty}^{t}{}_{(t)}\boldsymbol{\sigma}(\tau)\frac{\partial g(t-\tau)}{\partial\tau}\,d\tau, \tag{5.88}$$

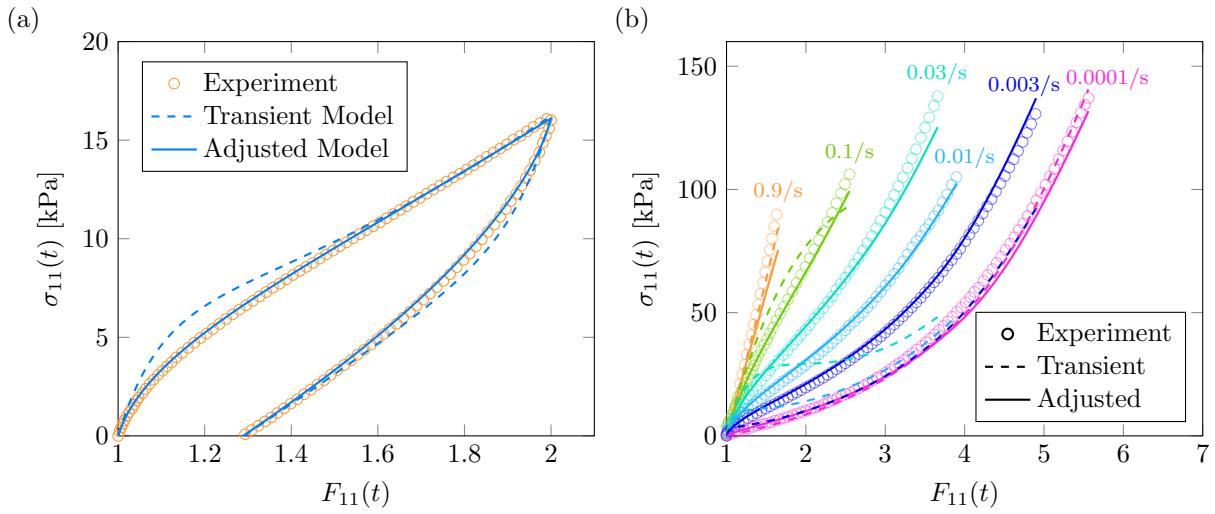


Figure 5.7: (a) Mechanical response of the PVA gel in Long et al. [167] under cyclic uniaxial tension at a rate of 0.03/s, with predictions provided by the transient network model and the model adjusted to use a relaxation function. (b) Mechanical response of the PVA gel of Mayumi et al. [46] under monotonic uniaxial tension at varying rates, with predictions provided again by both models.

where $g(t)$ is any relaxation function. Taking $g(t) = e^{-k_0 t}$ recovers the original transient network model. We instead utilize a non-exponential relaxation function [167] which in effect represents many timescales,

$$g(t) = \left[1 + (\alpha - 1)\frac{t}{t_R}\right]^{1/(1-\alpha)}, \tag{5.89}$$

where $\alpha > 1$ and t_R is the characteristic bond breaking time. We utilize the parameters reported by Guo et al. [168] for our adjusted model, which in our case are $\alpha = 2.6$, $t_R = 0.6$ s, $x_p = 10\%$ and $n/\beta = 24.15$ kPa. The resulting mechanical response in Fig. 5.7(a) is a near-perfect fit and a substantial improvement upon the transient model.

The PVA gel we are considering was studied under large deformation and over 4 orders of magnitude of strain rates by Mayumi et al. [46]. Our parameters change here since this material system is sensitive to synthesis conditions, often causing mterial parameters to change from batch-to-batch [167]. These experiments were modeled by Guo et al. [168], from which we obtain the relaxation function parameters $\alpha = 1.99$, $t_R = 3.23$ s, $x_p = 4.68\%$ and $n/\beta = 37.78$ kPa. We utilize Eqs. (5.85) and (5.86) with the EFJC model, fitting $N_b = 50$ and $\kappa = 40$ to the exponential hardening model used by Guo et al. [168]. For the transient network model we use the same $N_b = 50$ and $\kappa = 40$, but find $x_p = 5.85\%$ and $k_0 = 0.12$/s by fitting to the 0.0001/s and 0.1/s rate results, respectively. The modeling results in each case are shown in Fig. 5.7(b) with the experimental data. Overall, the adjusted model provides a reasonable prediction of the mechanical response over all strain rates. We find that the transient network model tends to perform poorly here at intermediate rates and thus in modeling the mechanical response of more dynamic networks at intermediate timescales. We attribute this to the insufficiency of the single timescale of the transient model in capturing the many timescales of the material observed in experiment.

5.4.3 General behavior

We now examine the behavior of the general model in comparison to the special cases we have just outlined, rate-independent reversible breaking and transient breaking. The Morse-FJC model is used in each case with the same parameters, apart from the approximations made to the reaction rate coefficient function for the special cases. The critical extension γ_c for both special cases is taken to be $\gamma_{\max} = 1.146$, the maximum extension where the chain remains intact. For the version of the transient model that does not neglect γ_c, see Appendix C.4.4. Using an exemplary set of parameters, we apply a series of ramps of rate $\dot{\epsilon}$ and holds to the deformation gradient, as shown in Fig. 5.8(a) as a function of the nondimensional time $\dot{\epsilon}t$. The total probability that a chain is intact, $P_{\mathrm{A}}^{\mathrm{tot}}(t)$, under this deformation is shown in Fig. 5.8(b). While a similar fraction of chains break under the first loading period for each case, a significant amount of reforming occurs under the following holding period for the general and transient case, in contrast with the irreversible case where reforming is neglected. In all cases, the unloading periods in the second half of the deformation history break a negligible fraction of chains. Overall, the transient case seems to provide a reasonable approximation of the general case for $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ here. The stress in Fig. 5.8(c), however, is substantially different for the two special cases versus the general case. Due to their neglect of the nontrivial shape of $k(\gamma)$, either special case underestimates the fraction of high-extension chains being broken and therefore tends to overestimate the stress. The reforming process negligibly affects the stress here since chains tend to reform towards $\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})$, i.e. the stress-free configuration, leading to a surprisingly similar stress response for the transient and irreversible models. During the holding periods we observe much more substantial stress relaxation in the general case compared to little in the transient case, which is again due to $k(\gamma)$ providing a region where chains break quickly but not instantaneously. This effect is clearly apparent from examining $\mathscr{P}_{\mathrm{A}}(\gamma_1, 0, 0; t)$ after the second hold period in Fig. 5.8(d), where γ_1 is the component of $\boldsymbol{\gamma}$ aligned with the loading direction. Fig. 5.8(d) shows that the special cases overestimate the fraction of chains at larger extensions and therefore both the stress and the rate of breaking chains. This higher rate of breaking in the general case provides a higher rate of reforming, which is why the general case shows the highest probability of chains near $\gamma = 0$.

Parametric studies

Next we conduct parametric studies to understand the dependencies of the general model. In each case we apply one cycle of uniaxial tension with strain rate $\dot{\epsilon}$ and maximum stretch 9, plotting the results as a function of the nondimensional time $\dot{\epsilon}t$. The nondimensional base parameters are $N_b = 1$, $N_b^{\#} = 8$, $\kappa = 200$, $\kappa^{\#} = 500$, $\varsigma = 1$, $\beta u_b = 100$, $\beta\Delta\Psi_0 = 5$, and $k_0/\dot{\epsilon} = 1/100$. First, we vary the nondimensional initial reaction rate $k_0/\dot{\epsilon}$, or equivalently, the nondimensional strain rate $\dot{\epsilon}/k_0$. Fig. 5.9(a) shows the nondimensional stress $\beta\sigma_{11}(t)/n$ as a function of the nondimensional time $\dot{\epsilon}t$. As k_0 increases relative to $\dot{\epsilon}$, we see a decrease in the time it takes for the network to yield as chains break more rapidly. Increasing k_0 also causes broken chains to reform more rapidly, which causes an increasing amount of compressive stress when fully returning the network to zero strain. This compressive stress results from chains reforming towards their stress-free equilibrium distribution while the deformation is still being applied to the network. The rate-independent irreversibly-breaking special case ($k_0 = 0$) is shown for reference, which still differs considerably from the lowest k_0 case where $k_0/\dot{\epsilon} = 10^{-4}$. For additional insight, we plot the fraction of intact chains within the network $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ as a function of the nondimensional time $\dot{\epsilon}t$ in Fig. 5.9(b). As k_0 increases relative to

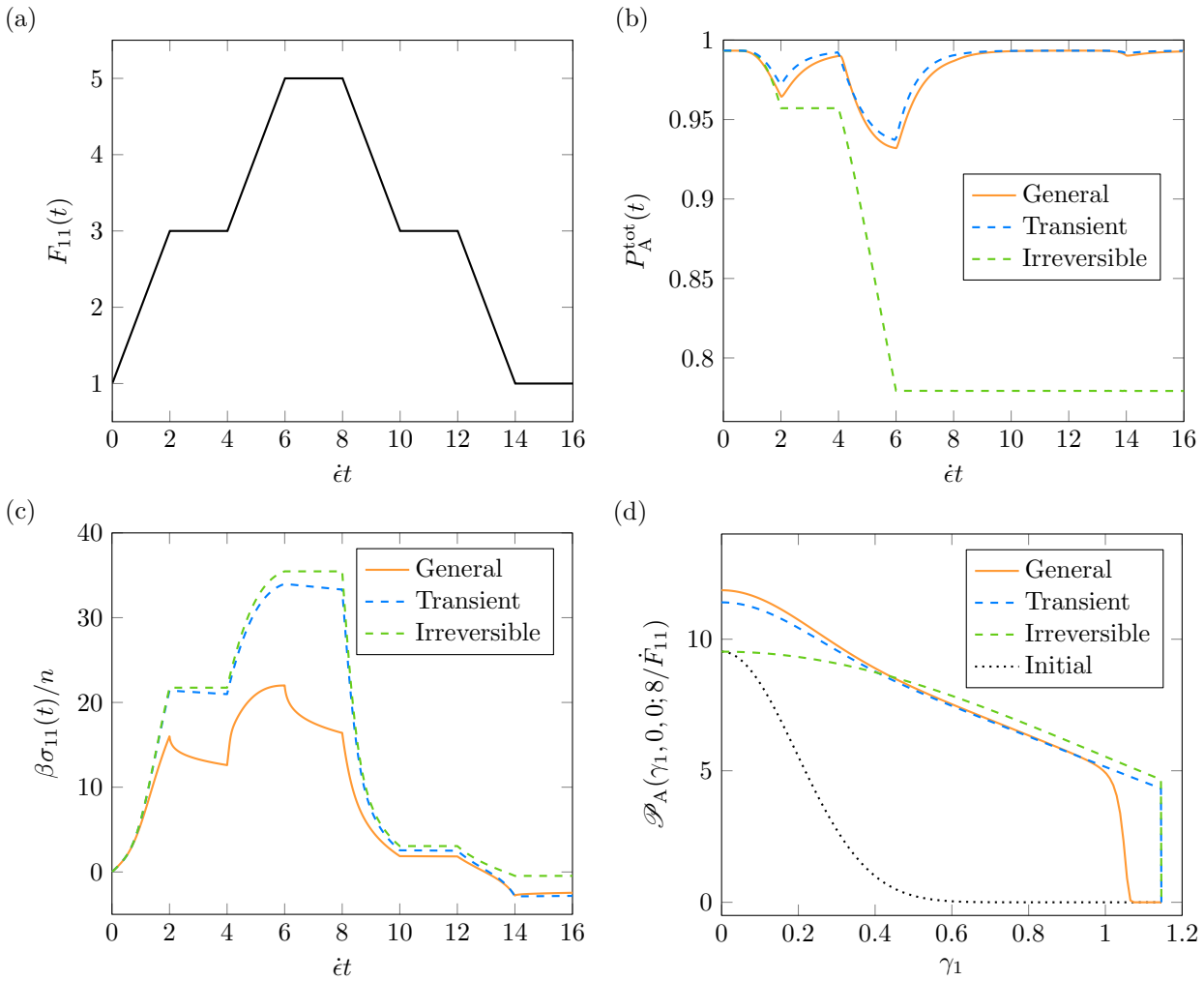


Figure 5.8: (a) The applied deformation $F_{11}(t)$ as a function of nondimensional time $\dot{\epsilon}t$, where $\dot{\epsilon}$ is a constant and the stress is uniaxial tension. (b) The total probability that a chain is intact, $P_{\mathrm{A}}^{\mathrm{tot}}(t)$, as function of nondimensional time for the general model and the two special cases. (c) The applied nondimensional stress $\beta\sigma_{11}(t)/n$ as a function of nondimensional time for the same cases. (d) The distribution of intact chains $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma}; t)$ aligned with the loading direction (along the γ_1-axis) at the halfway point $t = 8/\dot{\epsilon}$. The nondimensional parameters are $N_b = 1$, $N_b^{\#} = 8$, $\kappa = 200$, $\kappa^{\#} = 500$, $\varsigma = 1$, $\beta u_b = 100$, $\beta\Delta\Psi_0 = 5$, and $k_0/\dot{\epsilon} = 1/100$, with $\gamma_c = \gamma_{\max} = 1.146$ for the special cases.

$\dot{\epsilon}$, we see that chains reform more rapidly and correspondingly, more chains are intact at any time. When k_0 becomes sufficiently large it appears that $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ remains approximately constant at its equilibrium value $P_{\mathrm{A}}^{\mathrm{tot,eq}}$, however, this is not exactly true (see Appendix C.4.4). We see that even when k_0 is quite small compared to $\dot{\epsilon}$, reforming still takes place after a sufficient amount of chains are broken. This is because the total reforming rate is proportional to $\rho(t) = [1 - P_{\mathrm{A}}^{\mathrm{tot}}(t)]/P_{\mathrm{B}}^{\mathrm{tot,eq}}$, which spikes if appreciable amount chains break when $P_{\mathrm{B}}^{\mathrm{tot,eq}}$ is small, e.g. here where we have $P_{\mathrm{B}}^{\mathrm{tot,eq}} = 6.69 \times 10^{-3}$.

Second, we examine the behavior of the general model while varying the nondimensional stiffness of the unbreakable links $\kappa^{\#}$ and keeping that of the breakable link constant, $\kappa = 200$. Referring back to the single-chain mechanical response and the reaction rate coefficient function in Fig. 5.5, we recall that while increasing $\kappa^{\#}$ stiffens the chain near its full extension, it causes the chain to break more

rapidly and reduces the maximum extensibility due to an increasing amount of strain localization in the breakable link. Fig. 5.9(c) shows the nondimensional stress $\beta\sigma_{11}(t)/n$, where we see that the more rapid breaking resulting from increasing $\kappa^{\#}$ is manifested as more rapid yielding of the network. While larger $\kappa^{\#}$ causes the stress to increase at small deformations, smaller $\kappa^{\#}$ enables much higher stresses to be reached at large deformations due to increased maximum chain extensibility. The fraction of intact chains, shown in Fig. 5.9(d), verifies that as $\kappa^{\#}$ increases, chains break more rapidly as the deformation is applied. Overall, these results illustrate that for a given breakable bond, maximizing single-chain extensibility may be much more effective than maximizing chain backbone stiffness when it comes to strengthening networks. These results additionally illustrate that force-driven breaking of chains within the network is substantially increased after ensuring that the breakable bond is far less stiff than the rest of the chain. Lastly, we find that for $\kappa^{\#} \gg \kappa$ utilizing the relevant rigid-constraint single-chain model ($\kappa^{\#} = \infty$, which is the FJC model here) captures both single-chain results and macroscopic-level results.

Third, we examine the behavior of the general model while varying the number of unbreakable links $N_b^{\#}$. We keep the number of breakable links $N_b = 1$ constant, so we effectively vary the chain length. Our results are independent of whether the number density of chains, n, or number density of unbreakable links, $nN_b^{\#}$, is kept constant as $N_b^{\#}$ increases since we use the nondimensional stress. As $N_b^{\#}$ increases, the nondimensional stress, shown in Fig. 5.9(e), follows a similar trend we observed in Fig. 5.9(c) when decreasing $\kappa^{\#}$. Longer chains require less force to have the same end-to-end length as shorter chains, and additionally the average nondimensional end-to-end length at equilibrium decreases as chains become longer [50, 125]. Combined, these two effects allow a network of longer chains to deform more without breaking down and thus reach higher nondimensional stresses without yielding. This is verified by examining the fraction of intact chains, shown in Fig. 5.9(f), where we see less overall breaking as chains become longer. The results in the ideal chain limit ($N_b^{\#} = \infty$) are also shown in both Figs. 5.9(e) and 5.9(f), and are equivalent to the results obtained when using the Neo-Hookean model [125]. As $N_b^{\#}$ increases, the results of the general model matches the Neo-Hookean model for an increasing amount of time, but continued deformation of the network always causes the two to diverge as the finite-length chains in the general model stiffen and begin to break.

Fourth, we examine the behavior of the general model while varying the nondimensional breakable link energy βu_b. Referring back to the single-chain mechanical response in Fig. 5.3(b), we recall that increasing βu_b caused the maximum nondimensional force that a chain could support, $\eta_{\max} \equiv \sqrt{\kappa\beta u_b/8}$, to increase. We also recall that the overall mechanical response away from $\eta_{\max}$ and the reaction rate coefficient function tended not to change when varying βu_b. Fig. 5.10(a) shows the nondimensional stress $\beta\sigma_{11}(t)/n$ as a function of the nondimensional time $\dot{\epsilon}t$. As βu_b increases, the shape of the curve remains relatively unchanged while the overall stress level increases due to increasing $\eta_{\max}$. The fraction of intact chains within the network $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ as a function of the nondimensional time $\dot{\epsilon}t$ is shown in Fig. 5.10(b). From $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ we see that the breakdown of the network is only mildly lessened by increasing βu_b, which can be further understood after reconsidering the mechanical response in Fig. 5.3(b) and observing that the maximum extensibility $\gamma_{\max}$ also only mildly increases with βu_b. While these results show that the network is strengthened by increasing the energy of the breakable bond, it is unlikely that this energy would be controllable in the range of many factors of $\mathfrak{b}T = 1/\beta$. Correspondingly, the mechanics of the network are relatively insensitive to the breakable bond energy.

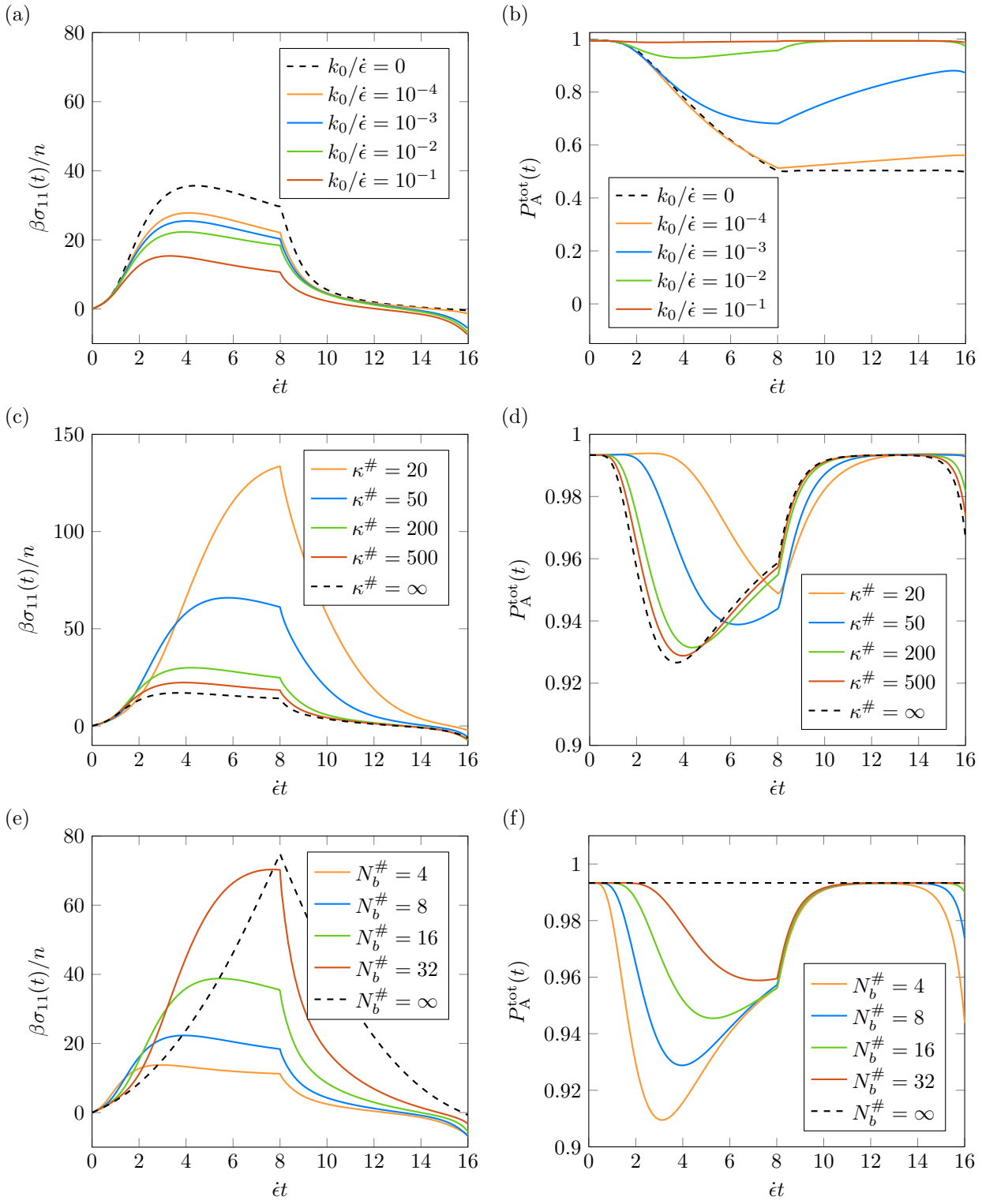


Figure 5.9: Parametric studies concerning rates and the unbreakable links, where we vary (a,b) $k_0/\dot{\epsilon}$, the nondimensional initial reaction rate, (c,d) $\kappa^{\#}$, the nondimensional stiffness of the unbreakable links, and (e,f) $N_b^{\#}$, the number of unbreakable links, while keeping the number of breakable links, $N_b = 1$, constant. For one cycle of uniaxial monotonic tension, the nondimensional stress, $\beta\sigma_{11}(t)/n$, and total probability that a chain is intact, $P_{\mathrm{A}}^{\mathrm{tot}}(t)$, are plotted as a function of the nondimensional time $\dot{\epsilon}t$.

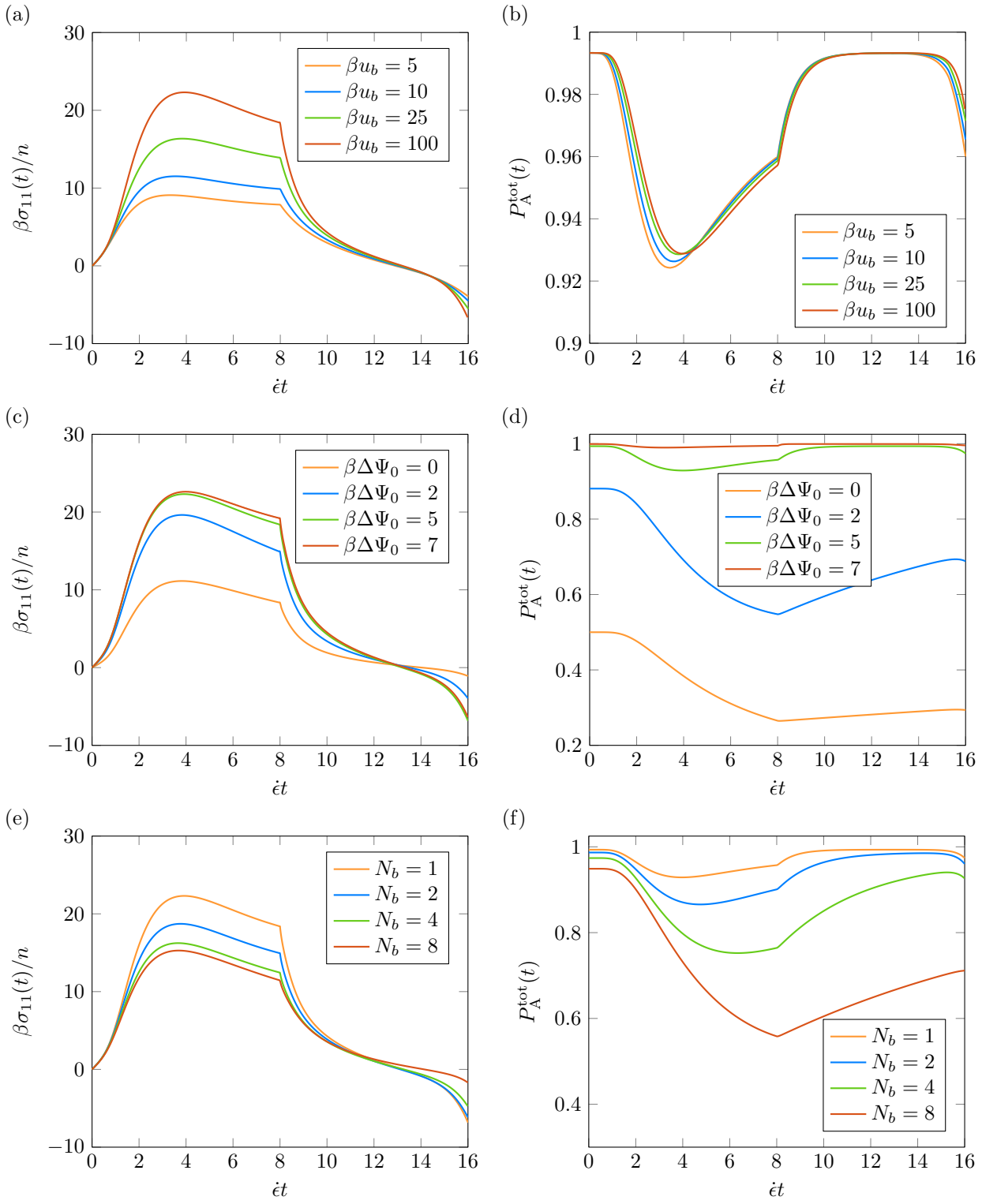


Figure 5.10: Parametric studies concerning the breakable link(s), where we vary (a,b) βu_b, the nondimensional breakable link energy, (c,d) $\beta\Delta\Psi_0$, the net Helmholtz free energy change when breaking a chain, and (e,f) N_b, the number of breakable links, while keeping the total number of links, $N_b + N_b^{\#} = 9$, constant. For one cycle of uniaxial monotonic tension, the nondimensional stress, $\beta\sigma_{11}(t)/n$, and total probability that a chain is intact, $P_{\mathrm{A}}^{\mathrm{tot}}(t)$, are plotted as a function of the nondimensional time $\dot{\epsilon}t$.

Fifth, we examine the behavior of the general model while varying $\beta\Delta\Psi_0$, the net Helmholtz free energy change when breaking a chain. This parameter directly controls the total probability at equilibrium that a chain is intact, $P_{\mathrm{A}}^{\mathrm{tot,eq}}$, or broken, $P_{\mathrm{B}}^{\mathrm{tot,eq}} = 1 - P_{\mathrm{A}}^{\mathrm{tot,eq}}$, via Eq. (5.70). Fig. 5.10(c) shows the nondimensional stress, where we see that increasing $\beta\Delta\Psi_0$ decreases the initial modulus (roughly $3nP_{\mathrm{A}}^{\mathrm{tot,eq}}/\beta$) as well as the overall stress. Both trends are almost entirely due to decreasing $P_{\mathrm{A}}^{\mathrm{tot,eq}}$: when scaling the nondimensional stress by $P_{\mathrm{A}}^{\mathrm{tot,eq}}$, the curves collapse on one another (not shown). Although it is not observed in the stress, $\beta\Delta\Psi_0$ also has a strong effect on the rate of reforming. The net reverse reaction rate coefficient function, $k(\boldsymbol{\gamma})\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})/P_{\mathrm{B}}^{\mathrm{tot,eq}}$, multiplies $P_{\mathrm{B}}^{\mathrm{tot}}(t)$ in Eq. (5.74). The rate of breaking relative to reforming is then $\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})/P_{\mathrm{B}}^{\mathrm{tot,eq}}$, which becomes large as $\beta\Delta\Psi_0$ increases. The stress is mostly unaffected by the rate of reforming since chains reform towards the equilibrium distribution $\mathscr{P}_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\gamma})$ where they tend not to contribute to the overall stress. Importantly, this reforming towards the stress-free equilibrium distribution does not appear to provide any appreciable stress reduction, although more rapid reforming does cause an increasing amount of compression when the strain is returned to zero. The fraction of intact chains, shown in Fig. 5.10(d), allows us to better examine the effects of the rate of reforming broken chains. As $\beta\Delta\Psi_0$ increases, more rapid reforming prevents a large percentage of broken chains to be reached at any time, and the equilibrium value (which also increases) is more quickly recovered after the loading portion. When $\beta\Delta\Psi_0$ is sufficiently large, the reforming of broken chains becomes so rapid that $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ appears to remain constant at its equilibrium value. This result might lead one to believe that for $\beta\Delta\Psi_0 \gg 1$, $\rho(t) = P_{\mathrm{B}}^{\mathrm{tot}}(t)/P_{\mathrm{B}}^{\mathrm{tot,eq}}$ could be approximated as $\rho(t) \sim 1$ within the solution for $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)$ in Eq. (5.75), thus avoiding computing the solution for $\rho(t)$ in Appendix C.4. Such an approximation fails entirely: $\rho(t)$ spikes and decays rapidly as chains break and are immediately reformed, playing a crucial role in the solution for $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)$. Taking $\rho(t) \sim 1$ then retrieves entirely different results where reforming is vastly under-predicted and $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ does not actually remain constant (not shown). This is unfortunate, because cases where $\beta\Delta\Psi_0 \gg 1$ tend to be the most computationally expensive and represent the cases where chains require considerable force to break but reform quickly afterward.

Sixth and lastly, we examine the behavior of the general model while varying the number of breakable links N_b. We keep the total number of links $N_b + N_b^{\#} = 9$ constant, so the chain length remains constant and the number of unbreakable links $N_b^{\#}$ varies accordingly. In effect, this varying N_b while keeping $N_b + N_b^{\#}$ constant varies the fraction of the chain that is breakable. The stress in Fig. 5.10(e) decreases mildly as N_b increases while retaining the same overall shape. This stress decrease is mostly due to chains breaking more quickly as N_b increases since the net reaction rate coefficient function is still $k(\gamma) = N_b k'(\gamma)$ scales directly with N_b. It is also due to the equilibrium fraction of intact chains $P_{\mathrm{A}}^{\mathrm{tot,eq}}$ decreasing as N_b increases, i.e. Eq. (5.70). This is directly observed in Fig. 5.10(f), where the evolution of the fraction of intact chains in the network, $P_{\mathrm{A}}^{\mathrm{tot}}(t)$, is shown. As N_b increases, chains break more rapidly and allow a larger fraction of chains to be broken overall. The total rate of reforming seems to increase with N_b, but this is simply due to more chains being broken, driving faster reforming due to the system being further out of equilibrium. Our results here verify that including more breakable bonds within a chain of a fixed contour length causes substantially increased bond breaking under a given deformation.

General model application

Now that our parametric study is finished, we apply the general model to a material system from the literature with force-sensitive reversible crosslinks. A tough, self-recovering hydrogel was synthesized by Zheng et al. [194] using metal-coordination complexes as reversible crosslinks. This system was modeled by Lin et al. [165], where the crosslink breaking rate was taken to increase as the network experienced more stress to account for force-sensitive breaking. We obtain parameters for our model as follows. The hydrogel was synthesized with 10% mole fraction of crosslinking monomers, so when taking $N_b = 1$ we obtain $N_b^{\#} = 9$ after taking each link to represent a monomer. We obtain $n/\beta = 0.48$ MPa from half of the reported shear modulus, 0.96 MPa. We take $k_0 = 2 \times 10^{-4}$/s from the force-free rate of breaking obtained for the model of Lin et al. [165], and similarly take $\beta\Delta\Psi_0 = 8.55$ in order to match total reforming rate of the model, $\hat{K} = 1$/s. Otherwise, we find $\varsigma = 1$, $\beta u_b = 100$, $\kappa = 200$, and $\kappa^{\#} = 400$ provides the best fit. The stress as a function of applied stretch is shown in Fig. 5.11 for one cycle of uniaxial monotonic tension, where different tests are performed to different maximum stretches. For the first test (to a stretch of 2.5), we find that our model accurately predicts the loading curve from the experiment but upon unloading overpredicts the recovery and thus underpredicts the dissipation. For the second test (to a stretch of 4), the model begins to better predict the growing amount of dissipation, but still overpredicts the recovery and begins to yield. For the third test (to a stretch of 5.5), the model continues to yield and deviates strongly from the experimental loading curve. Overall, since our model cannot create large amounts of dissipation without significantly breaking down the network, it is unable to capture the mechanical response observed in experiment due to the accompanying significant yielding. We attribute this to our model not accounting for the viscous deformation resulting from broken portions of the network freely flowing before reforming. This was not encountered earlier in Sec. 5.4.1 when modeling the multinetwork elastomer since the secondary networks provided integrity while the sacrificial network broke down. This viscous flow was included in the model of Lin et al. [165] and allowed it to make more accurate predictions. Here, the viscous flow would allow for increased dissipation without requiring the significant network breakdown that creates artificial yielding, and additionally would reduce the predicted amount of recovery.

5.5 Conclusion

We have accomplished a fundamental derivation leading to a constitutive model for the stress-strain behavior of elastomers with chain breaking that properly incorporates the statistical mechanics of a general single-chain model. We have shown that the single-chain mechanical response, the breaking and reforming kinetics, and the equilibrium distribution of chains in the network are all inextricably determined by the single-chain model Hamiltonian. Our meticulous formulation was seamlessly brought to the continuum scale, where we obtained the Cauchy stress entirely in terms of the applied deformation, the temperature, the network chain density, and the single-chain model. We showed that the second law of thermodynamics was automatically arbitrarily satisfied as a consequence of our statistical mechanical treatment. We introduced, developed, and studied the potential-supplemented freely jointed chain (uFJC) model. We then accomplished a valuable exact solution for the evolving distribution of chains in the network, applicable beyond the uFJC model.

Next, we developed two special cases of our constitutive model and applied them to exemplary polymer materials from the literature. In the case of the multinetwork elastomer, the rate-

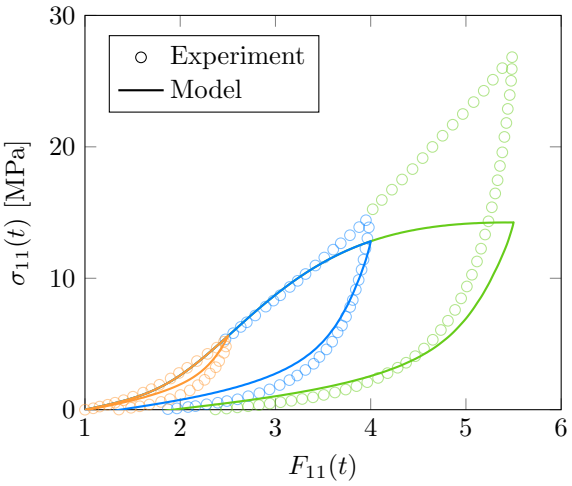


Figure 5.11: Stress as a function of stretch for one cycle of uniaxial monotonic tension, repeated for different maximum stretches. The experimental results for the metal-coordinated gel of Lin et al. [165] are shown with those predicted by the general model.

independent irreversible theory was mostly successful in modeling both the mechanical response of the material and the breakdown of the sacrificial network. In the case of the dual crosslink gel, the single timescale of the transient network model was mostly unable to capture the rate-dependent mechanical response of the material. We then examined the behavior of our constitutive model in the general case, where we contrasted it with the two special cases and performed several parametric studies to highlight how material performance could be tuned. Afterward we applied the general model to a hydrogel with force-sensitive reversible metal-coordinated crosslinks, where we found that the model was unable to capture the toughness of the material without artificially yielding. Overall, we observed the following: while chain breaking is a dominant feature in the large deformation of many elastomeric systems, related phenomena often become similarly important as the network breaks down, limiting the success of our current approach. For example, breaking bonds could increase the effective contour length of chains rather than simply reduce the amount of intact chains, similar to network alteration theory [195, 196]. This idea is supported by a recent molecular dynamics study, which determined that network breakdown in the same multinetwork elastomer we considered here was strongly tied to the evolving shortest contour length between crosslinks [197]. Another example: significant network breakdown – especially in the case of a single network material – seems to be accompanied by additional viscous flow not included in our model [165]. Nonetheless, our existing theory provides a strong foundation to build upon and include additional physical mechanisms. The presented approach is a robust method to obtain macroscopic constitutive functions in terms of molecular functions and parameters.

Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant No. CAREER-1653059. Calculations in this work used the Extreme Science and Engineering Discovery Environment (XSEDE) `Stampede2` cluster [124], which is supported by National Science Foundation Grant ACI-1548562.

Chapter 6

Conclusions and future outlook

This dissertation focused primarily on developing continuum constitutive models for elastomeric materials starting from the statistical mechanics of single polymer chains. These physically-founded models were developed to capture the large-deformation mechanical behavior of an emerging set of materials which contain polymer networks that are designed to benefit from bond breaking. The successful development and application of a general model framework for these materials constitute the most substantial advances made by this dissertation. Additional advances made by this dissertation were: a study of the important connections in the single-chain statistical mechanical description; a combined experimental and theoretical study of a metallopolymer using density functional theory calculations and experimental mechanical tests; the development of an asymptotic theory for the statistical thermodynamics of classical systems with stiff potentials, or equivalently, systems at sufficiently low temperatures.

In Chapter 2, density functional theory (DFT) was combined with experimental mechanical tests to investigate how the mechanical properties of a metallopolymer may be tuned using metal-ligand interactions at the crosslinks. A DFT calculation regimen was established, beginning with a geometry optimization and subsequent frequency analysis to obtain and verify the molecular structure of the crosslink. External forces were applied incrementally, the crosslink was re-optimized, and the extension was measured to obtain the mechanical response of the crosslink. Once the maximum force was reached, the dissociated structure was used as a guess to obtain and verify the broken crosslink structure. Equipped with the two ends of the breaking/reforming pathway, the transition state separating them was determined and was subsequently verified. The forward free energy barriers to breaking the crosslink were then calculated, along with the net free energy change when breaking. This calculation regimen obtained the mechanical response and breaking/reforming thermodynamics of the crosslinks, and was repeated while varying the number and type of ligands. It was determined that increasing numbers of bound ligands did mechanically weaken the crosslinks and reduce the thermodynamic barrier to breaking, which was borne out in mechanical experiments on the bulk materials. A separate binding energy analysis was also established, allowing the equilibrium populations of different numbers of bound ligands to be calculated as a function of the total ligands per metal center. These population calculations predicted a saturation of bound ligands that was also borne out in mechanical experiments. While these set of calculations provided direct insight when varying the number of ligands, it provided incomplete insight when varying the type of ligands. Specifically, the calculations predicted negligible effects when varying the ligand type, whereas mechanical experiments showed significant differences. These differences

were then attributed to varying ligand-environment interactions, which were not considered by the calculations, so this was only qualitatively explained. Correspondingly, future work should consider repeating the DFT calculation regimen with the crosslink in an explicit polymer environment. While this would be a considerable computational task, it would allow quantification of important variables like the average rate of reforming crosslinks as the ligand type changes. This could possibly be done using *ab initio* molecular dynamics (AIMD), Born-Oppenheimer molecular dynamics (BOMD), or hybrid quantum mechanics/molecular mechanics (QM/MM) in `Q-Chem` [120]. Other methods in other software packages could be useful, for example using DFT to train the reactive force field method `ReaxFF` [198]. More importantly than additional simulation, a continuum constitutive model is necessary to complete the investigation of this material system. Ideally, this model would include parameters and functions from the DFT analyses and capture the mechanical response of the material observed in experiment, allowing the conclusions from this work to be directly probed.

In Chapter 3, a large deformation constitutive model for polymer networks was developed from fundamental statistical mechanical principles. This work highlighted the inextricable connection between single-chain mechanical behavior and the equilibrium distribution of chains in the network, in addition to the correspondence between different single-chain thermodynamic ensembles. Using the extensible freely-jointed chain (EFJC) model, the effects of these distinctions were studied at both the molecular and continuum scale. Two popular approximation methods were considered along with the exact method, producing different results in general until the number of links in the chains became sufficiently large. When chains are considered to be infinitely long, the constitutive model in any case becomes the Neo-Hookean model. Notably, this work contributes a mechanistic hyperelastic constitutive model that is directly applicable to the large deformation mechanical response of elastomers. Other single-chain models could be applied in the framework, such as an extensible freely-rotating chain (EFRC) model. The performance trends of the two approximation methods would likely remain the same, but the obtained constitutive model could behave differently. More beneficial would be for future work to reconsider and study the results of the affine assumption. Non-affine formulations are important for polydisperse networks [75, 78, 80] and typically consist of the microsphere and maximal path constraint methods [68, 79]. These non-affine formulations have not yet been applied to distributions with varying end-to-end length, which would be necessary here. Most desirable would be the establishment of a statistical mechanical basis for connecting the macroscopic deformation to the evolution of the distribution of chain end-to-end vectors, but this is perhaps fundamentally inconsistent due to the noninteracting chains approximation. This approximation could also be reconsidered, perhaps utilizing the high-temperature perturbation theory of Zwanzig [138] to obtain approximations for the weak interactions between polymer chains in an elastomer. Such an approach would avoid computational intractability by still allowing the constitutive model to be formulated using single-chain models.

In Chapter 4, an asymptotic theory was developed for the statistical thermodynamics of classical systems with potential energies of interaction that are strong compared to available thermal energy. This theory acts as the low-temperature analog of the high-temperature theory of Zwanzig [138], adding a new tool to the sparse list of existing tools available for general systems [141]. The full system partition function was asymptotically approximated using the reference system (the full system with strong potentials replaced by athermal rigid constraints) and corrections related to the shape of the strong potentials at their minima. This lead to an asymptotically-correct approximation for the Helmholtz free energy. At leading order the theory retrieved the classical rigid-rotor-harmonic

oscillator approximation, where higher order approximations were then systematically-retrieved corrections related to anharmonicity and rotation-vibration coupling. This theory can be applied to other thermodynamic ensembles, such as the ensemble for isotensional molecular stretching experiments, but there are additional complications related to the force-dependent total potential energy minimum. These complications are surmountable, and future work will consider applying the theory to the stretching of single polymer chains. Simulations and exact solutions, when available, will be compared to theoretical asymptotic approximations of the single-chain mechanical response as the stiff bond potential parameters are varied. Future work will also consider applying the theory to general simulation, such as using it to improve upon numerical simulations incorporating rigid constraints, and will analyze the most general case where stiff degrees of freedom interact. Lastly, it should be possible to improve upon several past high-temperature approximations for fluids [139, 140]. This would be done by replacing the hard-sphere approximation for the repulsive part of the Lenard-Jones potential with an asymptotic treatment via this theory, in effect combining the low-temperature and high-temperature theories in a single composite approximation.

In Chapter 5, another statistical mechanical constitutive model for polymer networks was developed, this time including bond breaking and potential reforming. Classical transition state theory was extended to consider a continuous probability distribution of polymer chain end-to-end vectors. The mechanically-sensitive breaking rates were derived exactly using this extended transition state theory, and were shown to be connected to the single-chain mechanical response and equilibrium distribution of chains in the network. A thermodynamically-consistent Helmholtz free energy was then formulated for the network and utilized to develop the macroscopic constitutive theory, where the Cauchy stress was retrieved and the second law of thermodynamics was shown to be automatically arbitrarily satisfied. After specifying the external inputs and parameters (the deformation history, the temperature, and the number density of chains), the resulting constitutive framework requires only the single-chain Hamiltonian, i.e. the single-chain model. To demonstrate the model and apply it to relevant materials, the uFJC model was presented and developed, which is the freely-jointed chain (FJC) with stiff but flexible links with potential energy u. An asymptotic analysis obtained the single-chain mechanical response, which was then pushed through the framework in order to obtain each of the other single-chain statistical mechanical functions. The Morse potential was utilized specifically for u, and both the single-chain mechanical response and reaction rate coefficient function were studied across a range of molecular parameters. An original exact solution to the integro-partial differential equation governing the evolution of the network was presented; it is possible that this equation and its solution is applicable beyond this specific system. This exact solution was specialized for two cases: rate-independent irreversible breaking, and transient breaking. The first specialization is applied to the cyclic deformation of a mechanoluminescent multinetwork elastomer [11], where the model is almost entirely informed by the chemistry and provides excellent results. The second specialization, the transient network model [59, 60], was applied to the cyclic [167] and multi-rate [46] deformation of a poly-vinyl alcohol gel. The transient network model performed poorly at intermediate rates, and replacing the single exponential (single linear timescale) with a nonlinear relaxation function [168] showed that the model needed to consider many effective timescales in order to become accurate. The general model was then compared to both specializations, and parameter study valuable for design purposes was carried out across a range of different molecular parameters. The general theory was then applied to a polymer with force-sensitive reversible crosslinks [194], where the inaccuracy of the model over different cyclic stretches showed that viscous dissipation mechanisms were likely responsible for

much of the toughness exhibited by the material, instead of just bonds breaking and reforming. In Appendix C.5.2, the general theory was adjusted to account for chains remaining intact after a breaking reaction in order to model force-mediated molecular release in a hydrogel [199]. A similar adjustment could be made in order to model chain-lengthening reactions [13–15] that are now being utilized to toughen polymers [16]. This and other applications could require the consideration of non-equivalent reaction pathways within each chain, so an exact solution in that case – i.e. to the system of integro-differential equations in Eq. (C.133) – would be desired but is not yet known. More critically, future should could reconsider the four most fundamental assumptions made when constructing the general theory: the affine assumption, the noninteracting chains approximation, the transition state theory approximation, and the simple behavior of broken chains. The first two were discussed previously as they also pertain to the permanent network model in Chapter 3. Transition state theory approximates that the timescale of reaction is long, where the two regions of the phase space corresponding to each chemical species are distinct and reach a local thermodynamic equilibrium very fast [173]. Such an approximation is not always valid, for example: highly dynamic bonds could break and reform fast enough to near chain relaxation dynamics timescales. This was exhibited by the transient network model performing poorly in capturing the mechanical response of a highly dynamic network over many strain rates. Future work would then perform another derivation without the transition state theory assumption, seeking to provide a substantial improvement upon the transient network model. The fourth fundamental assumption was that the ends of broken chains do not interact until reforming and are then insensitive to extension. This assumption then caused the distribution of broken chains to be flat, unaffected by the deformation, and reform directly toward the intact equilibrium distribution. While this result is consistent with the existing literature [59, 60, 160, 161, 175], it may not be representative of the true physics and is likely creating model features that are not observed in experiment. For example, the scaled stress of the model can be nearly independent of the rate of reforming, and will always yield at some stretch though ∼100% of the chains may remain intact. Future work should reconsider the behavior of broken chains and how they reform in order to alleviate these non-intuitive features, which should aid in modeling force-sensitive reversible breaking in polymer networks. This could also involve viscous mechanisms related to chain slip accounting for much of the dissipation observed in cyclic stress experiments [165], as well as more complicated network evolution like network alteration theory [195, 196]. Finally, it would be beneficial for future work to develop expedient numerical methods that would make this model more suitable for finite element analyses. While the current implementation [149] is robust, accurate, and relatively efficient, it can often require large amounts of memory and time to complete a computation where doing so over many elements would immediately become impractical.

Appendix A

Supporting Information for Chapter 2

A.1 General

Unless otherwise noted, solvents and reagents were purchased and used without further purification. Nickel(II) acetate tetra hydrate, imidazole, 2-methyimidazole, and dimethylamine 2M in THF were purchased from Oakwood Chemical. Piperidine, pyridine, 2-carboxyethyl acrylate, Diphenyl(2,4,6-trimethylbenzoyl)phosphine oxide, and 2-Hydroxy-2-methylpropiophenone were purchased from Millipore-Sigma. 2-Hydroxyethyl acrylate was purchased from TCI America. Sylgard^{TM} 184 was purchased from Electron Microscopy Sciences. Infrared (IR) spectra were recorded in the range of 600–4000 cm^{-1} with 8 cm^{-1} resolution on a Bruker Vertex V80V Vacuum FTIR system in ATR mode under vacuum.

A.2 Polymer synthesis

A.2.1 Preparation of the acrylic formulation with Ni^{2+} crosslinker

A solution of nickel acetate tetra hydrate (34.9 gr, 140 mmol) in methanol (200 ml) was added to a mixture of 2-hydroxyethyl acrylate (246.7 gr, 2.124 mol) and 2-carboxyethyl acrylate (40.4 gr, 280 mmol). The acetic acid was distilled out under reduced pressure (50 mm Hg) at 65 °C. The formulation was stored in the dark at 4 °C and was used for the preparation of all the nickel crosslinked materials.

A.2.2 General procedure for neutral ligand attachment

The appropriate ligand in the indicated number of equivalents was added slowly to the Ni^{2+} acrylic formulation at 0 °C while stirring. The modified nickel formulation was cured according to the general UV curing procedure and was tested 24 hours after preparation.

A.2.3 General procedure for UV free radical polymerization

Diphenyl(2,4,6-trimethylbenzoyl)phosphine oxide (0.005 eq for every acrylic group) and 2-hydroxy-2- methylpropiophenone (0.017 eq for every acrylic group) were added to the acrylic formulation.

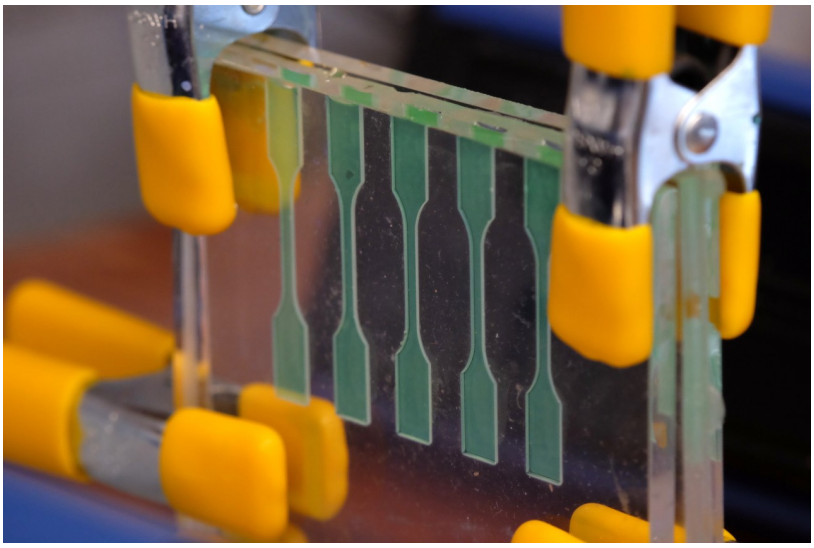


Figure A.1: Polymerization setup; dog-bone silicone mold pressed between two laminated glass plates.

The UV curable formulation was transferred to a dog-bone shaped silicone mold (Fig. A.1) pressed between two glass plates laminated with scotch tape (to prevent adhesion of the cured polymer to the glass). The formulation was cured by 365 nm UV light irradiation generated by two lamps (6 Watt 365 nm, VWR) for 45 minutes.

A.2.4 Preparation of the linear polymer poly(2-hydroxyethyl acrylate-co-2-carboxyethyl acrylate)

A mixture of 2-hydroxyethyl acrylate (6.15 gr, 53.0 mmol) and 2-carboxyethyl acrylate (1.01 gr, 7.0 mmol) was pretreated under reduced pressure (50 mm Hg) at 65 °C to mimic the same preparation conditions as the nickel crosslinked materials. The formulation was cured according to the general UV curing procedure and was tested 24 hours after preparation.

