

UC Merced

UC Merced Electronic Theses and Dissertations

Title

Mathematical Modeling of Biological Systems: Numerical Methods and Applications to Protein Dynamics and Prion Inheritance

Permalink

<https://escholarship.org/uc/item/3pf755ph>

ISBN

9798197864406

Author

Kucherova, Anna

Publication Date

2026-04-24

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, MERCED

**Mathematical Modeling of Biological Systems:
Numerical Methods and Applications to Protein
Dynamics and Prion Inheritance**

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Applied Mathematics

by

Anna Kucherova

Committee in charge:

Suzanne Sindi, Chair

Maxime Theillard

Arnold Kim

Tomas Rube

2026

The dissertation of Anna Kucherova is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

(Maxime Theillard)

(Arnold Kim)

(Tomas Rube)

(Suzanne Sindi, Chair)

University of California, Merced

2026

TABLE OF CONTENTS

	Signature Page	iii
	List of Figures	vii
	List of Tables	xiv
	Acknowledgements	xv
	Curriculum Vitae	xvi
	Abstract	xix
Chapter 1	Introduction	1
Chapter 2	Least Squares Discretization Method: A Robust and Versatile Meshless Paradigm for Elliptic PDEs	5
	2.1 Introduction	5
	2.2 LSQD	8
	2.2.1 Method Presentation & Implementation	8
	2.2.2 Constructing a Neighborhood	10
	2.2.3 Constructing & Inverting the System	11
	2.2.4 One-Dimensional Proof of Concept	12
	2.3 Extension to Higher Dimensions	13
	2.3.1 Basis Functions in 2D	15
	2.3.2 Neighborhood Construction	15
	2.4 Results	15
	2.4.1 Problem Setup	16
	2.4.2 Classical Boundary Conditions: Dirichlet, Neumann, and Robin	18
	2.4.3 Degenerate Cases	18
	2.4.4 Parameter Sensitivity Analysis	20
	2.4.5 Computational Time	21
	2.4.6 Comparison to Finite Volume Method	22
	2.4.7 Error Estimator	22
	2.5 Conclusion	23
Chapter 3	Computational Modeling of Protein Conformational Changes - Application to the Opening SARS-CoV-2 Spike	27
	3.1 Introduction	27
	3.2 Opening of the SARS-CoV-2 Spike Protein	29
	3.2.1 Simulated Trajectory	29
	3.2.2 Relative Opening Measurement	29

	3.2.3	Magnitude of the Conformation Change	30
	3.3	Mathematical Construction	30
	3.3.1	Numerical Method	33
	3.4	Results	34
	3.4.1	Spike Protein Core	36
	3.4.2	Electric Cloud	36
	3.4.3	Electric Field	36
	3.5	Conclusion	39
Chapter 4		Spatially Resolved Multi-Generation Modeling of Prion Aggregate Inheritance in Budding Yeast	42
	4.1	Introduction	42
	4.2	Mathematical Modeling	44
	4.2.1	Governing Equations	45
	4.2.2	Level Set Representation	46
	4.3	Numerical Method	47
	4.3.1	Finite Volume Discretization	47
	4.3.2	Geometry Representation and Grid Structure	48
	4.4	Simulation Setup and Parameter Calibration	49
	4.4.1	Simulation Parameters	49
	4.4.2	Diffusion Coefficients (FLIP fitting)	49
	4.4.3	Growth Rate Determination	52
	4.5	Preliminary Multi-Generation Results	54
	4.5.1	Inter-Generational Initialization	54
	4.5.2	Reaction-Free Validation	55
	4.5.3	Reaction-Diffusion System with a Nonlinear Source	55
	4.6	Lineage Construction and HPC Implementation	56
	4.6.1	Binary Lineage Representation	56
	4.6.2	Parallelization Strategy	57
	4.7	Lineage Simulation Results	58
	4.7.1	Full Lineage Trajectories	58
	4.7.2	Aggregate Mass Fraction	60
	4.7.3	Generation-Level Statistics	60
	4.7.4	Geometric Decay of Aggregate Burden Across Generations	67
	4.7.5	Lineage Dilution Across the Parameter Space	69
	4.8	Discussion	72
Chapter 5		Conclusion and future work	76
	5.1	Summary of Contributions	76
	5.1.1	A New Meshless Method for Elliptic PDEs (LSQD)	76
	5.1.2	Electrostatic Dynamics of the SARS-CoV-2 Spike Protein	77

5.1.3	Multi-Generation Prion Aggregate Inheritance in Budding Yeast	77
5.2	Limitations	78
5.3	Future Directions	79
5.4	Closing Remarks	80
	Bibliography	80
Appendix A	Appendix for Chapter 4	91

LIST OF FIGURES

Figure 2.1:	We represent the solution of a differential equation as a set of local expansions, denoted by $u_i(x)$, which are centered at arbitrary positions $x_{i=1..N}$. To determine the coefficients $\alpha_{i=0..N}^{q=1..Q}$ of these expansions, we create an overdetermined discrete system by evaluating the differential equation at its neighbors and enforcing continuity conditions between neighboring local representations. We define the final coefficient vector as the least squares solution to the overdetermined system.	8
Figure 2.2:	1D Method - solution representation for increasing spatial and polynomial resolution. We start from a randomly generated set of points $x_{i=1..10}$ and then recursively split the grid by adding one grid point exactly between each existing pair of direct neighbors. The exact solution is depicted as a dashed line.	12
Figure 2.3:	1D Method - (a) h-P convergence analysis for Poisson's equation with Dirichlet boundary conditions. L^∞ error and corresponding Estimated Order of Accuracy (EOA). (b) Condition number of the least squares matrix $A^T A$. (c) Condition number of the preconditioned least squares matrix, using an incomplete cholesky decomposition.	12
Figure 2.4:	Extending to higher dimensions: domain of interest Ω , and local vicinities Ω_i, Ω_j . The neighborhoods V_i, V_j are defined as the set of points contained in the corresponding vicinities.	14
Figure 2.5:	Illustration of neighborhood construction around a reference cell (in blue) for polynomial basis order $P = 4$. The process is carried out one layer at a time until enough neighbors are identified to satisfy the system size and independence conditions. When the reference cell lays next to the interface, the neighbors are chosen within Ω (bottom row).	16
Figure 2.6:	Grids and domains used for the convergence analysis - For each boundary condition type (Dirichlet, Neumann, Mixed Dirichlet Robin), enforced on a square (top rows) or the octofoil domain (bottom rows), we will consider both uniform (row 1 and 3) and adaptive grids (rows 2 and 4). This representation will be used to label all convergence figures hereafter. <i>Credit: this figure was created by Gbocho Terasaki.</i>	17
Figure 2.7:	Convergence of L^∞ global error for three boundary condition types.	19
Figure 2.8:	Local error convergence: Octofoil domain with Dirichlet boundary conditions on an adaptive grid. Error decays to zero both as polynomial order increases and as the grid is refined.	20

Figure 2.9:	Degenerate cases: (a) Robin and Dirichlet mixed boundary conditions, (b) Dirichlet boundary condition with highly negative μ .	21
Figure 2.10:	Neumann boundary condition on the octofoil domain with a uniform grid: impact of $\frac{a}{\mu}$ ratios. (a) Global error trend for increasing split number for each polynomial basis. (b) Impact of computational time on global error compared to FVM. (c) Illustration of LSQD vs FVM computational error for increasing grid refinement.	25
Figure 2.11:	Error Estimator Analysis on an Adaptive Grid. (a) Three distinct node locations and what is considered their respective surrounding/neighbor cells. (b) Comparison between computational order of accuracy (colored lines) and predicted order of accuracy (dashed lines). (c) Illustration of computational and predicted error for increasing polynomial order and grid refinement. (<i>Credit: Gbocho Terasaki helped create portions of this figure.</i>)	26
Figure 3.1:	SARS-CoV-2 Spike protein visualized in its closed (a) and open (b) states. The four pairs of atoms, along with the separating distances, used to quantify the gap opening are represented with straight solid colored lines.(c) The relative distance (Å) between arbitrarily chosen atom pairs are represented as a function of time. One atom is selected on the receptor-binding domain (RBD), and the other opposite the RBD across the opening of the spike protein. This distance is relative to the average distance between each respective atom pair (1,2,3, and 4 in the graph), hence the term 'relative distance'. (d) The root mean squared deviation (RMSD) is calculated as a function of time with frame 1 (closed state) as the reference frame.	31
Figure 3.2:	(a) Visualization of the top view of the Spike protein molecule (colored by monomer). Color denotes the three monomers (b) A cutaway diagram of a general molecular surface depicting the Van der Waals Surface (vdWS), solvent-excluded surface (SES), and solvent-accessible surface (SAS). (c) SES representation of the Spike protein oriented the same way as Figure (a) (top view). . . .	32
Figure 3.3:	(a) Evolution of the spike's Solvent Excluded Surface and electrostatic potential as the maximal resolution increases. (top down view in the closed configuration (frame 1)). (b – d) Convergence of the interface representation (e_Γ), surface normal electric potential (e_E), and surface potential (e_ψ).	35

Figure 3.4:	(a) Potential map of the SARS-Cov-2 spike open (frame 1), intermediate (frame 35), and close configuration (frame 71). (b) Evolution of the negatively charged core as the spike opens. (c) Repulsive electric cloud generated by the negatively charged core. (d – e) Characterization of the negatively charged core χ and repulsive electric cloud $\mathcal{C}$: relative size (d) and potential (e) variations over the entire molecular trajectory.	37
Figure 3.5:	(a) Portion of the ACE2 receptor displaying surface charge (left). ACE2 and RBD of the spike protein connected (right). (b) Binding site of ACE2 highlighted in green (left). Binding site of ACE2 highlighted within the ACE2 (orange) and RBD (yellow) connection (right). (c) SARS-Cov-2 spike in open conformation with RBD (yellow) and binding site, $\mathcal{B}$ (green) highlighted. (d – e) Characteristics of the binding site, $\mathcal{B}$, of the spike protein. (d) Relative variance of the area and absolute curvature along with average potential. (e) average positive and negative potential.	38
Figure 3.6:	Streamlines and direction of the electric field in the close (top) and open (bottom) configurations. As the spike opens, the streamlines emanating from the upper part of the molecule are observed to open as well. This effect is particularly noticeable for the lines emerging from the RBD domain (opening extremity on the right).	39
Figure 3.7:	Streamlines of the electric field ($\mathbf{E} = -\nabla\hat{\psi}$) in the close (left) and open (right) configurations over the entire computational domain. The portion of the streamlines where the electric field is pointing away from the protein (<i>i.e.</i> $\nabla\psi \cdot \mathbf{x} = -\mathbf{E} \cdot \mathbf{x} < 0$), and therefore negative charges would be drawn to the it, is depicted in blue.	40
Figure 4.1:	Three-dimensional simulation of a dividing yeast cell with a splitting nucleus. As time progresses, a daughter cell buds from the mother cell, grows, and ultimately detaches.	44
Figure 4.2:	FLIP data from [113]. Two conformations of the yeast prion protein ($[\text{PSI}+]^{\text{Weak}}$ and $[\text{PSI}+]^{\text{Strong}}$) are depicted in yellow and blue, respectively. The remaining data ($\text{psi}-$ and diffuse) are non-prion strains and are seen to diffuse into the daughter cell more easily.	50
Figure 4.3:	Fitting experimental data. The data provided is used to fit the decay model for all prion and non-prion strains. The diffusion which minimizes the error ($\ L\ _{\infty}$) between the model and experimental data, for each respective case, is emphasized as a black point.	51

Figure 4.4:	Model Fit: Comparison of how the experimental data fits the chosen model for each of the strains. Here, the respective optimal diffusion coefficients are used for the predicted intensity, meaning that 4.18 now has a specific value of D for each strain.	52
Figure 4.5:	Fraction of monomer mass in the daughter cell, $F_A(t)$, for growth rates $r \in \{0.1, 0.2, 0.4, 0.6, 0.8\}$, with diffusion coefficients fixed at their FLIP-fitted values. Lower growth rates fail to reach a steady state within the 190-minute simulation window, indicating that the monomer concentration in the daughter cell has not yet reached a steady state.. At $r = 0.8$, F_A plateaus by the end of the simulation, satisfying the steady-state criterion used to select this value for all preliminary simulations.	53
Figure 4.6:	Time progression of the cell geometry for the growth rate $r = 0.4$, for a system with no reaction, only diffusion and logistic growth. The timestep is defined as ts , they are, respectively, $t = 0, 20, 81, 190$ minutes.	54
Figure 4.7:	Two-generation lineage tree for the diffusion–growth system ($\gamma_{AB} = \gamma_{BA} = 0$, $r = 0.8$). As expected in the absence of aggregation, mother and daughter cells reach identical average concentrations once the daughter has grown to maturity.	55
Figure 4.8:	Three-generation lineage tree for the reaction–diffusion–logistic system. Diffusion coefficients are taken from the FLIP data fit in Section 4.4.2.	56
Figure 4.9:	Mean aggregate concentration ψ_B over time for all cells in a five-generation lineage tree, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.8$ fixed. Each curve corresponds to a single simulated cell; color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). Sharp downward steps mark division events, after which both mother and daughter enter a recovery phase. Trajectories are ordered by replicative age across all parameter combinations, with older-lineage cells carrying persistently higher aggregate burden. Corresponding figures for $r = 0.5$ and $r = 0.2$ are provided in Appendix A.	59

- Figure 4.10: Aggregate mass fraction $F_B(t) = 2\psi_B/(\psi_A + 2\psi_B)$ over time for all cells in a five-generation lineage, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.8$ fixed. Color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). At each division event ($t = 70$), the newly detached daughter cell exhibits a transient upward spike in F_B reflecting her small initial volume; as she grows and acquires additional monomer the fraction falls to its post-division level. The mother experiences a modest dilution at division but remains at a substantially higher F_B than the daughter, reflecting the asymmetric partitioning of aggregate at cytokinesis. Increasing γ_{BA} shifts the long-term balance from aggregate retention toward dilution across all rows, while increasing γ_{AB} raises the overall level of F_B across all columns. Generational ordering persists across all nine parameter combinations. Corresponding figures for $r = 0.5$ and $r = 0.2$ are provided in Appendix A. 61
- Figure 4.11: Box plots of final mean aggregate concentration ψ_B grouped by generation index, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.8$ fixed. Orange lines indicate the median; filled circles indicate the mean; open circles indicate outliers. Generation 0 contains only the single root cell and appears as a point. Median ψ_B decreases monotonically with generation index across all parameter combinations, confirming that lineage memory is robust to variation in both dimerization and degradation rates. Increasing γ_{BA} lowers overall burden while increasing γ_{AB} raises it; neither parameter disrupts the generational ordering. 64
- Figure 4.12: Box plots of final mean aggregate concentration ψ_B grouped by generation index, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.5$ fixed. 65
- Figure 4.13: Box plots of final mean aggregate concentration ψ_B grouped by generation index, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.2$ fixed. 66
- Figure 4.14: Geometric decay model $\psi_B(g) = a \cdot \alpha^g$ fitted to final-time aggregate concentrations across all 27 parameter combinations, organized as a 3×3 grid with γ_{AB} varying across rows and r varying across columns. Colored scatter points show generation means for each γ_{BA} value. Dashed lines show the fitted decay curve with annotated parameters a , α , and R^2 67

Figure 4.15: Full lineage trajectories for $\gamma_{AB} = 0.01$, $\gamma_{BA} = 0.008$, $r = 0.2$ extended to generation depth $G = 8$. Color encodes generation index from earliest (purple) to latest (yellow-green). The founder-line cell maintains persistently high aggregate burden while the youngest-daughter trajectories converge toward near-zero by generations 6–7, consistent with the geometric decay model prediction of $g^* \approx 6.5$	69
Figure 4.16: Heatmap of the dilution ratio $\mathcal{R}$ across the full parameter sweep, for $r \in \{0.2, 0.5, 0.8\}$ (panels, left to right), $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (x-axis), and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (y-axis). Values of $\mathcal{R}$ close to 1 indicate little divergence across the lineage; values close to 0 indicate strong dilution in the youngest-generation cells relative to the founder.	70
Figure A.1: Mean aggregate concentration ψ_B over time for all cells in a five-generation lineage tree, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.5$ fixed. Each curve corresponds to a single simulated cell; color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). Sharp downward steps mark division events, after which both mother and daughter enter a recovery phase. Trajectories are ordered by replicative age across all parameter combinations, with older-lineage cells carrying persistently higher aggregate burden.	92
Figure A.2: Mean aggregate concentration ψ_B over time for all cells in a five-generation lineage tree, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.2$ fixed. Each curve corresponds to a single simulated cell; color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). Sharp downward steps mark division events, after which both mother and daughter enter a recovery phase. Trajectories are ordered by replicative age across all parameter combinations, with older-lineage cells carrying persistently higher aggregate burden.	93