A.2.5 Preparation of the linear polymer poly(2-hydroxyethyl acrylate) with dispersion of $Ni(OAc)_2(H_2O)_4$

A solution of nickel acetate tetra hydrate (1.40 gr, 5.6 mmol) in methanol (10 ml) was added to 2-hydroxyethyl acrylate (9.87 gr, 85 mmol). The methanol was removed under reduced pressure (50 mm Hg) at 65 °C. The formulation was cured according to the general UV curing procedure and was tested 24 hours after preparation.

A.2.6 Preparation of the linear polymer poly(2-hydroxyethyl acrylate)

2-hydroxyethyl acrylate (9.87 gr, 85 mmol) was pretreated under reduced pressure (50 mm Hg) at 65 °C to mimic the same preparation conditions as the nickel crosslinked materials. The formulation was cured according to the general UV curing procedure and was tested 24 hours after preparation.

A.3 Dog-bone mold preparation

First, an acrylic mold was prepared. Five dog-bones (Fig. A.2) matching the desire final specimen shape were laser cut from 1.5 mm thick PMMA panel. The acrylic dog-bones were solvent welded to an acrylic plate with 5.5 mm separation between dog-bones in the acrylic mold. A silicone mold was then prepared by casting two parts PDMS resin (SYLGARD^{TM} 184) onto the acrylic mold. The resin was cured for 48 hours at room temperature and an additional 5 hours at 70 °C. The silicone molds were sprayed with EASE RELEASE^{TM} 200 (Smooth On Inc.) before each use.

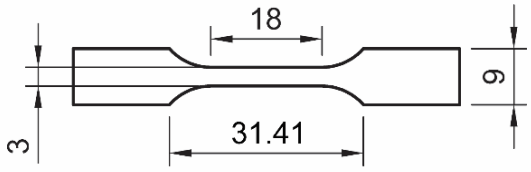


Figure A.2: Dog-bone dimensions in mm.

A.4 Tensile Test Methods

Tensile tests were performed on a Zwick-Roell Z010 system with a 500 N capacity load cell (X-Force HP, Zwick-Roell). Constant strain rate, cyclic, and stress relaxation tests were performed with displacement control using engineering strain rates and assuming an effective gage length of 28.00 mm. Both the cyclic and stress relaxation tests were conducted at an engineering strain rate of 0.114/s. For the cyclic test, the change in crosshead direction is specified by displacement when loaded and by force (0.1 N) when unloaded. Stress and strain were calculated from force and extension recorded during tensile tests. Engineering strain is given by $\varepsilon_{\text{eng}} = \Delta L/L_0$, where L_0 is the initial gage length obtained by linear finite element simulation of the tensile specimen geometry. Engineering stress is given by $\sigma_{\text{eng}} = F/A_0$ where F is the recorded force and A_0 is the original specimen cross-sectional area. Elastic moduli were determined by a linear fit to the stress-strain curve over the initial linear region. Yield stresses were determined by determining the point on the stress strain curve directly below the point of intersection between the line tangent to the initial linear region and the line tangent to the linear region immediately following the yield peak.

A.5 Glass Transition Temperature Determination

Glass transition temperatures (T_g) of the samples were measured by Dynamic Mechanical Analysis (DMA) Q800 (TA Instruments) with a film tension clamp under nitrogen atmosphere. The preload force was 0.01 N, and an oscillation strain 0.1% was performed on the sample with a frequency of 1 Hz. The sample was equilibrated at −30 °C for 2 min and then heated up to 150 °C with speed 3 °C/min.

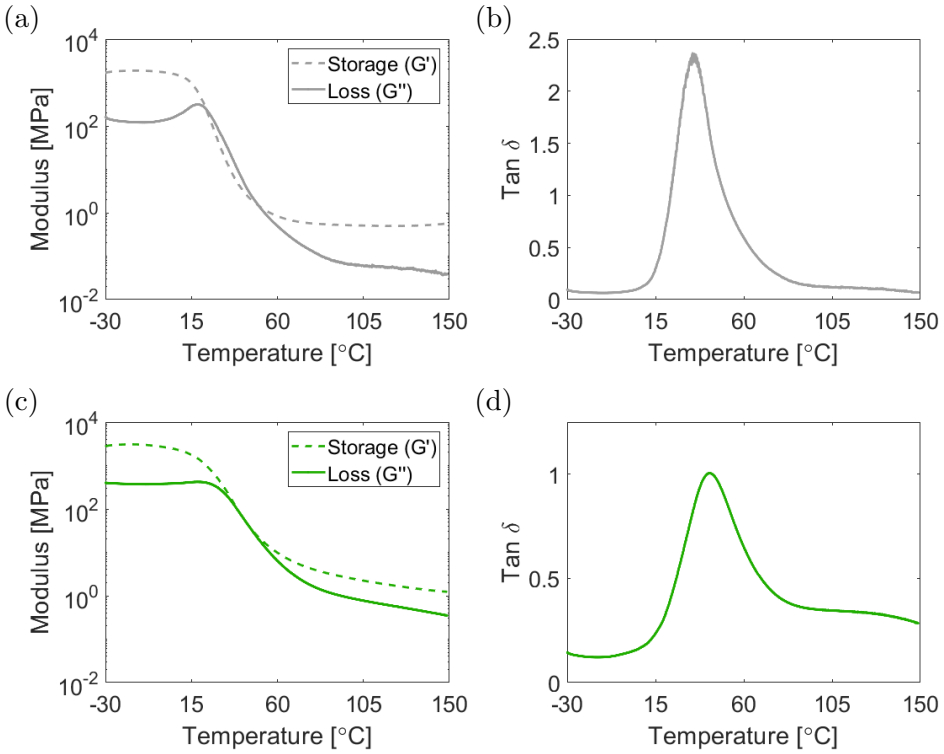


Figure A.3: Storage and loss modulus (left; a,c) and tan δ (right; b,d) for **Linear** (top; a,b) and **Ni** (bottom; c,d).

A.6 Theoretical backbone stiffness

The backbone stiffness between crosslinks was estimated by performing a molecular dynamics (MD) simulation of seven repeat units of poly(2-hydroxyethyl acrylate). All simulations were performed using the large-scale atomic/molecular massively parallel simulator (`LAMMPS`) [200] on the Extreme Science and Engineering Discovery Environment (XSEDE) `Stampede2` cluster [124]. A CVFF force field was used to define the interactions between the atoms in the polymer chain [201]. The chain was first equilibrated using Langevin dynamics for 0.1 ns (with a time step of 1.0 fs) at 500 K in the NVE ensemble and 0.1 ns (with a time step of 0.5 fs) at 500 K and atmospheric pressure in the NpT ensemble. This was followed by cooling the system at a constant rate from 500 K to 300 K over 0.25 ns (with a time step of 0.5 fs) in the NpT ensemble, followed by a 1.0 ns simulation (with a timestep of 0.5 fs) at 300 K and atmospheric pressure in the NpT ensemble. The polymer chain was then deformed for 1.0 ns (with a time step of 1.0 fs) in the NVE ensemble by fixing the carbon at one end of the chain and pulling on the carbon at the other end of the chain along the radial vector between them at a controlled force rate of 20.8 nN/ns. Langevin dynamics were used during the pulling step to maintain the temperature at 300 K.

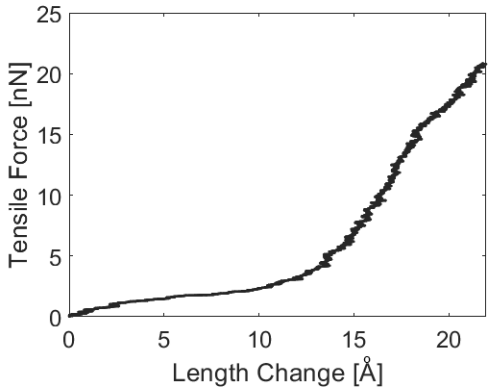


Figure A.4: Theoretically (MD) obtained mechanical response in the average length of backbone between crosslinks from which the initial stiffnesses is approximately 0.4 nN/Å and the maximum tangent stiffness is approximately 2.4 nN/Å.

A.7 Cyclic stress-strain response compared to monotonic loading

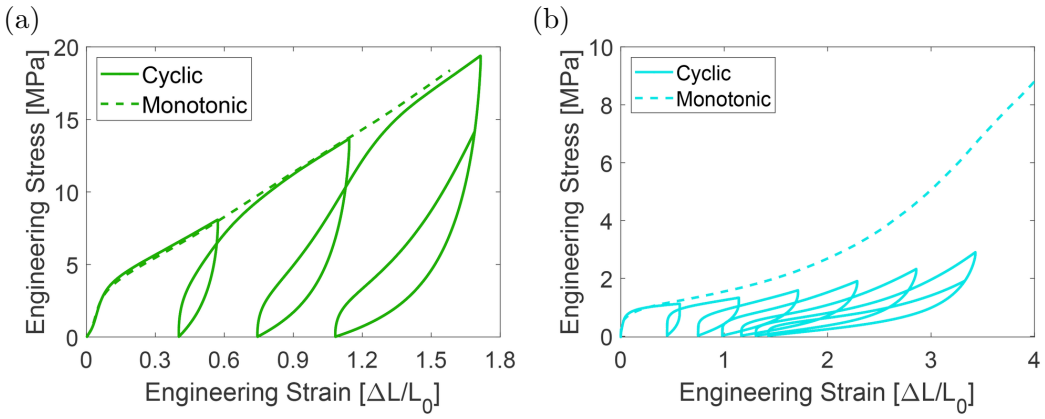


Figure A.5: Cyclic and monotonic response for (a) **Ni** and (b) **Ni-2Im**.

A.8 Computational details

All calculations were carried out using the Extreme Science and Engineering Discovery Environment (XSEDE) `Stampede2` cluster [124] and the U.S. Department of Defense High Performance Computing Modernization Program (HPCMP) `Mustang` and `Centennial` clusters.

A.8.1 Geometry optimizations of the model systems

All calculations were performed with the `Q-Chem` software package (`v5.1` development version) [120] using the ω`B97X-V` density functional approximation [121] and the `def2-TZVPP` basis set [122]. All SCF calculations were converged with a tight DIIS convergence threshold (`SCF_CONVERGENCE = 9`, corresponding to 10^{-9}) and integral threshold of (`THRESH = 14`, corresponding to 10^{-14}) unless otherwise noted. Symmetry was ignored during all calculations (`SYMMETRY = FALSE, SYM_IGNORE = TRUE`), and the force, displacement, and energy convergence criteria were set to minimum values (`GEOM_OPT_TOL_GRADIENT = 1`, `GEOM_OPT_TOL_DISPLACEMENT = 1`, `GEOM_OPT_TOL_ENERGY = 1`). Various nickel-ligand conformations and ligand rotations were sampled as starting points for geometry optimizations to ensure convergence to the global energy minimum of each nickel crosslink structure. In addition, numerous starting points for the second coordination sphere molecules were sampled (when appropriate) to obtain the lowest energy configurations. All calculations treated the electronic wavefunction as a high-spin triplet; for all imidazole containing compounds, we computed the corresponding singlet-triplet gaps to confirm a triplet ground state. During all calculations, the spin contamination of the wavefunction was minimal with $\langle S^2 \rangle$ values ranging from 2.0029 to 2.0043. Subsequent harmonic vibrational frequency analyses (using a numerical finite-difference scheme with `FDIFF_STEPSIZE = 100`) were performed to confirm that all optimized geometries were minima with no imaginary frequencies.

A.8.2 Applying external forces to the crosslink

The mechanical responses of the structures were then investigated by applying equal and opposite forces at the outer methyl carbon atoms using the External Force Explicitly Included (EFEI) algorithm [107]. The external force was incrementally increased in steps of 0.1 nN, and the geometry was re-optimized (as described above) until one of the nickel-carboxylate bonds ruptures and the complex dissociates from the applied forces. This rupturing event was defined as a failure in the geometry optimization procedure to converge after a total of 750 iterations.

A.8.3 Free energy barriers

The dissociated structures from the EFEI algorithm (i.e., the geometries at the point of rupture) were taken as starting points in subsequent force-free geometry optimizations to obtain the optimized configurations for each ruptured crosslink (as described above). To compute the corresponding free-energy barriers, the frozen string method (FSM) [202, 203] was employed to compute an approximate reaction pathway connecting the (force-free) optimized intact and ruptured structures. Subsequent transition state (TS) searches were performed on the maximum-energy configurations from the FSM pathways using the same force, displacement, and energy convergence criteria as above, and a maximum allowed step size of 0.05 (`GEOM_OPT_DMAX = 50`). Depending on the difficulty associated with finding the final TS, FSM calculations used 20-55 nodes and 6-20 orthogonal gradients. All TS structures were verified to have a single imaginary frequency (which connected the corresponding reactant and product) using the numerical Hessian procedure described above. Thermal contributions to the free energy were computed at a temperature of 298.15 K and a pressure of 1 atm using partition functions derived from the standard ideal gas (IG) and rigid rotor-harmonic oscillator (RRHO) approximations.

A.8.4 Consecutive ligand binding energies

The geometries used for computing consecutive binding free energies were optimized according to the procedure described above, based on starting configurations generated by systematically replacing each of the four water molecules initially bound to the nickel center with an imidazole ligand (see Fig. A.6). The replaced water molecules were taken to be infinitely separated from the complex (and not included in the second coordination sphere), which is needed to maintain size consistency in the binding free energies. Binding free energies for the consecutive addition of n ligands were then computed according to:

$$\Delta G_{\text{bind}}[n] = \left(G\left[\text{Ni(OAc)}_2\text{L}_n(\text{H}_2\text{O})_{4-n}\right]\right) - \left(G\left[\text{Ni(OAc)}_2\text{L}_{n-1}(\text{H}_2\text{O})_{4-(n-1)}\right] + G[\text{L}]\right), \tag{A.1}$$

where $G[X]$ is the free energy of the X species, and $n = 1, 2, 3, 4$ is the number of coordinating ligands (L). The results are reported in Table A.1 for the imidazole (Im), methylimidazole (MeIm), pyridine (Py), piperidine (Pipe), and dimethylamine (DMA) ligands.

A.8.5 Population analysis

To determine the number of imidazole ligands (Im) bound to each Ni center at equilibrium, we considered a finite system consisting of N metal centers (representing a polymer with N non-interacting

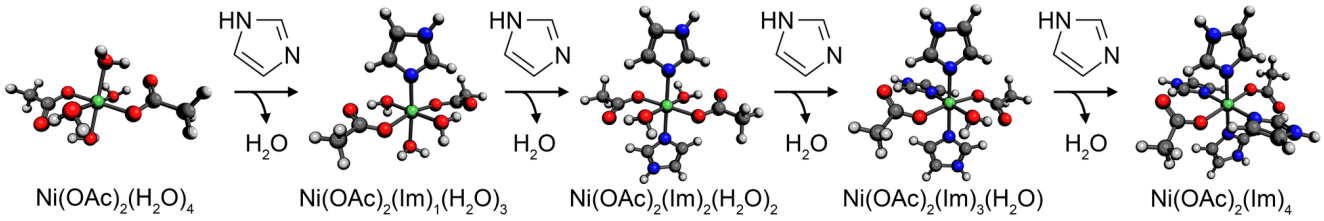


Figure A.6: Illustration of the procedure used to calculate the theoretical binding free energies for the consecutive addition of $n = 1$, 2, 3, 4 imidazole ligands.

nickel-carboxylate crosslinks). Each of these metal centers was taken to have the following molecular formula, $Ni(OAc)_2L_n(H_2O)_{4-n}$, and was allowed to reside in one of five states, ranging from no bound Im ligands ($n = 0$) to four bound Im ligands ($n = 4$). The corresponding free energies of each state (relative to the $n = 0$ state for which $\Delta G[0]$ was set to 0.0 kcal/mol) were taken as the sum of the consecutive binding free energies in Table A.1, i.e., $\Delta G[n] = \sum_k \Delta G_{\text{bind}}[k]$ for $1 \leq k \leq n$. With these $\Delta G[n]$ values in hand, we enumerated all possible energy levels for a system containing N metal centers, ranging from all metal centers in the $n = 0$ state to all metal centers in the $n = 4$ state. Enumerating energy levels (instead of explicitly enumerating system states) enables us to simulate larger system sizes by dramatically decreasing the computational effort associated with the enumeration. To construct Fig. 2.4(a) in the manuscript, we considered $m = 0$ to 5 equivalents of Im (in increments of 0.125). For a given m, we selected all of the enumerated levels which contain less than mN Im ligands for the analysis described below. Here we note that each energy level corresponds to all of the system states within the same permutation group. In other words, the degeneracy of each energy level arises from the permutational symmetry of the system, in which one can swap the states of any two Ni centers without changing the total system energy. To calculate the degeneracy of the jth energy level, g_i, we compute the number of distinct permutations via

$$g_j = \frac{N!}{N_{j,0}!\, N_{j,1}!\, N_{j,2}!\, N_{j,3}!\, N_{j,4}!}, \tag{A.2}$$

in which $N_{j,n}$ denotes the number of Ni centers bound to n Im ligands. It follows that the probability of occupying the jth energy level, P_j, is given by:

$$P_j = \frac{g_j\, e^{-\beta \Delta G_j}}{Q}, \tag{A.3}$$

Table A.1: Theoretical binding free energies (in kcal/mol) for the consecutive addition of a given ligand to the nickel center.

Ligand	$\mathbf{\Delta G_{bind}[1]}$	$\mathbf{\Delta G_{bind}[2]}$	$\mathbf{\Delta G_{bind}[3]}$	$\mathbf{\Delta G_{bind}[4]}$
Im	−4.60	−8.34	−0.80	0.53
MeIm	−5.15	−9.28	−0.64	-
Py	−4.79	−3.99	1.45	-
Pipe	−7.78	−6.83	0.87	-
DMA	−6.19	−5.57	1.23	-

in which $\beta = 1/k_{\mathrm{B}}T$ is the inverse temperature, k_{B} is Boltzmann's constant, T is the temperature, and the normalization Q is the canonical partition function,

$$Q = \sum_j g_j e^{-\beta \Delta G_j}. \tag{A.4}$$

In both of these expressions, we note that the total energy of the jth energy level is

$$\Delta G_j = \sum_n N_{j,n} \Delta G[n]. \tag{A.5}$$

With P_j in hand, the probability of occupying the nth Ni state, π_n, is then

$$\pi_n = \frac{1}{N} \sum_j N_{j,n} P_j, \tag{A.6}$$

in which the sum over j includes all energy levels that are possible when m equivalents of Im ligands are added to the system. In this work, we considered $N = 48$ Ni centers and $T = 298.15$ K, which yields π_n values that are converged to within $\sim 1\%$. As such, the error made with this finite ensemble is negligible when compared to the uncertainties in the computed $\Delta G[n]$ values.

A.9 Dynamic mechanical analysis for Ni-nIm materials

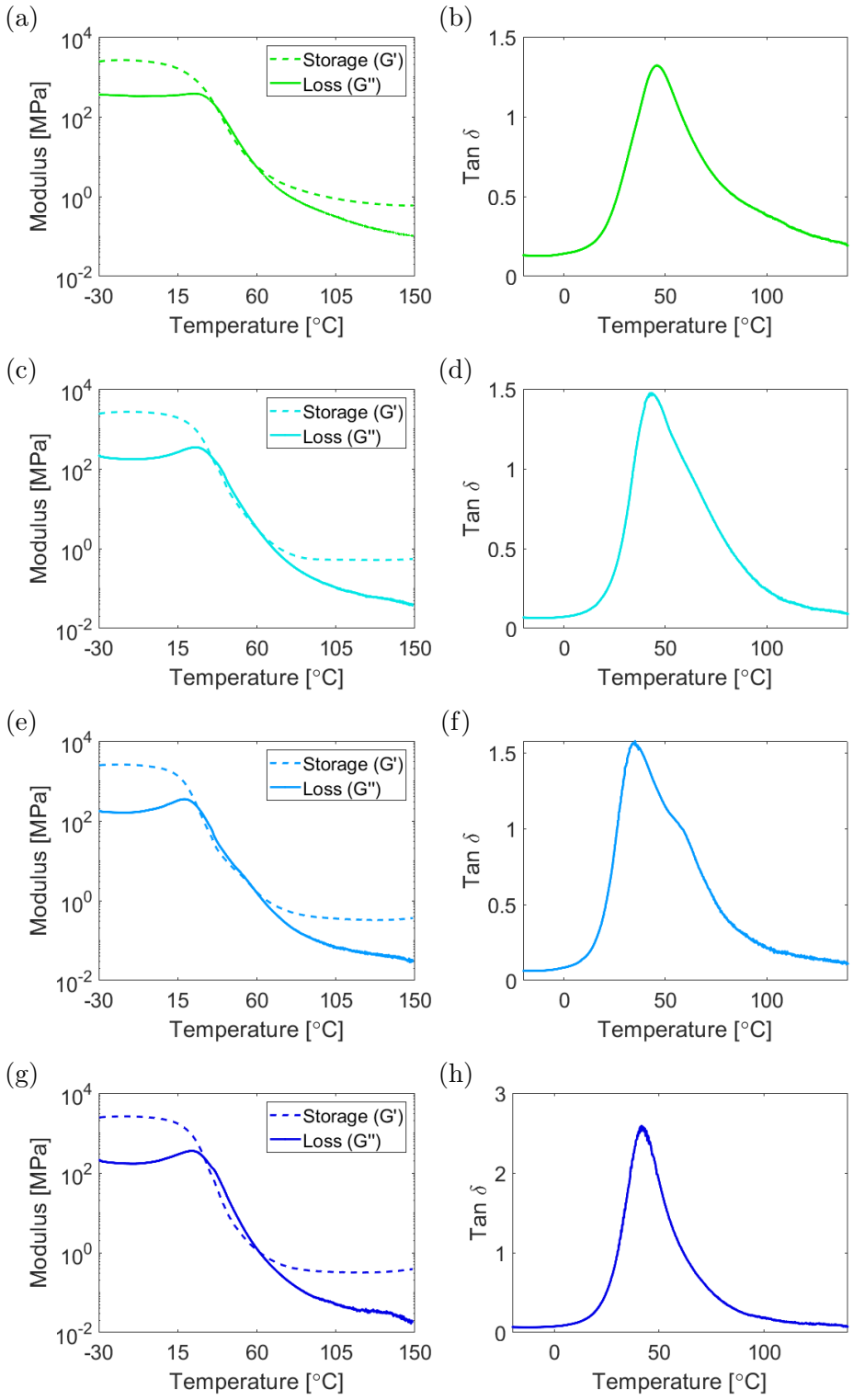


Figure A.7: Storage and loss modulus (left; a,c,e,g) and tan δ (right; b,d,f,h) for **Ni-1Im** (a,b), **Ni-2Im** (c,d), **Ni-3Im** (e,f), and **Ni-4Im** (g,h).

A.10 Infrared spectra and color of all materials

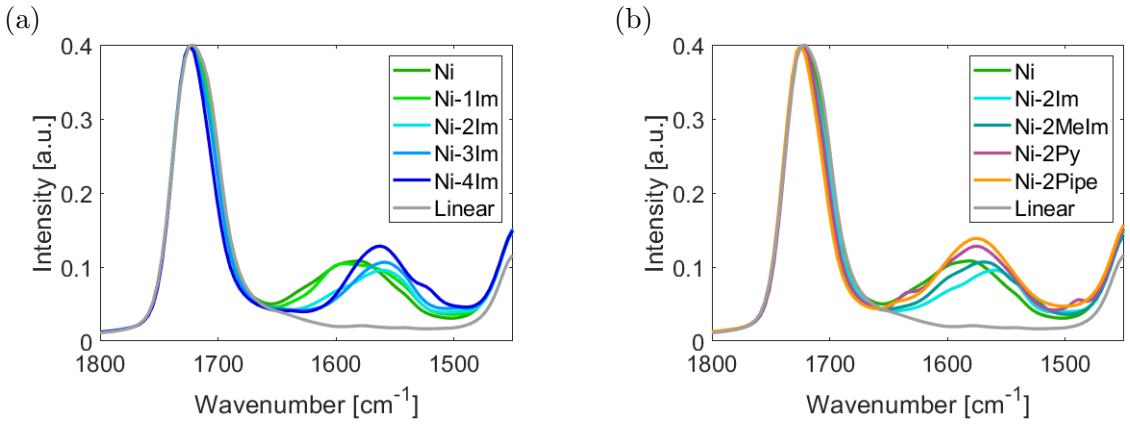


Figure A.8: IR spectra of polymers with (a) different numbers of Im equivalents per Ni center and (b) different types of ligands with 2 equivalents per Ni center.

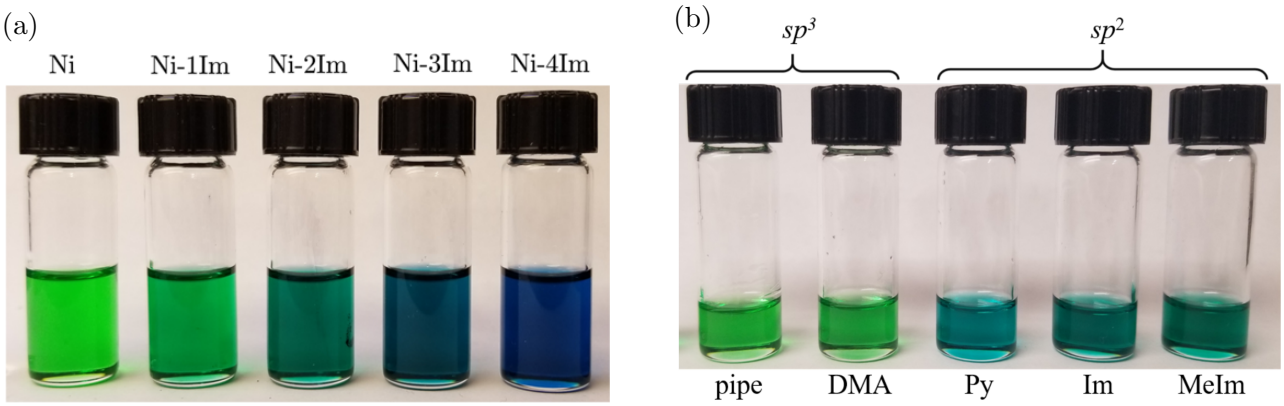


Figure A.9: (a) Nickel carboxylate crosslinkers in solution with increasing amounts of imidazole, and (b) differences in color for solutions of Ni-2Pipe and Ni-2DMA (sp^3 hybridized nitrogen; group II) vs. Ni- 2Py, Ni-Im and Ni-2MeIm (sp^2 hybridized nitrogen; group I). All solutions were prepared according to the general procedure for neutral ligand attachment (see Sec. A.2.2).

A.11 Additional mechanical data

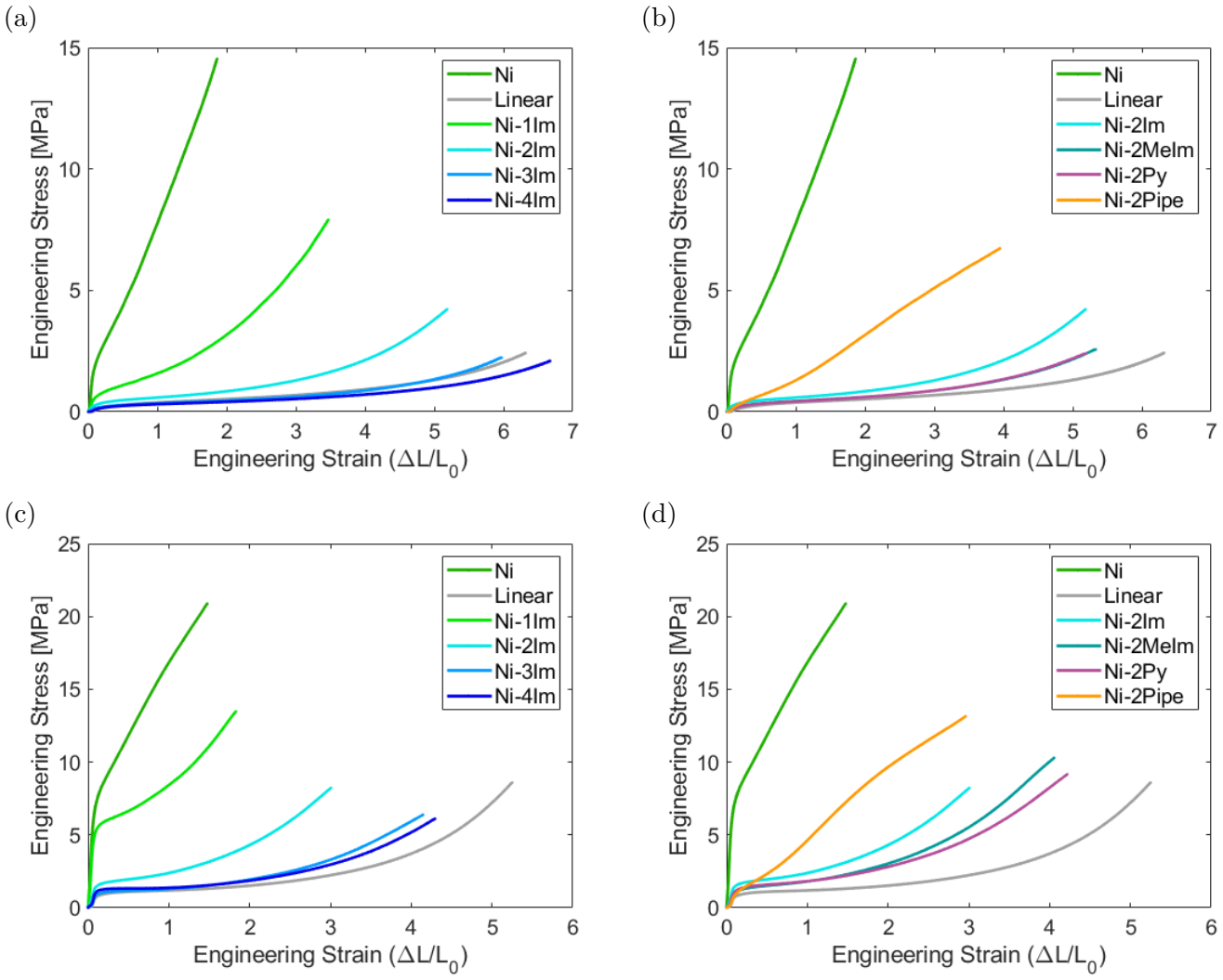


Figure A.10: Monotonic response at a rate of 0.023/s (top; a,b) and 0.57/s (bottom; c,d) when varying the number (left; a,c) and type (right; b,d) of ligands in the material. The **Ni-2DMA** material was not tested at these rates.

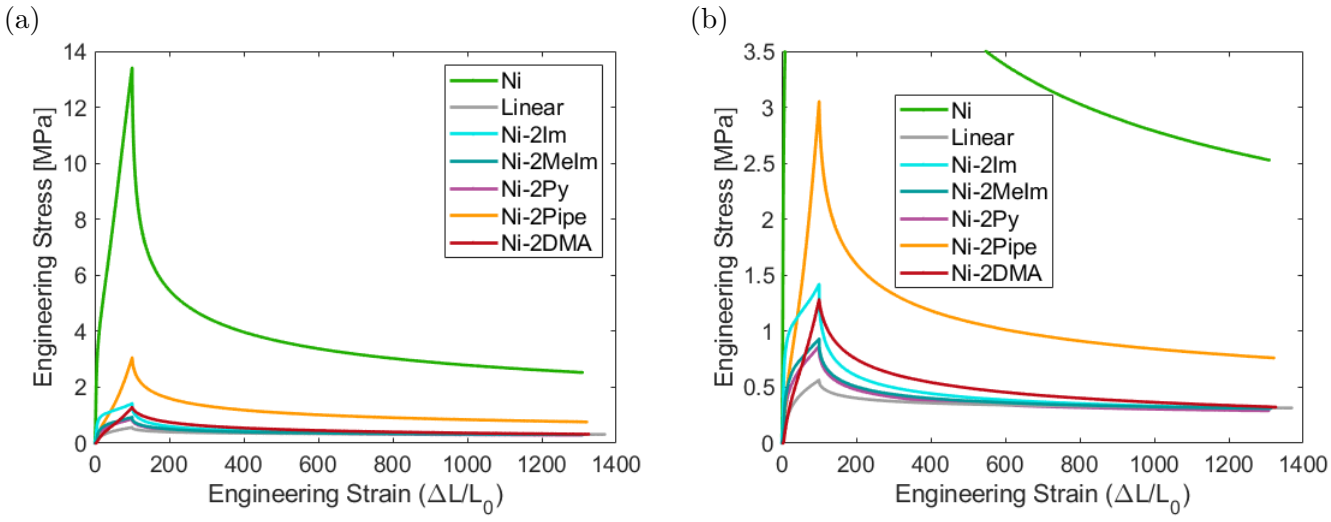


Figure A.11: Stress relaxation response when varying the type of ligands in the material (a) and zoomed in (b).

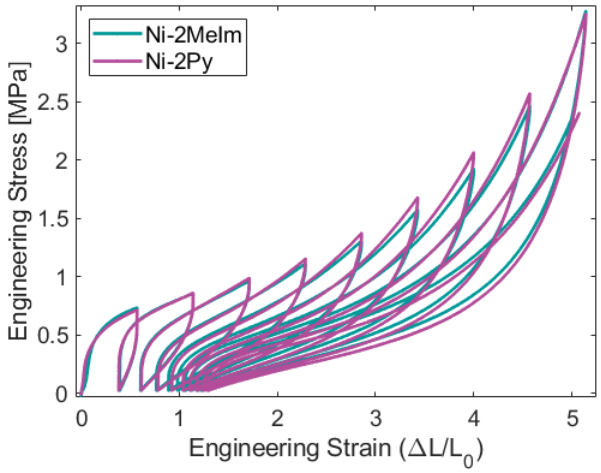


Figure A.12: Cyclic response for **Ni-2MeIm** and **Ni-2Py** over many cycles, showing remarkable similarity.

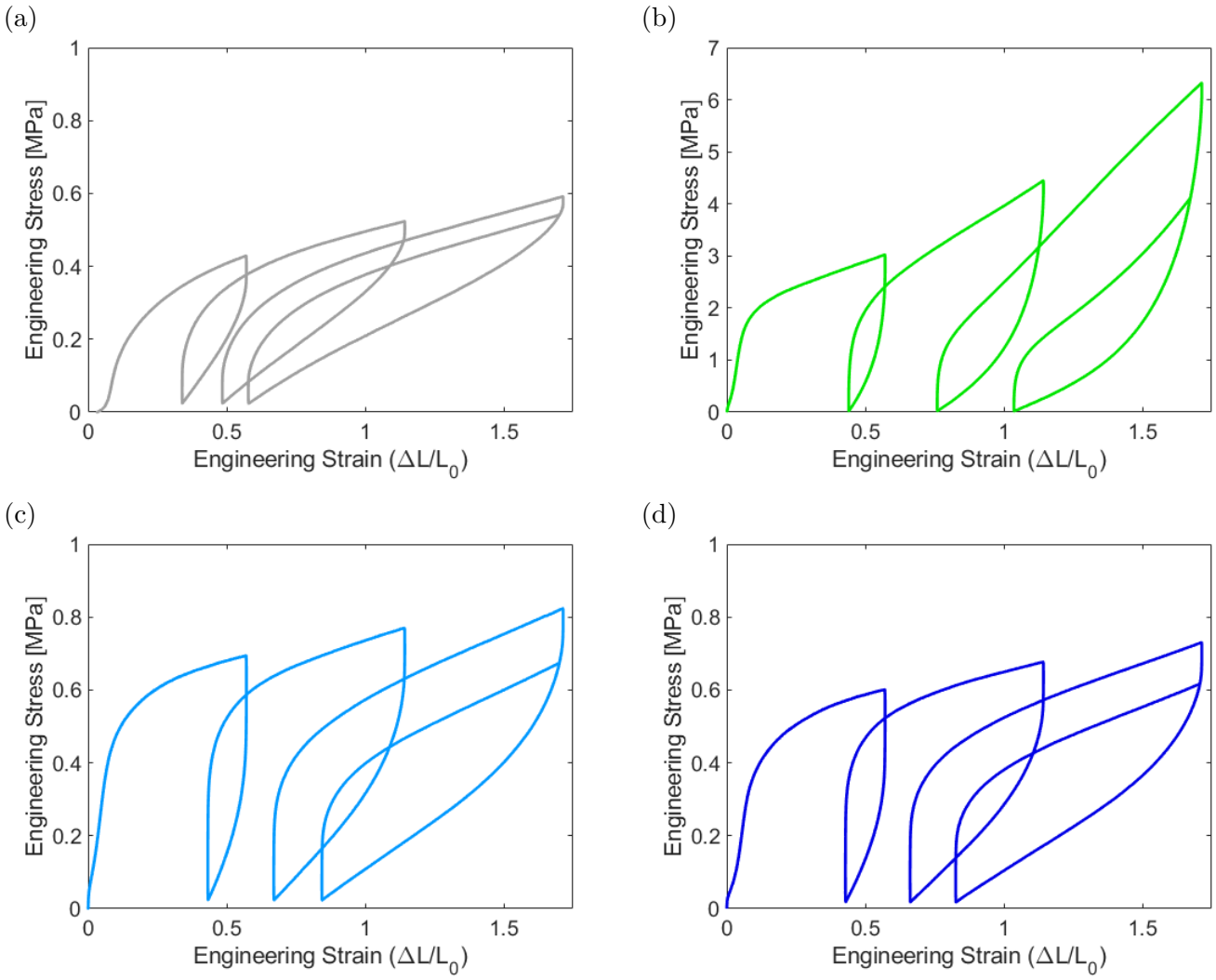


Figure A.13: Cyclic response for the (a) **Linear**, (b) **Ni-1Im**, (c) **Ni-3Im**, and (d) **Ni-4Im** materials.

Appendix B

Supporting Information for Chapter 3

B.1 Solution for the network distribution

We seek to analytically solve Eq. (3.36) in order to evaluate the probability distribution $P(\boldsymbol{\xi}, t)$ at any time under the deformation $\mathbf{F}(t)$. A new set of variables is taken: $\boldsymbol{\alpha} = (\boldsymbol{\xi}, t)^T$ with gradient $\nabla_{\boldsymbol{\alpha}} = (\partial/\partial\boldsymbol{\xi}, \partial/\partial t)^T$ and vector $\mathbf{b} = ([\mathbf{L}(t) \cdot \boldsymbol{\xi}], 1)^T$. We may then rewrite Eq. (3.36) as the concise linear partial differential equation

$$\mathbf{b}(\boldsymbol{\alpha}) \cdot \nabla_{\boldsymbol{\alpha}} P(\boldsymbol{\alpha}) = 0. \tag{B.1}$$

This type of partial differential equation can be solved using the method of characteristics. The characteristic solutions are parameterized by s in the system of first order linear ordinary differential equations given by

$$\frac{d\boldsymbol{\alpha}}{ds} = \mathbf{b}[\boldsymbol{\alpha}(s)], \tag{B.2a}$$

$$\frac{dP}{ds} = 0. \tag{B.2b}$$

Eq. (B.2a) is a vector equation with four components (three for $\boldsymbol{\xi}$, one for t). Consider the t component,

$$\frac{dt}{ds} = 1, \tag{B.3}$$

which simply shows that t and s differ by a constant, which we will now choose to be zero, thereby taking $t(s) = s$. The $\boldsymbol{\xi}$ components of Eq. (B.2a),

$$\frac{d\boldsymbol{\xi}}{ds} = \mathbf{L}(s) \cdot \boldsymbol{\xi}, \tag{B.4}$$

are solved after taking $\boldsymbol{\xi}(0) = \boldsymbol{\xi}_0$ and $\mathbf{F}(0) = \mathbf{1}$ by

$$\boldsymbol{\xi}(s) = \mathbf{F}(s) \cdot \boldsymbol{\xi}_0. \tag{B.5}$$

Eq. (B.2b) shows that P is constant when varying only s,

$$P\left[\boldsymbol{\xi}(s), t(s)\right] = \text{constant}, \tag{B.6}$$

where using Eq. (B.5) and $t = s$, and assuming we know P at some previous time $\tau < t$, we then have the solution

$$P\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}_0, t\right] = P\left[\mathbf{F}(\tau)\cdot\boldsymbol{\xi}_0, \tau\right]. \tag{B.7}$$

We are free to choose $\boldsymbol{\xi}_0 = \mathbf{F}^{-1}\cdot\boldsymbol{\xi}$ and retrieve

$$P\left(\boldsymbol{\xi}, t\right) = P\left[{}_{(\tau)}\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}, \tau\right], \tag{B.8}$$

where we have used the deformation at t relative to the deformation at a previous time previous time $\tau < t$, denoted as ${}_{(\tau)}\mathbf{F}(t)$, given by Paolucci [135] as

$${}_{(\tau)}\mathbf{F}(t) = \mathbf{F}(t)\cdot\mathbf{F}^{-1}(\tau). \tag{B.9}$$

Now, if we presume that the network is at equilibrium at time $\tau = 0$, we have $P(\boldsymbol{\xi}, 0) = P^{\text{eq}}(\boldsymbol{\xi})$ and ${}_{(0)}\mathbf{F}(t) = \mathbf{F}(t)$, and our solution then becomes

$$P(\boldsymbol{\xi}, t) = P^{\text{eq}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}\right], \tag{B.10}$$

which is Eq. (3.37) from the manuscript.

B.2 Retrieving the stress

Starting with Eqs. (3.38) and (3.44), we seek to retrieve a closed-form relation for the stress that does not include gradients of the network distribution of end-to-end vectors. We first relate the stress to the Helmholtz free energy using the hyperelastic formula (by way of neglecting dissipative stresses) and the solution for the distribution evolution in Eq. (3.36), where a spherical pressure term is included due to incompressibility. We then perform integration by parts, and after proving that the resulting boundary integral term is zero for relevant chain models, we retrieve the stress as an integral function of the network equilibrium distribution, the applied deformation, and the single chain mechanical response.

B.2.1 Simplifications and integration by parts

We begin by taking the time derivative of a given by Eq. (3.38) under constant temperature, so that it can later be used with Eq. (3.44),

$$\left(\frac{\partial a}{\partial t}\right)_T = n\iiint\left(\frac{\partial P}{\partial t}\right)_T \psi^*\, d^3\boldsymbol{\xi} - p\left(\frac{\partial J}{\partial t}\right)_T, \tag{B.11}$$

where we know the evolution of P from Eq. (3.36), we have chosen n to be constant due to the incompressibility constraint (we are free to consider the n derivatives to be nonzero and carry them through this derivation, but at the end they will be lumped into the pressure since they only produce spherical terms, thus leaving the same results). For the last term, we use

$$\left(\frac{\partial J}{\partial t}\right)_T = J\mathbf{1} : \mathbf{L}. \tag{B.12}$$

Substituting (with $J = 1$) the above into Eq. (3.44) and neglecting dissipative stresses (taking the inequality to be an equality) then shows that the stress must be

$$\boldsymbol{\sigma} = -n \iiint \left(\frac{\partial P}{\partial \boldsymbol{\xi}}\right) \psi^* \boldsymbol{\xi}\, d^3\boldsymbol{\xi} - p\mathbf{1}. \tag{B.13}$$

We now seek to rewrite the stress in a way that does not include gradients of P. We perform the integration by parts

$$\iiint \left(\frac{\partial P}{\partial \boldsymbol{\xi}}\right) \psi^* \boldsymbol{\xi}\, d^3\boldsymbol{\xi} = \iint P\psi^* \left(\frac{\boldsymbol{\xi}\boldsymbol{\xi}}{\xi}\right) d^2\boldsymbol{\xi} - \iiint P \left[\left(\frac{\partial \psi^*}{\partial \boldsymbol{\xi}}\right)\boldsymbol{\xi} + \psi^* \mathbf{1}\right] d^3\boldsymbol{\xi}, \tag{B.14}$$

where the double integral is along the boundary $\xi \to \infty$ with unit normal vector $\hat{\boldsymbol{\xi}} = \boldsymbol{\xi}/\xi$, and $d^2\boldsymbol{\xi}$ is the surface element. We are free to lump the spherical term into $p\mathbf{1}$ without loss of generality. Note that we have not taken into account that the molecular partition functions may depend on the volume – this dependence would produce more spherical terms that would also now be lumped into p. The stress is now written as

$$\boldsymbol{\sigma} = n \iiint P \left(\frac{\partial \psi^*}{\partial \boldsymbol{\xi}}\right) \boldsymbol{\xi}\, d^3\boldsymbol{\xi} - n \iint P\psi^* \left(\frac{\boldsymbol{\xi}\boldsymbol{\xi}}{\xi}\right) d^2\boldsymbol{\xi} - p\mathbf{1}.$$

If we then assume that ψ^* is rotationally symmetric, we receive the non-polar stress

$$\boldsymbol{\sigma} = n \iiint P \left(\frac{\partial \psi^*}{\partial \xi}\right) \left(\frac{\boldsymbol{\xi}\boldsymbol{\xi}}{\xi}\right) d^3\boldsymbol{\xi} - n \iint P\psi^* \left(\frac{\boldsymbol{\xi}\boldsymbol{\xi}}{\xi}\right) d^2\boldsymbol{\xi} - p\mathbf{1}. \tag{B.15}$$

Taking the term from the integration along the boundary to be zero, taking $p = p^{\mathrm{eq}} + \Delta p$, and using the solution for P in Eq. (3.37), we receive

$$\boldsymbol{\sigma}(t) = n \iiint P^{\mathrm{eq}} \left[\mathbf{F}^{-1}(t) \cdot \boldsymbol{\xi}\right] \left(\frac{\partial \psi^*}{\partial \xi}\right) \left(\frac{\boldsymbol{\xi}\boldsymbol{\xi}}{\xi}\right) d^3\boldsymbol{\xi} - \left[p^{\mathrm{eq}} + \Delta p(t)\right] \mathbf{1}, \tag{B.16}$$

which is Eq. (3.45) from the manuscript. In the following section we discuss the boundary integral stress term in depth.

B.2.2 The boundary integral stress term

We now consider the stress from the integration along the boundary in Eq. (B.15) in order to show that it is zero for arbitrary deformations as long as ψ^* satisfies certain growth criteria. Using Eq. (3.37) and taking $d\boldsymbol{\mathcal{S}} = (\boldsymbol{\xi}/\xi)\, d^2\boldsymbol{\xi}$, the boundary integral stress term is

$$\hat{\boldsymbol{\sigma}} = n \iint P^{\mathrm{eq}} \left(\mathbf{F}^{-1} \cdot \boldsymbol{\xi}\right) \psi^*(\boldsymbol{\xi}) \boldsymbol{\xi}\, d\boldsymbol{\mathcal{S}}. \tag{B.17}$$

We recall that ψ^* had been assumed to be rotationally symmetric, and the boundary is along $\xi \to \infty$, so we may take ψ^* out of the integrand and write

$$\hat{\boldsymbol{\sigma}} = n \lim_{\xi\to\infty} [\psi^*(\xi)] \iint P^{\text{eq}} \left(\mathbf{F}^{-1}\cdot\boldsymbol{\xi}\right) \boldsymbol{\xi}\, d\boldsymbol{\mathcal{S}}. \tag{B.18}$$

We take the change of variables $\boldsymbol{\xi} \mapsto \mathbf{F}\cdot\boldsymbol{\xi}$, where as pointed out by Paolucci [135], the surface element transforms as $d\boldsymbol{\mathcal{S}} \mapsto J\mathbf{F}^{-T}\cdot d\boldsymbol{\mathcal{S}}$. We also use $J = 1$ and then receive

$$\hat{\boldsymbol{\sigma}} = n \lim_{\xi\to\infty} [\psi^*(\xi)] \iint P^{\text{eq}} (\boldsymbol{\xi}) (\mathbf{F}\cdot\boldsymbol{\xi}) \mathbf{F}^{-T}\cdot d\boldsymbol{\mathcal{S}}. \tag{B.19}$$

Since ψ^* is rotationally symmetric, by the distribution-behavior correspondence in Eq. (3.13) P^{eq} is rotationally symmetric as well. We may then remove P^{eq} from the integrand along with the deformation terms for

$$\hat{\boldsymbol{\sigma}} = n \lim_{\xi\to\infty} \left[\psi^*(\xi) P^{\text{eq}}(\xi)\right] \left\{\mathbf{F}\mathbf{F}^{-T} : \iint \boldsymbol{\xi}\, d\boldsymbol{\mathcal{S}}\right\}, \tag{B.20}$$

and now we use $d\boldsymbol{\mathcal{S}} = \xi\boldsymbol{\xi}\, d\Omega$, where $d\Omega$ is the differential solid angle, and remove factors of ξ from the integrand to finally write $\hat{\boldsymbol{\sigma}}$ as

$$\hat{\boldsymbol{\sigma}} = \left(\frac{4\pi n}{3}\mathbf{1}\right) \lim_{\xi\to\infty} \left[\xi^3 \psi^*(\xi) P^{\text{eq}}(\xi)\right], \tag{B.21}$$

where the term in the parentheses was retrieved from

$$\mathbf{F}\mathbf{F}^{-T} : \iint \hat{\boldsymbol{\xi}}\hat{\boldsymbol{\xi}}\, d\Omega = \mathbf{F}\cdot\left(\frac{4\pi}{3}\mathbf{1}\right)\cdot\mathbf{F}^{-1} = \frac{4\pi}{3}\mathbf{1}. \tag{B.22}$$

So far we have shown that the stress contributed by $\hat{\boldsymbol{\sigma}}$ is spherical, and could therefore be lumped into the pressure term $-p\mathbf{1}$, as long as it is finite. We are then tasked with proving that the limit in Eq. (B.21) is finite, but we will instead prove it is zero. We will accomplish this by requiring that ψ^* grows sufficiently fast as $\xi\to\infty$, and using l'Hôpital's rule and the squeeze theorem. To start, we require that the growth of ψ^* as $\xi\to\infty$ is greater than that of a logarithm function, i.e.

$$\lim_{\xi\to\infty}\left[\frac{\psi^*(\xi)}{c\ln(\xi)}\right] = \infty, \qquad \forall c > 0. \tag{B.23}$$

This is required to guarantee that the denominator of Eq. (3.13) is finite, but we will use it here as well. Take Eq. (3.13) with the denominator now understood to be a finite constant, thus eliminating it, as we use the limit in Eq. (B.21) to define the functional of $\psi^*(\xi)$,

$$W(\psi^*) = \lim_{\xi\to\infty}\left[\frac{\xi^3\psi^*(\xi)}{e^{\beta\psi^*(\xi)}}\right], \tag{B.24}$$

where we can assume that $\beta > 0$. See from Eq. (B.23) that we have already required that ψ^* is positive as $\xi\to\infty$, so we have in turn required that $W(\psi^*) \geq 0$. Also see from Eq. (B.24) that $W(\psi^*) \leq W(c\ln\xi)$ for all ψ^* that satisfy the requirement from Eq. (B.23), so together we have

$$W(c\ln\xi) \geq W(\psi^*) \geq 0, \qquad \forall c > 0. \tag{B.25}$$

Next, let us consider the special (albeit prohibited for admissible ψ^*) case of $\psi^* = c\ln\xi$, where we repeatedly use l'Hôpital's rule to show

$$W(c\ln\xi) = \lim_{\xi\to\infty}\left[\frac{c\xi^3\ln\xi}{\xi^{c\beta}}\right], \tag{B.26}$$

$$= \lim_{\xi\to\infty}\left[\frac{6\xi^{3-c\beta}}{\beta(c\beta-1)(c\beta-2)(c\beta-3)}\right], \tag{B.27}$$

in order to see that $W(c\ln\xi) = 0$ for all $c\beta > 3$. Since Eq. (B.25) holds for all $c\beta > 3$, by the squeeze theorem we retrieve $W(\psi^*) = 0$ for any ψ^* that satisfies Eq. (B.23). Thus, the stress from the integration along the boundary must be zero, $\hat{\boldsymbol{\sigma}} = \mathbf{0}$, for any rotationally symmetric ψ^* that grows faster than logarithmically as $\xi \to \infty$. This is true for the EFJC model considered here.

B.3 Reduction to Neo-Hookean model

The Neo-Hookean model is retrieved when the ideal chain free energy in Eq. (3.31) and the corresponding equilibrium distribution in Eq. (3.32) are utilized. Substitution of these into the stress from Eq. (3.45) yields

$$\boldsymbol{\sigma} = \frac{n\mathfrak{b}T}{(2\pi)^{3/2}}\iiint \exp\left(-\frac{1}{2}\left\|\mathbf{F}^{-1}\cdot\mathbf{v}\right\|_2^2\right)\mathbf{v}\mathbf{v}\,d^3\mathbf{v} - [p^{\mathrm{eq}} + \Delta p]\,\mathbf{1}, \tag{B.28}$$

where we have made convenient use of the new non-dimensional variable

$$\mathbf{v} = \boldsymbol{\gamma}\sqrt{3c_\kappa N_b} = \boldsymbol{\xi}\sqrt{\frac{3c_\kappa}{N_b\ell_b^2}}, \tag{B.29}$$

and where $\|\mathbf{a}\|_2^2 = \mathbf{a}\cdot\mathbf{a}$. We may take the change of variables $\mathbf{v} \mapsto \mathbf{F}\cdot\mathbf{v}$, where $d^3\mathbf{v}$ is unaltered due to incompressibility, to instead write the stress as

$$\boldsymbol{\sigma} = \frac{n\mathfrak{b}T}{(2\pi)^{3/2}}\left(\mathbf{F}\mathbf{F}^T : \iiint e^{-v^2/2}\,\mathbf{v}\mathbf{v}\,d^3\mathbf{v}\right) - [p^{\mathrm{eq}} + \Delta p]\,\mathbf{1}. \tag{B.30}$$

This integral can be directly computed,

$$\iiint e^{-v^2/2}\,\mathbf{v}\mathbf{v}\,d^3\mathbf{v} = (2\pi)^{3/2}\,\mathbf{1}, \tag{B.31}$$

and we compute the same type of integral in order to retrieve $p^{\mathrm{eq}} = n\mathfrak{b}T$. Our end result is now seen to be

$$\boldsymbol{\sigma}(t) = n\mathfrak{b}T\,[\mathbf{B}(t) - \mathbf{1}] - \Delta p(t)\mathbf{1}, \tag{B.32}$$

which is exactly the incompressible Neo-Hookean model with shear modulus $n\mathfrak{b}T$.

B.4 Infinitesimal deformation

Here we seek to find the reduced form of the stress from Eq. (3.45) when an infinitesimal deformation $\mathbf{F} = \mathbf{1} + \mathbf{E}$ is applied, where $\mathbf{E}^2 \approx 0$. Incompressibility is now enforced via $\mathrm{tr}(\mathbf{E}) = 0$, and the

inverse is $\mathbf{F}^{-1} = \mathbf{1} - \mathbf{E}$. We take $P^{\mathrm{eq}}(\mathbf{F}^{-1} \cdot \boldsymbol{\xi})$ appearing in Eq. (3.45), which we can write in term of $\psi^*(\mathbf{F}^{-1} \cdot \boldsymbol{\xi})$ using Eq. (3.13), and is expanded as

$$\psi^*(\mathbf{F}^{-1} \cdot \boldsymbol{\xi}) = \psi^*(\boldsymbol{\xi}) - \mathbf{E} : \left(\frac{\partial \psi^*}{\partial \boldsymbol{\xi}} \boldsymbol{\xi} \right) + O(\mathbf{E}^2). \tag{B.33}$$

We substitute this into Eq. (3.13) to obtain

$$P^{\mathrm{eq}}(\mathbf{F}^{-1} \cdot \boldsymbol{\xi}) = P^{\mathrm{eq}}(\boldsymbol{\xi}) e^{\mathbf{E} : \left(\frac{\partial \beta \psi^*}{\partial \boldsymbol{\xi}} \boldsymbol{\xi} \right) + O(\mathbf{E}^2)}. \tag{B.34}$$

The equilibrium pressure p^{eq} is the form of the stress when $\mathbf{F} = \mathbf{1}$, so we will receive a term in the integrand of the following form, which is simplified for small $\mathbf{E}$ as

$$P^{\mathrm{eq}}(\boldsymbol{\xi}) \left[e^{\mathbf{E} : \left(\frac{\partial \beta \psi^*}{\partial \boldsymbol{\xi}} \boldsymbol{\xi} \right) + O(\mathbf{E}^2)} - 1 \right] = P^{\mathrm{eq}}(\boldsymbol{\xi}) \left[\mathbf{E} : \left(\frac{\partial \beta \psi^*}{\partial \boldsymbol{\xi}} \boldsymbol{\xi} \right) + O(\mathbf{E}^2) \right], \tag{B.35}$$

$$= \mathbf{E} : \left(-\frac{\partial P^{\mathrm{eq}}}{\partial \boldsymbol{\xi}} \boldsymbol{\xi} \right) + O(\mathbf{E}^2). \tag{B.36}$$

We recall that both P^{eq} and ψ^* are spherically symmetric, and substitute into Eq. (3.45) for

$$\boldsymbol{\sigma} = n \int_0^\infty \left(-\frac{\partial P^{\mathrm{eq}}}{\partial \xi} \right) \left(\frac{\partial \psi^*}{\partial \xi} \right) \xi^4 \, d\xi \mathbf{E} : \iint \hat{\boldsymbol{\xi}} \hat{\boldsymbol{\xi}} \hat{\boldsymbol{\xi}} \hat{\boldsymbol{\xi}} \, d\Omega - \Delta p \mathbf{1} + O(\mathbf{E}^2). \tag{B.37}$$

Now, after completing the integral

$$\mathbf{E} : \iint \hat{\boldsymbol{\xi}} \hat{\boldsymbol{\xi}} \hat{\boldsymbol{\xi}} \hat{\boldsymbol{\xi}} \, d\Omega = \frac{4\pi}{15} \left[2\mathbf{E} + \mathrm{tr}(\mathbf{E}) \mathbf{1} \right], \tag{B.38}$$

where $\mathrm{tr}(\mathbf{E}) = 0$ here, and then defining the shear modulus as

$$\mu = \frac{4\pi}{15} n \mathfrak{b} T \iiint \left(-\frac{\partial P^{\mathrm{eq}}}{\partial \xi} \right) \left(\frac{\partial \beta \psi^*}{\partial \xi} \right) \xi^4 \, d\xi, \tag{B.39}$$

we can finally write the stress as

$$\boldsymbol{\sigma}(t) = 2\mu \mathbf{E}(t) - \Delta p(t) \mathbf{1} + O(\mathbf{E}^2), \tag{B.40}$$

which are Eqs. (3.47) and (3.48) from the manuscript.

B.5 `Python` package

This section details the source code (a `Python` package constituted by three distinct modules) and several self-contained examples pertaining to the `Python` implementation [126] of the model [125] developed in Chapter 3 and Appendix B. The `Python` package is available on `GitHub` and `PyPI` (`pip install Buche_Silberstein_model_2020`). It was written for `Python 3`, and uses `numpy` and `scipy`. After installation, the package is best imported using:

```
from Buche_Silberstein_model_2020 import *
```

B.5.1 Basic Usage

The script below is a simple, concise example of using this package:

```
# Import the package
from Buche_Silberstein_model_2020 import *

# Create the constitutive model
model = Buche_Silberstein_model_2020(number_of_links = 25, nondimensional_link_stiffness = 50)

# Define the applied deformation and specify the boundary conditions
def F(t): return 1 + t
F.type = 'uniaxial', '11'

# Solve the constitutive model (for the other deformations components, the stress, etc.)
model.solve(F, [0, 3])
```

After importing the package, the constitutive model is created using:

```
model = Buche_Silberstein_model_2020(number_of_links =25, \
    nondimensional_link_stiffness =50)
```

The extensible freely-jointed chain (EFJC) model is automatically utilized. Note that the number of links in the chain (N_b) and the nondimensional link stiffness (κ) have been specified as 25 and 50, respectively. Also note the following optional keyword arguments:

- `shear_modulus`, which defaults to 1 (nondimensional stress);
- `bulk_modulus`, which defaults to 1000 times `shear_modulus` (effectively incompressible);
- `method`, which defaults to `'Gibbs_Legendre'` (other option is `'Gaussian-Gibbs-Legendre'`, and `'Helmholtz'` will be implemented in a future release);
- `num_grid_romb`, which defaults to `(1 + 2**9)` and controls the number of grid points used for completing spatial integrals.

The constitutive model also inherits the following single-chain functions from the EFJC model, each as a function of the nondimensional extension (i.e. `model.eta(2.3)` gives the nondimensional force from a nondimensional extension of 2.3):

- `eta`, the nondimensional force (Gibbs-Legendre method);
- `vartheta`, the nondimensional Helmholtz free energy per link (Gibbs-Legendre method);
- `P_eq`, the nondimensional equilibrium distribution (Gibbs-Legendre method);
- `g_eq`, the nondimensional equilibrium radial distribution function (Gibbs-Legendre method);
- `P_eq_Gaussian`, the nondimensional equilibrium distribution (Gibbs-Legendre method);
- `g_eq_Gaussian`, the nondimensional equilibrium radial distribution function (Gaussian-Gibbs-Legendre method).

After creating the constitutive model, we define the applied deformation and the associated traction boundary conditions:

```
def F(t): return 1 + t
F.type = 'uniaxial', '11'
```

This corresponds to uniaxial tension in the 1-direction. The solver will apply the boundary conditions and find the unknown deformation components. The solver also reduces spatial integration time (2D instead of 3D) through utilization of the symmetry conserved by the deformation in this case. The model is solved over a period of 3 seconds via

```
model.solve(F, [0, 3])
```

An array of time values for the solution to be evaluated at can be given instead of a timespan. The solution, attributed as model.solution, itself has several attributes (i.e. model.solution.t returns the attribute t):

- `t`, the times where the solution have been evaluated (shape N);
- `F`, the deformation gradient (shape 3 by 3 by N);
- `Cauchy_stress`, the true stress tensor (shape 3 by 3 by N);
- `nominal_stress`, the engineering stress tensor (shape 3 by 3 by N);
- `Hencky_strain`, the true strain tensor (shape 3 by 3 by N).

Parameter study

The example script below generates a large portion of Figs. 3.3(a) and 3.4(a) from Chapter 3:

```
# Import the package
from Buche_Silberstein_model_2020 import *

# Import matplotlib for plotting and saving
import matplotlib.pyplot as plt
def save_current_figure(xlabel, ylabel, name):
  plt.xlabel(xlabel)
  plt.ylabel(ylabel)
  plt.tight_layout()
  plt.show()
  fig = plt.gcf()
  plt.savefig(name)
  plt.close()

# Compare equilibrium radial distribution functions as the number of links increases
number_of_links_list = [3, 5, 10, 25]
gamma_plotting = np.linspace(0, 1.6, 1000)
for number_of_links in number_of_links_list:
  model = Buche_Silberstein_model_2020(number_of_links = number_of_links, nondimensional_link_stiffness = 50)
  plt.plot(gamma_plotting, model.g_eq(gamma_plotting), color = 'blue')
  plt.plot(gamma_plotting, model.g_eq_Gaussian(gamma_plotting), color = 'red')
plt.legend(['Gibbs-Legendre', 'Gaussian'])
save_current_figure('Nondimensional single-chain extension', 'Nondimensional radial distribution function', 'g_eq.png')

# Compare uniaxial tension results as the number of links increases
def F(t): return 1 + t
F.type = 'uniaxial', '11'
model_0 = Neo_Hookean()
model_0.solve(F, [0, 3])
for number_of_links in number_of_links_list[1:]:
  model_1 = Buche_Silberstein_model_2020(number_of_links = number_of_links, nondimensional_link_stiffness = 50)
  model_2 = Buche_Silberstein_model_2020(number_of_links = number_of_links, nondimensional_link_stiffness = 50, \
    method = 'Gaussian-Gibbs-Legendre')
```

```
  model_1.solve(F, [0, 3])
  model_2.solve(F, [0, 3])
  plt.plot(model_1.solution.F[0, 0, :], model_1.solution.Cauchy_stress[0, 0, :], color = 'blue')
  plt.plot(model_2.solution.F[0, 0, :], model_2.solution.Cauchy_stress[0, 0, :], color = 'red')
  if number_of_links == number_of_links_list[1]:
    plt.plot(model_0.solution.F[0, 0, :], model_0.solution.Cauchy_stress[0, 0, :], color = 'green')
plt.xlim([1, 4])
plt.ylim([0, 35])
plt.legend(['Gibbs-Legendre', 'Gaussian-Gibbs-Legendre', 'Neo-Hookean'])
save_current_figure('Applied stretch (uniaxial tension)', 'Nondimensional stress', 'stress.png')
```

B.5.2 Python Modules

The Python module file `miscellaneous.py` is printed below:

```
########################################################################################################################
# Miscellaneous functions and variables
########################################################################################################################

# Import libraries
import sys
import numpy as np
import scipy.linalg as la

# Use to avoid overflow
minimum_float = sys.float_info.min
maximum_float = sys.float_info.max
minimum_exponent = np.log(minimum_float)/np.log(10)
maximum_exponent = np.log(maximum_float)/np.log(10)

# The identity tensors
identity_tensor = np.diag([1, 1, 1])
fourth_rank_identity_tensor = np.tensordot(identity_tensor, identity_tensor, axes = 0)

# Function returning the deviatoric portion of a tensor
def dev(A):
  return A - A.trace()/3*identity_tensor

# Function to check symmetries of deformation gradient
def symmetry_check(F):
  diagonal_check = np.count_nonzero(F - np.diag(np.diagonal(F)))
  if diagonal_check == 0:
    if np.isclose(F[1, 1]**2, 1/F[0, 0]) and np.isclose(F[1, 1], F[2, 2]):
      return 'uniaxial'
    elif np.isclose(F[2, 2], 1/F[0, 0]**2) and np.isclose(F[1, 1], F[0, 0]):
      return 'equibiaxial'
    else:
      return 'diagonal'
  else:
    return 'none'
```

The Python module file `single_chain.py` is printed below:

```
########################################################################################################################
# General setup
########################################################################################################################

# Import miscellaneous functions and variables
from Buche_Silberstein_model_2020.miscellaneous import *

# Import libraries
import numpy as np
from scipy.integrate import quad
from scipy.optimize import root_scalar

########################################################################################################################
# General single-chain class
########################################################################################################################

class single_chain:

  # Initialization
  def __init__(self, **kwargs):

    # Default parameter values
    self.epsabs = 1e-3
    self.epsrel = 1e-3
    self.num_interp = int(3e3)
    self.interp_kind_1D = 'cubic'
    self.cutoff_for_log_over_sinh = 3e1
    self.cutoff_stretch_for_harmonic = 3
    self.N_b = 88
    self.kappa = 888

    # Retrieve specified parameters
```

```
    for key, value in kwargs.items():
      exec("self.%s = value" % key, {'self': self, 'value': value})

  # Function to invert a function
  def inv_fun_1D(self, x_query, fun, bounds = None):
    return root_scalar(lambda x: fun(x) - x_query, x0 = x_query*0.95, x1 = x_query/0.95).root

  # Function to avoid overflow when computing ln(x/sinh(x))
  def log_over_sinh(self, x):

    # Determine when argument is sufficiently large
    where_x_large = np.nan_to_num(x, nan = -1) > self.cutoff_for_log_over_sinh
    log_of_x_over_sinh_x = np.zeros(x.shape)

    # Use asymptotic relation valid for sufficiently large arguments
    if where_x_large.any():
      log_of_x_over_sinh_x[where_x_large] = np.log(2*x[where_x_large]) - x[where_x_large]

    # Compute analytically otherwise, and zero where argument is zero
    where_x_zero = x == 0
    where_compute = ~(where_x_large + where_x_zero)
    if where_compute.any():
      log_of_x_over_sinh_x[where_compute] = np.log(x[where_compute]/np.sinh(x[where_compute]))
    return log_of_x_over_sinh_x

  # Hyperbolic cotangent function
  def coth_safe(self, eta):
    eta = np.where(eta == 0, minimum_float, eta)
    return 1/np.tanh(eta)

  # Langevin function
  def Langevin(self, eta):
    eta = np.where(eta == 0, minimum_float, eta)
    return 1/np.tanh(eta) - 1/eta

  # Inverse Langevin function
  def inv_Langevin(self, gamma):
    return self.inv_fun_1D(gamma, self.Langevin)

################################################################################################################################
# Extensible freely-joined chain model
################################################################################################################################

class EFJC(single_chain):

  # For more information, see:
  #   Analytical results of the extensible freely jointed chain model
  #   Alessandro Fiasconaro and Fernando Falo
  # Physica A, 532, 121929 (2019)
  #   doi.org/10.1016/j.physa.2019.121929
  # See also:
  # Statistical mechanical constitutive theory of polymer networks:
  #   The inextricable links between distribution, behavior, and ensemble
  #   Michael R. Buche and Meredith N. Silberstein
  #   Physical Review E, 102, 012501 (2020)
  #   doi.org/10.1103/PhysRevE.102.012501

  # Initialization
  def __init__(self, **kwargs):

    # Retrieve default and specified parameter values
    single_chain.__init__(self, **kwargs)

    # Nondimensional mechanical response of a single chain
    def gamma_fun(eta):
      coth = self.coth_safe(eta)
      L = self.Langevin(eta)
      return L + eta/self.kappa*(1 + (1 - L*coth)/(1 + eta/self.kappa*coth))

    # Compute and store the inverted nondimensional mechanical response to interpolate from
    self.gamma_store = np.linspace(0, self.cutoff_stretch_for_harmonic, self.num_interp)
    self.eta_store = np.zeros(self.gamma_store.size)
    for i in range(1, len(self.gamma_store)):
      self.eta_store[i] = self.inv_fun_1D(self.gamma_store[i], gamma_fun)

    # Function to interpolate from the inverted nondimensional mechamical response of the chain
    def eta_fun(gamma):
      if isinstance(gamma, np.ndarray):
        eta_out = np.zeros(gamma.shape)
        harmonic_region = gamma > self.cutoff_stretch_for_harmonic
        eta_out[harmonic_region] = self.kappa*(gamma[harmonic_region] - 1)
        eta_out[~harmonic_region] = np.interp(gamma[~harmonic_region], self.gamma_store, self.eta_store)
      else:
        if gamma > self.cutoff_stretch_for_harmonic:
          eta_out = self.kappa*(gamma - 1)
        else:
          eta_out = np.interp(gamma, self.gamma_store, self.eta_store)
      return eta_out

    # Nondimensional Helmholtz free energy per link of a single chain
    def vartheta_fun(gamma):
```

```
      # Compute mechanical response
      eta = np.array(eta_fun(gamma))
      eta[eta == 0] = minimum_float

      # Compute nondimensional Helmholtz free energy per link
      coth = self.coth_safe(eta)
      L = self.Langevin(eta)
      return eta*L + self.log_over_sinh(eta) - np.log(1 + eta/self.kappa*coth) \
        + eta**2/self.kappa/2*(1/2 + (1 - L*coth)/(1 + eta/self.kappa*coth))

    # Nondimensional equilibrium distribution function
    def P_eq_fun(gamma, normalization = 1):

      # Compute nondimensional Helmholtz free energy per link
      vartheta = vartheta_fun(gamma)

      # Compute and return P_eq
      return np.exp(-self.N_b*vartheta)/normalization

    # Nondimensional equilibrium radial distribution function
    def g_eq_fun(gamma, normalization = 1):
      return 4*np.pi*gamma**2*P_eq_fun(gamma, normalization)

    # Normalize the equilibrium distribution
    P_eq_normalization = quad(g_eq_fun, 0, np.inf, epsabs = self.epsabs, epsrel = self.epsrel, full_output = 1)[0]

    # Compute average nondimensional end-to-end length at equilibrium
    integrand = lambda gamma: gamma*g_eq_fun(gamma, normalization = P_eq_normalization)
    self.average_gamma_eq = quad(integrand, 0, np.inf, epsabs = self.epsabs, epsrel = self.epsrel, full_output = 1)[0]

    # Nondimensional equilibrium (Gaussian) distribution function valid in the limit N_b -> inf
    def P_eq_Gaussian_fun(gamma):
      c_kappa = self.kappa*(self.kappa + 1)/(self.kappa*self.kappa + 6*self.kappa + 3)
      return (3*c_kappa*self.N_b/(2*np.pi))**(3/2)*np.exp(-3/2*c_kappa*self.N_b*gamma*gamma)

    # Nondimensional equilibrium radial (Gaussian) distribution function valid in the limit N_b -> inf
    def g_eq_Gaussian_fun(gamma, normalization = 1):
      return 4*np.pi*gamma**2*P_eq_Gaussian_fun(gamma)

    # Retain each single-chain function
    self.eta = eta_fun
    self.vartheta = vartheta_fun
    self.P_eq = lambda gamma: P_eq_fun(gamma, normalization = P_eq_normalization)
    self.g_eq = lambda gamma: g_eq_fun(gamma, normalization = P_eq_normalization)
    self.P_eq_Gaussian = lambda gamma: P_eq_Gaussian_fun(gamma)
    self.g_eq_Gaussian = lambda gamma: g_eq_Gaussian_fun(gamma)
```

The Python module file `constitutive_model.py` is printed below:

```
################################################################################################################################################
# General setup
################################################################################################################################################

# Import single-chain model library
from Buche_Silberstein_model_2020.single_chain import *

# Import libraries
import numpy as np
import scipy.linalg as la
from scipy.integrate import romb
from scipy.optimize import root_scalar

################################################################################################################################################
# Solve a model with no internal state variables
################################################################################################################################################

class solver:

  # Currently only implemented in F principal coordinates
  # Also assumes initially isotropic and some symmetry is preserved by the deformation
  def solve(self, F_applied, t_span, *args, **kwargs):

    # Residual function to apply traction boundary conditions
    if F_applied.type[0] == 'uniaxial':
      def guess(t):
        return 1/np.sqrt(F_applied(t))
      app_comp = int(F_applied.type[1][0]) - 1
      def residual(t, F_traction_free, return_F = False):
        F = np.zeros((3, 3))
        F[app_comp, app_comp] = F_applied(t)
        F[app_comp - 1, app_comp - 1] = F_traction_free
        F[app_comp - 2, app_comp - 2] = F_traction_free
        if return_F is True:
          return F
        else:
          return self.Cauchy_stress(t, F)[app_comp - 1, app_comp - 1]

    # Default to a certain number of points in time
    if len(t_span) == 2:
      t_span = np.linspace(t_span[0], t_span[1], 23)
    class solution:
```

```
      pass
    solution.t = t_span

    # Function to solve for F components using boundary conditions
    def F_fun(t):
      F_traction_free = root_scalar(lambda F: residual(t, F), x0 = guess(t)*0.95, x1 = guess(t)/0.95).root
      return residual(t, F_traction_free, return_F = True)

    # Recompute and store all components of deformation gradient
    solution.F = np.zeros((3, 3, len(solution.t)))
    for index_t in range(len(solution.t)):
      solution.F[:, :, index_t] = F_fun(solution.t[index_t])

    # Compute relevant stresses
    solution.Cauchy_stress = np.zeros((3, 3, len(solution.t)))
    solution.nominal_stress = np.zeros((3, 3, len(solution.t)))
    solution.Hencky_strain = np.zeros((3, 3, len(solution.t)))
    for index_t in range(len(solution.t)):
      solution.Cauchy_stress[:, :, index_t] = \
        self.Cauchy_stress(solution.t[index_t], solution.F[:, :, index_t])
      solution.nominal_stress[:, :, index_t] = \
        la.det(solution.F[:, :, index_t])*la.inv(solution.F[:, :, index_t]).dot(solution.Cauchy_stress[:, :, index_t])
      solution.Hencky_strain[:, :, index_t] = \
        la.logm(la.sqrtm(solution.F[:, :, index_t].dot(solution.F[:, :, index_t].T)))

    # Return solution object with t, F, and stresses as attributes
    self.solution = solution

################################################################################################################################
# General class for elastic models
################################################################################################################################

class elastic(solver):

  # Initialization
  def __init__(self, **kwargs):

    # Default parameter values
    self.shear_modulus = 1
    self.number_of_links = 88
    self.nondimensional_link_stiffness = 888
    self.num_grid_romb = 1 + 2**9
    self.method = 'Gibbs-Legendre'

    # Retrieve specified parameters
    for key, value in kwargs.items():
      exec("self.%s = value" % key, {'self': self, 'value': value})

  # Check if bulk modulus specified, otherwise ensure much larger than modulus
  def bulk_modulus_fun(self):
    try:
      self.bulk_modulus
    except AttributeError:
      return 1e3*self.shear_modulus
    else:
      return self.bulk_modulus

  # Default spherical term in Cauchy stress for effectively incompressible models
  def spherical_Cauchy_stress(self, F):
    J = la.det(F)
    return self.bulk_modulus_fun()*(J - 1)*identity_tensor

################################################################################################################################
# Neo-Hookean model
################################################################################################################################

class Neo_Hookean(elastic):

  # For more information, see:
  # Large elastic deformations of isotropic materials. I. Fundamental concepts.
  #   Ronald S. Rivlin
  #   Philosophical Transactions of the Royal Society of London.
  #   Series A, Mathematical and Physical Sciences, 240, 822 (1948)
  #   doi.org/10.1098/rsta.1948.0002

  # Initialization
  def __init__(self, **kwargs):

    # Retrieve default and specified parameter values
    elastic.__init__(self, **kwargs)

  # Stress due to the applied deformation
  def Cauchy_stress(self, *args):

    # Kinematics
    F = args[-1]
    J = la.det(F)
    B = F.dot(F.T)
    dev_B_bar = dev(B)/J**(2/3)

    # Constitutive relation for the stress
    return self.shear_modulus*dev_B_bar/J + self.spherical_Cauchy_stress(F)
```

```
########################################################################################################################################################
# Buche-Silberstein model
########################################################################################################################################################

class Buche_Silberstein_model_2020(elastic, EFJC):

  # For more information, see:
  # Statistical mechanical constitutive theory of polymer networks:
  #   The inextricable links between distribution, behavior, and ensemble
  #   Michael R. Buche and Meredith N. Silberstein
  #   Physical Review E, 102, 012501 (2020)
  #   doi.org/10.1103/PhysRevE.102.012501

  # Initialization
  def __init__(self, **kwargs):

    # Retrieve default and specified parameter values
    elastic.__init__(self, **kwargs)
    EFJC.__init__(self, N_b = self.number_of_links, kappa = self.nondimensional_link_stiffness)

    # Mark grid not initialized
    self.grid_initialized = False

    # Will implement the Helmholtz method here in the future
    if self.method == 'Helmholtz':
      pass
    elif self.method == 'Gibbs-Legendre':
      self.P_eq_used = self.P_eq
    elif self.method == 'Gaussian-Gibbs-Legendre':
      self.P_eq_used = self.P_eq_Gaussian

  def initialized_grid(self, symmetry):

    # Mark grid as initialized using the given symmetry
    self.grid_initialized = symmetry

    if symmetry == 'uniaxial':
      w = np.linspace(0, 1, 1 + self.num_grid_romb)[:-1]
      self.dw = w[1] - w[0]
      W_z, W_r = np.meshgrid(w, w)
      self.Z = np.arctanh(W_z)
      self.R = np.arctanh(W_r)
      GAMMA = np.sqrt(self.Z*self.Z + self.R*self.R)
      ELEMENT = 4*np.pi*self.R/(1 - W_z*W_z)/(1 - W_r*W_r)
      GEOM = np.zeros((self.num_grid_romb, self.num_grid_romb, 3, 3))
      GEOM[:, :, 0, 0] = self.Z*self.Z
      GEOM[:, :, 1, 1] = self.R*self.R/2
      GEOM[:, :, 2, 2] = self.R*self.R/2
      ETA_OVER_GAMMA = np.zeros(GAMMA.shape)
      ETA_OVER_GAMMA[GAMMA != 0] = self.eta(GAMMA[GAMMA != 0])/GAMMA[GAMMA != 0]
      self.ELEMENT_STRESS = (self.shear_modulus*self.number_of_links*ETA_OVER_GAMMA*ELEMENT)[:, :, None, None]*GEOM

    # No (None) symmetry
    else:
      w = np.linspace(-1, 1, 2 + self.num_grid_romb)[1:-1]
      self.dw = w[1] - w[0]
      W_x, W_y, W_z = np.meshgrid(w, w, w)
      self.X = np.arctanh(W_x)
      self.Y = np.arctanh(W_y)
      self.Z = np.arctanh(W_z)
      GAMMA = np.sqrt(self.X*self.X + self.Y*self.Y + self.Z*self.Z)
      ELEMENT = 1/(1 - W_x*W_x)/(1 - W_y*W_y)/(1 - W_z*W_z)
      GEOM = np.zeros((self.num_grid_romb, self.num_grid_romb, self.num_grid_romb, 3, 3))
      GEOM[:, :, :, 0, 0] = self.X*self.X
      GEOM[:, :, :, 0, 1] = self.X*self.Y
      GEOM[:, :, :, 0, 2] = self.X*self.Z
      GEOM[:, :, :, 1, 0] = self.Y*self.X
      GEOM[:, :, :, 1, 1] = self.Y*self.Y
      GEOM[:, :, :, 1, 2] = self.Y*self.Z
      GEOM[:, :, :, 2, 0] = self.Z*self.X
      GEOM[:, :, :, 2, 1] = self.Z*self.Y
      GEOM[:, :, :, 2, 2] = self.Z*self.Z
      ETA_OVER_GAMMA = np.zeros(GAMMA.shape)
      ETA_OVER_GAMMA[GAMMA != 0] = self.eta(GAMMA[GAMMA != 0])/GAMMA[GAMMA != 0]
      self.ELEMENT_STRESS = (self.shear_modulus*self.number_of_links*ETA_OVER_GAMMA*ELEMENT)[:, :, :, None, None]*GEOM

  # Stress due to the applied deformation
  def Cauchy_stress(self, *args):

    # Kinematics
    F = args[-1]
    J = la.det(F)
    F_bar = F/J**(1/3)

    if symmetry_check(F_bar) == 'uniaxial':

      # Only initialize the grid if it has not been initialized already using this symmetry
      if self.grid_initialized != 'uniaxial':
        self.initialized_grid('uniaxial')

      # Relatively-deformed coordinates
```

```
    GAMMA_F = np.sqrt((self.Z/F_bar[0, 0])**2 + F_bar[0, 0]*self.R*self.R)

    # Integrate for and return the stress
    INTEGRAND = self.P_eq_used(GAMMA_F)[:, :, None, None]*self.ELEMENT_STRESS
    return self.spherical_Cauchy_stress(F) + romb(romb(INTEGRAND, dx = self.dw, axis = 0), dx = self.dw, axis = 0)/J

  # General, no symmetry utilized
  else:

    # Only initialize the grid if it has not been initialized already using this symmetry
    if self.grid_initialized != None:
      self.initialized_grid(None)

    # Relatively-deformed coordinates
    F_bar_inv = la.inv(F_bar)
    X_F = F_bar_inv[0, 0]*self.X + F_bar_inv[0, 1]*self.Y + F_bar_inv[0, 2]*self.Z
    Y_F = F_bar_inv[1, 0]*self.X + F_bar_inv[1, 1]*self.Y + F_bar_inv[1, 2]*self.Z
    Z_F = F_bar_inv[2, 0]*self.X + F_bar_inv[2, 1]*self.Y + F_bar_inv[2, 2]*self.Z
    GAMMA_F = np.sqrt(X_F*X_F + Y_F*Y_F + Z_F*Z_F)

    # Integrate for and return the stress
    INTEGRAND = self.P_eq_used(GAMMA_F)[:, :, :, None, None]*self.ELEMENT_STRESS
    return self.spherical_Cauchy_stress(F) + \
      romb(romb(romb(INTEGRAND, dx = self.dw, axis = 0), dx = self.dw, axis = 0), dx = self.dw, axis = 0)/J
```

Appendix C

Supporting Information for Chapter 5

C.1 Simplification under transition state theory

Here we detail the mathematical operations involved with simplifying the reaction rates $\mathcal{R}'_j(\boldsymbol{\xi};t)$ and $\mathcal{R}''_j(\boldsymbol{\xi};t)$ in Eqs. (5.13) and (5.14) when applying the assumptions of transition state theory, resulting in Eqs. (5.24) and (5.25). We first substitute the primary assumption of transition state theory Eq. (5.16) into Eq. (5.13) for

$$\mathcal{R}'_j(\boldsymbol{\xi};t) = \frac{P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\mathfrak{q}P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})} \int \cdots \int e^{-\beta H(\Gamma)} \frac{p_j}{m_j} \Theta(p_j) \delta^3 \left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right] \delta\left(\ell_j^\ddagger - \ell_j\right) \prod_{\substack{i=1 \\ i\neq j}}^{M} \Theta\left(\ell_i^\ddagger - \ell_i\right) \, d\Gamma. \tag{C.1}$$

Similarly, we substitute Eq. (5.18) into Eq. (5.14) for

$$\mathcal{R}''_j(\boldsymbol{\xi};t) = \frac{P_{\mathrm{B}_j}(\boldsymbol{\xi};t)}{\mathfrak{q}P_{\mathrm{B}_j}^{\mathrm{eq}}(\boldsymbol{\xi})} \int \cdots \int e^{-\beta H(\Gamma)} \left(-\frac{p_j}{m_j}\right) \Theta(-p_j) \delta^3 \left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right] \delta\left(\ell_j^\ddagger - \ell_j\right) \prod_{\substack{i=1 \\ i\neq j}}^{M} \Theta\left(\ell_i^\ddagger - \ell_i\right) \, d\Gamma. \tag{C.2}$$

Note that we have used Eq. (5.3), which is $f^{\mathrm{eq}}(\Gamma) = e^{-\beta H(\Gamma)}/\mathfrak{q}$. We may complete the portion of these integrals related to the momentum along the reaction coordinate, which contributes a factor of $\mathfrak{b}T = 1/\beta$ in either case, and where we retain a delta function in order to keep the integration over the whole phase space:

$$\int \cdots \int e^{-\beta H(\Gamma)} \frac{p_j}{m_j} \Theta(p_j) \delta^3 \left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right] \delta\left(\ell_j^\ddagger - \ell_j\right) \prod_{\substack{i=1 \\ i \neq j}}^{M} \Theta\left(\ell_i^\ddagger - \ell_i\right) d\Gamma =$$

$$\int \cdots \int e^{-\beta H(\Gamma)} \left(-\frac{p_j}{m_j}\right) \Theta(-p_j) \delta^3 \left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right] \delta\left(\ell_j^\ddagger - \ell_j\right) \prod_{\substack{i=1 \\ i \neq j}}^{M} \Theta\left(\ell_i^\ddagger - \ell_i\right) d\Gamma =$$

$$\frac{1}{\beta} \int \cdots \int e^{-\beta H(\Gamma)} \delta\left(p_j\right) \delta^3 \left[\mathbf{R}(\Gamma) - \boldsymbol{\xi}\right] \delta\left(\ell_j^\ddagger - \ell_j\right) \prod_{\substack{i=1 \\ i \neq j}}^{M} \Theta\left(\ell_i^\ddagger - \ell_i\right) d\Gamma. \tag{C.3}$$

The remaining integral is the partition function of the transition state $\mathfrak{q}_{\ddagger_j}^*(\boldsymbol{\xi})$ in Eq. (5.23), and the reaction rates are now rewritten as

$$\mathcal{R}_j'(\boldsymbol{\xi}; t) = \frac{\mathfrak{q}_{\ddagger_j}^*(\boldsymbol{\xi})}{\beta \mathfrak{q} P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})} P_{\mathrm{A}}(\boldsymbol{\xi}; t), \tag{C.4}$$

$$\mathcal{R}_j''(\boldsymbol{\xi}; t) = \frac{\mathfrak{q}_{\ddagger_j}^*(\boldsymbol{\xi})}{\beta \mathfrak{q} P_{\mathrm{B}_j}^{\mathrm{eq}}(\boldsymbol{\xi})} P_{\mathrm{B}_j}(\boldsymbol{\xi}; t). \tag{C.5}$$

We use Eq. (5.19) for $\mathfrak{q} P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi}) = \mathfrak{q}_{\mathrm{A}}^*(\boldsymbol{\xi})$ and Eq. (5.20) for $\mathfrak{q} P_{\mathrm{B}_j}^{\mathrm{eq}}(\boldsymbol{\xi}) = \mathfrak{q}_{\mathrm{B}_j}^*(\boldsymbol{\xi})$; with the reaction rate coefficient functions $k_j'(\boldsymbol{\xi})$ and $k_j''(\boldsymbol{\xi})$ defined in Eq. (5.26) and Eq. (5.27), this allows us to obtain the simplified reaction rates $\mathcal{R}_j'(\boldsymbol{\xi}; t)$ and $\mathcal{R}_j''(\boldsymbol{\xi}; t)$ in Eqs. (5.24)–(5.25).

C.2 Extended derivations for the macroscopic theory

C.2.1 Macroscopically-obtained equilibrium

Here we show that the equilibrium probabilities $P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})$ and $P_{\mathrm{B}_j}^{\mathrm{tot,eq}}$ in Eqs. (5.38) and (5.39), respectively, may be obtained from the Helmholtz free energy density $a(t)$ in Eq. (5.49) through minimization. We use the Lagrange multiplier Λ to enforce the conservation requirement in Eq. (5.36) and write

$$a_\Lambda(t) \equiv a(t) - \Lambda \left[\iiint P_{\mathrm{A}}(\boldsymbol{\xi}; t) \, d^3\boldsymbol{\xi} + \sum_{j=1}^{M} P_{\mathrm{B}_j}^{\mathrm{tot}}(t) - 1 \right]. \tag{C.6}$$

We now take the functional derivative [204] of $a_\Lambda(t)$ with respect to the intact chain probability distribution $P_{\mathrm{A}}(\boldsymbol{\xi}; t)$ and each broken chain probability $P_{\mathrm{B}_j}^{\mathrm{tot}}(t)$. Evaluating the results at equilibrium and setting them equal to zero, we obtain the following:

$$\left(\frac{\delta a_\Lambda}{\delta P_{\mathrm{A}}(\boldsymbol{\xi};t)}\right)_{P^{\mathrm{tot}}_{\mathrm{B}_j}}\Bigg|^{\mathrm{eq}} = n\mu^{*,\mathrm{eq}}_{\mathrm{A}}(\boldsymbol{\xi}) - n\mathfrak{b}T - \Lambda = 0, \tag{C.7}$$

$$\left(\frac{\delta a_\Lambda}{\delta P^{\mathrm{tot}}_{\mathrm{B}_j}(t)}\right)_{P_{\mathrm{A}}}\Bigg|^{\mathrm{eq}} = n\mu^{\mathrm{eq}}_{\mathrm{B}_j} - n\mathfrak{b}T - \Lambda = 0. \tag{C.8}$$

We then see that all chemical potentials are equal at equilibrium, consistent with statistical thermodynamics [49]. Using Eq. (5.47), we find $P^{\mathrm{eq}}_{\mathrm{A}}(\boldsymbol{\xi})$ to be

$$P^{\mathrm{eq}}_{\mathrm{A}}(\boldsymbol{\xi}) = \frac{e^{\beta\Lambda/n+1}}{N}\,\mathfrak{q}^*_{\mathrm{A}}(\boldsymbol{\xi}), \tag{C.9}$$

and using Eq. (5.48) with Eq. (5.32) and Eq. (5.33), we find each $P^{\mathrm{tot,eq}}_{\mathrm{B}_j}$ to be

$$P^{\mathrm{tot,eq}}_{\mathrm{B}_j} = \frac{e^{\beta\Lambda/n+1}}{N}\,\mathfrak{q}_{\mathrm{A}} e^{-\beta\Delta\Psi_{0_j}}. \tag{C.10}$$

Recall that $\mathfrak{q}_{\mathrm{A}}$ is the integral of $\mathfrak{q}^*_{\mathrm{A}}(\boldsymbol{\xi})$ over all $\boldsymbol{\xi}$. In order to solve for the Lagrange multiplier Λ we apply the conservation requirement Eq. (5.36), to obtain

$$\frac{e^{\beta\Lambda/n+1}}{N}\,\mathfrak{q}_{\mathrm{A}} + \sum_{j=1}^{M}\frac{e^{\beta\Lambda/n+1}}{N}\,\mathfrak{q}_{\mathrm{A}} e^{-\beta\Delta\Psi_{0_j}} = 1, \tag{C.11}$$

which is then rearranged to solve for the entire factor

$$\frac{e^{\beta\Lambda/n+1}}{N} = \frac{1}{\mathfrak{q}_{\mathrm{A}}}\frac{1}{1+\sum_{j=1}^{M} e^{-\beta\Delta\Psi_{0_j}}}. \tag{C.12}$$

Substitution of this factor into Eq. (C.9) results in Eq. (5.38), and substitution into Eq. (C.10) results in Eq. (5.39).

C.2.2 Retrieving the stress

Starting from the hyperelastic form of the stress given in Eq. (5.57), here we retrieve the form of the stress in Eq. (5.59). To begin, we consider the evolution of the probability distributions due to the deformation from Eq. (5.40),

$$\left.\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial t}\right|_{\mathbf{F}} = -\frac{\partial}{\partial\boldsymbol{\xi}}\cdot\left[\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t)P_{\mathrm{A}}(\boldsymbol{\xi};t)\right], \tag{C.13}$$

and from Eq. (5.43), we have more simply

$$\left.\frac{\partial P^{\mathrm{tot}}_{\mathrm{B}_j}(t)}{\partial t}\right|_{\mathbf{F}} = 0. \tag{C.14}$$

We now integrate by parts using an extension of the divergence theorem, neglecting the probability of intact chains existing on the boundary $P_{\mathrm{A}}(\boldsymbol{\xi};t)|_{\partial_{\boldsymbol{\xi}}} \approx 0$ as we have previously discussed near the end of Sec. 5.2.1,

$$\iiint \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \left[\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t) P_{\mathrm{A}}(\boldsymbol{\xi};t)\right] \mu_{\mathrm{A}}^{*}(\boldsymbol{\xi};t)\, d^3\boldsymbol{\xi} = -\iiint P_{\mathrm{A}}(\boldsymbol{\xi};t) \frac{\partial \mu_{\mathrm{A}}^{*}(\boldsymbol{\xi};t)}{\partial \boldsymbol{\xi}} \cdot \dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t)\, d^3\boldsymbol{\xi}. \tag{C.15}$$

Next, we make the affine assumption $\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t) = \mathbf{L}(t) \cdot \boldsymbol{\xi}$, which is that the end-to-end vectors $\boldsymbol{\xi}$ are affinely deformed by the deformation gradient $\mathbf{F}(t)$ on average. Upon carrying out the derivatives involved in Eq. (5.57), the stress is

$$\boldsymbol{\sigma}(t) = n \iiint P_{\mathrm{A}}(\boldsymbol{\xi};t) \frac{\partial \mu_{\mathrm{A}}^{*}(\boldsymbol{\xi};t)}{\partial \boldsymbol{\xi}} \boldsymbol{\xi}\, d^3\boldsymbol{\xi} - p(t)\mathbf{1}. \tag{C.16}$$

The affine assumption allows us to expand Eq. (C.13), where the divergence of $\dot{\boldsymbol{\xi}}_{\mathrm{A}}(\boldsymbol{\xi};t)$ is zero since $\mathbf{1} : \mathbf{L}(t) = 0$ due to incompressibility, leaving only the term containing the gradient of $P_{\mathrm{A}}(\boldsymbol{\xi};t)$. We substitute this nonzero term into the evolution equation for $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ in Eq. (5.40) to obtain

$$\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial t} = -\left[\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial \boldsymbol{\xi}} \boldsymbol{\xi}\right] : \mathbf{L}(t) - \sum_{j=1}^{M} k_j'(\boldsymbol{\xi}) \left[P_{\mathrm{A}}(\boldsymbol{\xi};t) - \frac{P_{\mathrm{B}_j}^{\mathrm{tot}}(t)}{P_{\mathrm{B}_j}^{\mathrm{tot,eq}}} P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})\right], \tag{C.17}$$

which is Eq. (5.58) in the manuscript. The stress in Eq. (C.16) can be written as a function of the time-independent Helmholtz free energy of an intact chain $\psi_{\mathrm{A}}^{*}(\boldsymbol{\xi})$ rather than the time-dependent corresponding chemical potential $\mu_{\mathrm{A}}^{*}(\boldsymbol{\xi};t)$. The chemical potential of an intact chain $\mu_{\mathrm{A}}^{*}(\boldsymbol{\xi};t)$ is given by Eq. (5.47) in terms of the partition function $\mathfrak{q}_{\mathrm{A}}^{*}(\boldsymbol{\xi})$ and the probability distribution $P_{\mathrm{A}}(\boldsymbol{\xi};t)$. The principal thermodynamic connection formula allows the partition function to be written as a function of the Helmholtz free energy,

$$\mathfrak{q}_{\mathrm{A}}^{*}(\boldsymbol{\xi}) = e^{-\beta \psi_{\mathrm{A}}^{*}(\boldsymbol{\xi})}, \tag{C.18}$$

so we then expand the gradient of $\mu_{\mathrm{A}}^{*}(\boldsymbol{\xi};t)$ in Eq. (5.47) as

$$\frac{\partial \mu_{\mathrm{A}}^{*}(\boldsymbol{\xi};t)}{\partial \boldsymbol{\xi}} = \frac{\partial \psi_{\mathrm{A}}^{*}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} + \frac{\mathfrak{b}T}{P_{\mathrm{A}}(\boldsymbol{\xi};t)} \frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial \boldsymbol{\xi}}. \tag{C.19}$$

Substitution of the second term in Eq. (C.19) into the stress Eq. (C.16) results in

$$\boldsymbol{\sigma}(t) = n \iiint P_{\mathrm{A}}(\boldsymbol{\xi};t) \frac{\partial \psi_{\mathrm{A}}^{*}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \boldsymbol{\xi}\, d^3\boldsymbol{\xi} + n\mathfrak{b}T \iiint \frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial \boldsymbol{\xi}} \boldsymbol{\xi}\, d^3\boldsymbol{\xi} - p(t)\mathbf{1}. \tag{C.20}$$

The second term in Eq. (C.20) is simplified by again using integration by parts and $P_{\mathrm{A}}(\boldsymbol{\xi};t)|_{\partial_{\boldsymbol{\xi}\mathcal{A}}} \approx 0$,

$$n\mathfrak{b}T \iiint \frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial \boldsymbol{\xi}} \boldsymbol{\xi}\, d^3\boldsymbol{\xi} = -n\mathfrak{b}T P_{\mathrm{A}}^{\mathrm{tot}}(t)\mathbf{1}. \tag{C.21}$$

The pressure term in Eq. (C.20) and this term in Eq. (C.21) are both spherical. Since the pressure $p(t)$ is merely a Lagrange multiplier enforcing incompressibility, we take $p(t) + n\mathfrak{b}T P_{\mathrm{A}}^{\mathrm{tot}}(t) \mapsto p(t)$ without loss of generality. The stress in Eq. (C.20) is then

$$\boldsymbol{\sigma}(t) = n \iiint P_{\mathrm{A}}(\boldsymbol{\xi};t)\,\frac{\partial \psi^*_{\mathrm{A}}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\,\boldsymbol{\xi}\,d^3\boldsymbol{\xi} - p(t)\mathbf{1}, \tag{C.22}$$

which is Eq. (5.59) in the manuscript. This stress is then nondimensionalized as follows: we first rearrange the pressure term and scale by β/n for

$$\frac{\boldsymbol{\sigma}(t) + p(t)\mathbf{1}}{\beta/n} = \iiint P_{\mathrm{A}}(\boldsymbol{\xi};t)\,\frac{\partial \beta\psi^*_{\mathrm{A}}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\,\boldsymbol{\xi}\,d^3\boldsymbol{\xi}. \tag{C.23}$$

Next, we substitute in $\boldsymbol{\xi} = N_b\ell_b\boldsymbol{\gamma}$, using $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t) \equiv (N_b\ell_b)^3 P_{\mathrm{A}}(\boldsymbol{\xi};t)$ and then $P_{\mathrm{A}}(\boldsymbol{\xi};t)\,d^3\boldsymbol{\xi} = \mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)\,d^3\boldsymbol{\gamma}$,

$$\frac{\boldsymbol{\sigma}(t) + p(t)\mathbf{1}}{\beta/n} = \iiint \mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)\,\frac{\partial \beta\psi^*_{\mathrm{A}}(\boldsymbol{\gamma})}{\partial \gamma}\left(\frac{\boldsymbol{\gamma}\boldsymbol{\gamma}}{\gamma}\right) d^3\boldsymbol{\gamma}. \tag{C.24}$$

We then substitute in $\beta\psi^*_{\mathrm{A}}(\boldsymbol{\gamma}) = N_b\vartheta^*_{\mathrm{A}}(\gamma)$, where $\eta(\boldsymbol{\gamma}) = \partial\vartheta^*_{\mathrm{A}}(\gamma)/\partial\gamma$, to obtain Eq. (5.76). When an inhomogeneous chain (consists of both breakable and unbreakable links) is used for the uFJC model, the homogeneous chain contour length transforms, $N_b\ell_b \mapsto (N_b + \varsigma N_b^{\#})\ell_b$. To adjust our nondimensional representation of the stress in Eq. (5.76), we must then transform $\boldsymbol{\gamma}$ using $N_b\gamma \mapsto (N_b + \varsigma N_b^{\#})\gamma$, where $\mathscr{P}_{\mathrm{A}}(\boldsymbol{\gamma};t)\,d^3\boldsymbol{\gamma}$ is invariant. The net result within Eq. (5.76) is effectively

$$N_b\left(\frac{\boldsymbol{\gamma}\boldsymbol{\gamma}}{\gamma}\right) \mapsto \left(N_b + \varsigma N_b^{\#}\right)\left(\frac{\boldsymbol{\gamma}\boldsymbol{\gamma}}{\gamma}\right), \tag{C.25}$$

which taken within Eq. (5.76) produces Eq. (5.80).