- Figure A.3: Aggregate mass fraction $F_B(t) = 2\psi_B/(\psi_A + 2\psi_B)$ over time for all cells in a five-generation lineage, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.5$ fixed. Color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). At each division event ($t = 70$), the newly detached daughter cell exhibits a transient upward spike in F_B reflecting her small initial volume; as she grows and acquires additional monomer the fraction falls to its post-division level. The mother experiences a modest dilution at division but remains at a substantially higher F_B than the daughter, reflecting the asymmetric partitioning of aggregate at cytokinesis. Increasing γ_{BA} shifts the long-term balance from aggregate retention toward dilution across all rows, while increasing γ_{AB} raises the overall level of F_B across all columns. Generational ordering persists across all nine parameter combinations. 94
- Figure A.4: Aggregate mass fraction $F_B(t) = 2\psi_B/(\psi_A + 2\psi_B)$ over time for all cells in a five-generation lineage, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.2$ fixed. Color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). At each division event ($t = 70$), the newly detached daughter cell exhibits a transient upward spike in F_B reflecting her small initial volume; as she grows and acquires additional monomer the fraction falls to its post-division level. The mother experiences a modest dilution at division but remains at a substantially higher F_B than the daughter, reflecting the asymmetric partitioning of aggregate at cytokinesis. Increasing γ_{BA} shifts the long-term balance from aggregate retention toward dilution across all rows, while increasing γ_{AB} raises the overall level of F_B across all columns. Generational ordering persists across all nine parameter combinations. 95

LIST OF TABLES

Table 2.1:	Convergence summary - Average estimated order of convergence for various boundary conditions, configurations, and polynomial orders $P = 2, \dots, 5$	18
Table 3.1:	Problem parameters and non-dimensional numbers.	33

ACKNOWLEDGEMENTS

I owe the existence of this dissertation, in no small part, to two people who shaped my path before I knew where it was going. Dr. Arnold Kim convinced an undergraduate chemistry major with dreams of becoming a forensic scientist to give mathematics a try. I am not sure whether either of us knew what he was starting. Dr. Noemi Petra gave me my first taste of research and made graduate school feel not just possible but exciting. Between the two of them, my trajectory was set before I fully understood where it would lead.

That trajectory led me to my graduate advisor, Dr. Maxime Theillard, who took a chance on me at the start of a global pandemic and built something with me that neither of us could have anticipated. Even after moving to France, he remained a dedicated and generous mentor, and the foundation he helped me lay is present throughout this dissertation. When circumstances changed, Dr. Suzanne Sindi stepped in as my advisor with characteristic grace and rigor — she has known me since my undergraduate years, and her guidance, especially through the final stretch of this journey, meant more than I can adequately express. I am also grateful to Dr. Tomas Rube for joining my committee without hesitation and engaging genuinely with the work presented here. To my wonderful friend Daniel Miliate and my MACS research group colleagues: Drs. Masato Terasaki, Cayce Fylling, Adam Binswanger, Scott West, and Matthew Blomquist — thank you for the camaraderie, the conversations, and for making the day-to-day of graduate school something to look forward to. Daniel deserves special mention not only for the many hours spent thinking through the HPC pipeline that made Chapter 4 possible, but for being a longtime friend who has shown up for me in more ways than one throughout this journey.

To my parents and grandparents — your prayers, love, encouragement, and sacrifices made all of this possible. You had faith that I could do this even when I wasn't sure myself, and that belief carried me further than you know.

And to Alisa Kravchuk — my best friend of over a decade and my chosen sister. You have been beside me through every version of this journey: the excitement, the doubt, the late nights, the small victories that nobody else saw. You celebrated with me when things went right and steadied me when they didn't. A PhD is supposed to be a solitary endeavor, but it never felt that way because of you. This one is, unconditionally, for you, too.

This work was made possible in part by the generous support of UC Merced's Graduate Division, the School of Natural Sciences, and the Applied Mathematics Department. I gratefully acknowledge fellowship support from the Applied Mathematics Summer Fellowship (2021), the NSF RTG Fellowship (Grant No. DMS-1840265, 2023), and a Travel Fellowship (2025). Computational resources were provided by the Pinnacles and MERCED clusters at the University of California, Merced, supported by NSF Grants No. ACI-2019144 and No. ACI-1429783, respectively.

ANNA KUCHEROVA

University of California, Merced
Ph.D. Candidate | akuchero@ucmerced.edu

EDUCATION

University of California, Merced	Merced, CA
Ph.D. Applied Mathematics	May 2026
M.S. Applied Mathematics	May 2025
B.S. Applied Mathematical Sciences, Computation and Data Sciences	May 2020
<i>Magna Cum Laude</i>	

RESEARCH INTERESTS

Ph.D. candidate in Applied Mathematics with a focus on predictive modeling, large-scale computational experiments, and data-driven hypothesis testing. Experienced in end-to-end research workflows including model design, parameter exploration, simulation-based inference, and building reproducible computational pipelines (HPC, automation, version control). Strong interest in translating theoretical insight into robust quantitative research systems.

RESEARCH EXPERIENCE

Lawrence Berkeley National Laboratory	Berkeley, CA
<i>Joint Genome Institute</i>	May 2024 - July 2024
<i>Mentors: Dr. Daniela Cassol, Dr. Kjersten Fagnan, Jeff Froula</i>	

- Managed version-controlled code repositories on GitLab to streamline collaboration and reproducibility
- Developed an automated pipeline for multi-generational yeast cell division simulations using Bash scripting
- Enhanced expertise in high-performance computing (HPC) with a focus on parallelization and workflow automation

University of California, Merced	Merced, CA
<i>Graduate Student Researcher</i>	August 2020 - Present
<i>PIs: Dr. Suzanne Sindi, Dr. Maxime Theillard</i>	

<u>Prion Aggregation in Yeast Cells</u>	August 2022 - Present
--	-----------------------

- Developed a computational model for reaction-diffusion-logistic dynamics in evolving 3D yeast geometries
- Decomposed simulations into ensembles of ~ 60 independent single-core jobs, enabling scalable execution and efficient use of HPC resources
- Implemented task-based parallelism in C++ using OpenMP, while constraining cluster execution to one core per job for reproducibility
- Designed simulation experiments to study sensitivity to biological parameters and identify regime changes across many generations of cell division
- Evaluated model behavior via stability analysis, sensitivity to discretization and parameters, and validation on simplified test scenarios

<u>Least Squares Discretization Method (LSQD)</u>	May 2020 - May 2024
--	---------------------

- Designed and implemented a novel least squares discretization framework for elliptic PDEs with minimal user tuning
- Conducted systematic convergence and stability studies, achieving h - p convergence and numerical errors as low as 10^{-11} on benchmark problems
- Compared performance against classical numerical methods, demonstrating competitive runtime and robustness

<u>Modeling SARS CoV-2 Spike Protein Dynamics</u>	May 2020 - November 2020
--	--------------------------

- Created a computational pipeline for studying deforming proteins
- Analyzed SARS-CoV-2 spike from its closed to open conformation; calculated the electrostatic potential on and around the protein
- Modeled the electric field and electric potential on and around the spike protein
- Identified a likely receptor binding domain corresponding to the spike-ACE2 interaction

**Summer Undergraduate Research Institute
UROC REU Participant**

Mentor: Dr. Noemi Petra

Merced, CA
May 2019 - August 2019

- Investigated an inverse problem governed by PDEs
- Estimated a parameter from available noisy data via inversion techniques
- Explored connection between eigenfunctions of the Hessian and the PDE to show analytically and numerically that eigenvalues decay exponentially

**University of California, Merced
DESCARTES Scholars Program**

Mentors: Dr. Fabian Santiago, Dr. Arnold Kim

Merced, CA
May 2019 - August 2019

- Developed a simulated annealing based algorithm to solve an NP-hard scheduling problem
- Used unsupervised learning methods to generate optimal schedule with minimized faculty dissatisfaction
- Created cost effective procedure for generating unbiased schedules in minutes

**Lawrence Livermore National Laboratory
Data Science Challenge**

Mentor: Dr. Sabah Ul-Hasan

Livermore, CA
May 2019 - June 2019

- Trained and evaluated machine-learning models to predict antibody-antigen interactions, achieving ~79% predictive accuracy
- Assessed how sequence mutations affected predicted antibody-antigen interactions