C.2.3 Expressing the chemical dissipation

Starting from Eq. (5.53), here we retrieve the form of the dissipation due to the chemical reactions $\mathcal{D}_{\mathrm{rxn}}(t)$ that allows us to conclude $\mathcal{D}_{\mathrm{rxn}}(t) \geq 0$. We begin by expanding Eq. (5.53),

$$\mathcal{D}_{\mathrm{rxn}}(t) = -n \iiint \left.\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial t}\right|_{\mathrm{rxn}} \mu^*_{\mathrm{A}}(\boldsymbol{\xi};t)\,d^3\boldsymbol{\xi} - n\sum_{j=1}^{M} \left.\frac{\partial P^{\mathrm{tot}}_{\mathrm{B}_j}(t)}{\partial t}\right|_{\mathrm{rxn}} \mu_{\mathrm{B}_j}(t). \tag{C.26}$$

It is possible to write the time derivatives and chemical potentials in Eq. (C.26) strictly in terms of the original forward and reverse reactions rates, $\mathcal{R}'_j(\boldsymbol{\xi};t)$ and $\mathcal{R}''_j(\boldsymbol{\xi};t)$. Referring back to Eq. (5.15), we first write

$$\left.\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial t}\right|_{\mathrm{rxn}} = -\sum_{j=1}^{M}\left[\mathcal{R}'_j(\boldsymbol{\xi};t) - \mathcal{R}''_j(\boldsymbol{\xi};t)\right]. \tag{C.27}$$

Similarly with Eq. (5.43), we use $\mathcal{R}_j(\boldsymbol{\xi};t) = k'_j(\boldsymbol{\xi})P_{\mathrm{A}}(\boldsymbol{\xi};t)$ from Eq. (5.24) to write

$$\left.\frac{\partial P^{\mathrm{tot}}_{\mathrm{B}_j}(t)}{\partial t}\right|_{\mathrm{rxn}} = \iiint \left[\mathcal{R}'_j(\boldsymbol{\xi};t) - k'_j(\boldsymbol{\xi})P^{\mathrm{eq}}_{\mathrm{A}}(\boldsymbol{\xi})\,\frac{P^{\mathrm{tot}}_{\mathrm{B}_j}(t)}{P^{\mathrm{tot,eq}}_{\mathrm{B}_j}}\right] d^3\boldsymbol{\xi}. \tag{C.28}$$

Next, we use Eq. (5.28) for $k'_j(\boldsymbol{\xi})P^{\mathrm{eq}}_{\mathrm{A}}(\boldsymbol{\xi}) = k''_j(\boldsymbol{\xi})P^{\mathrm{eq}}_{\mathrm{B}_j}(\boldsymbol{\xi})$. We then use Eq. (5.35) to simplify

$$P^{\mathrm{eq}}_{\mathrm{B}_j}(\boldsymbol{\xi})\frac{P^{\mathrm{tot}}_{\mathrm{B}_j}(t)}{P^{\mathrm{tot,eq}}_{\mathrm{B}_j}} = P^{\mathrm{eq}}_{\mathrm{B}_j}(\boldsymbol{\xi})\frac{V_{\mathrm{B}_j}P_{\mathrm{B}_j}(\boldsymbol{\xi};t)}{V_{\mathrm{B}_j}P^{\mathrm{eq}}_{\mathrm{B}_j}(\boldsymbol{\xi})} = P_{\mathrm{B}_j}(\boldsymbol{\xi};t), \tag{C.29}$$

which with $\mathcal{R}''_j(\boldsymbol{\xi};t) = k''_j(\boldsymbol{\xi})P_{\mathrm{B}_j}(\boldsymbol{\xi};t)$ from Eq. (5.25) allows us to obtain

$$\left.\frac{\partial P^{\mathrm{tot}}_{\mathrm{B}_j}(t)}{\partial t}\right|_{\mathrm{rxn}} = \iiint \left[\mathcal{R}'_j(\boldsymbol{\xi};t) - \mathcal{R}''_j(\boldsymbol{\xi};t)\right]\, d^3\boldsymbol{\xi}. \tag{C.30}$$

With Eqs. (C.27) and (C.30), the dissipation in Eq. (C.26) now becomes

$$\mathcal{D}_{\mathrm{rxn}}(t) = n\sum_{j=1}^{M}\iiint \left[\mathcal{R}'_j(\boldsymbol{\xi};t) - \mathcal{R}''_j(\boldsymbol{\xi};t)\right]\left[\mu^*_{\mathrm{A}}(\boldsymbol{\xi};t) - \mu_{\mathrm{B}_j}(t)\right]\, d^3\boldsymbol{\xi} \tag{C.31}$$

Using both Eq. (5.47) and Eq. (5.48), we obtain the difference between the chemical potentials

$$\mu^*_{\mathrm{A}}(\boldsymbol{\xi};t) - \mu_{\mathrm{B}_j}(t) = \mathfrak{b}T\ln\left[\frac{\mathfrak{q}^*_{\mathrm{B}_j}}{\mathfrak{q}^*_{\mathrm{A}}(\boldsymbol{\xi})}\frac{P_{\mathrm{A}}(\boldsymbol{\xi};t)}{P^{\mathrm{tot}}_{\mathrm{B}_j}(t)/V_{\mathrm{B}_j}}\right]. \tag{C.32}$$

After noting $\mathfrak{q}^*_{\mathrm{B}_j}/\mathfrak{q}^*_{\mathrm{A}}(\boldsymbol{\xi}) = k'_j(\boldsymbol{\xi})/k''_j(\boldsymbol{\xi})$ using Eq. (5.28) and $P^{\mathrm{tot}}_{\mathrm{B}_j}(t)/V_{\mathrm{B}_j} = P_{\mathrm{B}_j}(\boldsymbol{\xi};t)$ using Eq. (5.35),

$$\mu^*_{\mathrm{A}}(\boldsymbol{\xi};t) - \mu_{\mathrm{B}_j}(t) = \mathfrak{b}T\ln\left[\frac{k'_j(\boldsymbol{\xi})P_{\mathrm{A}}(\boldsymbol{\xi};t)}{k''_j(\boldsymbol{\xi})P_{\mathrm{B}_j}(\boldsymbol{\xi};t)}\right], \tag{C.33}$$

which with Eq. (5.24) and Eq. (5.24) allows us to obtain

$$\mu^*_{\mathrm{A}}(\boldsymbol{\xi};t) - \mu_{\mathrm{B}_j}(t) = \mathfrak{b}T\ln\left[\frac{\mathcal{R}'_j(\boldsymbol{\xi};t)}{\mathcal{R}''_j(\boldsymbol{\xi};t)}\right]. \tag{C.34}$$

Combining Eqs. (C.31) and (C.34), we can then write the total dissipation $\mathcal{D}_{\mathrm{rxn}}$ succinctly as

$$\mathcal{D}_{\mathrm{rxn}}(t) = \sum_{j=1}^{M}\iiint \mathcal{D}^*_j(\boldsymbol{\xi};t)\, d^3\boldsymbol{\xi}, \tag{C.35}$$

where $\mathcal{D}^*_j(\boldsymbol{\xi};t)$, the dissipation density for the jth reaction occurring at the end-to-end vector $\boldsymbol{\xi}$, is

$$\mathcal{D}^*_j(\boldsymbol{\xi};t) \equiv n\mathfrak{b}T\left[\mathcal{R}'_j(\boldsymbol{\xi};t) - \mathcal{R}''_j(\boldsymbol{\xi};t)\right]\ln\left[\frac{\mathcal{R}'_j(\boldsymbol{\xi};t)}{\mathcal{R}''_j(\boldsymbol{\xi};t)}\right]. \tag{C.36}$$

Eqs. (C.35) and (C.36) are Eqs. (5.60) and (5.61) in the manuscript, respectively. The right-hand side of Eq. (C.35) is of the form $c(x-y)\ln(x/y)$, which is a quantity that is positive semidefinite for all $x > 0$ and $y > 0$ if $c \geq 0$. Since the reactions rates $\mathcal{R}'_j(\boldsymbol{\xi};t)$ and $\mathcal{R}''_j(\boldsymbol{\xi};t)$ are both positive definite, we have $[\mathcal{R}'_j(\boldsymbol{\xi};t) - \mathcal{R}''_j(\boldsymbol{\xi};t)]\ln[\mathcal{R}'_j(\boldsymbol{\xi};t)/\mathcal{R}''_j(\boldsymbol{\xi};t)] \geq 0$. For finite temperatures we also have $n\mathfrak{b}T > 0$, which with Eq. (C.36) then allows us to conclude that each $\mathcal{D}^*_j(\boldsymbol{\xi};t) \geq 0$, and therefore with Eq. (C.35) that $\mathcal{D}_{\mathrm{rxn}}(t) \geq 0$.

C.3 Extended details on implementing the uFJC model

C.3.1 Asymptotic approximation for the uFJC model

Here we obtain an asymptotic approximation for the single-chain mechanical response of the uFJC model. For nondimensional end-to-end lengths γ near and below unity, the mechanical response will closely match that of the EFJC (harmonic u) if $\kappa \gg 1$. Physically, $\kappa \gg 1$ represents that thermal energy is much smaller than the characteristic energy to begin stretching the link, so thermal sampling will be effectively restricted to where u is minimized, which is also where u is harmonic. If the force remains small enough, $\eta \ll 1$, nondimensional end-to-end lengths above unity will not be reached. The analytic expression we use for the single-chain mechanical response of the EFJC in the Gibbs (isotensional) ensemble is [57]

$$\gamma_{\text{EFJC}}(\eta) = \mathcal{L}(\eta) + \frac{\eta}{\kappa}\left[1 + \frac{1 - \mathcal{L}(\eta)\coth(\eta)}{1 + (\eta/\kappa)\coth(\eta)}\right], \tag{C.37}$$

where $\mathcal{L}(\eta) = \coth(\eta) - 1/\eta$ is the Langevin function. Since we will utilize $\kappa \gg 1$ in order to asymptotically approximate the mechanical response of the uFJC, we will expand Eq. (C.37) in a series that is valid as $\kappa \to \infty$. Using the Maclaurin series for $1/(1+x)$, we then have

$$\gamma_{\text{EFJC}}(\eta) = \mathcal{L}(\eta) + \frac{\eta}{\kappa}\left\{2 - \mathcal{L}(\eta)\coth(\eta) + [1 - \mathcal{L}(\eta)\coth(\eta)]\,\eta\coth(\eta)\sum_{n=1}^{\infty}(-\kappa)^{-n}\right\} \quad \text{as } \kappa \to \infty. \tag{C.38}$$

Now, we choose to make a first-order asymptotic approximation, only keeping $O(\kappa^{-1})$ terms,

$$\gamma_{\text{EFJC}}(\eta) \sim \mathcal{L}(\eta) + \frac{\eta}{\kappa}\left[2 - \mathcal{L}(\eta)\coth(\eta)\right] \quad \text{for } \kappa \gg 1. \tag{C.39}$$

For sufficiently low forces, $\eta \ll 1$, the mechanical response of the uFJC matches that of the EFJC with the same stiffness since in either case the link stretching will be restricted to the harmonic regime. Eq. (C.39) is then the $O(\kappa^{-1})$ asymptotic approximation of the single-chain mechanical response of the uFJC for $\eta \ll 1$,

$$\gamma(\eta) \sim \mathcal{L}(\eta) + \frac{\eta}{\kappa}\left[2 - \mathcal{L}(\eta)\coth(\eta)\right] \quad \text{for } \kappa \gg 1 \text{ and } \eta \ll 1. \tag{C.40}$$

Now, for $\kappa \gg 1$ and $\gamma \gtrsim 1$ the uFJC will remain aligned and the links will begin to be stretched directly. As γ continues to increase, the corresponding large forces required, $\eta \gg 1$, will be approximately due to stretching the links alone. In other words, the forces required to significantly stretch the stiff links will eclipse the entropically-based forces. In this limit the mechanical response of the chain is asymptotically given by that of the links,

$$\gamma(\eta) \sim \lambda(\eta) \quad \text{for } \kappa \gg 1 \text{ and } \eta \gg 1, \tag{C.41}$$

where $\lambda(\eta)$ is the stretch ℓ/ℓ_b of a link under the nondimensional force η, defined through

$$\eta = \beta\ell_b \left.\frac{\partial u(\ell)}{\partial \ell}\right|_{\ell = \ell_b\lambda(\eta)}. \tag{C.42}$$

We then have two asymptotic approximations when $\kappa \gg 1$ for the mechanical response in Eqs. (C.40) and (C.41) that we must match. We may do so using Prandtl's method of asymptotic matching [146, 205], which stipulates that the following must be true to obtain a composite approximation:

$$\lim_{\eta\to\infty} \gamma_{\eta\ll 1}(\eta) = \lim_{\eta\to 0} \gamma_{\eta\gg 1}(\eta). \tag{C.43}$$

Here the limits are $1+\eta/\kappa$ in either case, thus we satisfy Eq. (C.43). Prandtl's method also stipulates that this limit must be subtracted from the composite solution obtained when adding the two approximations, otherwise it would be accounted for twice. After subtracting this common part of $1+\eta/\kappa$, we obtain the composite first-order asymptotic approximation

$$\gamma_1(\eta) \sim \mathcal{L}(\eta) + \frac{\eta}{\kappa}\left[1 - \mathcal{L}(\eta)\coth(\eta)\right] + \lambda(\eta) - 1 \quad \text{for } \kappa \gg 1. \tag{C.44}$$

This mechanical response has three distinct terms. The first, $\mathcal{L}(\eta)$, is the entropic mechanical response of the FJC that dominates at low forces. The third, $\lambda(\eta)-1$, is based upon the mechanical response of the aligned chain $\lambda(\eta)$ that dominates at high forces. After noting that

$$\left|\eta[1 - \mathcal{L}(\eta)\coth(\eta)]\right| \leq 1 \quad \text{for all } \eta, \tag{C.45}$$

we see that the second term in Eq. (C.44) is essentially an $O(\kappa^{-1})$ correction. If κ is sufficiently high, this correction is negligible and we may take the even simpler (leading-order) approximation

$$\gamma_0(\eta) \sim \mathcal{L}(\eta) + \lambda(\eta) - 1 \quad \text{for } \kappa \gg 1, \tag{C.46}$$

which is the asymptotic approximation utilized in Eq. (5.64) of the manuscript.

When applying the Morse potential to the uFJC model mechanical response in Eq. (C.46), with link potential energy $u(\ell)$ given by Eq. (5.66), we first compute the link force

$$f = \frac{\partial u(\ell)}{\partial \ell} = \sqrt{2k_b u_b}\, e^{-\sqrt{k_b/2u_b}(\ell-\ell_b)}\left[1 - e^{-\sqrt{k_b/2u_b}(\ell-\ell_b)}\right], \tag{C.47}$$

and afterward nondimensionalize ($\eta \equiv \beta f \ell_b$, $\kappa \equiv \beta k_b \ell_b^2$, $\lambda \equiv \ell/\ell_b$) to obtain the nondimensional force

$$\eta = \sqrt{2\kappa\beta u_b}\, e^{-\sqrt{\kappa/2\beta u_b}[\lambda(\eta)-1]}\left\{1 - e^{-\sqrt{\kappa/2\beta u_b}[\lambda(\eta)-1]}\right\}. \tag{C.48}$$

We then choose the transition state stretch $\lambda_\ddagger \equiv 1+\ln(2)\sqrt{2\beta u_b/\kappa}$ corresponding to the maximum force $\eta_{\max} = \sqrt{\kappa\beta u_b/8}$. Solving Eq. (C.48) for $\lambda(\eta)$, we then obtain

$$\lambda(\eta) = 1 + \sqrt{\frac{2\beta u_b}{\kappa}}\,\ln\left[\frac{2}{1+\sqrt{1-\eta/\eta_{\max}}}\right] \quad \text{for } \eta \leq \eta_{\max} = \sqrt{\frac{\kappa\beta u_b}{8}}, \tag{C.49}$$

which is Eq. (5.67) from the manuscript. We plot the leading order $O(\kappa^0)$ and first-order corrected $O(\kappa^{-1})$ asymptotic approximations of the mechanical response from Eqs. (C.44) and (C.46) for the Morse-FJC for varying κ in Fig. C.1(a). We found that varying βu_b has little effect on the accuracy of the approximation of the mechanical response. Fig. C.1(a) shows that the $O(\kappa^{-1})$ correction provides a small contribution for $\kappa = 20$, a nearly negligible contribution for $\kappa = 50$, and essentially no contribution for $\kappa = 200$ and above. Higher order corrections may be obtained, but if the $O(\kappa^{-1})$

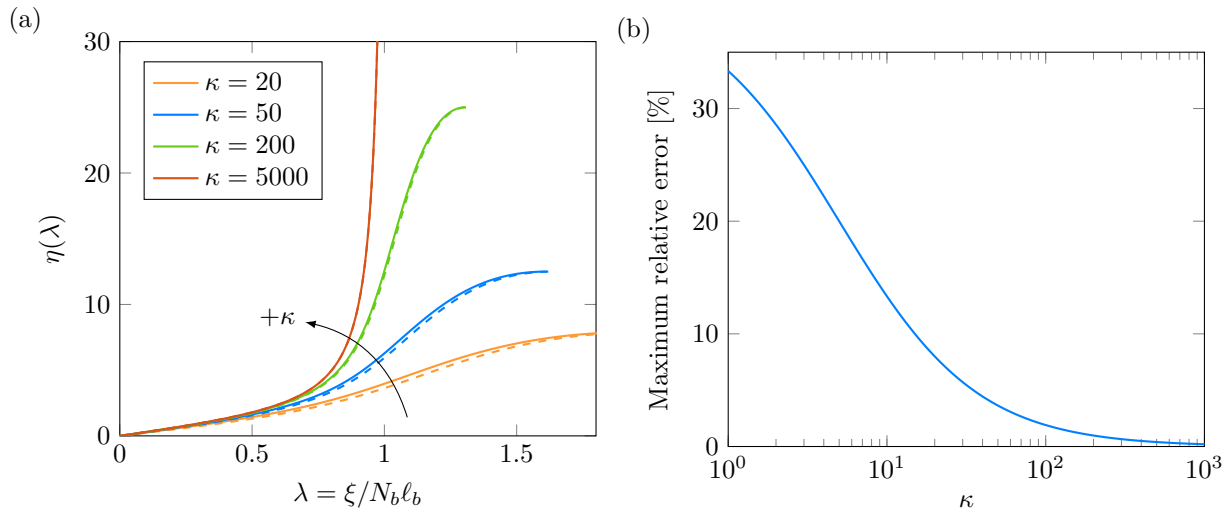


Figure C.1: (a) The nondimensional force $\eta = \beta f \ell_b$ as a function of the leading order (solid) and first-order corrected (dashed) asymptotic approximations of the corresponding nondimensional end-to-end length $\gamma = \xi/N_b\ell_b$ using the Morse-FJC model for $\beta u_b = 25$ and varying κ. (b) The maximum relative error $\max[e(\eta)]$ for $\beta u_b = 25$ and varying κ.

correction is negligible, these will surely be negligible. Going forward (and in the manuscript), we choose Eq. (C.46) as the asymptotic approximation of the mechanical response $\gamma(\eta)$ and treat the $O(\kappa^{-1})$ correction from Eq. (C.44) as an estimate of the error. The relative error $e \equiv |\gamma_1 - \gamma_0|/\gamma_1$ would then be

$$e(\eta) \sim \frac{(\eta/\kappa)\left[1 - \mathcal{L}(\eta)\coth(\eta)\right]}{\gamma_1(\eta)} \quad \text{for } \kappa \gg 1. \tag{C.50}$$

We plot the maximum (over η) of this relative error in Fig. C.1(b) as a function of κ. For smaller κ the maximum relative error can be quite large, indicating that the $O(\kappa^{-1})$ correction is (and perhaps higher order corrections are) necessary. As κ increases the maximum relative error rapidly shrinks, showing that it indeed becomes accurate to ignore the $O(\kappa^{-1})$ correction (and all higher order corrections) and utilize the Eq. (C.46) as the asymptotic approximation of the mechanical response $\gamma(\eta)$.

C.3.2 Obtaining various single-chain quantities for the uFJC model

Here we provide a full derivation of the single-chain quantities for the uFJC model leading up to the reaction rate coefficient function $k'(\gamma)$ in Eq. (5.71). We begin with the nondimensional configurational Helmholtz free energy per link $\vartheta^*_{\mathrm{A,con}}(\gamma)$, which under the Gibbs-Legendre approximation [125] may be obtained from the mechanical response $\gamma(\eta)$ via

$$\vartheta^*_{\mathrm{A,con}}(\gamma) \sim \eta\gamma(\eta) - \int \gamma(\eta)\, d\eta \quad \text{for } N_b \gg 1. \tag{C.51}$$

We then apply our asymptotic approximation for $\gamma(\eta)$ in Eq. (5.64) to obtain

$$\vartheta^*_{\mathrm{A,con}}(\gamma) = \ln\left\{\frac{\eta\exp[\eta\mathcal{L}(\eta)]}{\sinh(\eta)}\right\} + \beta u(\eta) - c_0, \tag{C.52}$$

where $c_0 = \ln(4\pi\ell_b^3\sqrt{2\pi/\kappa})$ is the constant of integration [57]. This constant is part of the constant prefactor of the partition function, and does not appear to produce the correct units in Eq. (5.69) only because we have scaled away Planck's constant $h = 1$. Starting from Eq. (5.69) and continuing in the following equations, note that functions of γ are often written in terms of η, where $\eta = \eta(\gamma)$ then represents inverting the mechanical response $\gamma(\eta)$ in Eq. (5.64) for the η corresponding to γ. Using the principal thermodynamic connection formula, $\mathfrak{q} = e^{-N_b\vartheta}$, the configuration partition function is

$$\mathfrak{q}^*_{\mathrm{A,con}}(\gamma) = \left[\ell_b\sqrt{\frac{2\pi}{\kappa}}\,e^{-\beta u(\eta)}\right]^{N_b}\mathfrak{q}^*_{\mathrm{FJC,con}}(\gamma), \tag{C.53}$$

where the configuration partition function of the FJC [50] is given by

$$\mathfrak{q}^*_{\mathrm{FJC,con}}(\gamma) = \left\{\frac{4\pi\ell_b^2\sinh(\eta)}{\eta\exp[\eta\mathcal{L}(\eta)]}\right\}^{N_b}. \tag{C.54}$$

After introducing the nondimensional equilibrium distribution $\mathscr{P}^{\mathrm{eq}}_{\mathrm{A}}(\gamma) \equiv (N_b\ell_b)^3 P^{\mathrm{eq}}_{\mathrm{A}}(\xi)$, Eq. (5.38) gives Eq. (5.70). The total probability that a chain is intact at equilibrium $P^{\mathrm{tot,eq}}_{\mathrm{A}}$ is the first factor in Eq. (5.70). Note that the momentum contribution $\mathfrak{q}^*_{\mathrm{A,mom}}$ to the partition function $\mathfrak{q}^*_{\mathrm{A}}(\gamma) = \mathfrak{q}^*_{\mathrm{A,mom}}\mathfrak{q}^*_{\mathrm{A,con}}(\gamma)$ is independent of γ and identical to $\mathfrak{q}^*_{\mathrm{B,mom}}$, causing it to vanish from Eq. (5.70). Additionally, note that the constant c_0 from Eq. (C.52) cancels when substituting in $\vartheta^*_{\mathrm{A,con}}(\gamma)$, showing c_0 is irrelevant (only the relative free energy matters) when computing $\mathscr{P}^{\mathrm{eq}}_{\mathrm{A}}(\gamma)$, which is why c_0 does not appear in Eq. (5.69) from the manuscript.

We now turn to the partition function of the transition state $\mathfrak{q}^*_{\ddagger}(\gamma)$, which is a chain of $N_b - 1$ intact links and a single link held rigidly in its transition state. Since our Gibbs-Legendre method allows us to treat each link independently, we may compute these partition function separately and multiply them. The transition state link partition function is that of the FJC with $\ell_b \mapsto \ell_\ddagger$, which also takes $\eta \mapsto \lambda_\ddagger\eta$ with $\lambda_\ddagger \equiv \ell_\ddagger/\ell_b$ due to the nondimensionalization. We then have

$$\mathfrak{q}^*_{\ddagger,\mathrm{con}}(\gamma) = e^{-\beta u(\ell_\ddagger)}\left\{\mathfrak{q}^*_{\mathrm{FJC,con}}(\gamma)\big|_{\ell_b=\ell_\ddagger}\,\mathfrak{q}^*_{\mathrm{A,con}}(\gamma)^{(N_b-1)}\right\}^{1/N_b}. \tag{C.55}$$

We then compute the ratio of $\mathfrak{q}^*_{\ddagger}(\gamma)$ to $\mathfrak{q}^*_{\mathrm{A}}(\gamma)$ in order to retrieve the reaction rate coefficient function $k'(\gamma)$ from Eq. (5.26). The ratio of the configuration contributions is found using Eqs. (C.53) and (C.55), where the contributions from the $N_b - 1$ non-transition state links cancel due to the link-independence that the Gibbs-Legendre method permits. The ratio of the momentum contributions similarly cancels except for the momentum degree of freedom from stretching the link. The contribution from this stretching degree of freedom is equivalent to that from the one-dimensional translation of the reduced mass $\nu = m/2$ of the link [49], so we then have

$$\frac{\mathfrak{q}^*_{\ddagger,\mathrm{mom}}(\gamma)}{\mathfrak{q}^*_{\mathrm{A,mom}}(\gamma)} = (2\pi\nu\mathfrak{b}T)^{-1/2}. \tag{C.56}$$

To obtain $k'(\gamma)$ from Eq. (5.26), we now multiply the ratio of $\mathfrak{q}^*_{\ddagger,\mathrm{con}}(\gamma)$ in Eq. (C.55) to $\mathfrak{q}^*_{\mathrm{A,con}}(\gamma)$ in Eq. (C.53) by $\mathfrak{q}^*_{\ddagger,\mathrm{mom}}(\gamma)/\mathfrak{q}^*_{\mathrm{A,mom}}(\gamma)$ in Eq. (C.56) and simplify. When taking the ratio of $\mathfrak{q}^*_{\ddagger,\mathrm{con}}(\gamma)$ to $\mathfrak{q}^*_{\mathrm{A,con}}(\gamma)$, all contributions from all links cancel except that from the link that may be in the transition state (analogous to the momentum contribution). After simplifying the ratio of $\mathfrak{q}^*_{\ddagger,\mathrm{con}}(\gamma)$ to $\mathfrak{q}^*_{\mathrm{A,con}}(\gamma)$ through powers of N_b, we have

$$\frac{\mathfrak{q}^*_{\ddagger,\mathrm{con}}(\gamma)}{\mathfrak{q}^*_{\mathrm{A,con}}(\gamma)} = \frac{e^{-\beta u(\ell_\ddagger)}\left[\mathfrak{q}^*_{\mathrm{FJC,con}}(\gamma)|_{\ell_b=\ell_\ddagger}\right]^{1/N_b}}{\mathfrak{q}^*_{\mathrm{A,con}}(\gamma)^{1/N_b}}. \tag{C.57}$$

The denominator is the partition function of a single uFJC link, while the numerator is the partition function of a single uFJC link in its transition state, which would then be that of a single FJC link of length $\ell_\ddagger$ (and correspondingly higher potential energy $u(\ell_\ddagger)$ due to stretching beyond the rest length ℓ_b). Expanding this result using Eq. (C.53), we obtain

$$\frac{\mathfrak{q}^*_{\mathrm{A,con}}(\gamma)}{\mathfrak{q}^*_{\ddagger,\mathrm{con}}(\gamma)} = \ell_b\sqrt{\frac{2\pi}{\kappa}}\, e^{-\beta[u(\ell_\ddagger)-u(\eta)]}\left[\frac{\mathfrak{q}^*_{\mathrm{FJC,con}}(\gamma)|_{\ell_b=\ell_\ddagger}}{\mathfrak{q}^*_{\mathrm{FJC,con}}(\gamma)}\right]^{1/N_b}, \tag{C.58}$$

which consists of three parts. The first part carries units of length with ℓ_b multiplying a unitless $\sqrt{2\pi/\kappa}$, where the units of length are a direct result of the single additional configurational degree of freedom (stretching) that a uFJC link has as opposed to the transition state link. The second part is the exponential function of the potential energy differences. The third part is the ratio of the FJC configurational partition functions of different lengths ($\ell_\ddagger$ and ℓ_b), purely entropic in nature. Multiplying Eq. (C.58) by Eq. (C.56), scaling by $\beta = 1/\mathfrak{b}T$ and simplifying, we receive the forward reaction rate coefficient function from Eq. (5.26),

$$k'(\gamma) = \frac{1}{2\pi}\sqrt{\frac{\kappa}{\nu\beta\ell_b^2}}\, e^{-\beta[u(\ell_\ddagger)-u(\eta)]}\left[\frac{\mathfrak{q}^*_{\mathrm{FJC,con}}(\gamma)|_{\ell_b=\ell_\ddagger}}{\mathfrak{q}^*_{\mathrm{FJC,con}}(\gamma)}\right]^{1/N_b}. \tag{C.59}$$

Next, we reference classical transition state theory [173] in order to make sense of Eq. (C.59). The quantity resulting from the square root function has units of frequency, so we define the attempt frequency $\omega_\ddagger$ as

$$\omega_\ddagger \equiv \sqrt{\frac{1}{\nu}\left.\frac{\partial^2 u}{\partial \ell^2}\right|_{\ell=\ell_b}} = \sqrt{\frac{\kappa}{\nu\beta\ell_b^2}}. \tag{C.60}$$

We simplify the last term in Eq. (C.59), proportional to the logarithm of the entropy barrier,

$$\left[\frac{\mathfrak{q}^*_{\mathrm{FJC,con}}(\gamma)|_{\ell_b=\ell_\ddagger}}{\mathfrak{q}^*_{\mathrm{FJC,con}}(\gamma)}\right]^{1/N_b} = \frac{\lambda_\ddagger \sinh(\lambda_\ddagger\eta)\exp[\eta\mathcal{L}(\eta)]}{\sinh(\eta)\exp[\lambda_\ddagger\eta\mathcal{L}(\lambda_\ddagger\eta)]}. \tag{C.61}$$

Combining the entropic and potential energy contributions in Eqs. (C.59) and (C.61), we obtain the Helmholtz free energy barrier $\Delta\Psi^*_\ddagger(\gamma)$ as

$$\Delta\Psi^*_\ddagger(\gamma) \equiv u(\ell_\ddagger) - u(\eta) - \mathfrak{b}T\ln\left\{\frac{\lambda_\ddagger \sinh(\lambda_\ddagger\eta)\exp[\eta\mathcal{L}(\eta)]}{\sinh(\eta)\exp[\lambda_\ddagger\eta\mathcal{L}(\lambda_\ddagger\eta)]}\right\}. \tag{C.62}$$

Writing $k'(\gamma)$ in Eq. (C.59) in terms of $\omega_\ddagger$ and $\Delta\Psi^*_\ddagger(\gamma)$, we obtain

$$k'(\gamma) = \frac{\omega_\ddagger}{2\pi}\, e^{-\beta\Delta\Psi^*_\ddagger(\gamma)}. \tag{C.63}$$

Eqs. (C.60)–(C.63) are equivalent to Eqs. (5.71)–(5.72) from the manuscript.

C.4 Solving for the distribution of intact chains

C.4.1 Obtaining and verifying the exact solution

Here we exactly solve Eq. (5.74), the evolution equation for the probability density distribution of intact chains $P_\mathrm{A}(\boldsymbol{\xi};t)$. Our solution is valid for all chain models with a single reaction coordinate or any number of identical reaction coordinates. Eq. (5.74) in terms of $\boldsymbol{\xi}$ is

$$\frac{\partial P_\mathrm{A}(\boldsymbol{\xi};t)}{\partial t} = -\left[\frac{\partial P_\mathrm{A}(\boldsymbol{\xi};t)}{\partial \boldsymbol{\xi}}\,\boldsymbol{\xi}\right] : \mathbf{L}(t) - k(\boldsymbol{\xi})\left\{P_\mathrm{A}(\boldsymbol{\xi};t) - \frac{P_\mathrm{A}^\mathrm{eq}(\boldsymbol{\xi})}{P_\mathrm{B}^\mathrm{tot,eq}}\left[1 - \iiint P_\mathrm{A}(\tilde{\boldsymbol{\xi}};t)\, d^3\tilde{\boldsymbol{\xi}}\right]\right\}. \tag{C.64}$$

We begin solving Eq. (C.64) by defining the new variable,

$$H(\boldsymbol{\xi};t) \equiv P_\mathrm{A}\left[\mathbf{F}(t)\cdot\boldsymbol{\xi};t\right] e^{\int_0^t k[\mathbf{F}(s)\cdot\boldsymbol{\xi}]\,ds}, \tag{C.65}$$

where $H(\boldsymbol{\xi};0) = P_\mathrm{A}(\boldsymbol{\xi};0)$ if $\mathbf{F}(0) = \mathbf{1}$. Substitution of Eq. (C.65) into Eq. (C.64) produces

$$\frac{\partial H(\boldsymbol{\xi};t)}{\partial t}\, e^{-\int_0^t k[\mathbf{F}(s)\cdot\boldsymbol{\xi}]\,ds} = \frac{k\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}\right] P_\mathrm{A}^\mathrm{eq}\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}\right]}{P_\mathrm{B}^\mathrm{tot,eq}}\left\{1 - \iiint H(\tilde{\boldsymbol{\xi}};t)\, e^{-\int_0^t k\left[\mathbf{F}(s)\cdot\tilde{\boldsymbol{\xi}}\right]ds}\, d^3\tilde{\boldsymbol{\xi}}\right\}. \tag{C.66}$$

Eq. (C.66) may be rearranged to cause the right-hand side to become a function of time only,

$$\frac{\partial H(\boldsymbol{\xi};t)}{\partial t}\,\frac{e^{-\int_0^t k[\mathbf{F}(s)\cdot\boldsymbol{\xi}]\,ds}}{k\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}\right] P_\mathrm{A}^\mathrm{eq}\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}\right]} = \frac{1}{P_\mathrm{B}^\mathrm{tot,eq}}\left\{1 - \iiint H(\tilde{\boldsymbol{\xi}};t)\, e^{-\int_0^t k\left[\mathbf{F}(s)\cdot\tilde{\boldsymbol{\xi}}\right]ds}\, d^3\tilde{\boldsymbol{\xi}}\right\}, \tag{C.67}$$

$$\equiv \rho(t), \tag{C.68}$$

which we have now defined as $\rho(t)$. We see from the right-hand side of Eq. (C.67) that this function happens to be $\rho(t) = P_\mathrm{B}^\mathrm{tot}(t)/P_\mathrm{B}^\mathrm{tot,eq}$. Rearranging and integrating Eq. (C.67), we then retrieve

$$H(\boldsymbol{\xi};t) = H(\boldsymbol{\xi};0) + \int_0^t \frac{k\left[\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right] P_\mathrm{A}^\mathrm{eq}\left[\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]}{e^{-\int_0^\tau k[\mathbf{F}(s)\cdot\boldsymbol{\xi}]\,ds}}\,\rho(\tau)\,d\tau, \tag{C.69}$$

where we now must determine a solution for $\rho(t)$ that is consistent with the solution in Eq. (C.69) by substituting it back into Eq. (C.67). Our results are simplified by introducing the reaction propagator $\Xi(\boldsymbol{\xi};t,\tau)$, which we define as

$$\Xi(\boldsymbol{\xi};t,\tau) \equiv \exp\left\{-\int_\tau^t k\left[{}_{(t)}\mathbf{F}(s)\cdot\boldsymbol{\xi}\right]\, ds\right\}, \tag{C.70}$$

where the relative deformation [135] is defined as

$$ {}_{(t)}\mathbf{F}(\tau) \equiv \mathbf{F}(\tau)\cdot\mathbf{F}^{-1}(t). \tag{C.71}$$

Next we define the kernel function $K(t,\tau)$, which we may rewrite in terms of the reaction propagator $\Xi(\boldsymbol{\xi};t,\tau)$ using the invariance (due to incompressibility) of $d^3\boldsymbol{\xi}$ to the transformation $\boldsymbol{\xi}\mapsto\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}$,

$$ K(t,\tau) \equiv \iiint \frac{k\left[\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right] P_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]}{P_{\mathrm{B}}^{\mathrm{tot,eq}} e^{\int_\tau^t k[\mathbf{F}(s)\cdot\boldsymbol{\xi}]\,ds}}\, d^3\boldsymbol{\xi}, \tag{C.72}$$

$$ = \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}} \iiint P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right] \frac{\partial\Xi(\boldsymbol{\xi};t,\tau)}{\partial\tau}\, d^3\boldsymbol{\xi}, \tag{C.73}$$

where $K(t,\tau>t)\equiv 0$. Then we similarly define the right-hand side function

$$ b(t) \equiv \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}}\left\{1 - \iiint H(\boldsymbol{\xi};0) e^{-\int_0^t k\left[\mathbf{F}(s)\cdot\tilde{\boldsymbol{\xi}}\right]ds}\, d^3\boldsymbol{\xi}\right\}, \tag{C.74}$$

$$ = \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}}\left\{1 - \iiint P_{\mathrm{A}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi};0\right]\, \Xi(\boldsymbol{\xi};t,0)\, d^3\boldsymbol{\xi}\right\}. \tag{C.75}$$

Now, substituting Eq. (C.69) into Eq. (C.67) and simplifying, we obtain the consistency condition

$$ \rho(t) + \int_0^t K(t,\tau)\rho(\tau)\, d\tau = b(t). \tag{C.76}$$

Eq. (C.76) is a linear Volterra integral equation of the second kind with eigenvalue -1.

We first consider the special case of motions with constant stretch history [134, 135] that have $\mathbf{L}$ = constant, which with $\mathbf{F}(0)=\mathbf{1}$ allow $\mathbf{F}(t)=e^{t\mathbf{L}}$ and therefore ${}_{(t_2)}\mathbf{F}(t_1)=\mathbf{F}(t_1-t_2)=\mathbf{F}^{-1}(t_2-t_1)$. This allows us to rewrite the kernel $K(t,\tau)$ in Eq. (C.73) to be of the convolution type [206] by depending only on the difference $t-\tau$,

$$ K(t-\tau) = \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}} \iiint P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi}) k'(\boldsymbol{\xi}) e^{-\int_0^{t-\tau} k[\mathbf{F}(s)\cdot\boldsymbol{\xi}]\,ds}\, d^3\boldsymbol{\xi}. \tag{C.77}$$

Kernels of the convolution type allow Eq. (C.76) to be solved using the Laplace transform $\mathfrak{L}$. After utilizing the convolution theorem twice [207], the solution may be written as

$$ \rho(t) = \int_0^t W(t-\tau) b(\tau)\, d\tau, \tag{C.78}$$

where the solution kernel (also of the convolution type) is given by

$$ W(t) = \mathfrak{L}^{-1}\left\{\frac{1}{1+\mathfrak{L}[K(t)]}\right\}. \tag{C.79}$$

The solution as $t\to\infty$ may be obtained without need for the inverse Laplace transform $\mathfrak{L}^{-1}$ and can be utilized to study the steady-state mechanical response under these special deformations [59].

For arbitrary incompressible deformation histories, the solution to Eq. (C.76), obtained using Picard's method of successive approximations [206, 208], is the Liouville-Neumann series

$$\rho(t) = b(t) + \sum_{m=1}^{\infty} (-1)^m \int_0^t K_m(t,\tau) b(\tau)\, d\tau. \tag{C.80}$$

The functions $K_m(t,\tau)$, where we begin with $K_1(t,\tau) \equiv K(t,\tau)$, are defined as

$$K_m(t,\tau) \equiv \int_\tau^t K(t,s) K_{m-1}(s,\tau)\, ds. \tag{C.81}$$

This series solution for $\rho(t)$ allows the solution for $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ and subsequently $\boldsymbol{\sigma}(t)$ in Eq. (5.59) to be written as series. This resulting series for $\boldsymbol{\sigma}(t)$ resembles the general viscoelastic constitutive equation for the stress of the integral type [134, 135]. However, in our case it is much more practical to construct $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ and integrate the single term for $\boldsymbol{\sigma}(t)$ afterward.

The Liouville-Neumann series in Eq. (C.80) converges for some total time interval $t \in [0, \mathcal{T}]$ if the kernel function $K(t,\tau)$ is square-integrable [208], which requires

$$\|K\|^2 \equiv \int_0^{\mathcal{T}} \int_0^t \left|K(t,\tau)\right|^2 d\tau\, dt < \infty. \tag{C.82}$$

Since $0 \leq \Xi(\boldsymbol{\xi};t,\tau) \leq 1$ in Eq. (C.73), we have $K \leq \hat{K}$ and thus $\|K\|^2 \leq \|\hat{K}\|^2$, where

$$\hat{K}(t,\tau) \equiv \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}} \iiint k\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right] P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]\, d^3\boldsymbol{\xi}, \tag{C.83}$$

so we may prove Eq. (C.82) by proving $\|\hat{K}\|^2 < \infty$. We transform $\boldsymbol{\xi} \mapsto {}_{(\tau)}\mathbf{F}(t)\cdot\boldsymbol{\xi}$ in order to see that $\hat{K}$ is not truly a function of t and τ and is more simply

$$\hat{K} = \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}} \iiint k(\boldsymbol{\xi}) P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})\, d^3\boldsymbol{\xi}, \tag{C.84}$$

which means $\|\hat{K}\|^2 < \infty$ is proven for a finite time interval, where $0 < P_{\mathrm{B}}^{\mathrm{tot,eq}} < 1$ and $k(\boldsymbol{\xi}) \propto k'(\boldsymbol{\xi})$, if the following is true:

$$\iiint k'(\boldsymbol{\xi}) P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})\, d^3\boldsymbol{\xi} < \infty. \tag{C.85}$$

If the positive semidefinite functions $k'(\boldsymbol{\xi})$ and $P^{\mathrm{eq}}(\boldsymbol{\xi})$ are prescribed without considering a chain model, this relation provides a constraint. On the other hand, if we utilize our statistical mechanical framework – namely Eqs. (5.26), (5.19), and (5.23) – we see Eq. (C.84) is a requirement that $\mathfrak{b}T\mathfrak{q}_\ddagger/\mathfrak{q}_{\mathrm{B}} < \infty$. This requirement is generally true for finite, nonzero temperatures because of the following. First, the momentum portions of the partition functions are known exactly and are finite. Second, the configuration portions in general have an upper bound that is powers of the volume (zero potential case), which is then finite if the volume is finite. Since the partition functions are positive-definite for nonzero temperatures and an appropriate Hamiltonian, the proof is complete. For the specific case of the Morse-FJC we have approximately considered in Sec. 5.3.1, we can immediately see that it is explicitly true: the integral of $\mathfrak{q}_\ddagger^*(\boldsymbol{\xi})$ over all end-to-end lengths exists and is finite and $\mathfrak{q}_{\mathrm{B}} = \mathfrak{q} - \mathfrak{q}_{\mathrm{A}}$ is a nonzero finite positive number.

While we have obtained the solution for $\rho(t)$ in Eq. (C.80) and have proven its convergence, we must now consider error estimates since we cannot sum to infinity in practice, and since it is unlikely that we will recognize what the series converges to for a given single-chain model. Including M terms in Eq. (C.80),

$$\rho(t) = b(t) + \sum_{m=1}^{M} (-1)^m \int_0^t K_m(t,\tau) b(\tau)\, d\tau + R_{M+1}(t), \tag{C.86}$$

where $R_{M+1}(t)$ is then the residual function. It can be shown [209] that the absolute value of the residual function in our case is bound by the inequality

$$|R_{M+1}(t)| \leq \frac{\left[\max|K(t,\tau)|\mathcal{T}\right]^{M+1} \max|\rho(t)|}{(M+1)!}. \tag{C.87}$$

The limit of $|R_{M+1}(t)|$ as $M \to \infty$ is then zero, further verifying the convergence of our series solution in Eq. (C.80). The maximum of $K(t,\tau) > 0$ over all t and $\tau \leq t$ is $\hat{K}$ in Eq. (C.84), and the maximum of $\rho(t) > 0$ is $1/P_{\mathrm{B}}^{\mathrm{tot,eq}}$ since the maximum of $P_{\mathrm{B}}^{\mathrm{tot}}(t)$ is unity. Therefore, we may bound the residual as

$$|R_{M+1}| \leq \frac{(\hat{K}\mathcal{T})^{M+1}}{P_{\mathrm{B}}^{\mathrm{tot,eq}}(M+1)!}. \tag{C.88}$$

The right-hand side of Eq. (C.88) is then our t-independent estimate for the residual which is computed once after the full time interval of interest $\mathcal{T}$ is specified. This estimate can be scaled by $\rho(t)$ for an estimate of the relative error at time t. We see two effective timescales – the total time interval $\mathcal{T}$ and the timescale $1/\hat{K}$, where $\hat{K}$ is actually the total reverse reaction rate coefficient. For $\mathcal{T} < 1/\hat{K}$ (short total time), the residual estimate in Eq. (C.88) rapidly becomes small as M increases, while for $\mathcal{T} > 1/\hat{K}$ (long total time) the residual estimate may require considerably large M to become small.

Now that our solution for $\rho(t)$ has been proven and error estimates in summing for it have been considered, we can go back to Eq. (C.69) and transform back using Eq. (C.65) to finally write the solution to Eq. (C.64),

$$P_{\mathrm{A}}(\boldsymbol{\xi};t) = P_{\mathrm{A}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi};0\right]\Xi(\boldsymbol{\xi};t,0) + \int_0^t P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]\frac{\partial\Xi(\boldsymbol{\xi};t,\tau)}{\partial\tau}\,\rho(\tau)\,d\tau. \tag{C.89}$$

Substitution of this solution back into Eq. (C.64) for verification shows that the first term in Eq. (C.89) is the homogeneous solution and the second term is the particular solution. For further verification, substitution of the solution Eq. (C.89) into the conservation requirement Eq. (5.73) retrieves again the integral equation for $\rho(t) = P_{\mathrm{B}}^{\mathrm{tot}}(t)/P_{\mathrm{B}}^{\mathrm{tot,eq}}$ from Eq. (C.76). As a final check, substitution of the solution Eq. (C.89) into the evolution equation for broken chains Eq. (5.43) retrieves (after simplifying and integrating the resulting ordinary integro-differential equation) the integral equation from Eq. (C.76) yet again. The homogeneous solution accounts for the decay of the initial distribution of chains as it is deformed and chains break. The particular solution accounts for broken chains reforming in time throughout the deformation history, which is why its integrand (a rate of reforming) is proportional to the total probability of broken chains through

$\rho(t) = P_{\mathrm{B}}^{\mathrm{tot}}(t)/P_{\mathrm{B}}^{\mathrm{tot,eq}}$. If the reaction propagator $\Xi(\boldsymbol{\xi};t,\tau)$ is treated as independent of end-to-end vector $\Xi(t,\tau)$ and instead constitutively prescribed, the integral of Eq. (C.89) over all end-to-end vectors for $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ resembles many recent models for the total number of intact bonds in a dynamic network [163, 166–168, 170, 172].

If the system is initially equilibrated before $t = 0$, which is $P_{\mathrm{A}}(\boldsymbol{\xi};t \leq 0) = P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})$ and $\mathbf{F}(t \leq 0) = \mathbf{1}$, we may use integration by parts and $\rho(t \leq 0) = 1$ to rewrite Eq. (C.89) as a single term,

$$P_{\mathrm{A}}(\boldsymbol{\xi};t) = \int_{-\infty}^{t} P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right] \frac{\partial\Xi(\boldsymbol{\xi};t,\tau)}{\partial\tau}\,\rho(\tau)\,d\tau, \tag{C.90}$$

which is equivalent to the solution from Eq. (5.75) of the manuscript.

C.4.2 Objectivity of the stress

Here we show that the relation for the Cauchy stress $\boldsymbol{\sigma}(t)$ in Eq. (5.59) obtained when substituting in the solution for $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ from Eq. (C.89) is objective. Under rigid rotations of the frame of reference [135], the relation for $\boldsymbol{\sigma}(t)$ (if objective) in the new frame is given by

$$\boldsymbol{\sigma}'(t) = \mathbf{Q}(t)\cdot\boldsymbol{\sigma}(t)\cdot\mathbf{Q}^{T}(t), \tag{C.91}$$

where $\mathbf{Q}(t)$ is a time-dependent orthogonal rotation tensor, i.e. $\mathbf{Q}^{-1}(t) = \mathbf{Q}^{T}(t)$ and $\det(\mathbf{Q}) = \pm 1$. The deformation gradient $\mathbf{F}(t)$ is not an objective tensor, and is instead a double vector:

$$\mathbf{F}'(t) = \mathbf{Q}(t)\cdot\mathbf{F}(t)\cdot\mathbf{Q}^{T}(0). \tag{C.92}$$

The end-to-end vector $\boldsymbol{\xi}$ is a time-independent reference coordinate with $\boldsymbol{\xi}' = \mathbf{Q}(0)\cdot\boldsymbol{\xi}$. While $\boldsymbol{\xi}$ is not objective, the deformed coordinate $\mathbf{F}(t)\cdot\boldsymbol{\xi}$ is objective: in the new frame it is given by

$$\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}\right]' = \mathbf{Q}(t)\cdot\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}\right]. \tag{C.93}$$

We can utilize this objective vector within the stress in Eq. (5.59) by taking $\boldsymbol{\xi} \to \mathbf{F}(t)\cdot\boldsymbol{\xi}$ in the integrand, where the element of integration is invariant to this transformation due to the incompressible deformation, and where $f(\boldsymbol{\xi}) \equiv \partial\psi^{*}(\boldsymbol{\xi})/\partial\xi$ is the force. The result is

$$\boldsymbol{\sigma}(t) = n\iiint P_{\mathrm{A}}\left[\mathbf{F}(t)\cdot\boldsymbol{\xi};t\right]\, f\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}\right]\left[\frac{\mathbf{F}(t)\cdot\boldsymbol{\xi}\boldsymbol{\xi}^{T}\cdot\mathbf{F}^{T}(t)}{\left\|\mathbf{F}(t)\cdot\boldsymbol{\xi}\right\|_2}\right] d^3\boldsymbol{\xi} - p(t)\mathbf{1}. \tag{C.94}$$

Note that we have assume that the single-chain model is spherically-symmetric in $\boldsymbol{\xi}$. Since $\mathbf{F}(t)\cdot\boldsymbol{\xi}$ is an objective vector, its norm $\left\|\mathbf{F}(t)\cdot\boldsymbol{\xi}\right\|_2$ is an objective scalar, which means that it is entirely invariant to rigid rotations of the frame of reference. This is made clear as follows: the norm is

$$\left\|\mathbf{F}(t)\cdot\boldsymbol{\xi}\right\|_2 = \sqrt{\boldsymbol{\xi}^{T}\cdot\mathbf{C}(t)\cdot\boldsymbol{\xi}}, \tag{C.95}$$

where $\mathbf{C}(t) = \mathbf{F}^{T}(t)\cdot\mathbf{F}(t)$ is the right Cauchy-Green strain tensor. Using Eq. (C.92), $\mathbf{C}(t)$ in new frame is given by $\mathbf{Q}(0)\cdot\mathbf{C}(t)\cdot\mathbf{Q}^{T}(0)$, and we then have the invariant quantity

$$\left[\boldsymbol{\xi}^{T}\cdot\mathbf{C}(t)\cdot\boldsymbol{\xi}\right]' = \boldsymbol{\xi}^{T}\cdot\mathbf{Q}^{T}(0)\cdot\mathbf{Q}(0)\cdot\mathbf{C}(t)\cdot\mathbf{Q}^{T}(0)\cdot\mathbf{Q}(0)\cdot\boldsymbol{\xi} = \boldsymbol{\xi}^{T}\cdot\mathbf{C}(t)\cdot\boldsymbol{\xi}, \tag{C.96}$$

which means that $\left\|\mathbf{F}(t)\cdot\boldsymbol{\xi}\right\|_2$ is invariant. Consequently, all spherically-symmetric scalar functions with argument $\mathbf{F}(t)\cdot\boldsymbol{\xi}$ are also invariant; this is also true for the cases where the deformed coordinate

is parameterized by τ or s. Thus, in Eq. (C.94), not only is $\|\mathbf{F}(t)\cdot\boldsymbol{\xi}\|_2$ invariant but $f\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}\right]$ is also invariant. We will next show that $P_{\mathrm{A}}\left[\mathbf{F}(t)\cdot\boldsymbol{\xi};t\right]$ is invariant using the invariance of spherically-symmetric scalar functions with argument $\mathbf{F}(t)\cdot\boldsymbol{\xi}$. The kernel function $K(t,\tau)$ in Eq. (C.73) when using $\boldsymbol{\xi}\to\mathbf{F}(t)\cdot\boldsymbol{\xi}$ is rewritten as

$$K(t,\tau)=\frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}}\iiint P_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]k\left[\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]e^{-\int_\tau^t k[\mathbf{F}(s)\cdot\boldsymbol{\xi}]\,ds}\,d^3\boldsymbol{\xi}. \tag{C.97}$$

Since $P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})$ and $k(\boldsymbol{\xi})$ are both spherically-symmetric, each of their instances in this relation for the kernel are invariant. Similarly, for the right-hand side function $b(t)$ in Eq. (C.75) we can write

$$b(t)=\frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}}\left\{1-\iiint P_{\mathrm{A}}(\boldsymbol{\xi};0)\,e^{-\int_0^t k[\mathbf{F}(s)\cdot\boldsymbol{\xi}]}\,d^3\boldsymbol{\xi}\right\}, \tag{C.98}$$

which is then invariant as long as the initial distribution $P_{\mathrm{A}}(\boldsymbol{\xi};0)$ is spherically-symmetric, which is true if it is the equilibrium distribution $P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})$. Since the solution for $\rho(t)$ in Eq. (C.80) is entirely related to $K(t,\tau)$ and $b(t)$, it is also invariant. The solution for $P_{\mathrm{A}}\left[\mathbf{F}(t)\cdot\boldsymbol{\xi};t\right]$ using Eq. (C.89) is

$$P_{\mathrm{A}}\left[\mathbf{F}(t)\cdot\boldsymbol{\xi};t\right]=P_{\mathrm{A}}(\boldsymbol{\xi};0)e^{-\int_0^t k[\mathbf{F}(s)\cdot\boldsymbol{\xi}]}+\int_0^t P_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]k\left[\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]e^{-\int_\tau^t k[\mathbf{F}(s)\cdot\boldsymbol{\xi}]\,ds}\,\rho(\tau)\,d\tau, \tag{C.99}$$

and since each function here on the right-hand side is invariant, $P_{\mathrm{A}}\left[\mathbf{F}(t)\cdot\boldsymbol{\xi};t\right]$ is invariant. We now refer back to Eq. (C.94) and apply the rigid rotation via $\mathbf{Q}(t)$, where we have shown each scalar function within the integrand is invariant. The result is

$$\begin{aligned}\boldsymbol{\sigma}'(t)=&n\iiint P_{\mathrm{A}}\left[\mathbf{F}(t)\cdot\boldsymbol{\xi};t\right]\,f\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}\right]\left[\frac{\mathbf{Q}(t)\cdot\mathbf{F}(t)\cdot\boldsymbol{\xi}\boldsymbol{\xi}^T\cdot\mathbf{F}^T(t)\cdot\mathbf{Q}^T(t)}{\|\mathbf{F}(t)\cdot\boldsymbol{\xi}\|_2}\right]d^3\boldsymbol{\xi}-p(t)\mathbf{1}, &\text{(C.100)}\\ =&\mathbf{Q}(t)\cdot\left\{n\iiint P_{\mathrm{A}}\left[\mathbf{F}(t)\cdot\boldsymbol{\xi};t\right]\,f\left[\mathbf{F}(t)\cdot\boldsymbol{\xi}\right]\left[\frac{\mathbf{F}(t)\cdot\boldsymbol{\xi}\boldsymbol{\xi}^T\cdot\mathbf{F}^T(t)}{\|\mathbf{F}(t)\cdot\boldsymbol{\xi}\|_2}\right]d^3\boldsymbol{\xi}-p(t)\mathbf{1}\right\}\cdot\mathbf{Q}^T(t), &\text{(C.101)}\\ =&\mathbf{Q}(t)\cdot\boldsymbol{\sigma}(t)\cdot\mathbf{Q}^T(t), &\text{(C.102)}\end{aligned}$$

which shows that $\boldsymbol{\sigma}(t)$ is indeed an objective tensor.

C.4.3 Computational considerations

We recognize that as the current time t grows, it becomes computationally prohibitive to store certain quantities (such as the reaction propagator) over the entire history. Fortunately, we are able to rewrite our solution at t in terms of the solution at any previous time, which allows us to periodically reset and resolve in order to satisfy memory requirements. For an intermediate time t_i obeying $0\le\tau\le t_i\le t$, the reaction propagator has the property

$$\Xi(\boldsymbol{\xi};t,\tau)=\Xi(\boldsymbol{\xi};t,t_i)\,\Xi\left[{}_{(t)}\mathbf{F}(t_i)\cdot\boldsymbol{\xi};t_i,\tau\right]. \tag{C.103}$$

This property allows the solution in Eq. (C.89) at time t to be written in terms of the solution at an intermediate time $0\le t_i\le t$ and the history from t_i to t,

$$P_{\mathrm{A}}(\boldsymbol{\xi};t) = P_{\mathrm{A}}\left[{}_{(t)}\mathbf{F}(t_i)\cdot\boldsymbol{\xi};t_i\right]\Xi(\boldsymbol{\xi};t,t_i) + \int_{t_i}^{t} P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]\frac{\partial\Xi(\boldsymbol{\xi};t,\tau)}{\partial\tau}\,\rho(\tau)\,d\tau. \tag{C.104}$$

When applying Eq. (C.103) to the solution for $\rho(t)$ in Eq. (C.80), two adjustments must be made. First, the bounds of integration in Eq. (C.80) must be $\tau\in[t_i,t]$. Second, $b(t)$ in Eq. (C.75) must be rewritten as

$$b(t) = \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}}\left\{1 - \iiint P_{\mathrm{A}}\left[{}_{(t)}\mathbf{F}(t_i)\cdot\boldsymbol{\xi};t_i\right]\,\Xi(\boldsymbol{\xi};t,t_i)\,d^3\boldsymbol{\xi}\right\}. \tag{C.105}$$

Next, we note that it is most computationally expedient to compute $\rho(t)$ in Eq. (C.80) by performing successive approximations [209]. This is as opposed to computing, storing, integrating, and summing each of the M functions $K_m(t,\tau)b(\tau)$, which is more computationally expensive. After starting with $\rho_0(t)\equiv b(t)$ and computing $K(t,\tau)$, storing both $b(t)$ and $K(t,\tau)$, we successively approximate the solution $\rho(t)$ by iterating

$$\rho_M(t) = b(t) - \int_0^t K(t,\tau)\rho_{M-1}(\tau)\,d\tau. \tag{C.106}$$

To computationally obtain the solution as $\rho(t)=\lim_{M\to\infty}\rho_M(t)$, we take finite M such that the residual in Eq. (C.88) falls below some specified tolerance. Finally, we note that exploiting the symmetries of $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ that are preserved over the deformation history additionally serves to alleviate computation expense when computing the integrals over $\boldsymbol{\xi}$. The uFJC model we utilize here is spherically-symmetric (only depend on $\|\boldsymbol{\xi}\|_2=\xi$), allowing many integrals to be reduced to one-dimensional integrals over the scalar end-to-end length ξ. For example, Eq. (C.85) for spherically-symmetric single-chain models is reducible to

$$\iiint k'(\boldsymbol{\xi})P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})\,d^3\boldsymbol{\xi} = 4\pi\int_0^\infty k'(\xi)P_{\mathrm{A}}^{\mathrm{eq}}(\xi)\,\xi^2\,d\xi. \tag{C.107}$$

While the time-independent single-chain functions will always retain their symmetry, the distribution $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ will in general not. Fortunately, many deformation histories of interest will preserve a portion of the original spherical symmetry: when applying uniaxial tension (not necessarily monotonic), where the relative deformation is given by

$${}_{(t)}\mathbf{F}(\tau) = \begin{pmatrix} \frac{F_{11}(\tau)}{F_{11}(t)} & 0 & 0 \\ 0 & \left[\frac{F_{11}(t)}{F_{11}(\tau)}\right]^{1/2} & 0 \\ 0 & 0 & \left[\frac{F_{11}(t)}{F_{11}(\tau)}\right]^{1/2} \end{pmatrix}, \tag{C.108}$$

the angular symmetry about the ξ_1-axis is preserved. This is made clear after writing

$$\left\|{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right\|_2 = \left\{\left[\frac{F_{11}(\tau)}{F_{11}(t)}\right]^2 \xi_1^2 + \frac{F_{11}(t)}{F_{11}(\tau)}\left(\xi_2^2+\xi_3^2\right)\right\}^{1/2}, \tag{C.109}$$

$$= \left\{\left[\frac{F_{11}(\tau)}{F_{11}(t)}\right]^2 z^2 + \frac{F_{11}(t)}{F_{11}(\tau)}\, r^2\right\}^{1/2}, \tag{C.110}$$

where $z \equiv \xi_1$ and $r \equiv (\xi_2^2+\xi_3^2)^{1/2}$ are the height and radius in cylindrical coordinates. Physically, the deformation stretches a sphere into a spheroid, which is symmetric about the z-axis along which the deformation is directed. We then write each single chain function in terms of z and r rather than simply ξ, such as $P_{\mathrm{A}}^{\mathrm{eq}}(z,r) \equiv P_{\mathrm{A}}^{\mathrm{eq}}[(z^2+r^2)^{1/2}]$. After doing so we simplify each of the three-dimensional integrals over $\boldsymbol{\xi}$ into two-dimensional integrals over z and r; note that we also exploit the symmetry about the z-plane. For example: for an initially-equilibrated system, the right-hand side function $b(t)$ from Eq. (C.75) is

$$b(t) \equiv \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}}\left\{1 - \iiint P_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}\right]\,\Xi(\boldsymbol{\xi};t,0)\,d^3\boldsymbol{\xi}\right\}, \tag{C.111}$$

$$= \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}}\left\{1 - 4\pi\int_0^\infty r\,dr\int_0^\infty dz\, P_{\mathrm{A}}^{\mathrm{eq}}\left[F_{11}^{-1}(t)z,\, F_{11}^{1/2}(t)r\right]\, e^{-\int_0^t ds\, k\left[\frac{F_{11}(s)}{F_{11}(t)}z,\left[\frac{F_{11}(t)}{F_{11}(s)}\right]^{1/2}r\right]}\right\}. \tag{C.112}$$

The kernel function $K(t,\tau)$ simplifies similarly, allowing $\rho(t)$ and afterward $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ to be evaluated with less computational expense. Further, $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ too retains a symmetry about ξ_1, which allows the stress in Eq. (5.59) to also be written in terms of a two-dimensional integral over z and r:

$$\sigma_{11}(t) = 2\pi n\int_0^\infty r\,dr\int_0^\infty dz\, P_{\mathrm{A}}(z,r;t)\, f(z,r)\,\frac{2z^2-r^2}{(z^2+r^2)^{1/2}}. \tag{C.113}$$

Note that we have also applied the traction-free boundary conditions and correspondingly solved for the pressure $p(t)$ in order to obtain Eq. (C.113).

C.4.4 Special cases

As we discussed in Sec. 5.4.1 and observed in Figs. 5.4–5.5, the reaction rate coefficient function $k'(\xi)$ often behaves as being constant at its initial value $k'(0)$ before suddenly becoming infinite beyond some critical extension ξ_c. For the net reaction rate coefficient function $k(\xi) = N_b k'(\xi)$, this is more specifically

$$k(\xi) \sim \begin{cases} k_0, & \xi \le \xi_c, \\ \infty, & \xi > \xi_c. \end{cases} \tag{C.114}$$

Applying this approximation to the reaction propagator $\Xi(\boldsymbol{\xi};t,\tau)$ in Eq. (C.70), we obtain

$$\Xi(\boldsymbol{\xi};t,\tau) \sim \Theta(\boldsymbol{\xi};t,\tau)e^{-k_0(t-\tau)} \quad \text{and} \quad \frac{\partial \Xi(\boldsymbol{\xi};t,\tau)}{\partial \tau} \sim \Theta(\boldsymbol{\xi};t,\tau)k_0 e^{-k_0(t-\tau)}, \tag{C.115}$$

where the yield function $\Theta(\boldsymbol{\xi};t,\tau)$ is defined as

$$\Theta(\boldsymbol{\xi};t,\tau) \equiv \begin{cases} 1, & \left\| {}_{(t)}\mathbf{F}(s)\cdot\boldsymbol{\xi} \right\|_2 \leq \xi_c \quad \forall s \in [\tau,t], \\ 0, & \text{otherwise.} \end{cases} \tag{C.116}$$

The yield function accounts for chains that have been broken via extension past ξ_c by assigning zero to any vector $\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}$ being sampled in $P_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}\right]$ that was outside or deformed outside the yield surface at $\|\boldsymbol{\xi}\|_2 = \xi_c$ during the deformation history. With Eq. (C.115), our solution for the distribution of intact chains $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ from Eq. (C.90) becomes

$$P_{\mathrm{A}}(\boldsymbol{\xi};t) = \int_{-\infty}^{t} P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right] \Theta(\boldsymbol{\xi};t,\tau)k_0 e^{-k_0(t-\tau)}\,\rho(\tau)\,d\tau. \tag{C.117}$$

This special case is transient chain breaking combined with a finite critical extension.

Rate-independent irreversible breaking

Applying Eq. (C.115) to our solution for the distribution of intact chains $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ from Eq. (C.89),

$$\begin{aligned} P_{\mathrm{A}}(\boldsymbol{\xi};t) =& P_{\mathrm{A}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi};0\right]\Theta(\boldsymbol{\xi};t,0)e^{-k_0(t-\tau)} \\ &+ \int_0^t P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]\Theta(\boldsymbol{\xi};t,\tau)k_0 e^{-k_0(t-\tau)}\,\rho(\tau)\,d\tau. \end{aligned} \tag{C.118}$$

We assume that the distribution is initially equilibrated, and then Eq. (C.118) is equivalent to Eq. (C.117). Now, we consider the special case where $k_0 \approx 0$, which simplifies Eq. (C.118) to

$$P_{\mathrm{A}}(\boldsymbol{\xi};t) = P_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}\right]\Theta(\boldsymbol{\xi};t,0). \tag{C.119}$$

Eq. (C.119) is equivalent to Eq. (5.81) from the manuscript and is the special case of the rate-independent irreversible breaking of chains.

Transient breaking

Next, we consider the special case where $\xi_c \to \infty$ but $k_0 > 0$, where Eq. (C.117) becomes

$$P_{\mathrm{A}}(\boldsymbol{\xi};t) = \int_{-\infty}^{t} P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right] k_0 e^{-k_0(t-\tau)}\,d\tau. \tag{C.120}$$

Eq. (C.120) is equivalent to Eq. (5.84) from the manuscript and is the special case where chains constantly break and reform, i.e. the transient network model. Note that an infinitely-extensible single-chain model (such as the ideal or EFJC models) must be utilized since $\xi_c \to \infty$. Also note that we have taken $\rho(t) = 1$ within the solution in Eq. (C.120), which means that the total number of intact chains $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ remains constant at $P_{\mathrm{A}}^{\mathrm{tot,eq}}$ for all time. This can be verified by integrating Eq. (5.74) over $\boldsymbol{\xi}$ with $k(\boldsymbol{\xi}) = k_0$ for an initially-equilibrated system. Integrating Eq. (C.120) over

$\boldsymbol{\xi}$ additionally will show that $P_{\mathrm{A}}^{\mathrm{tot}}(t) = P_{\mathrm{A}}^{\mathrm{tot,eq}}$ for all time as long as $P_{\mathrm{A}}^{\mathrm{tot}}(0) = P_{\mathrm{A}}^{\mathrm{tot,eq}}$. Further, we point out that $k(\boldsymbol{\xi}) = k_0$ is likely the only way to guarantee that the fraction of intact chains remains constant at its equilibrium value. For this to be true, we must have $\rho(t) = 1$ as well as $P_{\mathrm{A}}^{\mathrm{tot}}(t) = P_{\mathrm{A}}^{\mathrm{tot,eq}}$. Considering Eq. (C.76) with $\rho(t) = 1$, we reference Eq. (C.73) for $K(t,\tau)$ and Eq. (C.75) for $b(t)$ in simplifying the following:

$$\int_0^t K(t,\tau)\,d\tau = \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}} \int_0^t \iiint P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right] \frac{\partial \Xi(\boldsymbol{\xi};t,\tau)}{\partial \tau}\,d^3\boldsymbol{\xi}\,d\tau, \tag{C.121}$$

$$= \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}} \int_0^t \iiint \left(\frac{\partial}{\partial\tau}\left\{P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]\Xi(\boldsymbol{\xi};t,\tau)\right\} - \Xi(\boldsymbol{\xi};t,\tau)\frac{\partial}{\partial\tau}P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]\right) d^3\boldsymbol{\xi}\,d\tau, \tag{C.122}$$

$$= \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}} \iiint \left(P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi}) - P_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}\right]\Xi(\boldsymbol{\xi};t,0) - \int_0^t \Xi(\boldsymbol{\xi};t,\tau)\frac{\partial}{\partial\tau}P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]\,d\tau\right) d^3\boldsymbol{\xi}, \tag{C.123}$$

$$= b(t) - 1 - \frac{1}{P_{\mathrm{B}}^{\mathrm{tot,eq}}} \int_0^t \iiint \Xi(\boldsymbol{\xi};t,\tau)\left[\left.\frac{\partial P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})}{\partial\boldsymbol{\xi}}\right|_{\boldsymbol{\xi}={}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}} \cdot \mathbf{L}(\tau)\cdot{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right] d^3\boldsymbol{\xi}\,d\tau. \tag{C.124}$$

We can alternatively arrive at this result by integrating Eq. (C.89) over all $\boldsymbol{\xi}$, setting both $\rho(t) = 1$ and $P_{\mathrm{A}}^{\mathrm{tot}}(t) = P_{\mathrm{A}}^{\mathrm{tot,eq}}$, and similarly simplifying. Substituting Eq. (C.124) into Eq. (C.76), we find in general that $P_{\mathrm{A}}^{\mathrm{tot}}(t) = P_{\mathrm{A}}^{\mathrm{tot,eq}}$ is only satisfied when

$$\int_0^t \iiint \exp\left\{-\int_\tau^t k\left[{}_{(\tau)}\mathbf{F}(s)\cdot\boldsymbol{\xi}\right]\,ds\right\}\left[\frac{\partial P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})}{\partial\boldsymbol{\xi}}\cdot\mathbf{L}(\tau)\cdot\boldsymbol{\xi}\right] d^3\boldsymbol{\xi}\,d\tau = 0. \tag{C.125}$$

This condition may also be retrieved through simplifying the spatial integral of Eq. (C.90) with $\rho(t) = 1$ and setting the result equal to $P_{\mathrm{A}}^{\mathrm{tot,eq}}$. While Eq. (C.125) is indeed satisfied for the transient network model where $k(\boldsymbol{\xi}) = k_0$, it is unlikely to be satisfied in all the other cases where $k(\boldsymbol{\xi})$ is not constant. We therefore find that the transient network model is the only case of our model where $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ can be guaranteed to remain constant at its equilibrium value $P_{\mathrm{A}}^{\mathrm{tot,eq}}$.

C.5 Additional generalizations and extensions

Here we present some additional generalizations and extensions of the theory that were not included within the original manuscript [148]. The last extension was recently applied in: P. B. Jayathilaka, T. G. Molley, Y. Haung, M. S. Islam, M. R. Buche, M. N. Silberstein, J. J. Kruzic, and K. A. Kilian. Force-mediated molecule release from double network hydrogels. Submitted to Chem. Commun. (2021).

C.5.1 Multiple non-equivalent reaction pathways

We now consider the more general case of M non-equivalent reaction pathways, proceeding in a manner similar to Sec. C.4.1, but beginning with Eq. (5.58) rather than Eq. (5.74). Eq. (5.58) is

$$\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial t} = -\left[\frac{\partial P_{\mathrm{A}}(\boldsymbol{\xi};t)}{\partial \boldsymbol{\xi}}\,\boldsymbol{\xi}\right] : \mathbf{L}(t) - \sum_{j=1}^{M} k_j'(\boldsymbol{\xi})\Big[P_{\mathrm{A}}(\boldsymbol{\xi};t) - \rho_j(t) P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})\Big]. \tag{C.126}$$

where $\rho_j(t) \equiv P_{\mathrm{B}_j}^{\mathrm{tot}}(t)/P_{\mathrm{B}_j}^{\mathrm{tot,eq}}$ and the forward reaction rate coefficient functions $k_j'(\boldsymbol{\xi})$ are given by Eq. (5.26). Following each step in Sec. C.4.1 that does not involve solving for $\rho(t)$, and assuming we begin at equilibrium and $k_j'(\boldsymbol{\xi}) > 0$, we obtain a more general form of the solution in Eq. (C.89),

$$\begin{aligned} P_{\mathrm{A}}(\boldsymbol{\xi};t) =& P_{\mathrm{A}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi};0\right]\Xi(\boldsymbol{\xi};t,0) \\ &+ \int_0^t P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]\Xi(\boldsymbol{\xi};t,\tau)\left\{\sum_{j=1}^{M} k_j'\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]\rho_j(\tau)\right\} d\tau, \end{aligned} \tag{C.127}$$

where the reaction propagator $\Xi(\boldsymbol{\xi};t,\tau)$ is now written as

$$\Xi(\boldsymbol{\xi};t,\tau) \equiv \exp\left\{-\sum_{j=1}^{M}\int_{\tau}^{t} k_j'\left[{}_{(t)}\mathbf{F}(s)\cdot\boldsymbol{\xi}\right]\,ds\right\}. \tag{C.128}$$

Note that for M equivalent reaction pathways, Eq. (C.128) becomes Eq. (C.70) via $k(\boldsymbol{\xi}) \equiv Mk'(\boldsymbol{\xi})$, and Eq. (C.127) becomes Eq. (C.89) via $\rho_j(t) = \rho(t)$. We now require M equations for each of the M unknowns $\rho_j(t)$, which are given by Eq. (5.43), rewritten here as

$$\frac{\partial \rho_j(t)}{\partial t} = \frac{1}{P_{\mathrm{B}_j}^{\mathrm{tot,eq}}}\iiint k_j'(\boldsymbol{\xi})\Big[P_{\mathrm{A}}(\boldsymbol{\xi};t) - \rho_j(t)P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})\Big]d^3\boldsymbol{\xi}. \tag{C.129}$$

Referencing Eq. (C.84), note the total reverse reaction rate coefficients appearing in Eq. (C.129),

$$\hat{K}_j \equiv \frac{1}{P_{\mathrm{B}_j}^{\mathrm{tot,eq}}}\iiint k_j'(\boldsymbol{\xi})P_{\mathrm{A}}^{\mathrm{eq}}(\boldsymbol{\xi})\,d^3\boldsymbol{\xi}. \tag{C.130}$$

Substitution of the solution for $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ in Eq. (C.127) into Eq. (C.129), after defining the total forward rates of breaking initially-intact chains scaled by $P_{\mathrm{B}_j}^{\mathrm{tot,eq}}$,

$$h_j(t) \equiv \frac{1}{P_{\mathrm{B}_j}^{\mathrm{tot,eq}}}\iiint k_j'(\boldsymbol{\xi})P_{\mathrm{A}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi};0\right]\Xi(\boldsymbol{\xi};t,0)\,d^3\boldsymbol{\xi}, \tag{C.131}$$

as well as the kernel functions

$$W_{j\ell}(t,\tau) \equiv \frac{1}{P_{\mathrm{B}_j}^{\mathrm{tot,eq}}}\iiint k_j'(\boldsymbol{\xi})k_\ell'\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]P_{\mathrm{A}}^{\mathrm{eq}}\left[{}_{(t)}\mathbf{F}(\tau)\cdot\boldsymbol{\xi}\right]\Xi(\boldsymbol{\xi};t,\tau)\,d^3\boldsymbol{\xi}, \tag{C.132}$$

allows us to retrieve M equations for the unknowns $\rho_j(t)$,

$$\frac{\partial \rho_j(t)}{\partial t} = \int_0^t\left[\sum_{\ell=1}^{M} W_{j\ell}(t,\tau)\rho_\ell(\tau)\right]d\tau - \hat{K}_j\rho_j(t) + h_j(t). \tag{C.133}$$

Eq. (C.133) gives a coupled system of linear integro-differential equations, which upon solving for each $\rho_j(t)$ and subsequently utilizing Eq. (C.127) provides the solution for $P_{\mathrm{A}}(\boldsymbol{\xi};t)$. For verification,

when all reaction pathways are considered equivalent, Eq. (C.133) can be integrated in order to obtain Eq. (C.76). While we do not present an exact solution to the system given by Eq. (C.133), we note that it is easier to solve numerically than the original system, Eqs. (C.126) and (C.129).

C.5.2 Chemical reactions leaving chains intact

Here we consider the reactions $\mathrm{A} \leftrightharpoons \mathrm{B}_j$ to leave the chains intact, i.e. in the B_j states there is a nontrivial single-chain mechanical response and equilibrium distribution via a non-constant $\psi^*_{\mathrm{B}_j}(\boldsymbol{\xi})$. This case is applicable to many mechanochemically-active polymer networks, such as those containing force-triggered mechanophores that lengthen chains upon activation [16]. We track the distribution of chains in these states $P_{\mathrm{B}_j}(\boldsymbol{\xi};t)$ using the evolution law analogous to Eq. (5.29), which would be

$$\frac{\partial P_{\mathrm{B}_j}(\boldsymbol{\xi};t)}{\partial t} = \sum_{j=1}^{M} k'_j(\boldsymbol{\xi}) \left[P_{\mathrm{A}}(\boldsymbol{\xi};t) - \frac{P_{\mathrm{B}_j}(\boldsymbol{\xi};t)}{P^{\mathrm{eq}}_{\mathrm{B}_j}(\boldsymbol{\xi})} P^{\mathrm{eq}}_{\mathrm{A}}(\boldsymbol{\xi}) \right] - \frac{\partial}{\partial \boldsymbol{\xi}} \cdot \left[\dot{\boldsymbol{\xi}}_{\mathrm{B}_j}(\boldsymbol{\xi};t) P_{\mathrm{B}_j}(\boldsymbol{\xi};t) \right]. \tag{C.134}$$

With the affine assumption we have $\dot{\boldsymbol{\xi}}_{\mathrm{B}_j}(\boldsymbol{\xi};t) = \mathbf{L}(t)\cdot\boldsymbol{\xi}$. The stress in Eq. (5.59) is now

$$\boldsymbol{\sigma}(t) = n \iiint P_{\mathrm{A}}(\boldsymbol{\xi};t) \frac{\partial \psi^*_{\mathrm{A}}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \boldsymbol{\xi}\, d^3\boldsymbol{\xi} + n \iiint \left[\sum_{j=1}^{M} P_{\mathrm{B}_j}(\boldsymbol{\xi};t) \frac{\partial \psi^*_{\mathrm{B}_j}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \right] \boldsymbol{\xi}\, d^3\boldsymbol{\xi} - p(t)\mathbf{1}. \tag{C.135}$$

Rate-independent irreversible reaction

We now consider the special case of rate-independent irreversible (see Sec. C.4.4) reactions that leave chains intact. For simplicity (and in certain cases, without loss of generality), we will only consider a single reaction pathway ($M = 1$). In this case the evolution of $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ is decoupled from that of $P_{\mathrm{B}}(\boldsymbol{\xi};t)$, and we retrieve the solution or $P_{\mathrm{A}}(\boldsymbol{\xi};t)$ given by Eq. (C.119). The solution for $P_{\mathrm{B}}(\boldsymbol{\xi};t)$ can be found after integrating Eq. (C.134) under this special case using the solution for $P_{\mathrm{A}}(\boldsymbol{\xi};t)$, and we assume $P_{\mathrm{B}}(\boldsymbol{\xi};0) = 0$. More intuitively, $P_{\mathrm{B}}(\boldsymbol{\xi};t)$ can be calculated by including the end-to-end vectors that have crossed the yield surface, which would be

$$P_{\mathrm{B}}(\boldsymbol{\xi};t) = P^{\mathrm{eq}}_{\mathrm{A}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}\right]\Big[1 - \Theta(\boldsymbol{\xi};t,0)\Big]. \tag{C.136}$$

After defining the time-dependent effective end-to-end force vector

$$\mathbf{f}_{\mathrm{AB}}(\boldsymbol{\xi};t) \equiv \frac{\partial \psi^*_{\mathrm{A}}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \Theta(\boldsymbol{\xi};t,0) + \frac{\partial \psi^*_{\mathrm{B}}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \Big[1 - \Theta(\boldsymbol{\xi};t,0)\Big], \tag{C.137}$$

the stress in this special case can then be written as

$$\boldsymbol{\sigma}(t) = n \iiint P^{\mathrm{eq}}_{\mathrm{A}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}\right] \mathbf{f}_{\mathrm{AB}}(\boldsymbol{\xi};t)\, \boldsymbol{\xi}\, d^3\boldsymbol{\xi} - p(t)\mathbf{1}. \tag{C.138}$$

Neglecting the change in single-chain mechanical response

Next, we consider the additional specialty where the single-chain mechanical response changes negligibly after the rate-independent irreversible reaction takes place which leaves the chain intact. This effectively amounts to neglecting the change in length and stiffness of the active bond/monomer during the reaction. In this case we can write the stress in the same form as [125], which is that of a permanent network,

$$\boldsymbol{\sigma}(t) = n \iiint P_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}\right] \frac{\partial \psi^*(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\, \boldsymbol{\xi}\, d^3\boldsymbol{\xi} - p(t)\mathbf{1}, \tag{C.139}$$

where $\psi^*(\boldsymbol{\xi})$ is valid for all $\boldsymbol{\xi}$. The distribution of unreacted and reacted chains are still given by Eqs. (C.119) and (C.136), respectively, and we track the totals in each through integration, i.e.

$$P_{\mathrm{A}}^{\mathrm{tot}}(t) = \iiint P_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}\right] \Theta(\boldsymbol{\xi};t,0)\, d^3\boldsymbol{\xi}. \tag{C.140}$$

Application: force-mediated small molecule release in a double network hydrogel

We apply our extended-then-simplified theory to model the mechanical response and release percentage when deforming a double network hydrogel containing mechanophores for small molecule release [199]. We model the release reaction as a rate-independent irreversible reaction occurring at an extension (nondimensional end-to-end length) of γ_r, neglecting changes in single-chain mechanics. We also include irreversible chain breaking at the larger extension $\gamma_c > \gamma_r$. The release percentage $P_{\mathrm{R}}^{\mathrm{tot}}(t)$ is given by Eq. (C.140) when using γ_r within the reaction propagator, i.e.

$$P_{\mathrm{R}}^{\mathrm{tot}}(t) = \iiint P_{\mathrm{A}}^{\mathrm{eq}}\left[\mathbf{F}^{-1}(t)\cdot\boldsymbol{\xi}\right] \Theta_{\mathrm{R}}(\boldsymbol{\xi};t,0)\, d^3\boldsymbol{\xi}. \tag{C.141}$$

The distribution of intact chains is still given by Eq. (C.119), which is then used for stress in Eq. (5.59). Three model parameters were directly determined by the polymer chemistry: the number of links $N_b^{\#} = 85$ was determined by the average number of monomers between crosslinks; the volumetric swelling ratio $J = 10$ was estimated from the water content compared to the dry polymer content; and the pre-swelling shear modulus $n/\beta = 36.56$ kPa was determined such that the model matched the median modulus from experiment of 56 kPa. The nondimensional link stiffness $\kappa = 2000$ and $\gamma_c = 1$ were calibrated to the mechanical response of the material, while $\gamma_r = 0.575$ was then calibrated to the corresponding release percentage. We utilized the EFJC model for simplicity: since κ is large and γ_c is unity, the links of the intact chains are always in the harmonic regime, and therefore the uFJC model would obtain the same results as the EFJC model.

Uniaxial compression experiments were performed on material samples at room temperature to different maximum stresses, holding for 5 min and afterward measuring the release percentage throughout the entire specimen. The resulting mechanical responses and release percentages are shown in Fig. C.2, along with that predicted by the theory. We find reasonable overall agreement between experiment and theory for the mechanical response, where the theoretical response mostly lies within specimen variability. For the release percentage as a function of the applied stress, the theory does not predict the high amount of release at the smaller stress of 0.1 MPa, but predicts the overall sub-linear behavior and is accurate at the higher stresses of 0.5 MPa and 1 MPa. Another set of uniaxial compression experiments were performed: after the samples were compressed to 0.1 MPa

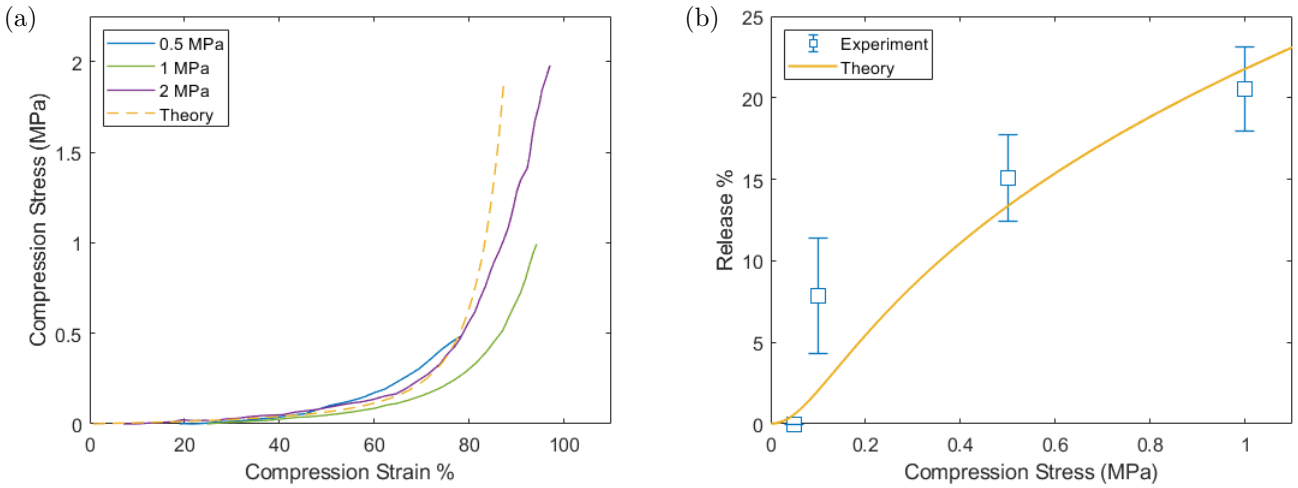


Figure C.2: (a) Engineering stress-strain results from the experiments and the theory. (b) Molecular release percentage as a function of the maximum compressive stress from the experiments and the theory. Plots from [199].

and the release percentage was measured, they were rotated by 90° and reloaded to 0.1 MPa, after which the additional amount release was measured. These results are shown in Fig. C.3 along with the theoretical results, which predict a similar trend in release reduction of the rotated samples. Our theory provides the following insight: for uniaxial compression, release primarily occurs in chains with end-to-end vectors transverse to the loading direction. Since a sizable portion of these end-to-end vectors is shared between the unrotated and rotated sample, less release will be observed when reloading the rotated sample to the same stress.

C.6 Python package

This section details the source code (a `Python` package constituted by four distinct modules) and several self-contained examples pertaining to the `Python` implementation [149] of the model [148] developed in Chapter 5 and Appendix C. The `Python` package is available on `GitHub` and `PyPI` (`pip install chain_breaking_polymer_networks`). It was written for `Python 3`, and uses some typical packages: `numpy`, `scipy`, and `matplotlib`. After installation, the package is best imported using:

```
from chain_breaking_polymer_networks import *
```

C.6.1 Python Modules

`single_chain`

The package contains a `single_chain` module of classes corresponding to different single-chain models. For example,

```
single_chain_model = ideal(N_b = 88)
```

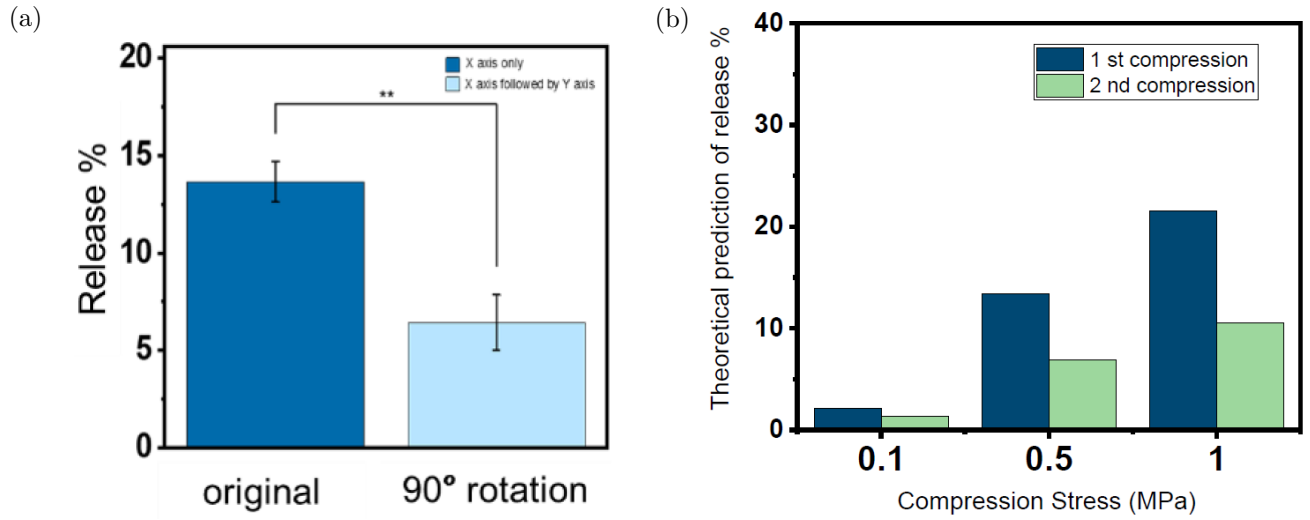


Figure C.3: (a) Molecular release percentage on a sample after first compression to 0.1 MPa, 90° rotation, and second compression to 0.1 MPa. (b) Theoretical molecular release percentage after first compression, 90° rotation, and second compression, for three different stress values. Plots from [199].

creates an ideal chain model with 88 links. The single-chain models are nearly fully nondimensional (model parameters, inputs and outputs for functions) and they use keyword arguments for the model parameters, though not all parameters are optional (see source code, or examples). The single-chain models contain four main functions:

- the nondimensional single-chain mechanical response, `eta(gamma)`, which is $\eta(\gamma)$;
- the nondimensional equilibrium probability density distribution of intact chain extensions, `P_A_eq(gamma)`, which is $P_{\mathrm{A}}^{\mathrm{eq}}(\gamma)$;
- the nondimensional equilibrium radial distribution function, `g_A_eq(gamma)`, which is $g_{\mathrm{A}}^{\mathrm{eq}}(\gamma)$;
- the net forward reaction rate coefficient function, `k(gamma)` (is the only function with units, 1/seconds), which is $k(\gamma)$.

Example: for the ideal chain model above,

```
single_chain_model.P_A_eq(0.88)
```

returns the probability density that a chain is both intact and at a nondimensional end-to-end length of 0.88, which is $P_{\mathrm{A}}^{\mathrm{eq}}(0.8)$.

The `Python` module file `single_chain.py` is printed below:

```
#########################################################################################################################
# General setup
#########################################################################################################################

# Import libraries
import sys
import numpy as np
from scipy.integrate import quad
from scipy.interpolate import interp1d
```

```
from scipy.optimize import minimize_scalar

# Interpolation parameters
num_interp = int(3e3)
interp_kind_1D = 'cubic'

# Numerical tolerance parameters
cutoff_for_log_over_sinh = 3e1
cutoff_stretch_for_harmonic_eta_EFJC = 3
minimum_exponent = np.log(sys.float_info.min)/np.log(10)
maximum_exponent = np.log(sys.float_info.max)/np.log(10)
eta_small = 10**minimum_exponent

# Function to invert a function
def inv_fun_1D(x_query, fun, bounds = None):

  # Change method depending on whether bounds are involved
  if bounds is None:
    return minimize_scalar(lambda x: np.abs(fun(x) - x_query)).x
  else:
    return minimize_scalar(lambda x: np.abs(fun(x) - x_query), bounds = bounds, method = 'bounded').x

# Function to create interpolation function from stored function
def interp_fun_1D(x_store, y_store):
  return interp1d(x_store, y_store, kind = interp_kind_1D, bounds_error = False, fill_value = np.nan)

# Function to avoid overflow when computing ln(x/sinh(x))
def log_over_sinh(x):

  # Determine when argument is sufficiently large
  where_x_large = np.nan_to_num(x, nan = -1) > cutoff_for_log_over_sinh
  log_of_x_over_sinh_x = np.zeros(x.size)

  # Use asymptotic relation valid for sufficiently large arguments
  if where_x_large.any():
    log_of_x_over_sinh_x[where_x_large] = np.log(2*x[where_x_large]) - x[where_x_large]

  # Compute analytically otherwise, and zero where argument is zero
  where_x_zero = x == 0
  where_compute = ~(where_x_large + where_x_zero)
  if where_compute.any():
    log_of_x_over_sinh_x[where_compute] = np.log(x[where_compute]/np.sinh(x[where_compute]))
  return log_of_x_over_sinh_x

# Hyperbolic cotangent function
def coth_safe(eta):
  eta = np.where(eta == 0, eta_small, eta)
  return 1/np.tanh(eta)

# Langevin function
def Langevin(eta):
  eta = np.where(eta == 0, eta_small, eta)
  return 1/np.tanh(eta) - 1/eta

########################################################################################################################
# Ideal chain model
########################################################################################################################

class ideal:

  # For more information, see:
  # Statistical mechanical constitutive theory of polymer networks:
  #   The inextricable links between distribution, behavior, and ensemble
  #   Michael R. Buche and Meredith N. Silberstein
  #   Physical Review E, 2021, 102, 012501
  #   doi.org/10.1103/PhysRevE.102.012501

  # Class initialization
  def __init__(self, **kwargs):

    # Default parameter values
    N_b = 88
    k_0 = np.exp(minimum_exponent)
    gamma_c = np.inf

    # Retrieve specified parameters
    for key, value in kwargs.items():
      if key == 'N_b':
        N_b = value
      elif key == 'k_0':
        if value > 0:
          k_0 = value
      elif key == 'gamma_c':
        gamma_c = value

    # Retain for certain purposes
    self.N_b = N_b
    self.k_0 = k_0
    self.gamma_c = gamma_c

    # Model-specific modifications
    self.P_A_tot_eq = 1
```

```
    self.gamma_TS = np.inf
    self.k = lambda gamma_in: k_0 + 0*gamma_in
    self.K_hat = k_0
    self.max_k_rev = k_0
    self.N_b_H = 0
    self.varsigma = 1

  # Nondimensional mechamical response of the chain
  def eta(self, gamma_in):
    return 3*gamma_in

  # Nondimensional equilibrium distribution function
  def P_A_eq(self, gamma_in, normalization = 1):
    return (gamma_in <= self.gamma_c)*(3*self.N_b/2/np.pi)**(3/2)*np.exp(-3/2*self.N_b*gamma_in**2)/normalization

  # Nondimensional equilibrium radial distribution function
  def g_A_eq(self, gamma_in, normalization = 1):
    return 4*np.pi*gamma_in**2*self.P_A_eq(gamma_in, normalization)

################################################################################################################################################
# Extensible freely-joined chain model
################################################################################################################################################

class EFJC:

  # For more information, see:
  #   Analytical results of the extensible freely jointed chain model
  #   Alessandro Fiasconaro and Fernando Falo
  # Physica A 2019, 532, 121929
  #   doi.org/10.1016/j.physa.2019.121929
  # See also:
  # Statistical mechanical constitutive theory of polymer networks:
  #   The inextricable links between distribution, behavior, and ensemble
  #   Michael R. Buche and Meredith N. Silberstein
  #   Physical Review E, 2021, 102, 012501
  #   doi.org/10.1103/PhysRevE.102.012501

  # Class initialization
  def __init__(self, **kwargs):

    # Default parameter values
    N_b = None
    kappa = None
    k_0 = np.exp(minimum_exponent)
    gamma_c = np.inf

    # Retrieve specified parameters
    for key, value in kwargs.items():
      if key == 'N_b':
        N_b = value
      elif key == 'k_0':
        if value > 0:
          k_0 = value
      elif key == 'kappa':
        kappa = value
      elif key == 'gamma_c':
        gamma_c = value

    # Check parameter specifications
    if N_b is None:
      sys.exit('Error: Need to specify N_b.')
    elif kappa is None:
      sys.exit('Error: Need to specify kappa.')

    # Retain for certain purposes
    self.N_b = N_b
    self.k_0 = k_0
    self.kappa = kappa
    self.gamma_c = gamma_c

    # Model-specific modifications
    self.P_A_tot_eq = 1
    self.gamma_TS = np.inf
    self.k = lambda gamma_in: k_0 + 0*gamma_in
    self.K_hat = k_0
    self.max_k_rev = k_0
    self.N_b_H = 0
    self.varsigma = 1

    # Nondimensional mechanical response of the chain
    def gamma_fun(eta):
      coth = coth_safe(eta)
      L = Langevin(eta)
      return L + eta/kappa*(1 + (1 - L*coth)/(1 + eta/kappa*coth))

    # Compute and store the inverted nondimensional mechanical response to interpolate from
    self.gamma_store = np.linspace(0, cutoff_stretch_for_harmonic_eta_EFJC, num_interp)
    self.eta_store = np.zeros(self.gamma_store.size)
    for i in range(1, len(self.gamma_store)):
      self.eta_store[i] = inv_fun_1D(self.gamma_store[i], gamma_fun)

    # Function to interpolate from the inverted nondimensional mechamical response of the chain
```

```
    self.eta_interp_fun = interp_fun_1D(self.gamma_store, self.eta_store)
    def eta_fun(gamma_in):
      if isinstance(gamma_in, np.ndarray):
        eta_out = np.zeros(gamma_in.shape)
        harmonic_region = gamma_in > cutoff_stretch_for_harmonic_eta_EFJC
        eta_out[harmonic_region] = kappa*(gamma_in[harmonic_region] - 1)
        eta_out[~harmonic_region] = self.eta_interp_fun(gamma_in[~harmonic_region])
      else:
        if gamma_in > cutoff_stretch_for_harmonic_eta_EFJC:
          eta_out = kappa*(gamma_in - 1)
        else:
          eta_out = self.eta_interp_fun(gamma_in)
      return eta_out

    # Nondimensional equilibrium distribution function
    def P_A_eq_fun(gamma_in, normalization = 1):

      # Compute mechanical response
      eta = np.array(eta_fun(gamma_in))
      eta[eta == 0] = eta_small

      # Compute nondimensional Helmholtz free energy per link
      coth = coth_safe(eta)
      L = Langevin(eta)
      vartheta = eta*L + log_over_sinh(eta) - np.log(1 + eta/kappa*coth) \
        + eta**2/kappa/2*(1/2 + (1 - L*coth)/(1 + eta/kappa*coth))

      # Compute P_A_eq below the yield surface
      return (gamma_in <= self.gamma_c)*np.exp(-N_b*vartheta)/normalization

    # Nondimensional equilibrium radial distribution function
    def g_A_eq_fun(gamma_in, normalization = 1):
      return 4*np.pi*gamma_in**2*P_A_eq_fun(gamma_in, normalization)

    # Normalize the equilibrium distribution
    P_A_eq_normalization = quad(g_A_eq_fun, 0, np.inf, full_output = 1)[0]/self.P_A_tot_eq

    # Compute and store the function to interpolate from
    self.P_A_eq_store = P_A_eq_fun(self.gamma_store, normalization = P_A_eq_normalization)
    self.P_A_eq_interp_fun = interp_fun_1D(self.gamma_store, self.P_A_eq_store)

  # Function to interpolate from the inverted nondimensional mechamical response of the chain
  def eta(self, gamma_in):
    if isinstance(gamma_in, np.ndarray):
      eta_out = np.zeros(gamma_in.shape)
      harmonic_region = gamma_in > cutoff_stretch_for_harmonic_eta_EFJC
      eta_out[harmonic_region] = self.kappa*(gamma_in[harmonic_region] - 1)
      eta_out[~harmonic_region] = self.eta_interp_fun(gamma_in[~harmonic_region])
    else:
      if gamma_in > cutoff_stretch_for_harmonic_eta_EFJC:
        eta_out = self.kappa*(gamma_in - 1)
      else:
        eta_out = self.eta_interp_fun(gamma_in)
    return eta_out

  # Function to interpolate from the stored nondimensional equilibrium distribution function
  def P_A_eq(self, gamma_in, normalization = 1):
    return (gamma_in <= self.gamma_c)*self.P_A_eq_interp_fun(gamma_in)/normalization

  # Function for the nondimensional equilibrium radial distribution function
  def g_A_eq(self, gamma_in, normalization = 1):
    return 4*np.pi*gamma_in**2*self.P_A_eq(gamma_in, normalization)

########################################################################################################################
# Morse potential-supplemented freely-joined chain model
########################################################################################################################

class Morse_FJC:

  # For more information, see:
  #   Chain breaking in the statistical mechanical constitutive theory of polymer networks
  #   Michael R. Buche and Meredith N. Silberstein
  # Preprint submitted to Journal of the Mechanics and Physics of Solids
  #   arxiv.org/abs/2104.08866

  # Class initialization
  def __init__(self, **kwargs):

    # Default parameter values
    N_b = None
    N_b_H = 0
    k_0 = np.exp(minimum_exponent)
    omega = None
    grumbo = None
    beta_u_b = None
    kappa = None
    kappa_H = None
    varsigma = 1
    beta_Delta_Psi_0 = 0
    gamma_c = None

    # Retrieve specified parameters
```

```
for key, value in kwargs.items():
  if key == 'N_b':
    N_b = value
  elif key == 'N_b_H':
    N_b_H = value
  elif key == 'k_0':
    if value > 0:
      k_0 = value
  elif key == 'omega':
    omega = value
  elif key == 'kappa':
    kappa = value
  elif key == 'grumbo':
    grumbo = value
  elif key == 'beta_u_b':
    beta_u_b = value
  elif key == 'kappa_H':
    kappa_H = value
  elif key == 'varsigma':
    varsigma = value
  elif key == 'beta_Delta_Psi_0':
    beta_Delta_Psi_0 = value
  elif key == 'gamma_c':
    gamma_c = value

# Check parameter specifications
if N_b is None:
  sys.exit('Error: Need to specify N_b in Morse_FJC().')
elif kappa is None:
  sys.exit('Error: Need to specify kappa in Morse_FJC().')
elif N_b_H == 0:
  kappa_H = 1 # kappa_H just has to be nonzero if N_b_H = 0
elif kappa_H is None:
  sys.exit('Error: Specify nonzero kappa_H when specifying nonzero N_b_H in Morse_FJC().')
elif N_b == 0:
  sys.exit('Error: For N_b = 0, use the class EFJC() instead of Morse_FJC().')
if beta_u_b is not None:
  if grumbo is not None:
    sys.exit('Error: Either gamma or beta_u_b need to be specified in Morse_FJC(), but not both.')
  else:
    grumbo = np.sqrt(kappa/2/beta_u_b)
elif beta_u_b is None:
  if grumbo is None:
    sys.exit('Error: Need to specify beta_u_b in Morse_FJC().')

# Retain for certain purposes
self.N_b = N_b
self.N_b_H = N_b_H
self.varsigma = varsigma
self.k_0 = k_0
self.gamma_c = gamma_c

# Automatic physical parameters
eta_max = kappa/4/grumbo
self.P_A_tot_eq = 1/(1 + N_b*np.exp(-beta_Delta_Psi_0))
self.P_B_tot_eq = 1 - self.P_A_tot_eq

# Check if Python thinks P_A_tot_eq or P_B_tot_eq = 0
if self.P_B_tot_eq**2 == 0:
  sys.exit('Error: P_B_tot_eq is smaller than precision allows; need to decrease beta_Delta_Psi_0 in Morse_FJC().')
elif self.P_A_tot_eq**2 == 0:
  sys.exit('Error: P_A_tot_eq is smaller than precision allows; need to increase beta_Delta_Psi_0 in Morse_FJC().')

# Nondimensional incremental mechanical response of the Morse potential
def Delta_lambda(eta):
  return np.log(2/(1 + np.sqrt(1 - eta/eta_max)))/grumbo

# Nondimensional Morse potential
def beta_u(eta):
  return kappa/2/grumbo**2*(1 - np.exp(-grumbo*Delta_lambda(eta) ))**2

# Nondimensional mechanical response of the chain
def gamma_fun(eta):
  r = varsigma*N_b_H/N_b
  eta_c = varsigma*eta
  return 1/(1 + r)*(Langevin(eta) + Delta_lambda(eta) + r*(Langevin(eta_c) + eta_c/kappa_H))

# Limiting stretch for an intact chain
self.gamma_TS = gamma_fun(eta_max)

# Ensure that gamma_c is below the limiting stretch for an intact chain
if gamma_c is not None:
  if gamma_c > self.gamma_TS:
    sys.exit('Error: Cannot have gamma_c > gamma_TS = ' + str(self.gamma_TS))

# Initial nondimensional free energy barrier to transition state
beta_Delta_Psi_TS_0 = kappa/8/grumbo**2 - 2*np.log(self.gamma_TS)

# Ensure either attempt frequency or k(0) have been specified
if omega is None:
  if k_0 is None:
    sys.exit('Error: Either omega or k_0 need to be specified in Morse_FJC().')
```

```
      else:
        omega = 2*np.pi/N_b*np.exp(beta_Delta_Psi_TS_0 + np.log(k_0))
    elif k_0 is None:
      k_0 = N_b/2/np.pi*np.exp(-beta_Delta_Psi_TS_0 + np.log(omega))
    else:
      sys.exit('Error: Either omega or k_0 need to be specified in Morse_FJC(), but not both.')

    # Compute and store the inverted nondimensional mechanical response to interpolate from
    self.gamma_store = np.linspace(0, self.gamma_TS, num_interp)
    self.eta_store = np.zeros(self.gamma_store.size)
    for i in range(len(self.gamma_store)):
      self.eta_store[i] = inv_fun_1D(self.gamma_store[i], gamma_fun, bounds = (0, eta_max))

    # Function to interpolate from the inverted nondimensional mechamical response of the chain
    self.eta = interp_fun_1D(self.gamma_store, self.eta_store)

    # Nondimensional equilibrium distribution function
    def P_A_eq_fun(gamma_in, normalization = 1):

      # Compute nondimensional Helmholtz free energy per link
      eta = self.eta(gamma_in)
      r_N = N_b_H/N_b
      eta_c = varsigma*eta
      vartheta = 1/(1 + r_N)*(eta/np.tanh(eta) - 1 + log_over_sinh(eta) + beta_u(eta) \
        + r_N*(eta_c/np.tanh(eta_c) - 1 + log_over_sinh(eta_c) + eta_c**2/2/kappa_H))

      # Avoid overflow before computing and returning P_A_eq
      exponent = -np.log(normalization) - (N_b + N_b_H)*vartheta
      exponent[exponent > maximum_exponent] = maximum_exponent
      exponent[exponent < minimum_exponent] = minimum_exponent
      return np.exp(exponent)

    # Nondimensional equilibrium radial distribution function
    def g_A_eq_fun(gamma_in, normalization = 1):
      return 4*np.pi*gamma_in**2*P_A_eq_fun(gamma_in, normalization)

    # Normalize the equilibrium distribution
    P_A_eq_normalization = quad(g_A_eq_fun, 0, self.gamma_TS, full_output = 1)[0]/self.P_A_tot_eq

    # Compute and store the function to interpolate from
    self.P_A_eq_store = P_A_eq_fun(self.gamma_store, normalization = P_A_eq_normalization)
    self.P_A_eq_interp_fun = interp_fun_1D(self.gamma_store, self.P_A_eq_store)

    # Reaction rate coefficient function
    if self.gamma_c is None:
      def k_fun(gamma_in):

        # Compute nondimensional Helmholtz free energy barrier
        eta = self.eta(gamma_in)
        adj_beta_Delta_Psi_TS = log_over_sinh(self.gamma_TS*eta) - log_over_sinh(eta) - beta_u(eta) \
            + self.gamma_TS*eta/np.tanh(self.gamma_TS*eta) - eta/np.tanh(eta)

        # Avoid overflow before computing and returning k
        exponent = np.log(k_0) - N_b*adj_beta_Delta_Psi_TS
        exponent[exponent > maximum_exponent] = maximum_exponent
        exponent[exponent < minimum_exponent] = minimum_exponent
        return np.exp(exponent)

      # Compute and store the function to interpolate from
      self.log_k_store = np.log(k_fun(self.gamma_store))
      self.log_k_interp_fun = interp_fun_1D(self.gamma_store, self.log_k_store)

    # Total reverse reaction rate coefficient
    if self.gamma_c is None:
      integrand_K_hat = lambda xi: k_fun(xi)*g_A_eq_fun(xi, normalization = P_A_eq_normalization)
      self.K_hat = quad(integrand_K_hat, 0, self.gamma_TS, full_output = 1)[0]/self.P_B_tot_eq
    else:
      self.K_hat = k_0*self.P_A_tot_eq/self.P_B_tot_eq

    # Maximum reverse reaction rate coefficient
    if self.gamma_c is None:
      fun = lambda xi: -k_fun(xi)*P_A_eq_fun(xi, normalization = P_A_eq_normalization)/self.P_B_tot_eq
      self.max_k_rev = -fun(minimize_scalar(fun, bounds = (0, self.gamma_TS), method = 'bounded').x)[0]
    else:
      self.max_k_rev = k_0*P_A_eq_fun(0)[0]/self.P_B_tot_eq

  # Function to interpolate from the stored nondimensional equilibrium distribution function
  def P_A_eq(self, gamma_in, normalization = 1):
    return self.P_A_eq_interp_fun(gamma_in)/normalization

  # Function for the nondimensional equilibrium radial distribution function
  def g_A_eq(self, gamma_in, normalization = 1):
    return 4*np.pi*gamma_in**2*self.P_A_eq(gamma_in, normalization)

  # Function to interpolate from the stored reaction rate coefficient function
  def k(self, gamma_in):
    if self.gamma_c is None:
      return np.exp(self.log_k_interp_fun(gamma_in))
    else:
      if isinstance(gamma_in, np.ndarray):
        k_out = self.k_0*np.ones(gamma_in.shape)
        k_out[gamma_in > self.gamma_c] = np.inf
```

```
    else:
      if gamma_in > self.gamma_c:
        k_out = np.inf
      else:
        k_out = self.k_0
    return k_out
```

`network`

The `network` module contains a few classes, most notably the `deform_network` class. Given an applied deformation, such as

```
def F(t): return 1 + t
```

the total testing time in seconds, the deformation mode (currently supports `'uniaxial'` and `'equibiaxial'` stress; the deformation is of course incompressible), and a single-chain model, the `deform_network` class is used to create a network model from the single-chain model. Here we apply uniaxial stress for 3 seconds:

```
network_model = deform_network(F, 'uniaxial', 3, single_chain_model, \
    ignore_yield = True, use_spatial_grid = False)
```

Since this initialization also prepares the solution method, many optional keyword arguments are available. In this example we ignore the breaking of chains via meeting a yield surface at some critical extension (the ideal chain model is infinitely extensible) using `ignore_yield = True`, and we choose to utilize quadrature for spatial integrations rather than a pre-specified spatial grid using `use_spatial_grid = False`; the converse in either case is the default. The examples that follow illustrate the critical aspects of creating the network model, and more information can be found in the helpful comments in the `network.py` file. The results (stress, total probability that a chain is intact, etc.) are solved for over the specified testing time using

```
results = network_model.solve(csv_directory ='./')
```

where the optional keyword argument here is used to write the results to a .csv file. The default is None (no .csv file is written).