PUBLICATIONS

1. A. Kucherova, S. Strango, S. Sukenik, M. Theillard. Computational modeling of protein conformational changes - Application to the opening SARS-CoV-2 spike. *Journal of Computational Physics*, 444:110591, 2021
2. A. Kucherova, G. M. Terasaki, S. Strango, M. Theillard. Least Squares Discretizations (LSQD): a robust and versatile meshless paradigm for solving elliptic PDEs. 2024 (*Submitted, In Review*) (*View on arXiv: [arXiv: 2406.13872v1](https://arxiv.org/abs/2406.13872v1)*)

CONTRIBUTED TALKS AND POSTER PRESENTATIONS

1. **Computational Modeling of Prion Aggregation Dynamics in Dividing Yeast Cells.** AMS Spring Western Sectional Meeting, San Luis Obispo, California, 2025
2. **Streamlining Computational Studies of Misfolding Proteins via High Performance Computing.** Joint Genome Institute, Lawrence Berkeley National Laboratory, 2024
3. **Modeling the Opening SARS-CoV-2 Spike: An Investigation of its Dynamic Electro-Geometric Properties.** SIAM Conference on Computational Science and Engineering, Amsterdam, The Netherlands, 2023
4. **Modeling the Opening SARS-CoV-2 Spike: An Investigation of its Dynamic Electro-Geometric Properties.** APS March Meeting, 2021 (Virtual)
5. **A Genetic Algorithm with Simulated Annealing Hybrid (GSA) for University Timetabling.** Nebraska Conference for Undergraduate Women in Mathematics (NCUWM), Lincoln, Nebraska, 2020
6. **A Genetic Algorithm with Simulated Annealing Hybrid (GSA) for University Timetabling.** Joint Mathematics Meetings (JMM), Denver, Colorado, 2020
7. **The Influence of Data on the Reconstruction of Parameters in an Inverse Problem.** Summer Undergraduate Research Symposium, UC Merced, 2019
8. **Selection and Engineering of Thermodynamically Relevant Antigen-Antibody Binding Features Using Machine Learning.** Lawrence Livermore National Laboratory, 2019

AWARDS

Applied Mathematics Travel Fellowship, 2025
Excellence in Teaching, 2024
NSF RTG Fellowship, 2023
Applied Mathematics Summer Fellowship, 2021
Outstanding Student Award, 2020
Chancellor's Honor List, 2020

UC Regent's Scholarship, 2016 – 2020
American Mathematical Society Travel Grant, 2019
DESCARTES Program Stipend, 2019
UROC REU Summer Fellowship, 2019
Chancellor's Honor List, 2019

TEACHING EXPERIENCE

University of California, Merced
Teaching Assistant

Merced, CA
August 2020 - Present

Over five years of experience designing and leading discussion sections, mentoring students, and supporting instruction across lower- and upper-division mathematics courses. Skilled in fostering inclusive, engaging learning environments that promote conceptual understanding and problem-solving confidence.

- Math 21 - Calculus I, 1 semester
- Math 32 - Probability and Statistics, 1 semester
- Math 130 - Numerical Analysis, 1 semester
- Math 132 - Numerical Methods for Differential Equations, 1 semester
- Math 141 - Linear Analysis, 6 semesters

University of California, Merced
Undergraduate Instructional Assistant

Merced, CA
September 2018 - June 2020

- Provided support for undergraduate math courses: all lower division and most upper division
- Implemented the Socratic Method of tutoring

TECHNICAL SKILLS AND COURSEWORK

Relevant Coursework: Bioinformatics; Scientific Computing (C++); Stochastic Processes; Modern Applied Statistics; Numerical Methods for Optimization; Numerical Linear Algebra; Partial Differential Equations
Programming: C++, MATLAB, R, Bash, OpenMP (task-based), L^AT_EX, Python (NumPy, Pandas)
Data & Systems: Linux/HPC (Slurm), Git, GitLab, Jupyter, SQL (in progress), automated experiment pipelines
Software & Tools: Qt, ParaView, CLion, VMD, Anaconda, OmniGraffle, Microsoft Office 365
Languages: English, Russian, Ukrainian

PROFESSIONAL AFFILIATIONS

Society for Industrial and Applied Mathematics (SIAM)

ABSTRACT OF THE DISSERTATION

Mathematical Modeling of Biological Systems: Numerical Methods and Applications to Protein Dynamics and Prion Inheritance

by Anna Kucherova

Doctor of Philosophy in Applied Mathematics

University of California Merced, 2026

Committee Chair: Suzanne Sindi

Scientific discovery depends on our ability to model the physical world accurately — and for many of the most consequential problems in biology and medicine, the geometry of the domain is not a detail to be approximated away but the very mechanism driving the phenomenon of interest. This dissertation develops and applies computational methods for partial differential equations on complex, irregular, and continuously deforming geometries, demonstrating that spatial resolution reveals biological insights that well-mixed or geometry-free approaches fundamentally cannot.

The first contribution is the Least Squares Discretization (LSQD) method, a meshless framework for solving elliptic PDEs on scattered point distributions without requiring a mesh, a weak formulation, or quadrature rules. The method achieves h-P convergence across a broad range of geometries, boundary conditions, and problem configurations, remains robust in degenerate cases where traditional formulations may be ill-posed, and includes a built-in error estimator, all while requiring minimal mathematical background to implement.

The second contribution applies this spatially resolved framework to characterize how the electrostatic surface potential of the SARS-CoV-2 spike protein evolves as it opens to expose its receptor binding domain. Solving the nonlinear Poisson-Boltzmann equation across 71 frames of a molecular dynamics trajectory, we find that geometric reorganization of the spike actively reshapes the electric field in a manner that may facilitate selective binding to the ACE2 receptor: a conclusion accessible only through spatially resolved computation on a deforming surface.

The third contribution develops the first spatially resolved, multi-generation computational model of prion aggregate inheritance in budding yeast. Solving a reaction-diffusion PDE system on a deforming cell geometry across a parallelized binary lineage tree, we demonstrate that aggregate burden decreases monotonically with generation index across numerous biochemical parameter combinations investigated, that this lineage memory is deterministic and geometric in origin, and that the per-generation dilution follows a geometric decay law whose parameters predict aggregate clearance in deeper simulations with remarkable accuracy.

Together these contributions establish that geometry is not merely a computational challenge to be overcome, but an essential feature of the science itself.

Introduction

Scientific discovery has long been driven by our ability to model the physical world mathematically. At the heart of this endeavor lies a deceptively simple challenge: most phenomena of interest are governed by partial differential equations (PDEs) posed on domains whose geometry is complex, irregular, or continuously evolving. From the electrostatic fields surrounding a deforming protein to the spatial dynamics of aggregating molecules inside a dividing cell, geometry is not a peripheral concern; instead, it is frequently the mechanism through which the underlying biology operates. A protein that changes shape as it binds a receptor, a cell that partitions its contents asymmetrically as it divides — in both cases, the spatial and geometric structure of the problem is inseparable from its scientific content.

Traditional numerical methods have provided the computational backbone of science and engineering for decades. Finite element, finite difference, and finite volume methods excel at solving PDEs on well-defined, static domains, and have produced transformative results across fluid dynamics, structural mechanics, and beyond [1]. Yet their reliance on body-fitted meshes or structured grids presents a fundamental barrier when the domain is geometrically intricate, changes over time, or must be resolved at a sharp interface between regions of qualitatively different physics. Generating a high-quality mesh for such problems is itself a significant undertaking, and for problems where the geometry deforms continuously, as in a dividing cell or an opening protein, remeshing at every timestep is computationally prohibitive. These limitations motivate two parallel lines of inquiry: the development of numerical methods that are geometry-agnostic by design, and the application of spatially resolved PDE frameworks to biological problems where existing well-mixed or geometry-free approaches are fundamentally insufficient.

This dissertation pursues both directions. The unifying theme across all three projects is the conviction that *geometry matters*: that spatial structure, domain shape, and interfacial dynamics are not details to be averaged away, but are essential features of the problems we study. On the methodological side, we develop a new meshless discretization method that solves elliptic PDEs on arbitrary point distributions without

requiring a mesh, a weak formulation, or quadrature rules. On the application side, we bring spatially resolved PDE computation to bear on two biological problems — the electrostatic dynamics of the SARS-CoV-2 spike protein and the multi generation inheritance of prion aggregates in yeast — where the geometry of the domain is not incidental to the science but central to it.

Contributions

A New Meshless Method for Elliptic PDEs (LSQD).