The `Python` module file `network.py` is printed below:

```
#########################################################################################################################
# General setup
#########################################################################################################################

# Import libraries
import sys
import numpy as np
from scipy.interpolate import interp1d, interp2d
from scipy.integrate import romb, simpson, dblquad

# Numerical parameters
array_factor_est = 8
tol_residual_rho = 1e-88
dblquad_epsabs = 1e-3
dblquad_epsrel = 1e-3
num_interp_quad_specialized_ignore_yield = int(1e2)
minimum_exponent = np.log(sys.float_info.min)/np.log(10)

#########################################################################################################################
# Deformation application class
```

```
#########################################################################################################################################

class deform_network:

  ###################################################################################################################################
  # Initialization
  ###################################################################################################################################

  def __init__(self, F, deformation_type, total_time_in_seconds, single_chain_model, relaxation_function = None, J_sw = 1, \
    max_F_dot = None, max_RAM_usage_in_bytes = None, nondimensional_timestep_suggestion = 1e-2, num_grid_suggestion = 129, \
    interp_kind_2D = 'quintic', use_spatial_grid = True, enumerate_full_arrays = True, \
    ignore_yield = False, ignore_reforming = False):

    # Initialize and retain certain variables
    self.csv_initialized = False
    self.ignore_yield = ignore_yield
    self.ignore_reforming = ignore_reforming
    self.use_spatial_grid = use_spatial_grid
    self.enumerate_full_arrays = enumerate_full_arrays
    self.initialized_single_chain_model = single_chain_model

    ###################################################################################################################################
    # Deformation
    ###################################################################################################################################

    # Retain certain variables
    self.F = F
    self.J_sw = J_sw
    self.deformation_type = deformation_type
    self.total_time_in_seconds = total_time_in_seconds

    # Estimate the maximum rate of deformation if not given
    if max_F_dot is None:
      t_temp = np.linspace(0, total_time_in_seconds, int(1e5))
      self.max_F_dot = np.max(np.abs(np.diff(F(t_temp))/np.diff(t_temp)))
    else:
      self.max_F_dot = max_F_dot

    # If gamma_c specified, best method to use may change
    if single_chain_model.gamma_c is None:
      self.use_specialized = False
    else:
      self.use_specialized = True

    ###################################################################################################################################
    # Spatial discretization or quadrature
    ###################################################################################################################################

    # Spatial integration using a grid
    if use_spatial_grid is True:

      # Retain the 2D interpolation kind
      self.interp_kind_2D = interp_kind_2D

      # Create the symmetry-conscious spatial grid
      self.num_grid = self.adjust_for_romb(num_grid_suggestion)
      self.z = np.linspace(0, single_chain_model.gamma_TS, self.num_grid)
      self.r = self.z
      self.dz = self.z[1] - self.z[0]
      self.dr = self.dz
      self.Z, self.R = np.meshgrid(self.z, self.r)
      ELL = np.sqrt(self.Z*self.Z + self.R*self.R)

      # Integration element specialized for stress calculation
      self.ELEMENT_stress = self.element_stress(self.Z, self.R, single_chain_model)

      # Adjust normalization of P_A_eq on the grid
      P_A_eq_ELL_non_normalized = np.nan_to_num(single_chain_model.P_A_eq(ELL), nan = 0)
      self.P_A_eq_normalization = self.integral_grid_d_3_xi(P_A_eq_ELL_non_normalized)/single_chain_model.P_A_tot_eq

      # Total reverse reaction rate coefficient on the grid
      P_A_eq_ELL = np.nan_to_num(single_chain_model.P_A_eq(ELL, normalization = self.P_A_eq_normalization), nan = 0)
      self.k_ELL = np.nan_to_num(single_chain_model.k(ELL), nan = 0)
      if single_chain_model.gamma_c is None:
        self.K_hat = self.integral_grid_d_3_xi(self.k_ELL*P_A_eq_ELL)/single_chain_model.P_B_tot_eq
      else:
        self.K_hat = single_chain_model.K_hat

      # Maximum reverse reaction rate coefficient on the grid
      if single_chain_model.gamma_c is None:
        self.max_k_rev = np.max(self.k_ELL*P_A_eq_ELL/single_chain_model.P_B_tot_eq)
      else:
        self.max_k_rev = single_chain_model.max_k_rev

    # Spatial integration using quadrature
    else:

      # Inherit from single_chain_model since will use the same integration scheme
      self.P_A_eq_normalization = 1
      self.k_0 = single_chain_model.k_0
      self.K_hat = single_chain_model.K_hat
      self.max_k_rev = single_chain_model.max_k_rev
```

```
    ########################################################################################################################
    # Relaxation function related setup
    ########################################################################################################################

    if relaxation_function is None:
      self.g_timescale = np.inf
      self.g = lambda t, tau: 1 + 0*t + 0*tau
      self.d_g_d_tau = lambda t, tau: 0*t + 0*tau
      self.g_K_hat = np.exp(minimum_exponent)
    else:
      self.g_timescale = relaxation_function.timescale
      self.g = relaxation_function.g
      self.d_g_d_tau = relaxation_function.d_g_d_tau
      try:
        self.g_K_hat = single_chain_model.P_A_tot_eq/single_chain_model.P_B_tot_eq/self.g_timescale
      except AttributeError:
        self.g_K_hat = np.exp(minimum_exponent)

    ########################################################################################################################
    # Time discretization
    ########################################################################################################################

    # Estimate timestep based on the smallest timescales
    timescales = np.array([1/self.max_F_dot, 1/self.K_hat, 1/self.max_k_rev, self.g_timescale, 1/self.g_K_hat])
    estimated_timestep = float(nondimensional_timestep_suggestion*np.min(timescales))

    # Enumerating full arrays requires large memory, so have to do it in chunks rather than over the full history
    if use_spatial_grid is True:

      # Memory considerations
      if max_RAM_usage_in_bytes is None:
        import psutil
        max_RAM_usage_in_bytes = psutil.virtual_memory().available
      max_array_numel = max_RAM_usage_in_bytes/8/array_factor_est
      if enumerate_full_arrays is True:
        max_num_time_chunk = np.floor(np.sqrt(max_array_numel/self.num_grid**2)).astype(int)
      else:
        max_num_time_chunk = np.floor(max_array_numel/self.num_grid**2).astype(int)

      # Chunk history and decrease timestep in order to satisfy memory requirements and Romberg integration
      self.num_chunks = 0
      self.num_time = 2*max_num_time_chunk
      while self.num_time > max_num_time_chunk:
        self.num_chunks += 1
        self.num_time = self.adjust_for_romb(total_time_in_seconds/estimated_timestep/self.num_chunks)
      self.timestep = total_time_in_seconds/self.num_chunks/(self.num_time - 1)

      # Enumerate time limits for each chunk
      t_lims_all = total_time_in_seconds/self.num_chunks*np.arange(0, self.num_chunks + 1, 1)
      self.t_lims = np.zeros((self.num_chunks, 2))
      for index_chunk in range(self.num_chunks):
        self.t_lims[index_chunk,:] = [t_lims_all[index_chunk], t_lims_all[index_chunk + 1]]

    # No memory considerations and corresponding history chunking since will not enumerate full arrays
    else:
      self.num_time = self.adjust_for_romb(total_time_in_seconds/estimated_timestep)
      self.timestep = total_time_in_seconds/(self.num_time - 1)

  ########################################################################################################################
  # Function to solve for results over the applied deformation history
  ########################################################################################################################

  def solve(self, display_progress = True, csv_directory = None, checkpoint_directory = None):

    # Methods using a grid for spatial integrals
    if self.use_spatial_grid is True:

      # Enumerate full arrays to mimimize computation time, chunking time history to satisfy memory requirements
      if self.enumerate_full_arrays is True:

        # Function to remove an initial point from results
        def remove_initial_point(results):
          results_out = list(results)
          for index in range(len(results)):
            results_out[index] = results[index][1:]
          return tuple(results_out)

        # Allocate results
        t = []
        F = []
        P_A_tot = []
        total_rate_break = []
        total_rate_reform = []
        beta_sigma_over_n = []

        # Loop over all chunks in the history
        for index_chunk in range(self.num_chunks):

          # Display progress if opted
          if display_progress is True:
            print('  On chunk', index_chunk + 1, 'of', self.num_chunks, end = '\r')
```

```
        # Equilibrium initial distribution for first chunk, or initial distribution from end of previous chunk
        if index_chunk == 0:
          results_chunk, P_A_end = self.compute_results_grid_chunky(self.t_lims[index_chunk,:])
        else:
          results_chunk, P_A_end = self.compute_results_grid_chunky(self.t_lims[index_chunk,:], P_A_0 = P_A_end)

        # Remove repeated points in time among chunks
        if index_chunk > 0 or self.J_sw != 1 or self.initialized_single_chain_model.gamma_c is not None:
          results_chunk = remove_initial_point(results_chunk)

        # Collect results
        t = np.append(t, results_chunk[0])
        F = np.append(F, results_chunk[1])
        P_A_tot = np.append(P_A_tot, results_chunk[2])
        total_rate_break = np.append(total_rate_break, results_chunk[3])
        total_rate_reform = np.append(total_rate_reform, results_chunk[4])
        beta_sigma_over_n = np.append(beta_sigma_over_n, results_chunk[5])
        results = t, F, P_A_tot, total_rate_break, total_rate_reform, beta_sigma_over_n

        # Create a checkpoint .csv if opted after each chunk
        if checkpoint_directory is not None:
          checkpoint(checkpoint_directory).create(t[-1], P_A_end)

    # Refrain from enumerating full arrays but requires no history chunking, takes longer
    else:
      results, P_A_end = self.compute_results_grid([0, self.total_time_in_seconds])

  # Methods using quadrature for spatial integrals
  else:

    # Efficient method in the special case for when k is constant and the single-chain model is infinitely-extensible
    if self.use_specialized is True and self.ignore_yield is True:
      results, P_A_end = self.compute_results_quad_specialized_ignore_yield([0, self.total_time_in_seconds])

    # Method for the general case
    else:
      results, P_A_end = self.compute_results_quad([0, self.total_time_in_seconds])

  # Append results to the .csv if opted
  if csv_directory is not None:
    if self.csv_initialized is False:
      results_csv_initialized = results_csv(csv_directory)
      self.csv_initialized = True
    results_csv_initialized.append(0, results)

  # Create a checkpoint .csv if opted
  if checkpoint_directory is not None:
    checkpoint(checkpoint_directory).create(t[-1], P_A_end)

  # Return the results
  return results

##########################################################################################################################
# Function for computing results using spatial quadrature
##########################################################################################################################

def compute_results_quad(self, t_span, P_A_0 = None):

  # Enumerate the time and relative deformation components
  t, F_zz_rel, F_rr_rel = self.enumerate_t_and_F_rel(t_span)
  total_time = t_span[-1] - t_span[0]
  num_time = len(t)

  # Function for the relatively-deformed coordinates
  def ell_rel(z, r, index_t, index_tau):
    return np.sqrt(z*z*F_zz_rel[index_t, index_tau]**2 + r*r*F_rr_rel[index_t, index_tau]**2)

  ########################################################################################################################
  # Time-dependent quantities for spatial integrals
  ########################################################################################################################

  # Function for the relatively-deformed initial distribution
  if P_A_0 is None:
    def P_A_0_rel_t(z, r, index_t):
      return self.initialized_single_chain_model.P_A_eq(ell_rel(z, r, index_t, 0)/self.J_sw**(1/3), \
        normalization = self.P_A_eq_normalization*self.J_sw)
  else:
    sys.exit('Error: Beginning from a nonequilibrium initial distribution not yet implemented.')

  # Function for the relatively-deformed equilibrium distribution
  def P_A_eq_rel(z, r, index_t, index_tau):
    return self.initialized_single_chain_model.P_A_eq(ell_rel(z, r, index_t, index_tau), \
      normalization = self.P_A_eq_normalization)

  # Function for the relatively-deformed reaction rate coefficient function
  def k_rel(z, r, index_t, index_tau):
    return self.initialized_single_chain_model.k(ell_rel(z, r, index_t, index_tau))

  # Function for the reaction propagator
  def Xi(z, r, index_t, index_tau):
    if index_t == index_tau:
```

```
      return 1 + 0*(z + r)
    else:
      integrand = np.zeros((index_t + 1 - index_tau))*(z + r)
      for index_s in range(index_t + 1 - index_tau):
        integrand[:,:,index_s] = k_rel(z, r, index_t, index_s)
      return np.exp(-self.integral_ds(integrand))

  # Function for the homogeneous solution for P_A
  def P_A_h(z, r, index_t):
    return np.nan_to_num(P_A_0_rel_t(z, r, index_t)*Xi(z, r, index_t, 0)*self.g(t[index_t], 0), nan = 0)

  # Function for the integrand of K
  def integrand_K(z, r, index_t, index_tau):
    return np.nan_to_num(P_A_eq_rel(z, r, index_t, index_tau)*Xi(z, r, index_t, index_tau)*( \
      k_rel(z, r, index_t, index_tau)*self.g(t[index_t], t[index_tau]) \
        + self.d_g_d_tau(t[index_t], t[index_tau])), nan = 0)

  ##################################################################################################################
  # Solve the integral equation
  ##################################################################################################################

  # Amount of initially-intact chains that have been broken
  P_B_tot_h = np.zeros(num_time)
  for index_t in range(num_time):
    P_B_tot_h[index_t] = 1 - self.integral_quad_d_3_xi(lambda z,r: P_A_h(z, r, index_t))

  # Integral equation only defined when P_B_tot_eq is nonzero
  if self.initialized_single_chain_model.P_B_tot_eq > 0 and self.ignore_reforming is False:

    # Kernel K(t,tau) and right-hand side b(t)
    b = self.initialized_single_chain_model.P_B_tot_eq*P_B_tot_h
    K = np.zeros((num_time, num_time))
    for index_t in range(num_time):
      for index_tau in range(index_t + 1):
        K[index_t, index_tau] = self.integral_quad_d_3_xi( \
          lambda z,r: integrand_K(z, r, index_t, index_tau))/self.initialized_single_chain_model.P_B_tot_eq

    # Successive approximations to retrieve rho(t)
    rho = self.solve_Volterra(K, b, total_time)

    # Total probability of broken chains
    P_B_tot = self.initialized_single_chain_model.P_B_tot_eq*rho

  # Integral equation undefined when P_B_tot_eq = 0; also for rate-independent irreversible breaking
  else:
    P_B_tot = P_B_tot_h

  ##################################################################################################################
  # Compute and return the results
  ##################################################################################################################

  # Total probability of intact chains
  P_A_tot = 1 - P_B_tot

  # Distribution at the end of the partition
  P_A_end = np.nan

  # Total breaking and reforming rates
  total_rate_reform = self.K_hat*(1 - P_A_tot)
  total_rate_break = np.gradient(P_A_tot)/np.gradient(t) - total_rate_reform

  # Nondimensional stress corresponding to applied the deformation
  beta_sigma_h_over_n = np.zeros(num_time)
  beta_sigma_p_over_n = np.zeros(num_time)
  for index_t in range(num_time):
    beta_sigma_h_over_n[index_t] = self.integral_quad_d_3_xi(lambda z,r: P_A_h(z, r, index_t), element = 'stress')
    if self.initialized_single_chain_model.P_B_tot_eq > 0 and self.ignore_reforming is False:
      integrand_beta_sigma_p_over_n = np.zeros(num_time)
      for index_tau in range(index_t + 1):
        integrand_beta_sigma_p_over_n[index_tau] = rho[index_tau]*self.integral_quad_d_3_xi( \
          lambda z,r: integrand_K(z, r, index_t, index_tau), element = 'stress')
      beta_sigma_p_over_n[index_t] = self.integral_d_tau(integrand_beta_sigma_p_over_n)
  beta_sigma_over_n = beta_sigma_h_over_n + beta_sigma_p_over_n

  # Return results
  results = t, self.F(t), P_A_tot, total_rate_break, total_rate_reform, beta_sigma_over_n
  return results, P_A_end

##################################################################################################################
# Function for computing results on a spatial grid
##################################################################################################################

def compute_results_grid(self, t_span, P_A_0 = None):

  # Enumerate the time and relative deformation components
  t, F_zz_rel, F_rr_rel = self.enumerate_t_and_F_rel(t_span)
  total_time = t_span[-1] - t_span[0]
  num_time = len(t)

  # Function for the relatively-deformed coordinates
  def ell_rel(index_t, index_tau):
    return np.sqrt(self.Z*self.Z*F_zz_rel[index_t, index_tau]**2 + self.R*self.R*F_rr_rel[index_t, index_tau]**2)
```

```
##########################################################################################################################
# Time-dependent quantities for spatial integrals
##########################################################################################################################

# Function for the relatively-deformed initial distribution
if P_A_0 is None:
  def P_A_0_rel_t(index_t):
    return self.initialized_single_chain_model.P_A_eq(ell_rel(index_t, 0)/self.J_sw**(1/3), \
      normalization = self.P_A_eq_normalization*self.J_sw)
else:
  sys.exit('Error: Beginning from a nonequilibrium initial distribution not yet implemented.')

# Function for the relatively-deformed equilibrium distribution
def P_A_eq_rel(index_t, index_tau):
  return self.initialized_single_chain_model.P_A_eq(ell_rel(index_t, index_tau), \
    normalization = self.P_A_eq_normalization)

# Function for the relatively-deformed reaction rate coefficient function
def k_rel(index_t, index_tau):
  return self.initialized_single_chain_model.k(ell_rel(index_t, index_tau))

# Function for the reaction propagator
def Xi(index_t, index_tau):
  if index_t == index_tau:
    return np.ones((self.num_grid, self.num_grid))
  else:
    integrand = np.zeros((self.num_grid, self.num_grid, index_t + 1 - index_tau))
    for index_s in range(index_t + 1 - index_tau):
      integrand[:,:,index_s] = k_rel(index_t, index_s)
    return np.exp(-self.integral_ds(integrand))

# Function for the homogeneous solution for P_A
def P_A_h(index_t):
  out = P_A_0_rel_t(index_t)*Xi(index_t, 0)*self.g(t[index_t], 0)
  out[np.isnan(out) + np.isinf(out)] = 0
  return out

# Function for the integrand of K
def integrand_K(index_t, index_tau):
  out = P_A_eq_rel(index_t, index_tau)*Xi(index_t, index_tau)*( \
    k_rel(index_t, index_tau)*self.g(t[index_t], t[index_tau]) + self.d_g_d_tau(t[index_t], t[index_tau]))
  out[np.isnan(out) + np.isinf(out)] = 0
  return out

##########################################################################################################################
# Solve the integral equation
##########################################################################################################################

# Amount of initially-intact chains that have been broken
P_B_tot_h = np.zeros(num_time)
for index_t in range(num_time):
  P_B_tot_h[index_t] = 1 - self.integral_grid_d_3_xi(P_A_h(index_t))

# Integral equation only defined when P_B_tot_eq is nonzero
if self.initialized_single_chain_model.P_B_tot_eq > 0 and self.ignore_reforming is False:

  # Kernel K(t,tau) and right-hand side b(t)
  b = self.initialized_single_chain_model.P_B_tot_eq*P_B_tot_h
  K = np.zeros((num_time, num_time))
  for index_t in range(num_time):
    for index_tau in range(index_t + 1):
      K[index_t, index_tau] = \
        self.integral_grid_d_3_xi(integrand_K(index_t,index_tau))/self.initialized_single_chain_model.P_B_tot_eq

  # Successive approximations to retrieve rho(t)
  rho = self.solve_Volterra(K, b, total_time)

  # Total probability of broken chains
  P_B_tot = self.initialized_single_chain_model.P_B_tot_eq*rho

# Integral equation undefined when P_B_tot_eq = 0; also for rate-independent irreversible breaking
else:
  P_B_tot = P_B_tot_h

##########################################################################################################################
# Compute and return the results
##########################################################################################################################

# Total probability of intact chains
P_A_tot = 1 - P_B_tot

# Distribution at the end of the partition
P_A_end = np.nan

# Total breaking and reforming rates
total_rate_reform = self.K_hat*(1 - P_A_tot)
total_rate_break = np.gradient(P_A_tot)/np.gradient(t) - total_rate_reform

# Nondimensional stress corresponding to applied the deformation
beta_sigma_h_over_n = np.zeros(num_time)
beta_sigma_p_over_n = np.zeros(num_time)
```

```
  for index_t in range(num_time):
    beta_sigma_h_over_n[index_t] = self.integral_grid_d_3_xi(P_A_h(index_t), element = 'stress')
    if self.initialized_single_chain_model.P_B_tot_eq > 0 and self.ignore_reforming is False:
      integrand_beta_sigma_p_over_n = np.zeros(num_time)
      for index_tau in range(index_t + 1):
        integrand_beta_sigma_p_over_n[index_tau] = \
          rho[index_tau]*self.integral_grid_d_3_xi(integrand_K(index_t, index_tau), element = 'stress')
      beta_sigma_p_over_n[index_t] = self.integral_d_tau(integrand_beta_sigma_p_over_n)
  beta_sigma_over_n = beta_sigma_h_over_n + beta_sigma_p_over_n

  # Return results
  results = t, self.F(t), P_A_tot, total_rate_break, total_rate_reform, beta_sigma_over_n
  return results, P_A_end

##########################################################################################################################################
# Function for computing results on a spatial grid; enumerates full arrays and uses vectorized operations
##########################################################################################################################################

def compute_results_grid_chunky(self, t_span, P_A_0 = None):

  # Enumerate the time and relative deformation components
  t, F_zz_rel, F_rr_rel = self.enumerate_t_and_F_rel(t_span)
  total_time = t_span[-1] - t_span[0]
  num_time = len(t)
  Delta_t = t - t[0]

  # Enumerate the relatively-deformed coordinates
  Z_rel = np.tensordot(self.Z, F_zz_rel, axes = 0)
  R_rel = np.tensordot(self.R, F_rr_rel, axes = 0)
  ELL_rel = np.sqrt(Z_rel*Z_rel + R_rel*R_rel)
  z_rel_t = np.tensordot(self.z, F_zz_rel[:,0], axes = 0)
  r_rel_t = np.tensordot(self.r, F_rr_rel[:,0], axes = 0)

  # Cleanup
  del F_zz_rel, F_rr_rel, Z_rel, R_rel

  ########################################################################################################################################
  # Time-dependent quantities for spatial integrals
  ########################################################################################################################################

  # Enumerate the relatively-deformed equilibrium distribution
  P_A_eq_rel = self.initialized_single_chain_model.P_A_eq(ELL_rel, normalization = self.P_A_eq_normalization)

  # Enumerate the relatively-deformed initial distribution
  if P_A_0 is None:
    P_A_0_rel_t = self.initialized_single_chain_model.P_A_eq(ELL_rel[:,:,:,0]/self.J_sw**(1/3), \
      normalization = self.P_A_eq_normalization*self.J_sw)
  else:
    P_A_0_rel_t = np.zeros((self.num_grid, self.num_grid, num_time))
    for index_t in range(num_time):
      P_A_0_rel_t[:,:,index_t] = self.interp_fun_2D(z_rel_t[:,index_t], r_rel_t[:,index_t], P_A_0)

  # Cleanup
  del z_rel_t, r_rel_t

  # Enumerate the relatively-deformed reaction rate coefficient function
  k_rel = self.initialized_single_chain_model.k(ELL_rel)

  # Cleanup
  del ELL_rel

  # Enumerate the reaction propagator
  Xi = np.zeros((self.num_grid, self.num_grid, num_time, num_time))
  for index_t in range(num_time):
    for index_tau in range(index_t + 1):
      Xi[:,:,index_t,index_tau] = \
        np.exp(-self.integral_ds(k_rel[:,:,index_t,index_tau:index_t + 1]))

  # Enumerate the relaxation function and its derivative
  g = np.zeros((num_time, num_time))
  d_g_d_tau = np.zeros((num_time, num_time))
  for index_t in range(num_time):
    for index_tau in range(index_t + 1):
      g[index_t, index_tau] = self.g(Delta_t[index_t], Delta_t[index_tau])
      d_g_d_tau[index_t, index_tau] = self.d_g_d_tau(Delta_t[index_t], Delta_t[index_tau])

  # Homogeneous solution for P_A
  P_A_h = Xi[:,:,:,0]*P_A_0_rel_t

  # Cleanup
  del P_A_0_rel_t

  # Integrand of K
  if self.initialized_single_chain_model.gamma_c is not None:
    k_rel[np.isinf(k_rel)] = 0
  integrand_K = P_A_eq_rel*Xi*(k_rel*g[None,None,:,:] + d_g_d_tau[None,None,:,:])

  # Cleanup
  del Xi, k_rel, P_A_eq_rel

  #
  P_A_h[np.isnan(P_A_h) + np.isinf(P_A_h)] = 0
```

```
    integrand_K[np.isnan(integrand_K) + np.isinf(integrand_K)] = 0

    ##########################################################################################################################
    # Solve the integral equation
    ##########################################################################################################################

    # Amount of initially-intact chains that have been broken
    P_B_tot_h = 1 - self.integral_grid_d_3_xi(P_A_h)

    # Integral equation only defined when P_B_tot_eq is nonzero
    if self.initialized_single_chain_model.P_B_tot_eq > 0 and self.ignore_reforming is False:

      # Kernel K(t,tau) and right-hand side b(t)
      K = self.integral_grid_d_3_xi(integrand_K)/self.initialized_single_chain_model.P_B_tot_eq
      b = P_B_tot_h/self.initialized_single_chain_model.P_B_tot_eq

      # Successive approximations to retrieve rho(t)
      rho = self.solve_Volterra(K, b, total_time)

      # Total probability of broken chains
      P_B_tot = self.initialized_single_chain_model.P_B_tot_eq*rho

    # Integral equation undefined when P_B_tot_eq = 0; also for rate-independent irreversible breaking
    else:
      P_B_tot = P_B_tot_h
      rho = np.zeros(num_time)

    # Integrand of particular solution for P_A
    integrand_P_A_p = rho*integrand_K

    # Cleanup
    del integrand_K

    ##########################################################################################################################
    # Compute and return the results
    ##########################################################################################################################

    # Total probability of intact chains
    P_A_tot = 1 - P_B_tot

    # Distribution of intact chains
    P_A = P_A_h + self.integral_d_tau(integrand_P_A_p)

    # Cleanup
    del P_A_h, integrand_P_A_p

    # Distribution at the end of the partition
    P_A_end = P_A[:,:,-1]

    # Total breaking and reforming rates
    total_rate_break = -self.integral_grid_d_3_xi(P_A*self.k_ELL[:,:,None])
    total_rate_reform = self.K_hat*(1 - P_A_tot)

    # Nondimensional stress corresponding to the applied deformation
    beta_sigma_over_n = self.integral_grid_d_3_xi(P_A, element = 'stress')

    # Return results
    results = t, self.F(t), P_A_tot, total_rate_break, total_rate_reform, beta_sigma_over_n
    return results, P_A_end

  ############################################################################################################################
  # Function for (one with k0 AND ignores yield)
  ############################################################################################################################

  def compute_results_quad_specialized_ignore_yield(self, t_span):

    # Enumerate the time and relative deformation components
    t, F_zz_rel, F_rr_rel = self.enumerate_t_and_F_rel(t_span)

    # Function for the relatively-deformed coordinates
    def ell_rel(z, r, index_t, index_tau):
      return np.sqrt(z*z*F_zz_rel[index_t, index_tau]**2 + r*r*F_rr_rel[index_t, index_tau]**2)

    ##########################################################################################################################
    # Time-dependent quantities for spatial integrals
    ##########################################################################################################################

    # Function for the relatively-deformed equilibrium distribution
    def P_A_eq_rel(z, r, index_t, index_tau):
      return np.nan_to_num(self.initialized_single_chain_model.P_A_eq(ell_rel(z, r, index_t, index_tau)/self.J_sw**(1/3),\
        normalization = self.P_A_eq_normalization*self.J_sw), nan = 0)

    # Function for the reaction propagator
    def Xi(t, tau):
      return np.exp(-self.k_0*(t - tau))

    # Function for relative time derivative of the reaction propagator
    def d_Xi_d_tau(t, tau):
      return Xi(t, tau)*self.k_0

    ##########################################################################################################################
    # Compute stress only at unique deformations to interpolate from
```

```
    ##########################################################################################################################

    # Choose deformation for stress response based on deformation mode
    if self.deformation_type == 'uniaxial':
      F_use = F_zz_rel

    # Store a limited number of unique deformations over the history
    F_unique = np.unique(F_use)
    indices = np.unique(np.round(np.linspace(0, len(F_unique) - 1, num_interp_quad_specialized_ignore_yield)).astype(int))
    F_store = F_unique[indices]

    # Compute the stress at these deformations to interpolate from
    beta_sigma_h0_over_n = np.zeros(len(F_store))
    for index in range(len(F_store)):
      index_t, index_tau = np.argwhere(F_store[index] == F_use)[0]
      beta_sigma_h0_over_n[index] = \
        self.integral_quad_d_3_xi(lambda z,r: P_A_eq_rel(z, r, index_t, index_tau), element = 'stress')

    ##########################################################################################################################
    # Compute and return the results
    ##########################################################################################################################

    # Function to interpolate from computed stress response
    interp_sigma_fun = interp1d(F_store, beta_sigma_h0_over_n, kind = 'cubic', bounds_error = True)

    # Homogeneous solution for the nondimensional stress
    beta_sigma_h_over_n = self.g(t, 0)*Xi(t, 0)*interp_sigma_fun(F_zz_rel[:,0])

    # Particular solution for the nondimensional stress
    beta_sigma_p_over_n = np.zeros(self.num_time)
    for index_t in range(self.num_time):
      integrand = interp_sigma_fun(F_zz_rel[index_t,:index_t + 1])*( \
        Xi(t[index_t], t[:index_t + 1])*self.d_g_d_tau(t[index_t], t[:index_t + 1]) \
        + d_Xi_d_tau(t[index_t], t[:index_t + 1])*self.g(t[index_t], t[:index_t + 1]))
      beta_sigma_p_over_n[index_t] = self.integral_d_tau(np.append(integrand, np.zeros(self.num_time - 1 - index_t)))

    # Return results
    beta_sigma_over_n = beta_sigma_h_over_n + beta_sigma_p_over_n
    P_A_tot = self.initialized_single_chain_model.P_A_tot_eq*np.ones(self.num_time)
    total_rate_reform = self.K_hat*(1 - P_A_tot)
    total_rate_break = -total_rate_reform
    results = t, self.F(t), P_A_tot, beta_sigma_h_over_n, beta_sigma_p_over_n, beta_sigma_over_n
    return results, np.nan

  ##############################################################################################################################
  # Function to adjust discretization for Romberg integration
  ##############################################################################################################################

  def adjust_for_romb(self, num_discretization, decrease = False):
    if ((np.log(num_discretization - 1)/np.log(2)).is_integer()):
      return int(round(num_discretization))
    else:
      n = 0
      dos_check = 3
      while dos_check >= 2:
        n += 1
        dos_check = (num_discretization - 1)**(1/n)
      if decrease is True and dos_check < 2:
        return int(1 + 2**(n - 1))
      else:
        return int(1 + 2**n)

  ##############################################################################################################################
  # Function for integration over the spatial grid
  ##############################################################################################################################

  def integral_grid_d_3_xi(self, FUN, element = None):
    if element is None:
      element = self.R
    elif element == 'stress':
      element = self.ELEMENT_stress
    if FUN.ndim == 2:
      return 4*np.pi*romb(romb(FUN*element, dx = self.dr, axis = 0), dx = self.dz, axis = 0)
    elif FUN.ndim == 3:
      return 4*np.pi*romb(romb(FUN*element[:,:,None], dx = self.dr, axis = 0), dx = self.dz, axis = 0)
    elif FUN.ndim == 4:
      return 4*np.pi*romb(romb(FUN*element[:,:,None,None], dx = self.dr, axis = 0), dx = self.dz, axis = 0)

  ##############################################################################################################################
  # Function for integration over continuous space
  ##############################################################################################################################

  def integral_quad_d_3_xi(self, fun, element = None):
    if element is None:
      def integrand(z, r):
        return fun(z, r)*r
    elif element == 'stress':
      def integrand(z, r):
        return fun(z, r)*self.element_stress(z, r, self.initialized_single_chain_model)
    if self.initialized_single_chain_model.gamma_c is None:
      lim = self.initialized_single_chain_model.gamma_TS
    else:
```

```
      lim = self.initialized_single_chain_model.gamma_c
    return 4*np.pi*dblquad(integrand, 0, lim, lambda r: 0, lambda r: np.sqrt(lim**2 - r**2), \
      epsabs = dblquad_epsabs, epsrel = dblquad_epsrel)[0]

  ##########################################################################################################################
  # Function for integration element specialized for stress calculation
  ##########################################################################################################################

  def element_stress(self, z, r, single_chain_model):
    ell = np.sqrt(z*z + r*r)
    eta = np.nan_to_num(single_chain_model.eta(ell), nan = 0)
    if isinstance(z, np.ndarray):
      eta_over_ell = np.zeros(z.shape)
      eta_over_ell[ell != 0] = eta[ell != 0]/ell[ell != 0]
    else:
      if ell == 0:
        eta_over_ell = 0
      else:
        eta_over_ell = eta/ell
    C = (single_chain_model.N_b + single_chain_model.varsigma*single_chain_model.N_b_H)/self.J_sw
    if self.deformation_type == 'uniaxial':
      return C*eta_over_ell*r*(z*z - r*r/2)
    elif self.deformation_type == 'equibiaxial' or deformation_type == 'simple_shear':
      return C*eta_over_ell*r*(r*r/2 - z*z)

  ##########################################################################################################################
  # Function to interpolate from a stored 2D function on the spatial grid
  ##########################################################################################################################

  def interp_fun_2D(self, z_query, r_query, FUN):
    return interp2d(self.z, self.r, FUN, kind = self.interp_kind_2D)(z_query, r_query)

  ##########################################################################################################################
  # Functions for integration in time
  ##########################################################################################################################

  def integral_ds(self, FUN):
    return simpson(FUN, dx = self.timestep, axis = -1, even = 'last')

  def integral_d_tau(self, FUN):
    return romb(FUN, dx = self.timestep, axis = -1)

  ##########################################################################################################################
  # Function to enumerate the time and relative deformation
  ##########################################################################################################################

  def enumerate_t_and_F_rel(self, t_span):

    # Enumerate the time
    t = np.linspace(t_span[0], t_span[-1], self.num_time)

    # Relative deformation gradient components
    if self.deformation_type == 'uniaxial':
      F_zz_rel = np.tensordot(1/self.F(t), self.F(t), axes = 0)
      F_rr_rel = np.tensordot(np.sqrt(self.F(t)), 1/np.sqrt(self.F(t)), axes = 0)
    elif self.deformation_type == 'equibiaxial':
      F_zz_rel = np.tensordot(self.F(t), 1/self.F(t), axes = 0)
      F_rr_rel = np.tensordot(1/self.F(t), self.F(t), axes = 0)
    elif self.deformation_type == 'simple_shear':
      pass

    return t, F_zz_rel, F_rr_rel

  ##########################################################################################################################
  # Function to solve the Volterra integral equation
  ##########################################################################################################################

  def solve_Volterra(self, K, b, total_time):
    M = 0
    rho = b
    residual_bound_rho = 1
    while residual_bound_rho > tol_residual_rho:
      M += 1
      rho = b - self.integral_d_tau(K*rho)
      residual_bound_rho = \
        (self.K_hat*total_time)**(M + 1)/self.initialized_single_chain_model.P_B_tot_eq/np.math.factorial(M + 1)
    return rho

##############################################################################################################################
# Checkpoint creation class
##############################################################################################################################

class checkpoint:

  # Initialization also clears any previous checkpoint
  def __init__(self, checkpoint_directory):
    self.checkpoint_directory_and_file = checkpoint_directory + 'checkpoint.csv'
    open(self.checkpoint_directory_and_file, 'w').close()

  # Function to create checkpoints
  def create(self, t_end, P_A_end):
    f = open(self.checkpoint_directory_and_file, 'a')
```

```
    f.write("%.8e" % t_end)
    for i in range(len(P_A_end[:,0])):
      f.write("\n")
      for j in range(len(P_A_end[0,:])):
        f.write("%.8e\t" % P_A_end[i,j])
    f.close()

  # Function to read checkpoints
  def read(self, existing_checkpoint_directory_and_file):
    pass

################################################################################################################################
# Results writing class
################################################################################################################################

class results_csv:

  # Initialization also clears any previous results
  def __init__(self, csv_directory):
    self.csv_directory_and_file = csv_directory + 'results.csv'
    open(self.csv_directory_and_file, "w").close()

  # Function to append results
  def append(self, index_chunk, results):
    f = open(self.csv_directory_and_file, "a")
    for index_t in range(len(results[0])):
      for index_results in range(len(results)):
        f.write("%.8e\t" % results[index_results][index_t])
      f.write("\n")
    f.close()
```

`relaxation_function`

The `relaxation_function` module contains several different classes corresponding to different relaxation functions, `g(t, tau)`, as attributes, as well as their derivatives, `d_g_d_tau(t, tau)`, and their corresponding loss and storage functions, `g_p(t, tau)` and `g_pp(t, tau)`. For information on currently-implemented relaxation functions, see: the chain relaxation dynamics of the Rouse model [210, 211], the additional transient bond dynamics of the Sticky Rouse model [212, 213], recent more complicated Sticky Rouse-based models [214, 215], and other relaxation functions [167, 168]. For example,

```
sticky_Rouse(N_b = 50, N_x = 5, t_0 = 4e-4, beta_E_A = 10)
```

returns an object that is the sticky Rouse model with 50 Kuhn monomers, 5 crosslinks per chain, a Kuhn monomer relaxation time of 0.0004 seconds, and a nondimensional crosslink dissociation energy of 10 [212, 214]. The relaxation function is an optional keyword argument when creating the network model (the default is None).

The `Python` module file `relaxation_function.py` is printed below:

```
################################################################################################################################
# General setup
################################################################################################################################

# Import libraries
import sys
import numpy as np
from scipy.integrate import quad

################################################################################################################################
# Relaxation function from Long et al. 2014
################################################################################################################################

class Long_et_al_2014:

  # For more information, see:
  #   Time Dependent Behavior of a Dual Cross-Link Self-Healing Gel: Theory and Experiments
  #   Rong Long, Koichi Mayumi, Costantino Creton, Tetsuharu Narita, and Chung-Yuen Hui
  #   Macromolecules 2014, 47, 7243-7250
  #   doi.org/10.1021/ma501290h
  # See also:
```

```
  # Mechanics of a Dual Cross-Link Gel with Dynamic Bonds: Steady State Kinetics and Large Deformation Effects
  #   Jingyi Guo, Rong Long, Koichi Mayumi, and Chung-Yuen Hui
  #   Macromolecules 2016, 49, 3497-3507
  # doi.org/10.1021/acs.macromol.6b00421
  # See also:
  #   Chain breaking in the statistical mechanical constitutive theory of polymer networks
  #   Michael R. Buche and Meredith N. Silberstein
  # Preprint submitted to Journal of the Mechanics and Physics of Solids
  #   arxiv.org/abs/2104.08866

  def __init__(self, **kwargs):

    # Default parameter values
    self.alpha = None
    self.t_R = None
    self.x_p = 0

    # Retrieve specified parameters
    for key, value in kwargs.items():
      if key == 'alpha':
        self.alpha = value
      elif key == 't_R':
        self.t_R = value
      elif key == 'x_p':
        self.x_p = value

    # Check parameter specifications
    if self.alpha is None:
      sys.exit("Error: Need to specify alpha for Long_et_al_2014().")
    elif self.alpha < 1:
      sys.exit("Error: Need to specify alpha > 1 for Long_et_al_2014().")
    elif self.t_R is None:
      sys.exit("Error: Need to specify t_R for Long_et_al_2014().")

    # Smallest timescale associated with this relaxation function
    self.timescale = self.t_R

  # Relaxation function
  def g(self, t, tau):
    return self.x_p + (1 - self.x_p)*(1 + (self.alpha - 1)*(t - tau)/self.t_R)**(1/(1 - self.alpha))

  # Relative time derivative of the relaxation function
  def d_g_d_tau(self, t, tau):
    return (1 - self.x_p)/self.t_R*(1 + (self.alpha - 1)*(t - tau)/self.t_R)**(self.alpha/(1 - self.alpha))

  # Storage function
  def g_p(self, omega):
    g_p_fun = lambda omega: \
      self.x_p + omega*quad(lambda s: (self.g(s, 0) - self.x_p)*np.sin(omega*s), 0, np.inf, full_output = 1)[0]
    return np.vectorize(g_p_fun)(omega)

  # Loss function
  def g_pp(self, omega):
    g_pp_fun = lambda omega: \
      omega*quad(lambda s: (self.g(s, 0) - self.x_p)*np.cos(omega*s), 0, np.inf, full_output = 1)[0]
    return np.vectorize(g_pp_fun)(omega)

##########################################################################################################################
# Relaxation function for the Rouse model
##########################################################################################################################

class Rouse:

  # For more information, see:
  #   A Theory of the Linear Viscoelastic Properties of Dilute Solutions of Coiling Polymers
  # Prince E Rouse
  # Journal of Chemical Physics, 1953, 21, 1272-1280
  # doi.org/10.1063/1.1699180
  # See also:
  # Extensions of the Rouse Theory of Viscoelastic Properties to Undiluted Linear Polymers
  # John D. Ferry, Robert F. Landel, and Malcolm L. Williams
  # Journal of Applied Physics, 1955, 26, 359
  # doi.org/10.1063/1.1721997
  # See also:
  #   Bridging experiments and theory:
  #   isolating the effects of metal-ligand interactions on viscoelasticity of reversible polymer networks
  #   Xinyue Zhang, Yuval Vidavsky, Sinai Aharonovich, Steven J. Yang, Michael R. Buche,
  #   Charles E. Diesendruck and Meredith N. Silberstein
  # Soft Matter, 2020, 16, 8591-8601
  # doi.org/10.1039/DOSM01115K

  def __init__(self, **kwargs):

    # Default parameter values
    self.N_b = None
    self.t_0 = None

    # Retrieve specified parameters
    for key, value in kwargs.items():
      if key == 'N_b':
        self.N_b = value
      elif key == 't_0':
```

```
        self.t_0 = value

    # Check parameter specifications
    if self.N_b is None:
      sys.exit("Error: Need to specify N_b for Rouse().")
    elif self.t_0 is None:
      sys.exit("Error: Need to specify t_0 for Rouse().")

    # Smallest timescale associated with this relaxation function
    self.timescale = self.t_0

  # Relaxation function
  def g(self, t, tau):
    g_out = np.zeros(omega.shape)
    for p in range(1, int(self.N_b + 1)):
      t_p = self.t_0*(self.N_b/p)**2
      g_out = g_out + np.exp(-(t - tau)/t_p)/self.N_b
    return g_out

  # Relative time derivative of the relaxation function
  def d_g_d_tau(self, t, tau):
    d_g_d_tau_out = np.zeros(omega.shape)
    for p in range(1, int(self.N_b + 1)):
      t_p = self.t_0*(self.N_b/p)**2
      d_g_d_tau_out = d_g_d_tau_out + np.exp(-(t - tau)/t_p)/self.N_b/t_p
    return d_g_d_tau_out

  # Storage function
  def g_p(self, omega):
    g_p_out = np.zeros(omega.shape)
    for p in range(1, int(self.N_b + 1)):
      t_p = self.t_0*(self.N_b/p)**2
      g_p_out = g_p_out + ((omega*t_p)**2/(1 + (omega*t_p)**2))/self.N_b
    return g_p_out

  # Loss function
  def g_pp(self, omega):
    g_pp_out = np.zeros(omega.shape)
    for p in range(1, int(self.N_b + 1)):
      t_p = self.t_0*(self.N_b/p)**2
      g_pp_out = g_pp_out + (omega*t_p/(1 + (omega*t_p)**2))/self.N_b
    return g_pp_out

########################################################################################################################
# Relaxation function for the sticky Rouse model
########################################################################################################################

class sticky_Rouse:

  # For more information, see:
  #   Ionomer dynamics and the sticky Rouse model
  # Quan Chen, Gregory J. Tudryn, and Ralph H. Colby
  # Journal of Rheology 2013, 57, 1441
  # doi.org/10.1122/1.4818868
  # See also:
  #   Bridging experiments and theory:
  #   isolating the effects of metal-ligand interactions on viscoelasticity of reversible polymer networks
  #   Xinyue Zhang, Yuval Vidavsky, Sinai Aharonovich, Steven J. Yang, Michael R. Buche,
  #   Charles E. Diesendruck and Meredith N. Silberstein
  # Soft Matter, 2020, 16, 8591-8601
  # doi.org/10.1039/DOSM01115K

  def __init__(self, **kwargs):

    # Default parameter values
    self.N_b = None
    self.N_x = None
    self.t_0 = None
    self.t_x = None
    self.beta_E_A = None
    self.G_x_over_G_0 = 0

    # Retrieve specified parameters
    for key, value in kwargs.items():
      if key == 'N_b':
        self.N_b = value
      elif key == 'N_x':
        self.N_x = value
      elif key == 't_0':
        self.t_0 = value
      elif key == 't_x':
        self.t_x = value
      elif key == 'beta_E_a':
        self.beta_E_a = value
        self.t_x = self.t_0*np.exp(self.beta_E_a)
      elif key == 'G_x_over_G_0':
        self.G_x_over_G_0 = value

    # Check parameter specifications
    if self.N_b is None:
      sys.exit("Error: Need to specify N_b for sticky_Rouse().")
    elif self.N_x is None:
```

```
      sys.exit("Error: Need to specify N_x for sticky_Rouse().")
    elif self.t_0 is None:
      sys.exit("Error: Need to specify t_0 for sticky_Rouse().")
    elif self.t_x is None:
      sys.exit("Error: Need to specify t_x for sticky_Rouse().")

    # Smallest timescale associated with this relaxation function
    self.timescale = np.min([self.t_0, self.t_x])

  # Relaxation function
  def g(self, t, tau):
    g_out = np.zeros(omega.shape)
    for p in range(int(self.N_x + 1), int(self.N_b + 1)):
      t_p = self.t_0*(self.N_b/p)**2
      g_out = g_out + np.exp(-(t - tau)/t_p)/self.N_b
    for p in range(1, int(self.N_x + 1)):
      t_p_x = self.t_x*(self.N_x/p)**2
      g_out = g_out + (1 + self.G_x_over_G_0)*np.exp(-(t - tau)/t_p_x)/self.N_b
    return g_out

  # Relative time derivative of the relaxation function
  def d_g_d_tau(self, t, tau):
    d_g_d_tau_out = np.zeros(omega.shape)
    for p in range(int(self.N_x + 1), int(self.N_b + 1)):
      t_p = self.t_0*(self.N_b/p)**2
      d_g_d_tau_out = d_g_d_tau_out + np.exp(-(t - tau)/t_p)/self.N_b/t_p
    for p in range(1, int(self.N_x + 1)):
      t_p_x = self.t_x*(self.N_x/p)**2
      d_g_d_tau_out = d_g_d_tau_out + (1 + self.G_x_over_G_0)*np.exp(-(t - tau)/t_p_x)/self.N_b/t_p_x
    return d_g_d_tau_out

  # Storage function
  def g_p(self, omega):
    g_p_out = np.zeros(omega.shape)
    for p in range(int(self.N_x + 1), int(self.N_b + 1)):
      t_p = self.t_0*(self.N_b/p)**2
      g_p_out = g_p_out + ((omega*t_p)**2/(1 + (omega*t_p)**2))/self.N_b
    for p in range(1, int(self.N_x + 1)):
      t_p_x = self.t_x*(self.N_x/p)**2
      g_p_out = g_p_out + (1 + self.G_x_over_G_0)*((omega*t_p_x)**2/(1 + (omega*t_p_x)**2))/self.N_b
    return g_p_out

  # Loss function
  def g_pp(self, omega):
    g_pp_out = np.zeros(omega.shape)
    for p in range(int(self.N_x + 1), int(self.N_b + 1)):
      t_p = self.t_0*(self.N_b/p)**2
      g_pp_out = g_pp_out + (omega*t_p/(1 + (omega*t_p)**2))/self.N_b
    for p in range(1, int(self.N_x + 1)):
      t_p_x = self.t_x*(self.N_x/p)**2
      g_pp_out = g_pp_out + (1 + self.G_x_over_G_0)*(omega*t_p_x/(1 + (omega*t_p_x)**2))/self.N_b
    return g_pp_out
```

`plotting`

The plotting module allows object for plotting to be quickly created, i.e.

```
plotter_object = plotter(plot_directory = './')
```

where the default for the optional keyword argument is shown (the current directory). The `plotter` class is the only instance requiring `matplotlib` in this package. The single-chain model functions can be plotted using

```
plotter_object.plot_single_chain(single_chain_model)
```

and all the results from deforming the network using

```
plotter_object.plot_results(network_model, results)
```

If only the nondimensional stress-stretch response of the network is desired, one can use

```
plotter_object.plot_results(None, results)
```

The `Python` module file `plotting.py` is printed below:

```
#############################################################################################################################################
# General setup
#############################################################################################################################################

# Import libraries
import numpy as np
import matplotlib.pyplot as plt

#############################################################################################################################################
# Plotting class
#############################################################################################################################################
class plotter:

  def __init__(self, plot_directory = './'):
    self.plot_directory = plot_directory

  # For general plotting
  def save_current_figure(self, xlabel, ylabel, name):
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.tight_layout()
    plt.show()
    fig = plt.gcf()
    plt.savefig(self.plot_directory + name)
    plt.close()

  ###########################################################################################################################################
  # Function for plotting deformation
  ###########################################################################################################################################

  def plot_deformation(self, F, total_time_in_seconds, csv_directory = None):

    # Enumerate time
    t_temp = np.linspace(0, total_time_in_seconds, int(1e5))

    # Plot the deformation
    fig = plt.figure()
    plt.plot(t_temp, F(t_temp))
    self.save_current_figure('$t$ [seconds]', '$F(t)$', 'F(t).png')

    # Save values to .csv if opted
    if csv_directory is not None:
      self.save_csv(csv_directory + 'deformation.csv', t_temp, F(t_temp))

  ###########################################################################################################################################
  # Function for plotting single-chain functions
  ###########################################################################################################################################

  def plot_single_chain(self, single_chain_model, J_sw = None, csv_directory = None):

    # Enumerate stretch
    if np.isinf(single_chain_model.gamma_TS):
      gamma_plot = np.linspace(0, 1.5, 1000)
    else:
      gamma_plot = np.linspace(0, single_chain_model.gamma_TS, 1000)

    # Plot the nondimensional mechanical response
    fig = plt.figure()
    plt.plot(gamma_plot, single_chain_model.eta(gamma_plot))
    self.save_current_figure('$\gamma$', '$\eta(\gamma)$', 'eta.png')

    # Plot the nondimensional equilibrium distribution
    fig = plt.figure()
    plt.plot(gamma_plot, single_chain_model.P_A_eq(gamma_plot), label = 'original')
    if J_sw is not None:
      plt.plot(gamma_plot, single_chain_model.P_A_eq(gamma_plot/J_sw**(1/3))/J_sw, label = 'swollen')
      plt.legend()
    self.save_current_figure('$\gamma$', r'$\mathscr{P}_A^{eq}(\gamma)$', 'P_A_eq.png')

    # Plot the nondimensional equilibrium radial distribution function
    fig = plt.figure()
    plt.plot(gamma_plot, single_chain_model.g_A_eq(gamma_plot), label = 'original')
    if J_sw is not None:
      plt.plot(gamma_plot, single_chain_model.g_A_eq(gamma_plot/J_sw**(1/3))/J_sw**(1/3), label = 'swollen')
      plt.legend()
    self.save_current_figure('$\gamma$', r'$\mathscr{g}_A^{eq}(\gamma)$', 'g_A_eq.png')

    # Plot the reaction rate coefficient function
    if np.all(np.isclose(single_chain_model.k(gamma_plot[:-2]), single_chain_model.k_0)) == False:
      fig = plt.figure()
      plt.plot(gamma_plot, single_chain_model.k(gamma_plot))
      plt.yscale('log')
      self.save_current_figure('$\gamma$', '$k(\gamma)$ [1/seconds]', 'k.png')

    # Save values to .csv if opted
    if csv_directory is not None:
```

```
    y_tuple = single_chain_model.eta(gamma_plot), single_chain_model.P_A_eq(gamma_plot), \
      single_chain_model.g_A_eq(gamma_plot), single_chain_model.k(gamma_plot)
    self.save_csv(csv_directory + 'single_chain.csv', gamma_plot, y_tuple)

##########################################################################################################################
# Function to save plot values to .csv
##########################################################################################################################

def save_csv(self, csv_file, x, y_tuple):

  # Initialization also clears any previous results
  open(csv_file, "w").close()

  # Save values
  f = open(csv_file, "a")
  for index in range(len(x)):
    f.write("%.8e\t" % x[index])
    if isinstance(y_tuple, tuple):
      for y in y_tuple:
        f.write("%.8e\t" % y[index])
    else:
      f.write("%.8e\t" % y_tuple[index])
    f.write("\n")
  f.close()

##########################################################################################################################
# Function for plotting results
##########################################################################################################################

def plot_results(self, deformation_object, results, use_nominal = False, n_over_beta = None, \
  data_F_stress = None, F_stress_1 = None, F_stress_2 = None, F_stress_3 = None):

  # Retrieve results
  t = results[0]
  F = results[1]
  beta_sigma_over_n = results[5]

  # Simpler plotting if given None as the deformation_object
  if deformation_object is not None:

    # Plot the component of the stress solution if ignoring chain breaking
    if deformation_object.ignore_yield is True:

      # Plot the homogeneous and particular solutions for the stress
      beta_sigma_h_over_n = results[3]
      beta_sigma_p_over_n = results[4]
      fig = plt.figure()
      plt.plot(F, beta_sigma_h_over_n, label = 'homogeneous')
      plt.plot(F, beta_sigma_p_over_n, label = 'particular')
      plt.plot(F, beta_sigma_over_n, label = 'total')
      plt.legend()
      self.save_current_figure('$F(t)$', r'$\beta\sigma(t)/n$', 'sigma_hpt(F).png')

      fig = plt.figure()
      plt.plot(t, beta_sigma_h_over_n, label = 'homogeneous')
      plt.plot(t, beta_sigma_p_over_n, label = 'particular')
      plt.plot(t, beta_sigma_over_n, label = 'total')
      plt.legend()
      self.save_current_figure('$t$ [seconds]', r'$\beta\sigma(t)/n$', 'sigma_hpt(t).png')

    # Typical plotting
    else:

      # Plot total probability of intact chains
      P_A_tot = results[2]
      fig = plt.figure()
      plt.plot(F, deformation_object.initialized_single_chain_model.P_A_tot_eq + 0*t, linestyle = 'dashed')
      plt.plot(F, P_A_tot)
      self.save_current_figure('$F(t)$', '$P_\mathrm{A}^\mathrm{tot}(t)$', 'P_A_tot(F).png')

      fig = plt.figure()
      plt.plot(t, deformation_object.initialized_single_chain_model.P_A_tot_eq + 0*t, linestyle = 'dashed')
      plt.plot(t, P_A_tot)
      self.save_current_figure('$t$ [seconds]', '$P_\mathrm{A}^\mathrm{tot}(t)$', 'P_A_tot(t).png')

      # Plot total rate of breaking and reforming chains
      total_rate_break = results[3]
      total_rate_reform = results[4]
      fig = plt.figure()
      plt.plot(F, total_rate_break, label = 'breaking rate')
      plt.plot(F, total_rate_reform, label = 'reforming rate')
      plt.plot(F, total_rate_reform + total_rate_break, label = 'net rate')
      plt.legend()
      self.save_current_figure('$F(t)$', r'$\frac{d}{dt}\,P_\mathrm{A}^\mathrm{tot}(t)$', 'd_P_A_tot_dt(F).png')

      fig = plt.figure()
      plt.plot(t, total_rate_break, label = 'breaking rate')
      plt.plot(t, total_rate_reform, label = 'reforming rate')
      plt.plot(t, total_rate_reform + total_rate_break, label = 'net rate')
      plt.legend()
      self.save_current_figure('$t$ [seconds]', \
        r'$\frac{d}{dt}\,P_\mathrm{A}^\mathrm{tot}(t)$', 'd_P_A_tot_dt(t).png')
```

```
    # Plot the stress
    fig = plt.figure()
    if n_over_beta is None:
      n_over_beta = 1
      y_label = r'$\beta\sigma(t)/n$'
    else:
      y_label = r'$\sigma(t)$'

    if use_nominal is True:
      y_label = 'nominal ' + y_label
      stress = n_over_beta*beta_sigma_over_n/F
    else:
      stress = n_over_beta*beta_sigma_over_n

    # Plot the stress data
    if data_F_stress is not None:
      plt.plot(data_F_stress[0], data_F_stress[1], 'o')

    # Plot any extra stress
    if F_stress_1 is not None:
      if use_nominal is True:
        sigma = F_stress_1[1]/F_stress_1[0]
      else:
        sigma = F_stress_1[1]
      plt.plot(F_stress_1[0], sigma, '--')
    if F_stress_2 is not None:
      if use_nominal is True:
        sigma = F_stress_2[1]/F_stress_2[0]
      else:
        sigma = F_stress_2[1]
      plt.plot(F_stress_2[0], sigma, '--')
    if F_stress_3 is not None:
      if use_nominal is True:
        sigma = F_stress_3[1]/F_stress_3[0]
      else:
        sigma = F_stress_3[1]
      plt.plot(F_stress_3[0], sigma, '--')

    # Plot the stress as a function of deformation
    plt.plot(F, stress)
    self.save_current_figure('$F(t)$', y_label, 'sigma(F).png')

    # Plot the stress as a function of time
    if deformation_object is not None:
      fig = plt.figure()
      plt.plot(t, stress)
      self.save_current_figure('$t$ [seconds]', y_label, 'sigma(t).png')
```

C.6.2 Simple Example

This subsection details the following example script demonstrating basic usage of this package:

```
# Import the library
from chain_breaking_polymer_networks import *

# Create the single-chain model
single_chain_model = Morse_FJC(N_b = 1, N_b_H = 8, kappa = 2e2, kappa_H = 5e2, beta_u_b = 1e2, k_0 = 1e-2, beta_Delta_Psi_0 = 2)

# Plot the single-chain model
plotter_object = plotter()
plotter_object.plot_single_chain(single_chain_model)

# Define the deformation
strain_rate = 1
maximum_strain = 8
total_time_in_seconds = 2*maximum_strain/strain_rate
def F(t):
  return 1 + strain_rate*t*np.heaviside(maximum_strain - strain_rate*t, 0.5) \
    + (2*maximum_strain - strain_rate*t)*np.heaviside(strain_rate*t - maximum_strain, 0.5)

# Plot the deformation
plotter_object.plot_deformation(F, total_time_in_seconds)

# Apply the deformation and solve
network_model = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model, num_grid_suggestion = 513)
results = network_model.solve(csv_directory = './')

# Plot the results
plotter_object.plot_results(network_model, results)
```

After importing the package, executing the line

```
single_chain_model = Morse_FJC(N_b =1, N_b_H = 8, kappa = 2e2, \
    kappa_H = 5e2, beta_u_b = 1e2, k_0 = 1e-2, beta_Delta_Psi_0 = 2)
```

we create the Morse-FJC single-chain model consisting of:

- 1 breakable link with nondimensional stiffness 200 and energy 100,
- 8 unbreakable links with nondimensional stiffness 500,
- an initial reaction rate coefficient of 0.01/s, and
- a nondimensional free energy change of 2 when breaking.

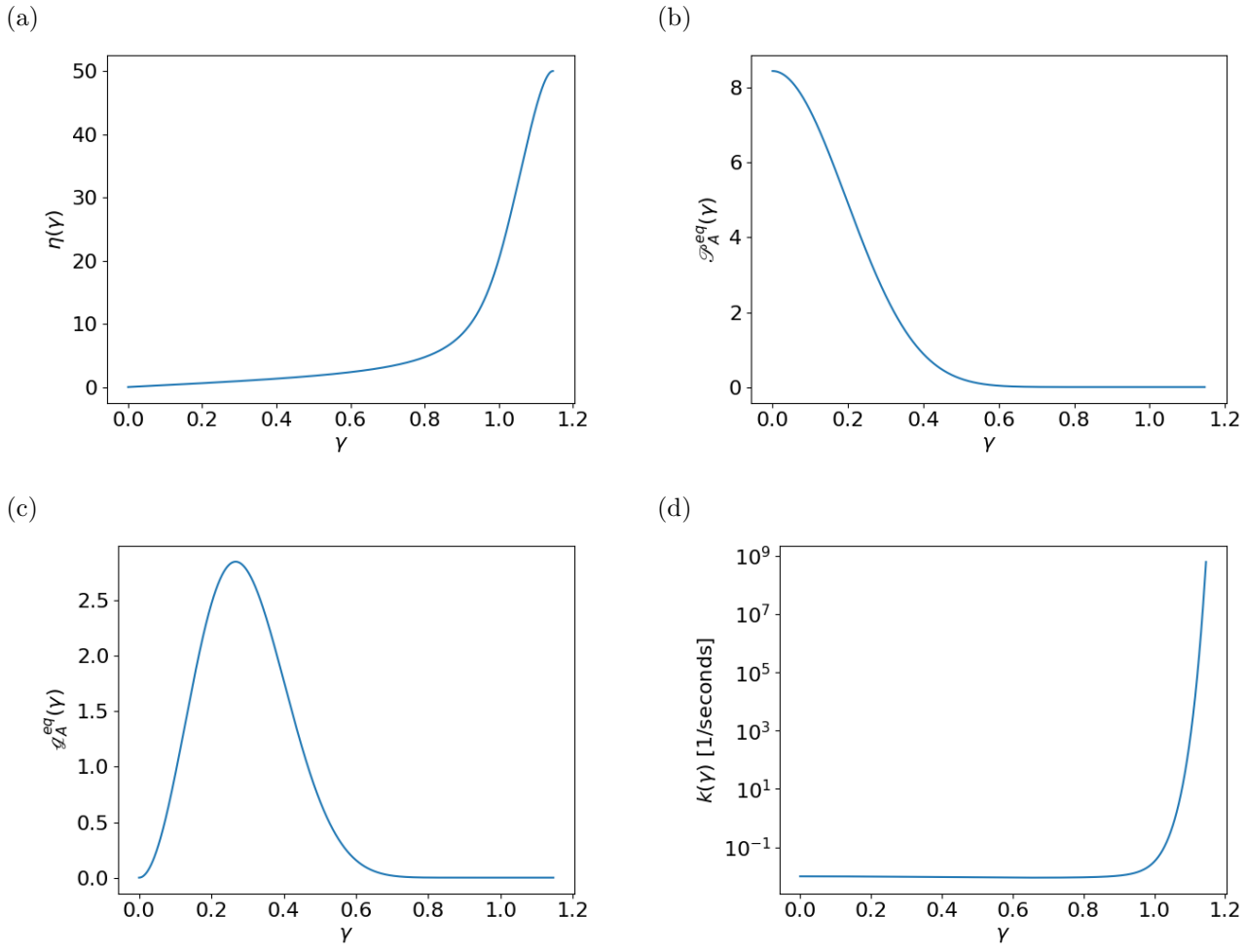


Figure C.4: Plots of single-chain functions generated by the `plot_single_chain` attribute of the `plotter` object. (a) the nondimensional single-chain mechanical response $\eta(\gamma)$, (b) the nondimensional equilibrium distribution of intact chains $\mathscr{P}_{\mathrm{A}}(\gamma)$, (c) the nondimensional equilibrium radial distribution of intact chains $4\pi\gamma^2\mathscr{P}_{\mathrm{A}}(\gamma)$, and (d) the net forward reaction rate coefficient function $k(\gamma)$, each as a function of the nondimensional end-to-end length γ.

The plotting object is then created and used to plot (saving the images, shown in Fig. C.4, in the local directory) each relevant single-chain function via:

```
plotter_object = plotter()
plotter_object.plot_single_chain(single_chain_model)
```

Choosing uniaxial stress (default in `deform_network`), the applied stretch is specified as a function of time here:

```
strain_rate = 1
maximum_strain = 8
total_time_in_seconds = 2*maximum_strain/strain_rate
def F(t):
    return 1 + strain_rate*t*np.heaviside(maximum_strain - strain_rate*t, 0.5)\
        + (2*maximum_strain - strain_rate*t)* \
           np.heaviside(strain_rate*t - maximum_strain, 0.5)
```

This corresponds to a normalized strain rate until a stretch of 9 is reached, and subsequently reversing the rate until at a stretch of unity. Using the same plotting object, this applied stretch is plotted (see Fig. C.5) as a function of time:

```
plotter_object.plot_deformation(F, total_time_in_seconds)
```

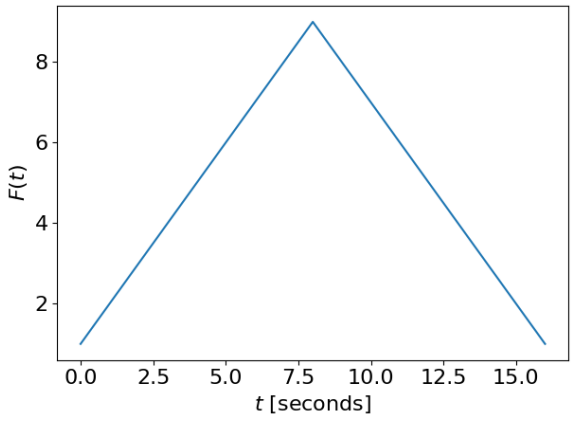


Figure C.5: The applied deformation F as a function of time t from `plotter.plot_deformation`.

The model for the network is created by simultaneously specifying the applied stretch and traction boundary condition, the total time, the single-chain model, and increasing the suggestion for the number of spatial grid points in each direction:

```
network_model = deform_network(F, 'uniaxial', total_time_in_seconds, \
    single_chain_model, num_grid_suggestion = 513)
```

The network model is solved and results are returned as a tuple, while also writing these results to a .csv file in the local directory:

```
results = network_model.solve(csv_directory ='./')
```

Using the same plotting object, these results (stress, total intact-chain probability, total breaking/reforming/net rate) are plotted (see Fig. C.6):

```
plotter_object.plot_results(network_model, results)
```

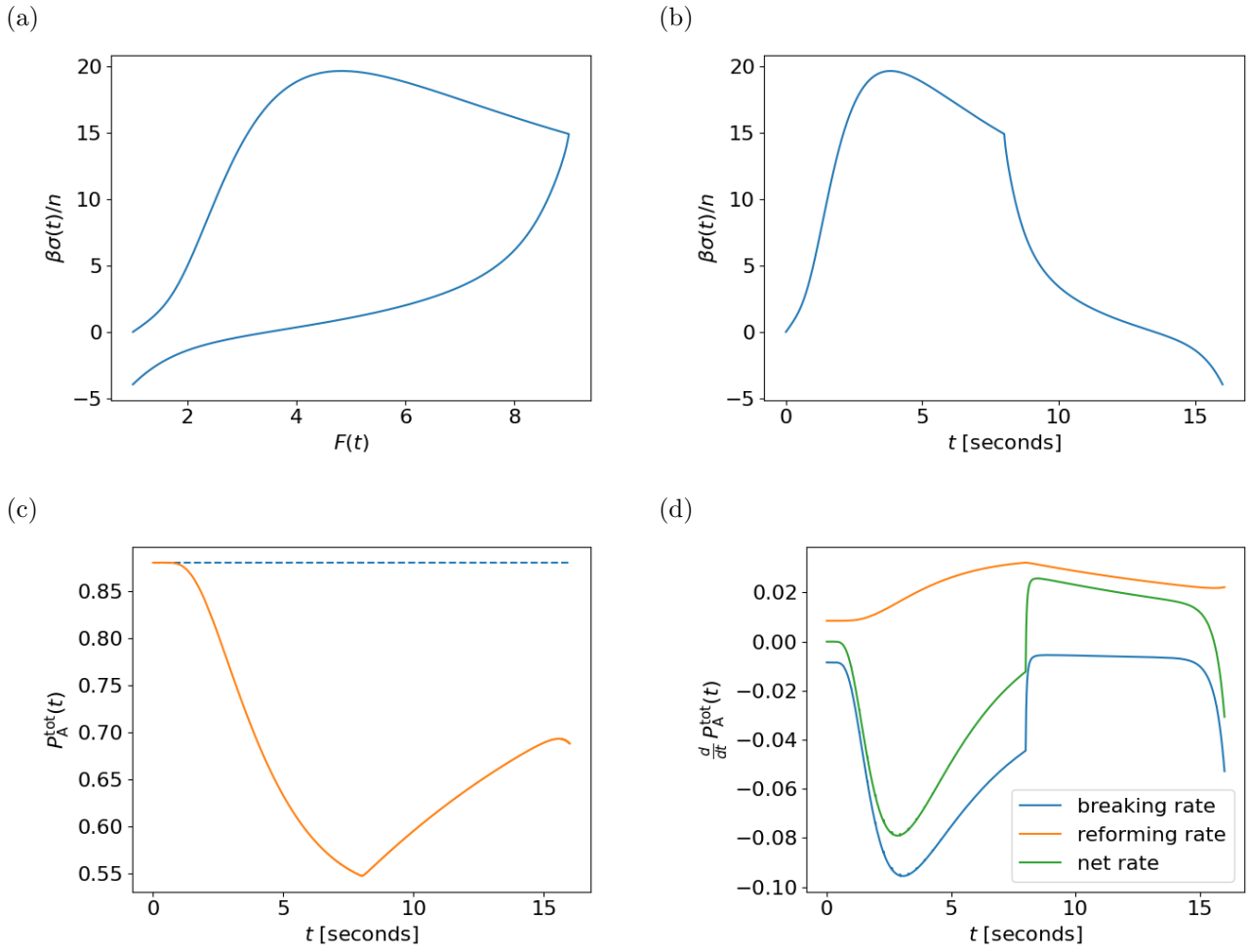


Figure C.6: Plots of the various results generated by the `plotter` object after solving the network model. (a) The nondimensional stress $\beta\sigma/n$ as a function of the deformation F and (b) of the time t, (c) the intact chain fraction $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ as a function of t, and (d) the total breaking (forward), reforming (reverse), and net (forward + reverse) rate of change of $P_{\mathrm{A}}^{\mathrm{tot}}(t)$ as a function of t.

C.6.3 Archived scripts

Here we list each `Python` script used to obtain the macroscopic theoretical results in Figs. 5.6–5.11. Below is the script corresponding to Fig. 5.6:

```
# Import the package
from chain_breaking_polymer_networks import *

# Create the single-chain model
single_chain_model = Morse_FJC(N_b = 1, N_b_H = 38, beta_u_b = 61.57, kappa = 9000,
  kappa_H = 6000, k_0 = 1e-88, beta_Delta_Psi_0 = 5, varsigma = 4)

# Volumetric swelling ratio
J_sw = 15.625

# Plot the single-chain model
output_directory = 'output_TN/'
plotter_object = plotter(plot_directory = output_directory)
plotter_object.plot_single_chain(single_chain_model, J_sw = J_sw, csv_directory = output_directory)

# Define the deformation
lambda_dot = 0.025
total_time_in_seconds = 15/lambda_dot
def F(t):
  return 1 + lambda_dot*t*np.heaviside(0.5 - lambda_dot*t, 0.5) \
    + (1 - lambda_dot*t)*np.heaviside(lambda_dot*t - 0.5, 0.5)*np.heaviside(1 - lambda_dot*t, 0.5) \
    + (lambda_dot*t - 1)*np.heaviside(lambda_dot*t - 1, 0.5)*np.heaviside(2 - lambda_dot*t, 0.5) \
    + (3 - lambda_dot*t)*np.heaviside(lambda_dot*t - 2, 0.5)*np.heaviside(3 - lambda_dot*t, 0.5) \
    + (lambda_dot*t - 3)*np.heaviside(lambda_dot*t - 3, 0.5)*np.heaviside(4.5 - lambda_dot*t, 0.5) \
    + (6 - lambda_dot*t)*np.heaviside(lambda_dot*t - 4.5, 0.5)*np.heaviside(6 - lambda_dot*t, 0.5) \
    + (lambda_dot*t - 6)*np.heaviside(lambda_dot*t - 6, 0.5)*np.heaviside(8 - lambda_dot*t, 0.5) \
    + (10 - lambda_dot*t)*np.heaviside(lambda_dot*t - 8, 0.5)*np.heaviside(10 - lambda_dot*t, 0.5) \
    + (lambda_dot*t - 10)*np.heaviside(lambda_dot*t - 10, 0.5)*np.heaviside(12.5 - lambda_dot*t, 0.5) \
    + (15 - lambda_dot*t)*np.heaviside(lambda_dot*t - 12.5, 0.5)*np.heaviside(15 - lambda_dot*t, 0.5)

# Plot the deformation
plotter_object.plot_deformation(F, total_time_in_seconds, csv_directory = output_directory)

# Apply the deformation and solve
deformation_object = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model, J_sw = J_sw)
results = deformation_object.solve(csv_directory = output_directory)

# Combine with a Neo-Hookean model for the filler network
E = 1.5
n_over_beta_1 = 0.2
n_over_beta_23 = E/3 - n_over_beta_1
F = results[1]
sigma_1 = n_over_beta_1/single_chain_model.P_A_tot_eq*results[5]
F_stress_23 = F, n_over_beta_23*(F**2 - 1/F)
F_stress_tot = F, F_stress_23[1] + sigma_1

# Plot the results with the experimental data
filename = 'data/ducrot_fig4c_stress.csv'
data_F_stress = np.genfromtxt(filename, usecols = (0)), np.genfromtxt(filename, usecols = (1))
plotter_object.plot_results(deformation_object, results, n_over_beta = n_over_beta_1, \
  data_F_stress = data_F_stress, F_stress_1 = F_stress_23, F_stress_2 = F_stress_tot)
```

Below is the script corresponding to the transient model in Fig. 5.7(a):

```
# Import the package
from chain_breaking_polymer_networks import *

# Create the single-chain model
single_chain_model = ideal(k_0 = 0.37)

# Plot the single-chain model
output_directory = 'output_Long_transient/'
plotter_object = plotter(plot_directory = output_directory)
plotter_object.plot_single_chain(single_chain_model)

# Define the deformation
lambda_dot = 0.03
Delta_lambda = 1
total_time_in_seconds = 2*Delta_lambda/lambda_dot
def F(t):
  return 1 + lambda_dot*t*np.heaviside(Delta_lambda/lambda_dot - t, 0.5) \
    + (2*Delta_lambda - lambda_dot*t)*np.heaviside(t - Delta_lambda/lambda_dot, 0.5)

# Plot the deformation
plotter_object.plot_deformation(F, total_time_in_seconds)

# Apply the deformation and solve
deformation_object = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model, \
  max_F_dot = lambda_dot, use_spatial_grid = False, ignore_yield = True)
results = deformation_object.solve(csv_directory = output_directory)

# Post-processing
n_over_beta = 24.15
```

```
fraction = 0.16
F = results[1]
sigma_1 = n_over_beta*(1 - fraction)*results[5]
sigma_2 = n_over_beta*fraction*(F**2 - 1/F)
sigma_tot = sigma_1 + sigma_2
F_stress_2 = F, sigma_2
F_stress_tot = F, sigma_tot

# Plot the results with the experimental data
data_F_stress = np.genfromtxt('data/long2014_2e.csv', usecols = (0)), np.genfromtxt('data/long2014_2e.csv', usecols = (1))
plotter_object.plot_results(deformation_object, results, n_over_beta = n_over_beta, use_nominal = True, \
  data_F_stress = data_F_stress, F_stress_1 = F_stress_tot, F_stress_2 = F_stress_2)
```

Below is the script corresponding to the adjusted model in Fig. 5.7(a):

```
# Import the package
from chain_breaking_polymer_networks import *

# Create the single-chain model
single_chain_model = ideal()

# Create the relaxation function
relaxation_function = Long_et_al_2014(alpha = 2.6, t_R = 0.6, x_p = 0.1)

# Plot the single-chain model
output_directory = 'output_Long_relax/'
plotter_object = plotter(plot_directory = output_directory)
plotter_object.plot_single_chain(single_chain_model)

# Define the deformation
Delta_lambda = 1
lambda_dot = 0.03
total_time_in_seconds = 2*Delta_lambda/lambda_dot
def F(t):
  return 1 + lambda_dot*t*np.heaviside(Delta_lambda/lambda_dot - t, 0.5) \
    + (2*Delta_lambda - lambda_dot*t)*np.heaviside(t - Delta_lambda/lambda_dot, 0.5)

# Plot the deformation
plotter_object.plot_deformation(F, total_time_in_seconds)

# Apply the deformation and solve
deformation_object = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model, \
  relaxation_function = relaxation_function, max_F_dot = lambda_dot, use_spatial_grid = False, ignore_yield = True)
results = deformation_object.solve(csv_directory = output_directory)

# Plot the results with the experimental data
data_F_stress = np.genfromtxt('data/long2014_2e.csv', usecols = (0)), np.genfromtxt('data/long2014_2e.csv', usecols = (1))
plotter_object.plot_results(deformation_object, results, n_over_beta = 24.15, use_nominal = True, data_F_stress = data_F_stress)
```

Below is the script corresponding to the transient model in Fig. 5.7(b):

```
# Import the package
from chain_breaking_polymer_networks import *

# Create the single-chain models
x_p = 5.85e-2
single_chain_model_1 = EFJC(N_b = 50, kappa = 40, k_0 = 0.12)
single_chain_model_2 = EFJC(N_b = 50, kappa = 40)

# Plot the single-chain model
output_directory = 'output_PVA_transient/'
plotter_object = plotter(plot_directory = output_directory)
plotter_object.plot_single_chain(single_chain_model_1)

# Loop over all deformation rates
order = [5, 1, 4, 0, 3, 2]
lambda_dot_all   = [0.90, 0.10, 0.03, 0.01, 3e-3, 1e-4]
Delta_lambda_all = [0.65, 1.55, 2.66, 2.90, 3.90, 4.56]
for index in order:

  # Define the deformation
  lambda_dot = lambda_dot_all[index]
  Delta_lambda = Delta_lambda_all[index]
  total_time_in_seconds = Delta_lambda/lambda_dot
  def F(t):
    return 1 + lambda_dot*t

  # Take larger timesteps for very slow deformation rates
  if lambda_dot == 0.003:
    nondim_dt = 1e0
  elif lambda_dot == 0.0001:
    nondim_dt = 5e1
  else:
    nondim_dt = 1e-2

  # Output file names
  output_directory_index = output_directory + str(lambda_dot) + '_'
  plotter_object = plotter(plot_directory = output_directory_index)
```

```
  # Apply the deformation and solve
  results_1 = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model_1, \
    max_F_dot = lambda_dot, use_spatial_grid = False, ignore_yield = True, \
    nondimensional_timestep_suggestion = nondim_dt).solve(csv_directory = output_directory_index + '1_')
  results_2 = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model_2, \
    max_F_dot = lambda_dot, use_spatial_grid = False, ignore_yield = True, \
    ).solve(csv_directory = output_directory_index + '2_')

  # Combine the results
  results_2_5th_interp = np.interp(results_1[1], results_2[1], results_2[5])
  results = results_1[0], results_1[1], results_1[2], results_1[3], results_1[4], \
    x_p*results_2_5th_interp + (1 - x_p)*results_1[5]

  # Plot the results with the experimental data
  filename = 'data/mayumi_3a_0p' + str(lambda_dot)[2:] + '.csv'
  data_F_stress = np.genfromtxt(filename, usecols = (0)), np.genfromtxt(filename, usecols = (1))
  plotter_object.plot_results(None, results, n_over_beta = 37.78, use_nominal = True, data_F_stress = data_F_stress)
```

Below is the script corresponding to the adjusted model in Fig. 5.7(b):

```
# Import the package
from chain_breaking_polymer_networks import *

# Create the single-chain model
single_chain_model = EFJC(N_b = 50, kappa = 40)

# Create the relaxation function
relaxation_function = Long_et_al_2014(alpha = 3.23, t_R = 1.99, x_p = 4.68e-2)

# Plot the single-chain model
output_directory = 'output_PVA/'
plotter_object = plotter(plot_directory = output_directory)
plotter_object.plot_single_chain(single_chain_model)

# Loop over all deformation rates
order = [4, 0, 1, 2, 3, 5]
lambda_dot_all   = [0.90, 0.10, 0.03, 0.01, 3e-3, 1e-4]
Delta_lambda_all = [0.65, 1.55, 2.66, 2.90, 3.90, 4.56]
for index in order:

  # Define the deformation
  lambda_dot = lambda_dot_all[index]
  Delta_lambda = Delta_lambda_all[index]
  total_time_in_seconds = Delta_lambda/lambda_dot
  def F(t):
    return 1 + lambda_dot*t

  # Take larger timesteps for very slow deformation rates
  if lambda_dot == 0.003:
    nondim_dt = 1e0
  elif lambda_dot == 0.0001:
    nondim_dt = 5e1
  else:
    nondim_dt = 1e-2

  # Output file names
  output_directory_index = output_directory + str(lambda_dot) + '_'
  plotter_object = plotter(plot_directory = output_directory_index)

  # Apply the deformation and solve
  deformation_object = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model, \
    relaxation_function = relaxation_function, max_F_dot = lambda_dot, use_spatial_grid = False, ignore_yield = True, \
    nondimensional_timestep_suggestion = nondim_dt)
  results = deformation_object.solve(csv_directory = output_directory_index)

  # Plot the results with the experimental data
  filename = 'data/mayumi_3a_0p' + str(lambda_dot)[2:] + '.csv'
  data_F_stress = np.genfromtxt(filename, usecols = (0)), np.genfromtxt(filename, usecols = (1))
  plotter_object.plot_results(None, results, n_over_beta = 37.78, use_nominal = True, data_F_stress = data_F_stress)
```

Below is the script corresponding to the general model in Fig. 5.8:

```
# Import the package
from chain_breaking_polymer_networks import *

# Create the single-chain model
single_chain_model = Morse_FJC(N_b = 1, N_b_H = 8, kappa = 200, kappa_H = 500, \
  beta_u_b = 100, k_0 = 1e-2, varsigma = 1, beta_Delta_Psi_0 = 5)

# Plot the single-chain model
output_directory = 'output_11/'
plotter_object = plotter(plot_directory = output_directory)
plotter_object.plot_single_chain(single_chain_model)

# Define the deformation
Delta = 2
total_time_in_seconds = 8*Delta
def F(t):
  return 1 + t*np.heaviside(Delta - t, 0.5) + Delta*np.heaviside(t - Delta, 0.5) \
```

```
    + (t - 2*Delta)*np.heaviside(t - 2*Delta, 0.5)*np.heaviside(3*Delta - t, 0.5) + Delta*np.heaviside(t - 3*Delta, 0.5) \
    + (4*Delta - t)*np.heaviside(t - 4*Delta, 0.5)*np.heaviside(5*Delta - t, 0.5) - Delta*np.heaviside(t - 5*Delta, 0.5) \
    + (6*Delta - t)*np.heaviside(t - 6*Delta, 0.5)*np.heaviside(7*Delta - t, 0.5) - Delta*np.heaviside(t - 7*Delta, 0.5)

# Plot the deformation
plotter_object.plot_deformation(F, total_time_in_seconds)

# Apply the deformation and solve
deformation_object = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model, num_grid_suggestion = 1025)
results = deformation_object.solve(csv_directory = output_directory, checkpoint_directory = output_directory)

# Plot the results
plotter_object.plot_results(deformation_object, results)
```

Below is the script corresponding to the transient model (with critical extension) in Fig. 5.8:

```
# Import the package
from chain_breaking_polymer_networks import *

# Create the single-chain model
single_chain_model = Morse_FJC(N_b = 1, N_b_H = 8, kappa = 200, kappa_H = 500, \
  beta_u_b = 100, k_0 = 1e-2, varsigma = 1, beta_Delta_Psi_0 = 5, gamma_c = 1.1459)

# Plot the single-chain model
output_directory = 'output_12/'
plotter_object = plotter(plot_directory = output_directory)
plotter_object.plot_single_chain(single_chain_model)

# Define the deformation
Delta = 2
total_time_in_seconds = 8*Delta
def F(t):
  return 1 + t*np.heaviside(Delta - t, 0.5) + Delta*np.heaviside(t - Delta, 0.5) \
    + (t - 2*Delta)*np.heaviside(t - 2*Delta, 0.5)*np.heaviside(3*Delta - t, 0.5) + Delta*np.heaviside(t - 3*Delta, 0.5) \
    + (4*Delta - t)*np.heaviside(t - 4*Delta, 0.5)*np.heaviside(5*Delta - t, 0.5) - Delta*np.heaviside(t - 5*Delta, 0.5) \
    + (6*Delta - t)*np.heaviside(t - 6*Delta, 0.5)*np.heaviside(7*Delta - t, 0.5) - Delta*np.heaviside(t - 7*Delta, 0.5)

# Plot the deformation
plotter_object.plot_deformation(F, total_time_in_seconds)

# Apply the deformation and solve
deformation_object = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model, num_grid_suggestion = 1025)
results = deformation_object.solve(csv_directory = output_directory, checkpoint_directory = output_directory)

# Plot the results
plotter_object.plot_results(deformation_object, results)
```

Below is the script corresponding to the rate-independent irreversible model in Fig. 5.8:

```
# Import the package
from chain_breaking_polymer_networks import *

# Create the single-chain model
single_chain_model = Morse_FJC(N_b = 1, N_b_H = 8, kappa = 200, kappa_H = 500, \
  beta_u_b = 100, k_0 = 1e-88, varsigma = 1, beta_Delta_Psi_0 = 5, gamma_c = 1.1459)

# Plot the single-chain model
output_directory = 'output_13/'
plotter_object = plotter(plot_directory = output_directory)
plotter_object.plot_single_chain(single_chain_model)

# Define the deformation
Delta = 2
total_time_in_seconds = 8*Delta
def F(t):
  return 1 + t*np.heaviside(Delta - t, 0.5) + Delta*np.heaviside(t - Delta, 0.5) \
    + (t - 2*Delta)*np.heaviside(t - 2*Delta, 0.5)*np.heaviside(3*Delta - t, 0.5) + Delta*np.heaviside(t - 3*Delta, 0.5) \
    + (4*Delta - t)*np.heaviside(t - 4*Delta, 0.5)*np.heaviside(5*Delta - t, 0.5) - Delta*np.heaviside(t - 5*Delta, 0.5) \
    + (6*Delta - t)*np.heaviside(t - 6*Delta, 0.5)*np.heaviside(7*Delta - t, 0.5) - Delta*np.heaviside(t - 7*Delta, 0.5)

# Plot the deformation
plotter_object.plot_deformation(F, total_time_in_seconds)

# Apply the deformation and solve
deformation_object = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model, num_grid_suggestion = 1025)
results = deformation_object.solve(csv_directory = output_directory, checkpoint_directory = output_directory)

# Plot the results
plotter_object.plot_results(deformation_object, results)
```

Below is the base script corresponding to the parameter studies in Figs. 5.9 and 5.10:

```
# Import the package
from chain_breaking_polymer_networks import *

# Create the single-chain model
single_chain_model = Morse_FJC(N_b = 1, N_b_H = 8, kappa = 200, kappa_H = 500, \
  beta_u_b = 100, k_0 = 1e-2, varsigma = 1, beta_Delta_Psi_0 = 5)
```

```
# Plot the single-chain model
output_directory = 'output_18/100_'
plotter_object = plotter(plot_directory = output_directory)
plotter_object.plot_single_chain(single_chain_model)

# Define the deformation
Delta = 8
total_time_in_seconds = 2*Delta
def F(t):
  return 1 + t*np.heaviside(Delta - t, 0.5) + (2*Delta - t)*np.heaviside(t - Delta, 0.5)

# Plot the deformation
plotter_object.plot_deformation(F, total_time_in_seconds)

# Apply the deformation and solve
deformation_object = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model, num_grid_suggestion = 1025)
results = deformation_object.solve(csv_directory = output_directory)

# Plot the results
plotter_object.plot_results(deformation_object, results)
```

Below is the script corresponding to Fig. 5.11:

```
# Import the package
from chain_breaking_polymer_networks import *

# Create the single-chain model
single_chain_model = Morse_FJC(N_b = 1, N_b_H = 9, kappa = 200, kappa_H = 400, \
  beta_u_b = 100, k_0 = 2e-4, beta_Delta_Psi_0 = 8.55)

# Plot the single-chain model
output_directory = 'output_Cu/1.5_'
plotter_object = plotter(plot_directory = output_directory)
plotter_object.plot_single_chain(single_chain_model)

# Define the deformation
Delta = 1.5
lambda_dot = 8.4/60
total_time_in_seconds = 2*Delta/lambda_dot
def F(t):
  return 1 + lambda_dot*t*np.heaviside(total_time_in_seconds/2 - t, 0.5) \
    + (2*Delta - lambda_dot*t)*np.heaviside(t - total_time_in_seconds/2, 0.5)

# Plot the deformation
plotter_object.plot_deformation(F, total_time_in_seconds)

# Apply the deformation and solve
deformation_object = deform_network(F, 'uniaxial', total_time_in_seconds, single_chain_model, max_F_dot = lambda_dot, \
  nondimensional_timestep_suggestion = 1e-1)
results = deformation_object.solve(csv_directory = output_directory)

# Plot the results with the experimental data
filename = 'data/lin_5b_all.csv'
data_F_stress = np.genfromtxt(filename, usecols = (0)), np.genfromtxt(filename, usecols = (1))
plotter_object.plot_results(deformation_object, results, \
  n_over_beta = 0.96/2, use_nominal = True, data_F_stress = data_F_stress)
```

C.7 Specialized `Python` package

In order to allow theoretical modeling of the rotation and reloading of the mechanophore-containing double network samples in the last part of Sec. C.5.2 (see Figs. C.2 and C.3), a specialized `Python` implementation of the model was created. In this specialized implementation, symmetry is neither used to compute spatial integrals nor is it used to calculate the unknown components of the deformation gradient. Instead, it employs a solver to iteratively compute the stress given the past deformation history and vary the unknown deformation gradient components until the traction boundary conditions are satisfied. It is currently only applicable when using the EFJC model and rate-independent irreversible breaking, in addition to the deformation gradient existing in principal coordinates. In the future it will be fully generalized, although certain cases will quickly become computationally prohibitive, so the original implementation is still highly recommended.