The first contribution of this dissertation is the Least Squares Discretization (LSQD) method, a meshless framework for solving elliptic PDEs on scattered point distributions with arbitrary geometry and boundary conditions. Rather than requiring a mesh, a weak formulation, or quadrature rules, LSQD represents the solution as a family of overlapping local polynomial expansions. The PDE, boundary conditions, and continuity conditions are enforced pointwise and assembled into an overdetermined linear system, whose least squares solution defines the approximation. The method achieves h-P convergence on both uniform and highly adaptive non-graded quadtree grids, handles Dirichlet, Neumann, and Robin boundary conditions, and remains robust in degenerate cases where traditional weak formulations may be ill-posed. A built-in a posteriori error estimator is also proposed, enabling local error assessment without access to an analytical solution. The method requires minimal mathematical background to implement and scales with the number of grid points, making it a practical and accessible alternative to conventional PDE solvers for problems involving complex or irregular geometries. This work is the subject of a manuscript currently available on arXiv [2], with the author of the dissertation as the first author.

Electrostatic Dynamics of the SARS-CoV-2 Spike Protein.

The second contribution applies spatially resolved PDE computation to a problem at the intersection of structural biology and computational biophysics: characterizing how the electrostatic potential of the SARS-CoV-2 spike protein evolves as it transitions from its closed to open conformation. The spike protein must undergo a dramatic conformational change to expose its receptor binding domain (RBD) and bind the human ACE2 receptor, a process critical to viral entry. To characterize the electrostatic consequences of this transition, we combine Markov state model molecular dynamics trajectories from the Folding@home [3] initiative with a high-resolution numerical solver for the nonlinear Poisson-Boltzmann equation, implemented on adaptive octree grids with a sharp level set representation of the protein surface. Solving the governing PDE independently at each of the 71 frames of the opening trajectory, we produce dynamical electrostatic potential maps that reveal dramatic rearrangements of the electric field during spike opening. Most notably, a negatively charged core is exposed as the spike opens, generating a repulsive electric cloud that nevertheless does not envelop the binding site; instead the RBD becomes more positively charged and the

electric field transitions from a globally attracting configuration to one focused specifically around the exposed binding domain. These findings suggest that the geometric reorganization of the spike protein during opening may actively facilitate selective binding to the negatively charged ACE2 receptor. This work was published in the *Journal of Computational Physics* in 2021 [4], with the dissertation author as first author.

Multi-Generation Prion Aggregate Inheritance in Budding Yeast.

The third contribution develops the first spatially resolved, multi-generation computational model of prion aggregate inheritance in budding yeast. Prion aggregates in dividing yeast cells are known to be inherited asymmetrically, with aggregates preferentially retained in the mother cell: a phenomenon established experimentally to arise from diffusion-limited partitioning within the confining geometry of the dividing cell rather than from active transport [5]. Existing mathematical models treat the intracellular environment as spatially homogeneous, prescribing asymmetric partitioning as a parameter rather than deriving it from the underlying spatial dynamics. Building on the finite volume framework of Heydari *et al.* [6], which demonstrated that cell geometry alone can produce asymmetric partitioning in a single division event, we extend this approach to simulate explicit binary lineage trees to generation five and beyond. The model incorporates diffusion of monomer and aggregate species, reversible dimerization, and a logistic growth term, with diffusion coefficients calibrated from fluorescence loss in photobleaching data. A parallelized HPC pipeline automates the inter-generational handoff and distributes simulations across a computing cluster, enabling a structured sweep of biochemical parameters. Across all parameter regimes investigated, the simulations demonstrate that aggregate burden is stratified by replicative age, that sibling correlations are perfect reflecting the deterministic nature of the PDE framework, and that lineage memory is a robust geometric phenomenon that persists regardless of the specific biochemical parameters chosen.

Organization

The remainder of this dissertation is organized as follows. Chapter 2 presents the LSQD method in full, including its theoretical formulation, neighborhood construction, convergence analysis on a variety of grid types and boundary conditions, and a comparison to the finite volume method. Chapter 3 describes the electrostatic dynamics study of the SARS-CoV-2 spike protein, covering the molecular trajectory, the Poisson-Boltzmann mathematical framework, the numerical method and its validation, and the electrostatic results characterizing the the opening of the spike. Chapter 4 develops the multi-generation prion aggregation model, including the governing reaction–diffusion equations, the level set geometry representation, the conservative finite volume discretization, the lineage construction and HPC parallelization strategy, and the full lineage simulation results across a structured parameter sweep. Chapter 5

concludes the dissertation with a synthesis of contributions, a discussion of limitations, and directions for future work.

Least Squares Discretization Method: A Robust and Versatile Meshless Paradigm for Elliptic PDEs

2.1 Introduction

In the ever-evolving landscape of computational science and engineering, the quest for accurate, efficient, and versatile numerical methods remains paramount. Traditional numerical approaches, such as finite element methods, finite difference methods, and finite volume methods, have long dominated the sphere of simulating complex physical phenomena. However, their reliance on elements, structured grids, or finite volumes often presents challenges when handling irregular geometries, dynamic domains, or intricate boundary conditions. In response to these limitations, a trailblazing paradigm emerged - meshless methods [7, 8].

Meshless methods began gaining traction in the 1990s [9, 10, 11, 12, 13, 14]. In recent years, meshless methods have witnessed a surge in popularity across a multitude of disciplines, including computational fluid dynamics, structural mechanics, electromagnetics, and biomedical engineering [15, 16, 17, 18]. Their inherent attributes (*e.g.* mesh generation simplicity, capacity to handle large deformations/distortions and complex geometries [19], etc.), have proven a versatile alternative to the traditional discretization techniques.

Some of the pioneering meshless methods fall into the following categories: kernel-based [7, 14, 20], Galerkin [11, 21, 22, 9, 23, 10, 24, 25, 26, 27, 28], and Finite Difference methods [29, 30]. Meshless Galerkin methods such as the Element-free Galerkin [11, 21], h-p clouds and h-p methods [22], Meshless Local Petrov-Galerkin [9, 10], the Natural Element Method [23, 24, 25], the Voronoi finite volume method [26, 27], and partition of unity method [28], all rely on the weak formulation of the partial differential equation (PDE). Thus, although they are meshless, these methods are not truly “geometry-free” as the weak formulation involves spatial integrals. This

requires approximations of definite integrals, or quadrature rules, over — potentially — intricate geometries. These geometries, themselves, may reap extravagant computational costs (*e.g.* Voronoi partition for the Meshless Natural Element method [23, 24, 25] or any Voronoi Finite Volume Method [26, 27]). Undeniably, the theoretical foundation for the aforementioned methods allows for a comprehensive mathematical study, but may serve as a deterrent for interested non-experts.

Although commonly used in scattered data interpolation [31], Kansa became one of the first to utilize radial basis functions (RBF) to solve fluid dynamics problems [13]. Theoretical condition number estimates indicated that some meshless methods using RBFs may be unstable [32]. Fortunately, Kansa was able to successfully circumvent the ill-conditioned system by using improved multi-quadrics and adaptive techniques [33]. Kansa’s method represents the solution as a sum of global radial functions, centered at a set of points. The coefficients in this decomposition are obtained by interpolating the PDE at these points. While in general, the interpolation system can be ill-posed (Mairhuber-Curtis theorem [34, 35]), when using appropriate radial functions, this system can be proven to be non-singular [36, 37]. Not solely is it straightforward to implement, this method is particularly well-suited for problems involving complex curved surfaces (*e.g.* such as reaction-diffusion patterns over biological entities such as frog species [38], red blood cells [39], or 3-D spherical shells [40] used in geophysical calculations). It has since been applied in the computational fluid dynamics realm [41] such as the solution of Navier–Stokes equations [42], porous media flow [43], and the solution of solid-liquid phase change problems [44]. Unfortunately, the convergence of Kansa’s method is still unproven. Because the representation is global, the interpolation system is full and consequently its inversion rapidly becomes prohibitive. To address this limitation, domain decomposition techniques can be employed [45], but these abandon the meshless nature of the domain. Using local RBFs [46], the local Kansa method [43, 47] offers an alternative approach to reduce the complexity. However, this approach still relies on a domain decomposition to distinguish between inner and border nodes, further requiring functional iterations on either node sets.

One of the most pervasive methods in computational fluid dynamics is the finite point method — a meshfree method for solving PDEs on scattered distributions of points. This method is considered a generalization of the finite difference method, created for arbitrary irregular grids [29, 48]. Despite the fact that this method requires a much lower mathematical background, it entails a more tedious implementation as the discretization coefficients are obtained from high-order Taylor expansions over non-uniform neighborhoods [30]. With the Radial Basis Function Finite Difference (RBF-FD [49]), these coefficients are computed at each point by interpolating the solution over a small neighborhood using a radial basis enriched with polynomial functions. The RBF ensures good conditioning while the polynomial enrichment grants convergence. The radial basis function-based differential quadrature approach (RBF-DQ [50]) proposes a similar strategy and has been employed to simulate compressible flows [50] and heat conduction [51] problems.

Despite offering significant advantages (*e.g.* flexibility in handling complex

geometries, avoiding computational overhead associated with mesh generation), meshless methods possess considerable limitations. One notable limitation is that most meshless methods must use background cells to integrate a weak form over the entire problem domain. The requirement of background cells for integration makes the method not “truly” meshless. Another challenge lies in enforcing boundary conditions accurately, especially so, as the degree of freedom may not be adapted to the problem geometry. As pointed out by Atluri and Zhu in their work on meshless local Petrov-Galerkin methods, ensuring the satisfaction of boundary conditions in meshless methods can be non-trivial, potentially leading to inaccuracies in the solution; they attempt to mitigate this by implementing a penalty function [9]. Additionally, the lack of a structured mesh can result in reduced convergence rates and increased computational costs, particularly in problems with high gradients or regions of varying solution behavior [11]. This introduced the need for a consistency technique involving the combination of derivatives in different directions. Furthermore, the computational expense of constructing and solving large systems of equations can be prohibitive for certain applications, limiting the scalability of meshless methods. This issue has to be ameliorated by obtaining weighted orthogonal basis functions via Schmidt orthogonalization [52]. These discrete limitations underscore the importance of carefully assessing the suitability of meshless methods for specific numerical analysis tasks and considering alternative approaches where appropriate. Clearly, it would be a quixotic undertaking to create an encyclopedic inventory for the trajectory of all meshless methods, so for an exhaustive analysis, we direct interested readers to the following reviews [53, 54, 55].

In amalgamating the desired advantages of meshless methods: accessibility, simplicity, h-P convergence, and low complexity, while simultaneously relinquishing the need for rigorous theoretical validation, we present the Least Squares Discretization method (LSQD). This approach only assumes that around each point, the solution can be decomposed on a local basis. No weak or strong formulation of the PDE, nor numerical analysis theory is needed to discretize the problem, instead, the PDE and boundary conditions are point-wise evaluated, and the local expansions are connected through pointwise continuity conditions. All of these conditions are expressed as linear equations involving evaluations of the known local basis functions and the unknown decomposition coefficients. They are grouped into a rectangular linear system, from which the vector of unknown coefficients is defined as the least squares solution. The method is, therefore, entirely meshfree, requires minimal mathematical and programming background, and scales with the number of grid points.

Our computational exploration demonstrates how this method can be easily implemented to approximate solutions of elliptic PDEs paired with arbitrary boundary conditions enforced on non-trivial geometries. We observe the solution converging as the spatial resolution and the maximum polynomial order increase (h-P convergence), on both uniform and arbitrary highly non-graded tree structures. Because our method is meshless in nature, the underlying set of points can be chosen as virtually anything. In particular, the points do not have to conform to the problem geometry or be aligned in

any specific pattern. This facilitates the construction and manipulation of the data structure and offers unbounded solution adaptivity and error control strategies.

This work is part of a manuscript, written in collaboration with Gbocho Terasaki and Prof. Maxime Theillard; the method was developed with the help of former student Selma Strango. An earlier version of the manuscript can be found on arXiv [2], with the dissertation author as first author. This chapter is organized as follows. In section 2.2, we present the LSQD method in the general context and discuss its practical implementation. The method is illustrated with a simple one-dimensional example. In section 2.3 we discuss the extension of the method to higher dimensions and propose an implementation on non-graded quadtree grids. Convergence results are provided in section 2.4 for a variety of grid and boundary configurations. The conclusion is in section 2.5.

2.2 LSQD

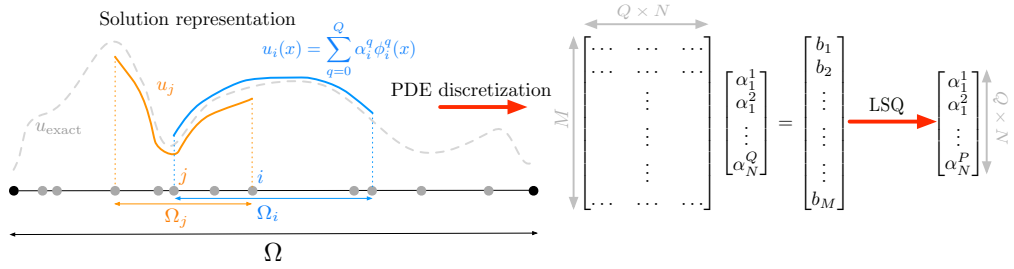


Figure 2.1: We represent the solution of a differential equation as a set of local expansions, denoted by $u_i(x)$, which are centered at arbitrary positions $x_{i=1..N}$. To determine the coefficients $\alpha_{i=0..N}^{q=1..Q}$ of these expansions, we create an overdetermined discrete system by evaluating the differential equation at its neighbors and enforcing continuity conditions between neighboring local representations. We define the final coefficient vector as the least squares solution to the overdetermined system.

2.2.1 Method Presentation & Implementation

For the LSQD method, consider the following elliptic problem:

$$au - \mu \Delta u = f(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega \subseteq \mathbb{R}^d, \quad f(\mathbf{x}) \in \mathcal{C}^0(\Omega) \quad (2.1)$$

with generalized (e.g. Dirichlet, Neumann, Robin, etc.)¹ boundary conditions on the contour $\partial\Omega$, i.e.

$$\beta u(\mathbf{x}) + \gamma \nabla u(\mathbf{x}) \cdot \mathbf{n} = g(\mathbf{x}), \quad \forall \mathbf{x} \in \partial\Omega, \quad g(\mathbf{x}) \in C^0(\Omega), \quad (2.2)$$

where $\mathbf{n}$ is the outward point normal to $\partial\Omega$. The parameters of the elliptic problem, a and μ , are positive and the solution is assumed to be continuously differentiable. We are interested in constructing a discrete approximation of the solution $u(\mathbf{x})$, using a set of points $(\mathbf{x}_1, \dots, \mathbf{x}_N) \in \Omega$. To each point $\mathbf{x}_i$, we associate a set of neighbors V_i and define the vicinity Ω_i as the convex hull of this set. For each vicinity Ω_i , we approximate the solution using an expansion over a local smooth basis $(\phi_i^1(\mathbf{x}), \dots, \phi_i^Q(\mathbf{x}))$

$$u(\mathbf{x}) \approx \sum_{q=1}^Q \alpha_i^q \phi_i^q(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_i. \quad (2.3)$$

Thus, by estimating the coefficients α_i^q , at each Ω_i , we hope to construct a family of local approximations to the solution $u(\mathbf{x})$.

Determining α_i^q

To find the coefficients α_i^q , we evaluate the PDE (2.1), the continuity conditions (for $\mathbf{u}$ and its first order derivatives), and the boundary condition (2.2) for the above local representation. For each node $\mathbf{x}_i$, we consider each neighbor pair $(\mathbf{x}_i, \mathbf{x}_j)$ with $\mathbf{x}_j \in V_i$ and construct the linear equations for α_i^q using the following procedure with three main steps:

1. **Partial Differential Equation:** we choose a point $\mathbf{x}_{ij}^{PDE} \in \Omega_i$ at which we evaluate the PDE for the expansion (2.3).

$$a \sum_{q=1}^Q \alpha_i^q \phi_i^q(\mathbf{x}_{ij}^{PDE}) - \mu \sum_{q=1}^Q \alpha_i^q \Delta \phi_i^q(\mathbf{x}_{ij}^{PDE}) = f(\mathbf{x}_{ij}^{PDE}). \quad (2.4)$$

2. **Continuity conditions:** we choose a point $\mathbf{x}_{ij}^C \in \Omega_i$ at which we enforce that the expansions centered at $\mathbf{x}_i$ and $\mathbf{x}_j$ are identical

$$\sum_{q=1}^Q \alpha_i^q \phi_i^q(\mathbf{x}_{ij}^C) = \sum_{q=1}^Q \alpha_j^q \phi_j^q(\mathbf{x}_{ij}^C), \quad (2.5)$$

and that the expansions for all first-order derivatives ∂_d are also identical at that point

$$\sum_{q=1}^Q \alpha_i^q \partial_d \phi_i^q(\mathbf{x}_{ij}^C) = \sum_{q=1}^Q \alpha_j^q \partial_d \phi_j^q(\mathbf{x}_{ij}^C). \quad (2.6)$$

¹For Dirichlet boundary conditions: $\gamma = 0$. For Neumann boundary conditions: $\beta = 0$. For Robin boundary conditions: $\beta \neq 0$ and $\gamma \neq 0$.