The Python module file single_chain.py for the specialized implementation is printed below:

```
################################################################################################################################################
# General setup
################################################################################################################################################

# Import libraries
import sys
import numpy as np
from scipy.integrate import quad
from scipy.interpolate import interp1d
from scipy.optimize import minimize_scalar

# Interpolation parameters
num_interp = int(3e3)
interp_kind_1D = 'cubic'

# Numerical tolerance parameters
cutoff_for_log_over_sinh = 3e1
cutoff_stretch_for_harmonic_eta_EFJC = 3
minimum_exponent = np.log(sys.float_info.min)/np.log(10)
maximum_exponent = np.log(sys.float_info.max)/np.log(10)
eta_small = 10**minimum_exponent

# Function to invert a function
def inv_fun_1D(x_query, fun, bounds = None):

  # Change method depending on whether bounds are involved
  if bounds is None:
    return minimize_scalar(lambda x: np.abs(fun(x) - x_query)).x
  else:
    return minimize_scalar(lambda x: np.abs(fun(x) - x_query), bounds = bounds, method = 'bounded').x

# Function to create interpolation function from stored function
def interp_fun_1D(x_store, y_store):
  return interp1d(x_store, y_store, kind = interp_kind_1D, bounds_error = False, fill_value = np.nan)

# Function to avoid overflow when computing ln(x/sinh(x))
def log_over_sinh(x):

  # Determine when argument is sufficiently large
  where_x_large = np.nan_to_num(x, nan = -1) > cutoff_for_log_over_sinh
  log_of_x_over_sinh_x = np.zeros(x.shape)

  # Use asymptotic relation valid for sufficiently large arguments
  if where_x_large.any():
    log_of_x_over_sinh_x[where_x_large] = np.log(2*x[where_x_large]) - x[where_x_large]

  # Compute analytically otherwise, and zero where argument is zero
  where_x_zero = x == 0
  where_compute = ~(where_x_large + where_x_zero)
  if where_compute.any():
    log_of_x_over_sinh_x[where_compute] = np.log(x[where_compute]/np.sinh(x[where_compute]))
  return log_of_x_over_sinh_x

# Hyperbolic cotangent function
def coth_safe(eta):
  eta = np.where(eta == 0, eta_small, eta)
  return 1/np.tanh(eta)

# Langevin function
def Langevin(eta):
  eta = np.where(eta == 0, eta_small, eta)
  return 1/np.tanh(eta) - 1/eta

################################################################################################################################################
# Extensible freely-joined chain model
################################################################################################################################################

class EFJC:

  # For more information, see:
  #   Analytical results of the extensible freely jointed chain model
  #   Alessandro Fiasconaro and Fernando Falo
  # Physica A 2019, 532, 121929
  #   doi.org/10.1016/j.physa.2019.121929
  # See also:
  # Statistical mechanical constitutive theory of polymer networks:
  #   The inextricable links between distribution, behavior, and ensemble
  #   Michael R. Buche and Meredith N. Silberstein
  #   Physical Review E, 2021, 102, 012501
  #   doi.org/10.1103/PhysRevE.102.012501

  # Class initialization
  def __init__(self, **kwargs):

    # Default parameter values
    N_b = None
    kappa = None
    k_0 = np.exp(minimum_exponent)
    gamma_c = np.inf
```

```
# Retrieve specified parameters
for key, value in kwargs.items():
  if key == 'N_b':
    N_b = value
  elif key == 'k_0':
    if value > 0:
      k_0 = value
  elif key == 'kappa':
    kappa = value
  elif key == 'gamma_c':
    gamma_c = value

# Check parameter specifications
if N_b is None:
  sys.exit('Error: Need to specify N_b.')
elif kappa is None:
  sys.exit('Error: Need to specify kappa.')

# Retain for certain purposes
self.N_b = N_b
self.k_0 = k_0
self.kappa = kappa
self.gamma_c = gamma_c

# Model-specific modifications
self.P_A_tot_eq = 1
self.P_B_tot_eq = 0
self.K_hat = k_0
self.max_k_rev = k_0
self.N_b_H = 0
self.varsigma = 1
self.gamma_TS = self.gamma_c
self.k = lambda gamma_in: k_0 + 0*gamma_in

# Nondimensional mechanical response of the chain
def gamma_fun(eta):
  coth = coth_safe(eta)
  L = Langevin(eta)
  return L + eta/kappa*(1 + (1 - L*coth)/(1 + eta/kappa*coth))

# Compute and store the inverted nondimensional mechanical response to interpolate from
self.gamma_store = np.linspace(0, cutoff_stretch_for_harmonic_eta_EFJC, num_interp)
self.eta_store = np.zeros(self.gamma_store.size)
for i in range(1, len(self.gamma_store)):
  self.eta_store[i] = inv_fun_1D(self.gamma_store[i], gamma_fun)

# Function to interpolate from the inverted nondimensional mechamical response of the chain
self.eta_interp_fun = interp_fun_1D(self.gamma_store, self.eta_store)
def eta_fun(gamma_in):
  if isinstance(gamma_in, np.ndarray):
    eta_out = np.zeros(gamma_in.shape)
    harmonic_region = gamma_in > cutoff_stretch_for_harmonic_eta_EFJC
    eta_out[harmonic_region] = self.kappa*(gamma_in[harmonic_region] - 1)
    eta_out[~harmonic_region] = self.eta_interp_fun(gamma_in[~harmonic_region])
  else:
    if gamma_in > cutoff_stretch_for_harmonic_eta_EFJC:
      eta_out = self.kappa*(gamma_in - 1)
    else:
      eta_out = self.eta_interp_fun(gamma_in)
  return eta_out

# Nondimensional equilibrium distribution function
def P_A_eq_fun(gamma_in, normalization = 1):

  # Compute mechanical response
  eta = np.array(eta_fun(gamma_in))
  eta[eta == 0] = eta_small

  # Compute nondimensional Helmholtz free energy per link
  coth = coth_safe(eta)
  L = Langevin(eta)
  vartheta = eta*L + log_over_sinh(eta) - np.log(1 + eta/kappa*coth) \
    + eta**2/kappa/2*(1/2 + (1 - L*coth)/(1 + eta/kappa*coth))

  # Compute P_A_eq below the yield surface
  return (gamma_in <= self.gamma_c)*np.exp(-N_b*vartheta)/normalization

# Nondimensional equilibrium radial distribution function
def g_A_eq_fun(gamma_in, normalization = 1):
  return 4*np.pi*gamma_in**2*P_A_eq_fun(gamma_in, normalization)

# Normalize the equilibrium distribution
P_A_eq_normalization = quad(g_A_eq_fun, 0, np.inf, full_output = 1)[0]/self.P_A_tot_eq

# Retain each single-chain function
self.eta = eta_fun
self.P_A_eq = lambda gamma_in: P_A_eq_fun(gamma_in, normalization = P_A_eq_normalization)
self.g_A_eq = lambda gamma_in: g_A_eq_fun(gamma_in, normalization = P_A_eq_normalization)
```

The Python module file network.py for the specialized implementation is printed below:

```
##########################################################################################################################################
# General setup
##########################################################################################################################################

# Import libraries
import sys
import copy
import numpy as np
from scipy.integrate import romb
from scipy.optimize import root_scalar

##########################################################################################################################################
# Deformation application class
##########################################################################################################################################

class deform_network_no_symmetry:

  ########################################################################################################################################
  # Initialization
  ########################################################################################################################################

  def __init__(self, single_chain_model, J_sw = 1, num_grid_suggestion = 129):

    # Inherit single_chain_model
    self.scm = single_chain_model

    # Volumetric swelling ratio
    self.J_sw = J_sw

    # Spatial grid octant (F diagonal, i.e. no rotation)
    self.num_grid = self.adjust_for_romb(num_grid_suggestion)
    self.x = np.linspace(0, self.scm.gamma_TS, self.num_grid)
    self.y = self.x
    self.z = self.x
    self.dx = self.x[1] - self.x[0]
    self.dy = self.dx
    self.dz = self.dx
    self.X, self.Y, self.Z = np.meshgrid(self.x, self.y, self.z)
    self.ELL = np.sqrt(self.X*self.X + self.Y*self.Y + self.Z*self.Z)

    # Integration element specialized for stress calculation
    self.ELEMENT_stress = self.element_stress(self.X, self.Y, self.Z, self.scm)

    # Adjust normalization of P_A_eq on the grid
    P_A_eq_ELL_non_normalized = np.nan_to_num(self.scm.P_A_eq(self.ELL), nan = 0)
    self.P_A_eq_normalization = self.integral_grid_d_3_xi(P_A_eq_ELL_non_normalized)/self.scm.P_A_tot_eq

  ########################################################################################################################################
  # Function to compute the results given a deformation and yield function
  ########################################################################################################################################

  def compute_results(self, index_t, F, update_yield_function = False):

    # Enumerate the relatively-deformed initial distribution
    F_inv = np.linalg.inv(F)
    P_A_0_rel_t = self.scm.P_A_eq(self.ELL_rel(F_inv)/self.J_sw**(1/3))/self.P_A_eq_normalization/self.J_sw
    P_A_0_rel_t[np.isnan(P_A_0_rel_t)] = 0

    # Enumerate the yield function
    ELL_rel = np.zeros((self.num_grid, self.num_grid, self.num_grid, index_t + 1))
    ELL_rel[:, :, :, -1] = self.ELL
    for index_index_t in range(index_t):
      ELL_rel[:, :, :, index_index_t] = self.ELL_rel(self.F[:, :, index_index_t]*F_inv)
    yield_function = np.trapz(ELL_rel > self.scm.gamma_c, x = self.t[:index_t + 1], axis = -1) == 0

    # Homogeneous solution for P_A
    P_A = P_A_0_rel_t*yield_function

    # Total probability that a chain is intact
    P_A_tot = self.integral_grid_d_3_xi(P_A)

    # Nondimensional stress corresponding to the applied deformation
    beta_sigma_over_n_plus_p = np.diag([ \
        self.integral_grid_d_3_xi(P_A, element = 'stress', component = [1, 1]), \
        self.integral_grid_d_3_xi(P_A, element = 'stress', component = [2, 2]), \
        self.integral_grid_d_3_xi(P_A, element = 'stress', component = [3, 3])  \
      ])

    # Return results
    return P_A_tot, beta_sigma_over_n_plus_p

  ########################################################################################################################################
  # Function to solve for results over the applied deformation history
  ########################################################################################################################################

  def solve(self, total_time_in_seconds, F_applied, timestep_in_seconds = None, n_over_beta = None):

    # Enumerate time discretization
    if timestep_in_seconds is None:
      self.timestep_in_seconds = total_time_in_seconds/25
```

```
    else:
      self.timestep_in_seconds = timestep_in_seconds
    num_time = 1 + np.ceil(total_time_in_seconds/self.timestep_in_seconds).astype(int)
    self.t = np.linspace(0, total_time_in_seconds, num_time)

    # Verify maximum array size is acceptable (memory) before starting
    A = np.zeros((self.num_grid, self.num_grid, self.num_grid, num_time))
    del A

    # Allocate results
    self.F = np.zeros((3, 3, num_time))
    self.P_A_tot = np.zeros(num_time)
    self.p = np.zeros(num_time)
    self.beta_sigma_over_n = np.zeros((3, 3, num_time))
    self.beta_sigma_nominal_over_n = np.zeros((3, 3, num_time))

    # Loop through time
    for index_t in range(num_time):

      # When appling F_11(t)
      if F_applied.component(self.t[index_t]) == 11:
        F_11 = F_applied(self.t[index_t])
        def fun(F_33):
          F_22 = 1/(F_11*F_33)
          beta_sigma_over_n_plus_p = self.compute_results(index_t, np.diag([F_11, F_22, F_33]))[-1]
          return beta_sigma_over_n_plus_p[1, 1] - beta_sigma_over_n_plus_p[2, 2]

      # When applying F_22(t)
      elif F_applied.component(self.t[index_t]) == 22:
        F_22 = F_applied(self.t[index_t])
        def fun(F_33):
          F_11 = 1/(F_22*F_33)
          beta_sigma_over_n_plus_p = self.compute_results(index_t, np.diag([F_11, F_22, F_33]))[-1]
          return beta_sigma_over_n_plus_p[0, 0] - beta_sigma_over_n_plus_p[2, 2]

      # Solve for F_33(t) using incompressibility and the traction-free boundary conditions
      if index_t == 0:
        F_33 = 1
      F_33 = root_scalar(fun, x0 = F_33*0.95, x1 = F_33/0.95).root

      # Compute F_22(t) or F_11(t) and thus retrieve F(t)
      if F_applied.component(self.t[index_t]) == 11:
        F_22 = 1/(F_11*F_33)
      elif F_applied.component(self.t[index_t]) == 22:
        F_11 = 1/(F_22*F_33)
      F = np.diag([F_11, F_22, F_33])

      # Compute and store the results, update yield function
      P_A_tot, beta_sigma_over_n_plus_p = self.compute_results(index_t, F, update_yield_function = True)
      self.F[:,:, index_t] = F
      self.P_A_tot[index_t] = P_A_tot
      self.p[index_t] = beta_sigma_over_n_plus_p[2, 2]
      self.beta_sigma_over_n[:, :, index_t] = beta_sigma_over_n_plus_p - self.p[index_t]*np.diag([1, 1, 1])
      self.beta_sigma_nominal_over_n[:, :, index_t] = np.linalg.det(self.F[:,:, index_t]) \
        *np.linalg.inv(self.F[:,:, index_t]).dot(self.beta_sigma_over_n[:, :, index_t])

    # Compute and store other results
    if n_over_beta is not None:
      self.n_over_beta = n_over_beta
      self.sigma = n_over_beta*self.beta_sigma_over_n
      self.sigma_nominal = n_over_beta*self.beta_sigma_nominal_over_n

##########################################################################################################################
# Function to adjust discretization for Romberg integration
##########################################################################################################################

def adjust_for_romb(self, num_discretization, decrease = False):
  if ((np.log(num_discretization - 1)/np.log(2)).is_integer()):
    return int(round(num_discretization))
  else:
    n = 0
    dos_check = 3
    while dos_check >= 2:
      n += 1
      dos_check = (num_discretization - 1)**(1/n)
    if decrease is True and dos_check < 2:
      return int(1 + 2**(n - 1))
    else:
      return int(1 + 2**n)

##########################################################################################################################
# Function for integration over the spatial grid
##########################################################################################################################

def integral_grid_d_3_xi(self, FUN, element = 1, component = None):
  if element == 'stress':
    element = self.ELEMENT_stress[:, :, :, component[0] - 1, component[1] - 1]
  if FUN.ndim == 3:
    return 8*romb(romb(romb(FUN*element, \
      dx = self.dx, axis = 0), dx = self.dy, axis = 0), dx = self.dz, axis = 0)
  elif FUN.ndim == 4:
    return 8*romb(romb(romb(FUN*element[:,:,:,None], \
```

```
      dx = self.dx, axis = 0), dx = self.dy, axis = 0), dx = self.dz, axis = 0)
  elif FUN.ndim == 5:
    return 8*romb(romb(romb(FUN*element[:,:,:,None,None], \
      dx = self.dx, axis = 0), dx = self.dy, axis = 0), dx = self.dz, axis = 0)

##########################################################################################################################################
# Function for integration element specialized for stress calculation
##########################################################################################################################################

def element_stress(self, x, y, z, single_chain_model):
  ell = np.sqrt(x*x + y*y + z*z)
  eta = np.nan_to_num(single_chain_model.eta(ell), nan = 0)
  if isinstance(z, np.ndarray):
    eta_over_ell = np.zeros(z.shape)
    eta_over_ell[ell != 0] = eta[ell != 0]/ell[ell != 0]
  else:
    if ell == 0:
      eta_over_ell = 0
    else:
      eta_over_ell = eta/ell
  C = (single_chain_model.N_b + single_chain_model.varsigma*single_chain_model.N_b_H)/self.J_sw
  element_out = np.zeros((ell.shape[0], ell.shape[1], ell.shape[2], 3, 3))
  element_out[:, :, :, 0, 0] = C*eta_over_ell*x*x
  element_out[:, :, :, 0, 1] = C*eta_over_ell*x*y
  element_out[:, :, :, 0, 2] = C*eta_over_ell*x*z
  element_out[:, :, :, 1, 0] = C*eta_over_ell*y*x
  element_out[:, :, :, 1, 1] = C*eta_over_ell*y*y
  element_out[:, :, :, 1, 2] = C*eta_over_ell*y*z
  element_out[:, :, :, 2, 0] = C*eta_over_ell*z*x
  element_out[:, :, :, 2, 1] = C*eta_over_ell*z*y
  element_out[:, :, :, 2, 2] = C*eta_over_ell*z*z
  return element_out

##########################################################################################################################################
# Function to return relatively-deformed coordinates
##########################################################################################################################################

def ELL_rel(self, F_rel):
  X_rel = F_rel[0, 0]*self.X + F_rel[0, 1]*self.Y + F_rel[0, 2]*self.Z
  Y_rel = F_rel[1, 0]*self.X + F_rel[1, 1]*self.Y + F_rel[1, 2]*self.Z
  Z_rel = F_rel[2, 0]*self.X + F_rel[2, 1]*self.Y + F_rel[2, 2]*self.Z
  return np.sqrt(X_rel*X_rel + Y_rel*Y_rel + Z_rel*Z_rel)
```

References

[1] L. R. G. Treloar. *The Physics of Rubber Elasticity*. Clarendon Press, Cambridge, UK, 1949.

[2] J. P. Gong, Y. Katsuyama, T. Kurokawa, and Y. Osada. Double-network hydrogels with extremely high mechanical strength. Adv. Mater. **15**, 14 (2003).

[3] T. Nakajima, H. Furukawa, Y. Tanaka, T. Kurokawa, Y. Osada, and J. P. Gong. True chemical structure of double network hydrogels. Macromolecules **42**, 6 (2009).

[4] J. P. Gong. Why are double network hydrogels so tough? Soft Matter **6**, 12 (2010).

[5] T. Nakajima, H. Sato, Y. Zhao, S. Kawahara, T. Kurokawa, K. Sugahara, and J. P. Gong. A universal molecular stent method to toughen any hydrogels based on double network concept. Adv. Func. Mater. **22**, 21 (2012).

[6] E. Ducrot and C. Creton. Characterizing large strain elasticity of brittle elastomeric networks by embedding them in a soft extensible matrix. Adv. Func. Mater. **26**, 15 (2016).

[7] T. Nakajima, Y. Ozaki, R. Namba, K. Ota, Y. Maida, T. Matsuda, T. Kurokawa, and J. P. Gong. Tough double-network gels and elastomers from the nonprestretched first network. ACS Macro Lett. **8**, 11 (2019).

[8] T. Matsuda, T. Nakajima, and J. P. Gong. Fabrication of tough and stretchable hybrid double-network elastomers using ionic dissociation of polyelectrolyte in nonaqueous media. Chem. Mater. **31**, 10 (2019).

[9] J. Murai, T. Nakajima, T. Matsuda, K. Tsunoda, T. Nonoyama, T. Kurokawa, and J. P. Gong. Tough double network elastomers reinforced by the amorphous cellulose network. Polymer **178**, (2019).

[10] Y. Chen, A. Spiering, S. Karthikeyan, G. W. Peters, E. Meijer, and R. P. Sijbesma. Mechanically induced chemiluminescence from polymers incorporating a 1, 2-dioxetane unit in the main chain. Nat. Chem. **4**, 7 (2012).

[11] E. Ducrot, Y. Chen, M. Bulters, R. P. Sijbesma, and C. Creton. Toughening elastomers with sacrificial bonds and watching them break. Science **344**, 6180 (2014).

[12] P. Millereau, E. Ducrot, J. M. Clough, M. E. Wiseman, H. R. Brown, R. P. Sijbesma, and C. Creton. Mechanics of elastomeric molecular composites. Proc. Natl. Acad. Sci. U.S.A. **115**, 37 (2018).

[13] S. Garcia-Manyes and A. E. Beedle. Steering chemical reactions with force. Nat. Rev. Chem. **1**, 11 (2017).

[14] Z. Chen, J. A. Mercer, X. Zhu, J. A. Romaniuk, R. Pfattner, L. Cegelski, T. J. Martinez, N. Z. Burns, and Y. Xia. Mechanochemical unzipping of insulating polyladderene to semiconducting polyacetylene. Science **357**, 6350 (2017).

[15] Y. Tian, X. Cao, X. Li, H. Zhang, C.-L. Sun, Y. Xu, W. Weng, W. Zhang, and R. Boulatov. A Polymer with Mechanochemically Active Hidden Length. J. Am. Chem. Soc. **142**, 43 (2020).

[16] Z. Wang, X. J. Zheng, T. Ouchi, T. Kouznetsova, H. Beech, S. Av-Ron, B. Bowser, S. Wang, J. Johnson, J. Kalow, B. Olsen, J. P. Gong, M. Rubinstein, and S. Craig. Toughening Hydrogels Through Force-triggered Chemical Reactions that Lengthen Polymer Strands. ChemRxiv, 13514377 (2021).

[17] P. Cordier, F. Tournilhac, C. Soulié-Ziakovic, and L. Leibler. Self-healing and thermoreversible rubber from supramolecular assembly. Nature **451**, 7181 (2008).

[18] C.-H. Li, C. Wang, C. Keplinger, J.-L. Zuo, L. Jin, Y. Sun, P. Zheng, Y. Cao, F. Lissel, C. Linder, et al. A highly stretchable autonomous self-healing elastomer. Nat. Chem. **8**, 6 (2016).

[19] J.-F. Mei, X.-Y. Jia, J.-C. Lai, Y. Sun, C.-H. Li, J.-H. Wu, Y. Cao, X.-Z. You, and Z. Bao. A highly stretchable and autonomous self-healing polymer based on combination of pt··· pt and π–π interactions. Macromol. Rapid Commun. **37**, 20 (2016).

[20] J.-Y. Sun, X. Zhao, W. R. Illeperuma, O. Chaudhuri, K. H. Oh, D. J. Mooney, J. J. Vlassak, and Z. Suo. Highly stretchable and tough hydrogels. Nature **489**, 7414 (2012).

[21] E. Khare, N. Holten-Andersen, and M. J. Buehler. Transition-metal coordinate bonds for bioinspired macromolecules with tunable mechanical properties. Nat. Rev. Mater. **6**, (2021).

[22] D. W. Balkenende, S. Coulibaly, S. Balog, Y. C. Simon, G. L. Fiore, and C. Weder. Mechanochemistry with metallosupramolecular polymers. J. Am. Chem. Soc. **136**, 29 (2014).

[23] S. C. Grindy, R. Learsch, D. Mozhdehi, J. Cheng, D. G. Barrett, Z. Guan, P. B. Messersmith, and N. Holten-Andersen. Control of hierarchical polymer mechanics with bioinspired metal-coordination dynamics. Nat. Mater. **14**, 12 (2015).

[24] D. Mozhdehi, J. A. Neal, S. C. Grindy, Y. Cordeau, S. Ayala, N. Holten-Andersen, and Z. Guan. Tuning dynamic mechanical response in metallopolymer networks through simultaneous control of structural and temporal properties of the networks. Macromolecules **49**, 17 (2016).

[25] E. Filippidi, T. R. Cristiani, C. D. Eisenbach, J. H. Waite, J. N. Israelachvili, B. K. Ahn, and M. T. Valentine. Toughening elastomers using mussel-inspired iron-catechol complexes. Science **358**, 6362 (2017).

[26] Y. Vidavsky, S. Bae, and M. N. Silberstein. Modulating metallopolymer mechanical properties by controlling metal ligand crosslinking. J. Polym. Sci., Part A: Polym. Chem. **56**, 11 (2018).

[27] Y. Sha, Y. Zhang, E. Xu, C. W. McAlister, T. Zhu, S. L. Craig, and C. Tang. Generalizing metallocene mechanochemistry to ruthenocene mechanophores. Chem. Sci. **10**, 19 (2019).

[28] Y. Vidavsky, M. R. Buche, Z. M. Sparrow, X. Zhang, S. J. Yang, R. A. DiStasio Jr., and M. N. Silberstein. Tuning the mechanical properties of metallopolymers via ligand interactions: A combined experimental and theoretical study. Macromolecules **53**, 6 (2020).

[29] D. Montarnal, M. Capelot, F. Tournilhac, and L. Leibler. Silica-like malleable materials from permanent organic networks. Science **334**, 6058 (2011).

[30] M. Capelot, M. M. Unterlass, F. Tournilhac, and L. Leibler. Catalytic control of the vitrimer glass transition. ACS Macro Lett. **1**, 7 (2012).

[31] W. Denissen, G. Rivero, R. Nicolaÿ, L. Leibler, J. M. Winne, and F. E. Du Prez. Vinylogous urethane vitrimers. Adv. Func. Mater. **25**, 16 (2015).

[32] M. Röttger, T. Domenech, R. van der Weegen, A. Breuillac, R. Nicolaÿ, and L. Leibler. High-performance vitrimers from commodity thermoplastics through dioxaborolane metathesis. Science **356**, 6333 (2017).

[33] C. J. Kloxin, T. F. Scott, B. J. Adzima, and C. N. Bowman. Covalent adaptable networks (CANs): a unique paradigm in cross-linked polymers. Macromolecules **43**, 6 (2010).

[34] X. Chen, M. A. Dam, K. Ono, A. Mal, H. Shen, S. R. Nutt, K. Sheran, and F. Wudl. A thermally re-mendable cross-linked polymeric material. Science **295**, 5560 (2002).

[35] C. J. Kloxin and C. N. Bowman. Covalent adaptable networks: smart, reconfigurable and responsive network systems. Chem. Soc. Rev. **42**, 17 (2013).

[36] V. Yesilyurt, M. J. Webber, E. A. Appel, C. Godwin, R. Langer, and D. G. Anderson. Injectable self-healing glucose-responsive hydrogels with pH-regulated mechanical properties. Adv. Mater. **28**, 1 (2016).

[37] H. Frisch, K. Mundsinger, B. L. Poad, S. J. Blanksby, and C. Barner-Kowollik. Wavelength-gated photoreversible polymerization and topology control. Chem. Sci. **11**, 10 (2020).

[38] M. Podgórski, S. Mavila, S. Huang, N. Spurgin, J. Sinha, and C. N. Bowman. Thiol–Anhydride Dynamic Reversible Networks. Angew. Chem. Int. Ed. **59**, 24 (2020).

[39] L. Carlsson, S. Rose, D. Hourdet, and A. Marcellan. Nano-hybrid self-crosslinked PDMA/silica hydrogels. Soft Matter **6**, 15 (2010).

[40] K. J. Henderson, T. C. Zhou, K. J. Otim, and K. R. Shull. Ionically cross-linked triblock copolymer hydrogels with high strength. Macromolecules **43**, 14 (2010).

[41] W.-C. Lin, W. Fan, A. Marcellan, D. Hourdet, and C. Creton. Large strain and fracture properties of poly (dimethylacrylamide)/silica hybrid hydrogels. Macromolecules **43**, 5 (2010).

[42] D. C. Tuncaboylu, M. Sari, W. Oppermann, and O. Okay. Tough and self-healing hydrogels formed via hydrophobic interactions. Macromolecules **44**, 12 (2011).

[43] M. A. Haque, T. Kurokawa, G. Kamita, and J. P. Gong. Lamellar bilayers as reversible sacrificial bonds to toughen hydrogel: hysteresis, self-recovery, fatigue resistance, and crack blunting. Macromolecules **44**, 22 (2011).

[44] T. L. Sun, T. Kurokawa, S. Kuroda, A. B. Ihsan, T. Akasaki, K. Sato, M. A. Haque, T. Nakajima, and J. P. Gong. Physical hydrogels composed of polyampholytes demonstrate high toughness and viscoelasticity. Nat. Mater. **12**, 10 (2013).

[45] T. Narita, K. Mayumi, G. Ducouret, and P. Hébraud. Viscoelastic properties of poly (vinyl alcohol) hydrogels having permanent and transient cross-links studied by microrheology, classical rheometry, and dynamic light scattering. Macromolecules **46**, 10 (2013).

[46] K. Mayumi, A. Marcellan, G. Ducouret, C. Creton, and T. Narita. Stress–strain relationship of highly stretchable dual cross-link gels: separability of strain and time effect. ACS Macro Lett. **2**, 12 (2013).

[47] S. Rose, A. Dizeux, T. Narita, D. Hourdet, and A. Marcellan. Time dependence of dissipative and recovery processes in nanohybrid hydrogels. Macromolecules **46**, 10 (2013).

[48] W. Kuhn and F. Grün. Beziehungen zwischen elastischen Konstanten und Dehnungsdoppelbrechung hochelastischer Stoffe. Kolloid-Zeitschrift **101**, 3 (1942).

[49] D. A. McQuarrie. *Statistical Mechanics*. University Science Books, Mill Valley, CA, 2000.

[50] M. Rubinstein and R. H. Colby. *Polymer Physics*. Oxford University Press, New York, 2003.

[51] M. Doi and S. F. Edwards. *The Theory of Polymer Dynamics*. Oxford University Press, Oxford, UK, 1988.

[52] M. C. Wang and E. Guth. Statistical theory of networks of non-Gaussian flexible chains. J. Chem. Phys. **20**, 7 (1952).

[53] F. Manca, S. Giordano, P. L. Palla, R. Zucca, F. Cleri, and L. Colombo. Elasticity of flexible and semiflexible polymers with extensible bonds in the Gibbs and Helmholtz ensembles. J. Chem. Phys. **136**, 15 (2012).

[54] P. Flory and D. Yoon. Moments and distribution functions for polymer chains of finite length. I. Theory. J. Chem. Phys. **61**, 12 (1974).

[55] D. Yoon and P. Flory. Moments and distribution functions for polymer chains of finite length. II. Polymethylene chains. J. Chem. Phys. **61**, 12 (1974).

[56] P. J. Flory. Molecular theory of rubber elasticity. Polymer **20**, 11 (1979).

[57] A. Fiasconaro and F. Falo. Analytical results of the extensible freely jointed chain model. Physica A **532**, (2019).

[58] M. S. Green and A. V. Tobolsky. A new approach to the theory of relaxing polymeric media. J. Chem. Phys. **14**, 2 (1946).

[59] F. Tanaka and S. Edwards. Viscoelastic properties of physically crosslinked networks. I. Transient network theory. Macromolecules **25**, 5 (1992).

[60] F. J. Vernerey, R. Long, and R. Brighenti. A Statistically Based Continuum Theory for Polymers with Transient Networks. J. Mech. Phys. Solids **107**, (2017).

[61] G. I. Bell. Models for the specific adhesion of cells to cells. Science **200**, 4342 (1978).

[62] Y. Mao, B. Talamini, and L. Anand. Rupture of polymers by chain scission. Extreme Mech. Lett. **13**, (2017).

[63] H. M. James and E. Guth. Theory of the elastic properties of rubber. J. Chem. Phys. **11**, 10 (1943).

[64] P. J. Flory and J. Rehner Jr. Statistical mechanics of cross-linked polymer networks I. Rubberlike elasticity. J. Chem. Phys. **11**, 11 (1943).

[65] L. R. G. Treloar. The elasticity of a network of long-chain molecules.–III. Trans. Faraday Soc. **42**, (1946).

[66] E. M. Arruda and M. C. Boyce. A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials. J. Mech. Phys. Solids **41**, (1993).

[67] Y. Mao and L. Anand. Fracture of elastomeric materials by crosslink failure. J. of Appl. Mech. **85**, 8 (2018).

[68] C. Miehe, S. Göktepe, and F. Lulei. A micro-macro approach to rubber-like materials–Part I: the nonaffine micro-sphere model of rubber elasticity. J. Mech. Phys. Solids **52**, 11 (2004).

[69] M. N. Silberstein, L. D. Cremar, B. A. Beiermann, S. B. Kramer, T. J. Martinez, S. R. White, and N. R. Sottos. Modeling mechanophore activation within a viscous rubbery network. J. Mech. Phys. Solids **63**, (2014).

[70] L. R. G. Treloar. The photoelastic properties of short-chain molecular networks. Trans. Faraday Soc. **50**, (1954).

[71] P. Wu and E. van der Giessen. On improved network models for rubber elasticity and their applications to orientation hardening in glassy polymers. J. Mech. Phys. Solids **41**, 3 (1993).

[72] C. Storm, J. J. Pastore, F. C. MacKintosh, T. C. Lubensky, and P. A. Janmey. Nonlinear elasticity in biological gels. Nature **435**, 7039 (2005).

[73] Q. Wang, G. R. Gossweiler, S. L. Craig, and X. Zhao. Mechanics of mechanochemically responsive elastomers. J. Mech. Phys. Solids **82**, (2015).

[74] S. R. Lavoie, P. Millereau, C. Creton, R. Long, and T. Tang. A continuum model for progressive damage in tough multinetwork elastomers. J. Mech. Phys. Solids **125**, (2019).

[75] B. Li and N. Bouklas. A variational phase-field model for brittle fracture in polydisperse elastomer networks. Int. J. Solids Struct. **182**, (2020).

[76] M. Tehrani and A. Sarvestani. Effect of chain length distribution on mechanical behavior of polymeric networks. Eur. Polym. J. **87**, (2017).

[77] M. Tehrani, Z. Ghalamzan, and A. Sarvestani. Polydispersity controls the strength of semi-flexible polymer networks. Phys. Biol. **15**, 6 (2018).

[78] J. Mulderrig, B. Li, and N. Bouklas. Affine and non-affine microsphere models for chain scission in polydisperse elastomer networks. Mech. Mater. **160**, (2021).

[79] M. Tkachuk and C. Linder. The maximal advance path constraint for the homogenization of materials with random network microstructure. Philos. Mag. **92**, 22 (2012).

[80] E. Verron and A. Gros. An equal force theory for network models of soft materials with arbitrary molecular weight distribution. J. Mech. Phys. Solids **106**, (2017).

[81] F. J. Vernerey, R. Brighenti, R. Long, and T. Shen. Statistical damage mechanics of polymer networks. Macromolecules **51**, 17 (2018).

[82] B. D. Coleman and W. Noll. The Thermodynamics of Elastic Materials with Heat Conduction and Viscosity. Arch. Rat. Mech. Anal **13**, (1963).

[83] B. D. Coleman and M. E. Gurtin. Thermodynamics with internal state variables. J. Chem. Phys. **47**, 2 (1967).

[84] N. S. Schauser, G. E. Sanoja, J. M. Bartels, S. K. Jain, J. G. Hu, S. Han, L. M. Walker, M. E. Helgeson, R. Seshadri, and R. A. Segalman. Decoupling bulk mechanics and mono-and multivalent ion transport in polymers based on metal–ligand coordination. Chem. Mater. **30**, 16 (2018).

[85] R. A. Musgrave, A. D. Russell, D. W. Hayward, G. R. Whittell, P. G. Lawrence, P. J. Gates, J. C. Green, and I. Manners. Main-chain metallopolymers at the static–dynamic boundary based on nickelocene. Nat. Chem. **9**, 8 (2017).

[86] Y. Sha, Y. Zhang, E. Xu, Z. Wang, T. Zhu, S. L. Craig, and C. Tang. Quantitative and mechanistic mechanochemistry in ferrocene dissociation. ACS Macro Lett. **7**, 10 (2018).

[87] C. Rüttiger, M. Appold, H. Didzoleit, A. Eils, C. Dietz, R. W. Stark, B. Stühn, and M. Gallei. Structure Formation of Metallopolymer-Grafted Block Copolymers. Macromolecules **49**, 9 (2016).

[88] A. C. Jackson, F. L. Beyer, S. C. Price, B. C. Rinderspacher, and R. H. Lambeth. Role of metal–ligand bond strength and phase separation on the mechanical properties of metallopolymer films. Macromolecules **46**, 14 (2013).

[89] A. C. Jackson, S. D. Walck, K. E. Strawhecker, B. G. Butler, R. H. Lambeth, and F. L. Beyer. Metallopolymers containing excess metal–ligand complex for improved mechanical properties. Macromolecules **47**, 13 (2014).

[90] S. Mavila, C. E. Diesendruck, S. Linde, L. Amir, R. Shikler, and N. G. Lemcoff. Polycyclooctadiene complexes of rhodium (I): direct access to organometallic nanoparticles. Angew. Chem. **125**, 22 (2013).

[91] A. Levy, R. Feinstein, and C. E. Diesendruck. Mechanical unfolding and thermal refolding of single-chain nanoparticles using ligand–metal bonds. Angew. Chem. **141**, 18 (2019).

[92] W. C. Yount, D. M. Loveless, and S. L. Craig. Small-molecule dynamics and mechanisms underlying the macroscopic mechanical properties of coordinatively cross-linked polymer networks. J. Am. Chem. Soc. **127**, 41 (2005).

[93] Y.-L. Rao, A. Chortos, R. Pfattner, F. Lissel, Y.-C. Chiu, V. Feig, J. Xu, T. Kurosawa, X. Gu, C. Wang, M. He, J. W. Chung, and Z. Bao. Stretchable self-healing polymeric dielectrics cross-linked through metal–ligand coordination. J. Am. Chem. Soc. **138**, 18 (2016).

[94] C.-H. Li and J.-L. Zuo. Self-healing polymers based on coordination bonds. Nat. Chem. **32**, 27 (2020).

[95] N. Holten-Andersen, M. J. Harrington, H. Birkedal, B. P. Lee, P. B. Messersmith, K. Y. C. Lee, and J. H. Waite. pH-induced metal-ligand cross-links inspired by mussel yield self-healing polymer networks with near-covalent elastic moduli. Proc. Natl. Acad. Sci. U.S.A. **108**, 7 (2011).

[96] S. C. Grindy and N. Holten-Andersen. Bio-inspired metal-coordinate hydrogels with programmable viscoelastic material functions controlled by longwave UV light. Soft Matter **13**, 22 (2017).

[97] J. T. Auletta, G. J. LeDonne, K. C. Gronborg, C. D. Ladd, H. Liu, W. W. Clark, and T. Y. Meyer. Stimuli-responsive iron-cross-linked hydrogels that undergo redox-driven switching between hard and soft states. Macromolecules **48**, 6 (2015).

[98] M. Di Giannantonio, M. A. Ayer, E. Verde-Sesto, M. Lattuada, C. Weder, and K. M. Fromm. Triggered metal ion release and oxidation: ferrocene as a mechanophore in polymers. Angew. Chem. Int. Ed. **57**, 35 (2018).

[99] A. Piermattei, S. Karthikeyan, and R. P. Sijbesma. Activating catalysts with mechanical force. Nat. Chem. **1**, 2 (2009).

[100] J. M. Paulusse and R. P. Sijbesma. Selectivity of mechanochemical chain scission in mixed palladium (II) and platinum (II) coordination polymers. Chem. Commun., 37 (2008).

[101] S. Karthikeyan, S. L. Potisek, A. Piermattei, and R. P. Sijbesma. Highly efficient mechanochemical scission of silver-carbene coordination polymers. J. Am. Chem. Soc. **130**, 45 (2008).

[102] P. Chen, Q. Li, S. Grindy, and N. Holten-Andersen. White-light-emitting lanthanide metallogels with tunable luminescence and reversible stimuli-responsive properties. J. Am. Chem. Soc. **137**, 36 (2015).

[103] G. A. Filonenko and J. R. Khusnutdinova. Dynamic phosphorescent probe for facile and reversible stress sensing. Adv. Mater. **29**, 22 (2017).

[104] G. A. Filonenko, D. Sun, M. Weber, C. Müller, and E. A. Pidko. Multicolor Organometallic Mechanophores for Polymer Imaging Driven by Exciplex Level Interactions. J. Am. Chem. Soc. **141**, 24 (2019).

[105] M. K. Beyer. The mechanical strength of a covalent bond calculated by density functional theory. J. Chem. Phys. **112**, 17 (2000).

[106] M. T. Ong, J. Leiding, H. Tao, A. M. Virshup, and T. J. Martínez. First principles dynamics and minimum energy pathways for mechanochemical ring opening of cyclobutene. J. Am. Chem. Soc. **131**, 18 (2009).

[107] J. Ribas-Arino, M. Shiga, and D. Marx. Understanding covalent mechanochemistry. Angew. Chem. Int. Ed. **48**, 23 (2009).

[108] K. Wolinski and J. Baker. Theoretical predictions of enforced structural changes in molecules. Mol. Phys. **107**, 22 (2009).

[109] T. Stauch and A. Dreuw. Quantum chemical strain analysis for mechanochemical processes. Acc. Chem. Res. **50**, 4 (2017).

[110] D. A. Davis, A. Hamilton, J. Yang, L. D. Cremar, D. Van Gough, S. L. Potisek, M. T. Ong, P. V. Braun, T. J. Martínez, S. R. White, et al. Force-induced activation of covalent bonds in mechanoresponsive polymeric materials. Nature **459**, 7243 (2009).

[111] T. Bettens, M. Alonso, P. Geerlings, and F. De Proft. Implementing the mechanical force into the conceptual DFT framework: understanding and predicting molecular mechanochemical properties. Phys. Chem. Chem. Phys. **21**, 14 (2019).

[112] J. Müller and B. Hartke. ReaxFF reactive force field for disulfide mechanochemistry, fitted to multireference ab initio data. J. Chem. Theory Comput. **12**, 8 (2016).

[113] T. Stauch and A. Dreuw. Force-induced retro-click reaction of triazoles competes with adjacent single-bond rupture. Chem. Sci. **8**, 8 (2017).

[114] M. Krupička, P. Dopieralski, and D. Marx. Unclicking the Click: Metal-Assisted Mechanochemical Cycloreversion of Triazoles Is Possible. Angew. Chem. **129**, 27 (2017).

[115] T. Stauch and A. Dreuw. Predicting the efficiency of photoswitches using force analysis. J. Phys. Chem. Lett. **7**, 7 (2016).

[116] J.-C. Eloi, L. Chabanne, G. R. Whittell, and I. Manners. Metallopolymers with emerging applications. Mater. Today **11**, 4 (2008).

[117] D. Edwards and R. Hayward. Transition metal acetates. Can. J. Chem. **46**, 22 (1968).

[118] T. Ishii, S. Tsuboi, G. Sakane, M. Yamashita, and B. K. Breedlove. Universal spectrochemical series of six-coordinate octahedral metal complexes for modifying the ligand field splitting. Dalt. Trans. **0**, 4 (2009).

[119] J. E. Bauman Jr. and J. C. Wang. Imidazole complexes of nickel (II), copper (II), zinc (II), and silver (I). Inorg. Chem. **3**, 3 (1964).

[120] Y. Shao, Z. Gan, E. Epifanovsky, A. T. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, X. Feng, et al. Advances in molecular quantum chemistry contained in the Q-Chem 4 program package. Mol. Phys. **113**, 2 (2015).

[121] N. Mardirossian and M. Head-Gordon. ωB97X-V: A 10-parameter, range-separated hybrid, generalized gradient approximation density functional with nonlocal correlation, designed by a survival-of-the-fittest strategy. Phys. Chem. Chem. Phys. **16**, 21 (2014).

[122] F. Weigend and R. Ahlrichs. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. Phys. Chem. Chem. Phys. **7**, 18 (2005).

[123] C. Wu, A. Alqahtani, S. Sangtarash, A. Vezzoli, H. Sadeghi, C. M. Robertson, C. Cai, C. J. Lambert, S. J. Higgins, and R. J. Nichols. In situ formation of H-bonding imidazole chains in break-junction experiments. Nanoscale **12**, 14 (2020).

[124] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, et al. XSEDE: accelerating scientific discovery. Comput. Sci. Eng. **16**, 5 (2014).

[125] M. R. Buche and M. N. Silberstein. Statistical mechanical constitutive theory of polymer networks: The inextricable links between distribution, behavior, and ensemble. Phys. Rev. E **102**, 012501 (2020).

[126] M. R. Buche. `Buche_Silberstein_model_2020`. Zenodo (2021).

[127] L. Livadaru, R. Netz, and H. Kreuzer. Stretching response of discrete semiflexible polymers. Macromolecules **36**, 10 (2003).

[128] O. Kratky and G. Porod. Rntgenuntersuchung Gelster Fadenmolekle. Recl. Trav. Chim. Pays-Bas **68**, (1949).

[129] J. Kierfeld, O. Niamploy, V. Sa-Yakanit, and R. Lipowsky. Stretching of semiflexible polymers with elastic bonds. Eur. Phys. J. E **14**, 1 (2004).

[130] S. R. Lavoie, R. Long, and T. Tang. Modeling the Mechanics of Polymer Chains with Deformable and Active Bonds. J. Phys. Chem. B **124**, 1 (2019).

[131] R. M. Neumann. Nonequivalence of the stress and strain ensembles in describing polymer-chain elasticity. Phys. Rev. A **31**, 5 (1985).

[132] R. M. Neumann. Comment on "Elasticity of flexible and semiflexible polymers with extensible bonds in the Gibbs and Helmholtz ensembles". J. Chem. Phys. **138**, 15 (2013).

[133] F. Manca, S. Giordano, P. L. Palla, F. Cleri, and L. Colombo. Response to "Comment on 'Elasticity of flexible and semiflexible polymers with extensible bonds in the Gibbs and Helmholtz ensembles"'. J. Chem. Phys. **138**, 15 (2013).

[134] C. Truesdell and W. Noll. *The Non-linear Field Theories of Mechanics*. Springer, Berlin, 2004.

[135] S. Paolucci. *Continuum Mechanics and Thermodynamics of Matter*. Cambridge University Press, Cambridge, UK, 2016.

[136] T. Ooura and M. Mori. The double exponential formula for oscillatory functions over the half infinite interval. J. Comput. Appl. Math. **38**, 1-3 (1991).

[137] D. H. Bailey. `MPFUN2015`: A thread-safe arbitrary precision computation package. Technical paper, `davidhbailey.com` (2015).

[138] R. W. Zwanzig. High-temperature equation of state by a perturbation method. I. Nonpolar gases. J. Chem. Phys. **22**, 8 (1954).

[139] J. A. Barker and D. Henderson. Perturbation theory and equation of state for fluids. II. A successful theory of liquids. J. Chem. Phys. **47**, 11 (1967).

[140] J. D. Weeks, D. Chandler, and H. C. Andersen. Role of repulsive forces in determining the equilibrium structure of simple liquids. J. Chem. Phys. **54**, 12 (1971).

[141] J. Sethna. *Statistical Mechanics: Entropy, Order Parameters, and Complexity*. Oxford University Press, USA, New York, 2021.

[142] B. Talamini, Y. Mao, and L. Anand. Progressive damage and rupture in polymers. J. Mech. Phys. Solids **111**, (2018).

[143] T. Yang, K. M. Liechti, and R. Huang. A multiscale cohesive zone model for rate-dependent fracture of interfaces. J. Mech. Phys. Solids **145**, (2020).

[144] Q. Guo and F. Zaïri. A micromechanics-based model for deformation-induced damage and failure in elastomeric media. Int. J. Plast. **140**, (2021).

[145] N. Bleistein and R. A. Handelsman. *Asymptotic Expansions of Integrals*. Holt, Rinehart and Winston, New York, 1975.

[146] C. M. Bender and S. A. Orszag. *Advanced Mathematical Methods for Scientists and Engineers I: Asymptotic Methods and Perturbation Theory*. Springer Science & Business Media, New York, 2013.

[147] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. U.S. Government printing office, Washington, D.C., 1964.

[148] M. R. Buche and M. N. Silberstein. Chain breaking in the statistical mechanical constitutive theory of polymer networks. `arXiv:2104.08866 [cond-mat.soft]`. Accepted at J. Mech. Phys. Solids (2021).

[149] M. R. Buche. `chain_breaking_polymer_networks`. Zenodo (2021).

[150] S. R. Lavoie, R. Long, and T. Tang. A rate-dependent damage model for elastomers at large strain. Extreme Mech. Lett. **8**, (2016).

[151] M. Bacca, C. Creton, and R. M. McMeeking. A model for the Mullins effect in multinetwork elastomers. J. Appl. Mech. **84**, 12 (2017).

[152] V. Morovati and R. Dargazany. Micro-mechanical modeling of the stress softening in double-network hydrogels. Int. J. Solids Struct. **164**, (2019).

[153] D. Zhong, Y. Xiang, J. Liu, Z. Chen, H. Zhou, H. Yu, S. Qu, and W. Yang. A constitutive model for multi network elastomers pre-stretched by swelling. Extreme Mech. Lett. **40**, (2020).

[154] X. Zhao. A theory for large deformation and damage of interpenetrating polymer networks. J. Mech. Phys. Solids **60**, 2 (2012).

[155] V. Morovati, M. A. Saadat, and R. Dargazany. Necking of double-network gels: Constitutive modeling with microstructural insight. Phys. Rev. E **102**, 062501 (2020).

[156] F. Tanaka and S. Edwards. Viscoelastic properties of physically crosslinked networks. J. Non-Newton. Fluid. Mech. **43**, 2-3 (1992).

[157] P. Flory. Elasticity of polymer networks cross-linked in states of strain. Trans. Faraday Soc. **56**, (1960).

[158] D. Thomas. Limitations of the Tobolsky 'two network' theory in the interpretation of stress-relaxation data in rubbers. Polymer **7**, 3 (1966).

[159] H. Fricker. On the theory of stress relaxation by cross-link reorganization. Nat. Rev. Mater. **335**, 1602 (1973).

[160] F. J. Vernerey. Transient response of nonlinear polymer networks: A kinetic theory. J. Mech. Phys. Solids **115**, (2018).

[161] R. Brighenti, F. J. Vernerey, and F. Artoni. Rate-dependent failure mechanism of elastomers. Int. J. Mech. Sciences **130**, (2017).

[162] T. Shen and F. J. Vernerey. Rate-dependent fracture in transient networks. J. Mech. Phys. Solids **143**, (2020).

[163] F. Meng, R. H. Pritchard, and E. M. Terentjev. Stress relaxation, dynamics, and plasticity of transient polymer networks. Macromolecules **49**, 7 (2016).

[164] K. Yu, A. Xin, and Q. Wang. Mechanics of self-healing polymer networks crosslinked by dynamic bonds. J. Mech. Phys. Solids **121**, (2018).

[165] J. Lin, S. Y. Zheng, R. Xiao, J. Yin, Z. L. Wu, Q. Zheng, and J. Qian. Constitutive behaviors of tough physical hydrogels with dynamic metal-coordinated bonds. J. Mech. Phys. Solids **139**, (2020).

[166] C.-Y. Hui and R. Long. A constitutive model for the large deformation of a self-healing gel. Soft Matter **8**, 31 (2012).

[167] R. Long, K. Mayumi, C. Creton, T. Narita, and C.-Y. Hui. Time dependent behavior of a dual cross-link self-healing gel: Theory and experiments. Macromolecules **47**, 20 (2014).

[168] J. Guo, R. Long, K. Mayumi, and C.-Y. Hui. Mechanics of a dual cross-link gel with dynamic bonds: Steady state kinetics and large deformation effects. Macromolecules **49**, 9 (2016).

[169] T. Lu, Z. Wang, J. Tang, W. Zhang, and T. Wang. A pseudo-elasticity theory to model the strain-softening behavior of tough hydrogels. J. Mech. Phys. Solids **137**, (2020).

[170] R. Long, H. J. Qi, and M. L. Dunn. Modeling the mechanics of covalently adaptable polymer networks with temperature-dependent bond exchange reactions. Soft Matter **9**, 15 (2013).

[171] K. N. Long. The mechanics of network polymers with thermally reversible linkages. J. Mech. Phys. Solids **63**, (2014).

[172] X. Sun, H. Wu, and R. Long. Thermomechanics of a temperature sensitive covalent adaptable polymer with bond exchange reactions. Soft Matter **12**, 43 (2016).

[173] R. Zwanzig. *Nonequilibrium Statistical Mechanics*. Oxford University Press, New York, 2001.

[174] R. Zwanzig. Memory effects in irreversible thermodynamics. Phys. Rev. **124**, 4 (1961).

[175] Q. Guo and R. Long. *Mechanics of Polymer Networks with Dynamic Bonds*. *Self-Healing and Self-Recovering Hydrogels*. Ed. by C. Creton and O. Okay. Springer, Switzerland, 2020.

[176] I. Müller and T. Ruggeri. *Rational extended thermodynamics*. Springer Science & Business Media, New York, 2013.

[177] I. Prigogine. *Introduction to Thermodynamics of Irreversible Processes*. Interscience, New York, 1967.

[178] J. M. Powers. *Combustion Thermodynamics and Dynamics*. Cambridge University Press, Cambridge, New York, 2016.

[179] P. M. Morse. Diatomic molecules according to the wave mechanics. II. Vibrational levels. Phys. Rev. **34**, 1 (1929).

[180] F. Manca, S. Giordano, P. L. Palla, and F. Cleri. On the equivalence of thermodynamics ensembles for flexible polymer chains. Physica A **395**, (2014).

[181] W. Kauzmann and H. Eyring. The viscous flow of large molecules. J. Am. Chem. Soc. **62**, 11 (1940).

[182] S. Zhurkov. Kinetic concept of the strength of solids. Int. J. Fract. **26**, 4 (1984).

[183] M. N. Silberstein, K. Min, L. D. Cremar, C. M. Degen, T. J. Martinez, N. R. Aluru, S. R. White, and N. R. Sottos. Modeling mechanophore activation within a crosslinked glassy matrix. J. Appl. Phys. **114**, 2 (2013).

[184] O. K. Dudko, G. Hummer, and A. Szabo. Intrinsic rates and activation free energies from single-molecule pulling experiments. Phys. Rev. Lett. **96**, 10 (2006).

[185] H. A. Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. Physica **7**, 4 (1940).

[186] P. Schwaderer, E. Funk, F. Achenbach, J. Weis, C. Bräuchle, and J. Michaelis. Single-molecule measurement of the strength of a siloxane bond. Langmuir **24**, 4 (2008).

[187] G. Lake and A. Thomas. The strength of highly elastic materials. Proc. R. Soc. Lond. **300**, 1460 (1967).

[188] J. M. Clough, C. Creton, S. L. Craig, and R. P. Sijbesma. Covalent bond scission in the Mullins effect of a filled elastomer: real-time visualization with mechanoluminescence. Adv. Func. Mater. **26**, 48 (2016).

[189] L. Mullins. Effect of stretching on the properties of rubber. Rubber Chem. Technol. **21**, 2 (1948).

[190] R. E. Webber, C. Creton, H. R. Brown, and J. P. Gong. Large strain hysteresis and mullins effect of tough double-network hydrogels. Macromolecules **40**, 8 (2007).

[191] L. Grebikova, P. Maroni, B. Zhang, A. D. Schlüter, and M. Borkovec. Single-molecule force measurements by nano-handling of individual dendronized polymers. ACS nano **8**, 3 (2014).

[192] L. Grebikova, M. Radiom, P. Maroni, A. D. Schlüter, and M. Borkovec. Recording stretching response of single polymer chains adsorbed on solid substrates. Polymer **102**, (2016).

[193] F. Meng, M. O. Saed, and E. M. Terentjev. Elasticity and relaxation in full and partial vitrimer networks. Macromolecules **52**, 19 (2019).

[194] S. Y. Zheng, H. Ding, J. Qian, J. Yin, Z. L. Wu, Y. Song, and Q. Zheng. Metal-coordination complexes mediated physical hydrogels with high toughness, stick–slip tearing behavior, and good processability. Macromolecules **49**, 24 (2016).

[195] G. Marckmann, E. Verron, L. Gornet, G. Chagnon, P. Charrier, and P. Fort. A theory of network alteration for the Mullins effect. J. Mech. Phys. Solids **50**, 9 (2002).

[196] G. Chagnon, E. Verron, G. Marckmann, and L. Gornet. Development of new constitutive equations for the Mullins effect in rubber using the network alteration theory. Int. J. Solids Struct. **43**, 22-23 (2006).

[197] Y. Yin, N. Bertin, Y. Wang, Z. Bao, and W. Cai. Topological origin of strain induced damage of multi-network elastomers by bond breaking. Extreme Mech. Lett. **40**, (2020).

[198] T. P. Senftle, S. Hong, M. M. Islam, S. B. Kylasa, Y. Zheng, Y. K. Shin, C. Junkermeier, R. Engel-Herbert, M. J. Janik, H. M. Aktulga, et al. The ReaxFF reactive force-field: development, applications and future directions. npj Comput. Mater. **2**, 1 (2016).

[199] P. B. Jayathilaka, T. G. Molley, Y. Haung, M. S. Islam, M. R. Buche, M. N. Silberstein, J. J. Kruzic, and K. A. Kilian. Force-mediated molecule release from double network hydrogels. Submitted to Chem. Commun. (2021).

[200] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. J. Comput. Phys. **117**, 1 (1995).

[201] P. Dauber-Osguthorpe, V. A. Roberts, D. J. Osguthorpe, J. Wolff, M. Genest, and A. T. Hagler. Structure and energetics of ligand binding to proteins: Escherichia coli dihydrofolate reductase-trimethoprim, a drug-receptor system. Proteins **4**, 1 (1988).

[202] A. Behn, P. M. Zimmerman, A. T. Bell, and M. Head-Gordon. Efficient exploration of reaction paths via a freezing string method. J. Chem. Phys. **135**, 22 (2011).

[203] S. Mallikarjun Sharada, P. M. Zimmerman, A. T. Bell, and M. Head-Gordon. Automated transition state searches without evaluating the Hessian. J. Chem. Theory Comput. **8**, 12 (2012).

[204] M. Giaquinta and S. Hildebrandt. *Calculus of Variations I*. Springer, Berlin, 2004.

[205] J. M. Powers and M. Sen. *Mathematical Methods in Engineering*. Cambridge University Press, New York, 2015.

[206] R. P. Kanwal. *Linear Integral Equations*. Springer Science & Business Media, New York, 2013.

[207] M. Rahman. *Integral Equations and Their Applications*. WIT press, Boston, MA, 2007.

[208] J. A. Cochran. *The Analysis of Linear Integral Equations*. McGraw-Hill, New York, 1972.

[209] W. V. Lovitt. *Linear Integral Equations*. McGraw-Hill, New York, 1924.

[210] P. E. Rouse Jr. A theory of the linear viscoelastic properties of dilute solutions of coiling polymers. J. Chem. Phys. **21**, 7 (1953).

[211] J. D. Ferry, R. F. Landel, and M. L. Williams. Extensions of the Rouse theory of viscoelastic properties to undiluted linear polymers. J. Appl. Phys. **26**, 4 (1955).

[212] Q. Chen, G. J. Tudryn, and R. H. Colby. Ionomer dynamics and the sticky Rouse model. J. Rheo. **57**, 5 (2013).

[213] Q. Chen, C. Huang, R. Weiss, and R. H. Colby. Viscoelasticity of reversible gelation for ionomers. Macromolecules **48**, 4 (2015).

[214] X. Zhang, Y. Vidavsky, S. Aharonovich, S. J. Yang, M. R. Buche, C. E. Diesendruck, and M. N. Silberstein. Bridging experiments and theory: isolating the effects of metal–ligand interactions on viscoelasticity of reversible polymer networks. Soft Matter **16**, 37 (2020).

[215] N. Jiang, H. Zhang, P. Tang, and Y. Yang. Linear Viscoelasticity of Associative Polymers: Sticky Rouse Model and the Role of Bridges. Macromolecules **53**, 9 (2020).