3. **Boundary conditions:** if $\mathbf{x}_i$ is close to the contour $\partial\Omega$, we find a point $\mathbf{x}_i^{BC} \in \partial\Omega$, at which we enforce the boundary condition for the local expansion (2.3).

$$\beta \sum_{q=1}^Q \alpha_i^q \phi_i^q(\mathbf{x}_i^{BC}) + \gamma \sum_{q=1}^Q \alpha_i^q \nabla \phi_i^q(\mathbf{x}_i^{BC}) \cdot \mathbf{n} = g(\mathbf{x}_i^{BC}). \quad (2.7)$$

After grouping all of these equations, we are able to form a rectangular system of height M and width $Q \times N$.

$$\begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ & \vdots & \\ & \vdots & \\ & \vdots & \\ \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} \alpha_1^1 \\ \alpha_1^2 \\ \vdots \\ \alpha_N^Q \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ \vdots \\ \vdots \\ b_M \end{bmatrix} \Leftrightarrow A\alpha = b. \quad (2.8)$$

If the matrix A is full rank, we construct the coefficient vector α as the least squares solution:

$$\alpha = \underset{x}{\operatorname{argmin}} \|Ax - b\|_2 \Leftrightarrow A^T A \alpha = A^T b. \quad (2.9)$$

If the matrix A is not full-rank, we can consider the stabilized problem

$$(A^T A + \epsilon I) \alpha = A^T b, \quad (2.10)$$

which is well-known to be invertible for any $\epsilon > 0$. Nevertheless, we must be cognizant of two extremities: choosing a large ϵ will affect the accuracy of the method, but choosing its value to be too small may not serve the intended purpose and leave the system singular to machine precision.

2.2.2 Constructing a Neighborhood

Although with this method there is no restriction on the basis selection, a naive choice could lead to singular least squares systems, making it impossible to construct the solution (without stabilization (2.10)). Let us note that for the remainder of this study, at each point $\mathbf{x}_i$, we will use polynomial functions that are centered at the point $\mathbf{x}_i$ and rescaled by the size of the neighborhood V_i . Hereafter, the maximum polynomial order of the basis will be denoted by P and the total number of elements in the basis will be denoted by Q .

From the general method description, we see that it is essential to construct a full-rank matrix A , while keeping A as compact and sparse as possible for computational efficiency. Because the matrix structure and size are primarily dictated

by the neighborhood configurations, it is imperative to understand how many neighbors are needed in each neighborhood, how to construct them, and what properties they should satisfy. The following are the three chief properties that each neighborhood must satisfy:

- i **System Size** - The total number of equations must be greater than or equal to the total number of basis functions (*i.e.* the number of unknown coefficients α_i^q).
- ii **Connectivity** - The neighbors' adjacency matrix must be invertible. This means that you should be able to travel from any point $\mathbf{x}_i$ to any other point $\mathbf{x}_j$ whilst traveling along a neighboring connection.
- iii **Independence** - The neighborhood should contain points with at least $P + 1$ unique coordinates in each direction.

Condition (i) ensures that the matrix A is full rank. The second condition (ii) ensures that all the neighborhoods are connected, and therefore, that the computation of each basis's coefficients depends on all other coefficients. If this condition is not satisfied, there will be clusters of nodes and, consequently, basis coefficients that are separated from each other, violating the global nature of the elliptic equation (2.1). The third condition (iii) reinforces linear independence between the coefficient equations. Note that even if the problem only varies in one direction, the matrix A must still be full rank. In this scenario, the number of linearly independent basis functions is simply $P + 1$ and so we need at least the same amount of unique coordinates in the direction that varies.

We point out that the neighborhoods do not have to be symmetric, meaning that if $\mathbf{x}_j$ is a neighbor of $\mathbf{x}_i$ (*i.e.* $\mathbf{x}_j \in V_i$), it is not necessary for $\mathbf{x}_i$ to be a neighbor of $\mathbf{x}_j$. Furthermore, they do not all have to be the same size, nor abide by specific geometric configurations. In practice, however, it is natural to keep them local and compact.

2.2.3 Constructing & Inverting the System

One of the principal advantages of our method is that the matrix A is straightforward to construct. Once the neighborhoods have been built, we loop over all neighboring pairs and evaluate the PDE, continuity, and boundary conditions. The matrix assembly only requires evaluation of the basis function at the evaluation points. No quadrature rules, finite difference formulas, nor any other numerical analysis algorithms are required.

Assuming that the Least Squares problem is non-singular, the solution vector can be computed using a preconditioned conjugate gradient method (since $A^T A$ is symmetric), or using the LSQR method [56, 57] to avoid forming $A^T A$. Forming $A^T A$ may allow us to construct a wider range of preconditioners, and so it is, *a priori*, unclear which of these options will be computationally more efficient. It should be noted that the matrices A , A^T and $A^T A$ are sparse and can be constructed in $\mathcal{O}(NQ^2)$; therefore, disregarding the neighborhood construction, the total computational cost will scale as the complexity of the linear solver, typically $\mathcal{O}(NQ^2\sqrt{\kappa})$, with κ being the condition number of $A^T A$.

2.2.4 One-Dimensional Proof of Concept

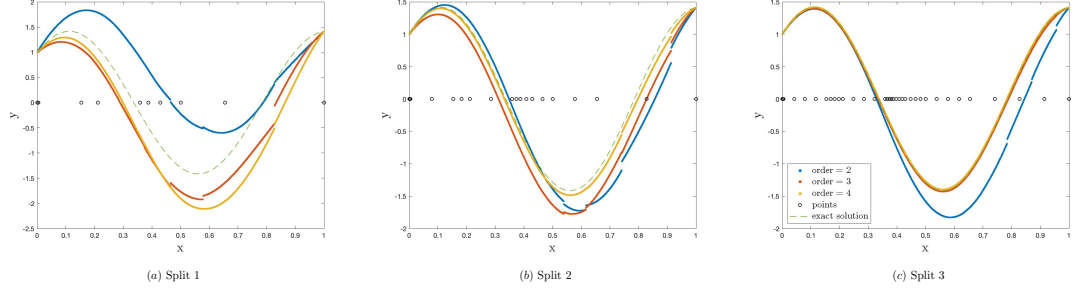


Figure 2.2: 1D Method - solution representation for increasing spatial and polynomial resolution. We start from a randomly generated set of points $\mathbf{x}_{i=1..10}$ and then recursively split the grid by adding one grid point exactly between each existing pair of direct neighbors. The exact solution is depicted as a dashed line.

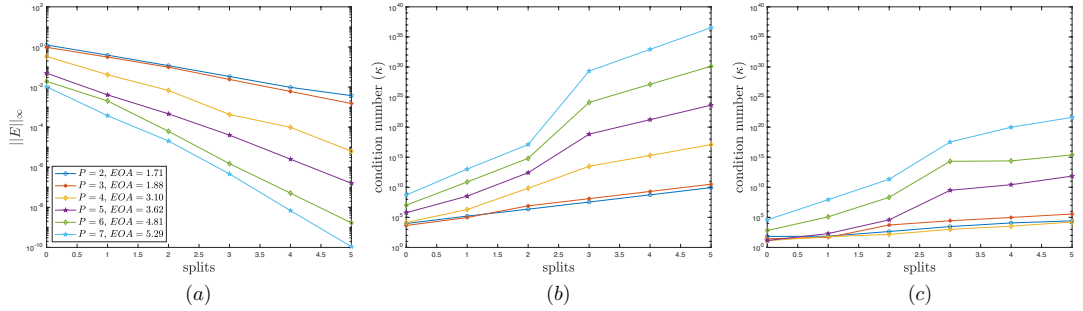


Figure 2.3: 1D Method - (a) h-P convergence analysis for Poisson's equation with Dirichlet boundary conditions. L^∞ error and corresponding Estimated Order of Accuracy (EOA). (b) Condition number of the least squares matrix $A^T A$. (c) Condition number of the preconditioned least squares matrix, using an incomplete cholesky decomposition.

We consider the following Poisson problem

$$\begin{aligned} -\Delta u &= f(x) & \forall x \in \Omega = [0, 1], \\ u(x) &= g(x) & \forall x \in \partial\Omega. \end{aligned} \quad (2.11)$$

In order to study the performance of the method we choose the exact solution to be

$$u_{\text{exact}}(x) = \sin(7x) + \cos(7x)$$

and we set $f(x)$ and $g(x)$ accordingly. All local basis function sets are chosen to be centered and rescaled polynomials, *i.e.* for each x_i and over each corresponding neighborhood V_i , we define

$$\phi_i^q(x_i) = \left(\frac{x - x_i}{|V_i|} \right)^{q-1}, \quad |V_i| = \max_{j \in V_i} |x_i - x|, \quad x \in V_i, \quad q = \{1, \dots, Q = P + 1\} \quad (2.12)$$

Each neighborhood V_i is first created by choosing the node $\mathbf{x}_i$ itself, its closest left and right neighbors. This ensures that the neighborhoods overlap with at least another neighborhood in each direction; therefore the connectivity condition ii is satisfied. Additionally, we need at least $P + 1$ different neighbors in V_i to satisfy both the system size i and independence iii conditions. So, if $P > 2$, we add the remaining $P - 2$ closest points to the neighborhood. As a result, each neighborhood will contain $\eta = \max(3, P + 1)$ elements. We choose the evaluation points to be $\mathbf{x}_{ij}^{PDE} = \mathbf{x}_{ij}^C = \mathbf{x}_j$.

Once again, we emphasize the fact that advanced numerical analysis techniques are not required to construct our system, nor to obtain our solution (e.g. no Taylor expansions, convoluted integrals, or quadrature rules). Our implementation contains around 100 lines of Matlab code and the least squares system is solved using the simple backslash solve command (i.e. $\alpha = A^T A \backslash A^T b$). The backslash command for general square matrices uses LU decomposition; however for symmetric matrices (which is our case) the Cholesky decomposition is used instead. The results can be seen in Figure 2.2.

To study the h-P convergence of the method, we initialize the points $x_{i=1..10}$ at random and then recursively add midpoints between adjacent neighbors to double the local spatial resolution. We will refer to splits as the number of subdivisions performed from the original points set, and point out that the local resolution scales like $h^{-\text{splits}}$, where h is the original local spatial resolution. Simultaneously, we vary the polynomial order using $P = 2, 3, 4$. The error is computed by comparing the exact solution and the local approximation at each point. We will focus on the maximum error only.

Convergence results are presented in Figure 2.3, where we see the error diminishing both as the resolution and the polynomial order increase. The condition number – indicative of the computational time needed to solve the least squares problem – increases with both the resolution and maximum polynomial order. Again, our philosophy is to build a method that can easily be understood and implemented; we are willing to have the computer take on the bulk of the work if it entails that we do less. Indeed, for this example, we see that our method’s simplicity comes at the price of inflated computational time. With the system being symmetric, we are able to precondition it using a standard incomplete Cholesky decomposition, which, for this example, drastically reduces the condition number. Typically, the condition number of the preconditioned system is only the square root of the unpreconditioned one.

2.3 Extension to Higher Dimensions

Despite the one-dimensional example serving as an obligatory proof of concept in terms of convergence, the full capabilities of our method, such as dealing with highly adaptive data structures, arbitrary geometries, and boundary conditions, only manifest themselves in higher dimensions.

For our method, the extension to higher dimensions is straightforward from both the theoretical and practical perspectives. As illustrated in Figure 2.4, the domain of

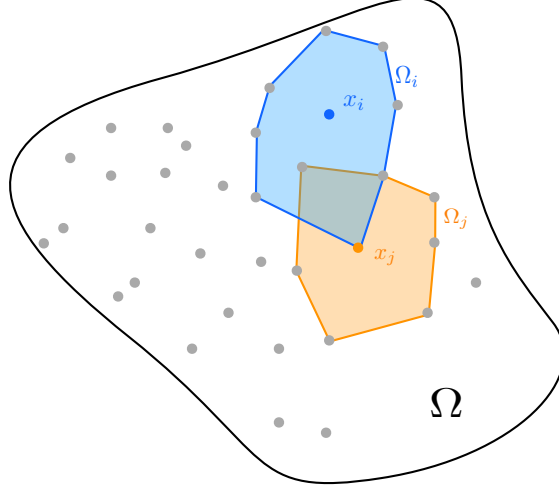


Figure 2.4: Extending to higher dimensions: domain of interest Ω , and local vicinities Ω_i, Ω_j . The neighborhoods V_i, V_j are defined as the set of points contained in the corresponding vicinities.

interest Ω is still represented by a set of points $\mathbf{x}_{i=1..N}$. The solution is still decomposed over a local basis $\phi_i^1(\mathbf{x}).. \phi_i^Q(\mathbf{x})$ around the set of points, and the coefficients for these expansions are obtained by evaluating the partial differential equations, boundary, and continuity conditions over local neighborhoods, as underscored in section 2.2.1. The construction of the linear system remains trivial, and the properties of the final system are unchanged ($A^T A$ is symmetric, positive definite).

The main change in extending to higher dimensions difference is the neighborhood construction - the shape and size of the neighborhoods both change. Furthermore, because the derivative continuity condition will be evaluated in each direction, the total number of equation written per neighboring pair will differ, and thus the minimal neighborhood size. So, in dimension d , for a neighborhood V_i composed of η neighbors (including $\mathbf{x}_i$), we will evaluate

- η PDE equations,
- $\eta - 1$ continuity conditions,
- $d(\eta - 1)$ first-order derivative continuity conditions ($\eta - 1$ for each partial derivative),

for a total of $\eta(d + 2) - d - 1$ equations. Since we want the matrix A to be full rank, we will systematically pick the number of basis of local basis function Q such that

$$Q \leq \eta(d + 2) - d - 1. \quad (2.13)$$

2.3.1 Basis Functions in 2D

We chose our basis function as product of one dimensional rescaled polynomials, around $\mathbf{x}_i = (x_i, y_i)$:

$$\phi_i^q(\mathbf{x}) = \left(\frac{x - x_i}{|V_i^x|} \right)^{p_x} \left(\frac{y - y_i}{|V_i^y|} \right)^{p_y} \quad \forall \mathbf{x} = (x, y) \in \Omega_i, \quad (2.14)$$

where the rescaling coefficients are

$$|V_i^x| = \max_{\mathbf{x} \in V_i} |x_i - x|, \quad |V_i^y| = \max_{\mathbf{y} \in V_i} |y_i - y|. \quad (2.15)$$

The total number of basis function will be

$$Q = \frac{(P+1)(P+2)}{2}, \quad (2.16)$$

and based on condition (2.13), we will need to choose the neighborhood size η such that

$$\eta \geq \frac{(P+1)(P+2) + 6}{8}. \quad (2.17)$$

2.3.2 Neighborhood Construction

Figure 2.5 illustrates the construction of the neighborhood V_i for any given point $\mathbf{x}_i$. We start by adding the point itself (depicted in blue) to the neighborhood. Next, the algorithm recursively checks for nearby cells in a counterclockwise fashion, such that it looks left first ($-x$ direction), followed by below ($-y$), then right ($+x$), and finally up ($+y$). For each neighbors $\mathbf{v} \in V_i$, and in each direction, we find one neighboring cell center and add it to the candidate set (depicted in pink). If $\mathbf{v}$ lies next to the interface, it may not have neighbors inside Ω in at least one direction and so we only add neighbors in Ω to V_i (see bottom row of Figure 2.5) and store the outside neighbor in a ghost set G_i that will be used to find evaluation points for the boundary condition.

On the first pass, we add the entire candidate set to the neighborhood to ensure that the connectivity condition (ii) is met. On the subsequent passes, while the other two conditions are not met, we go through the candidates set and add a candidate $\mathbf{c}$ to the neighborhood if either: the system size condition (i), (2.17), is not met or the independence condition (iii) is not met in at least a direction d and the coordinate c_d of the candidate in that direction is not already present in the neighborhood.

2.4 Results

To explore our method, we considered multiple parameter settings. Presented here are convergence results for various domain types, levels of mesh refinement, boundary condition types, and numerous formulations of Eqn. 2.1 (by changing the coefficients). Furthermore, we present an alternative way to compute a local error estimate – one that is directly built-in to our numerical method. Finally, we discuss the implications of Neumann boundary conditions with a computational time analysis.

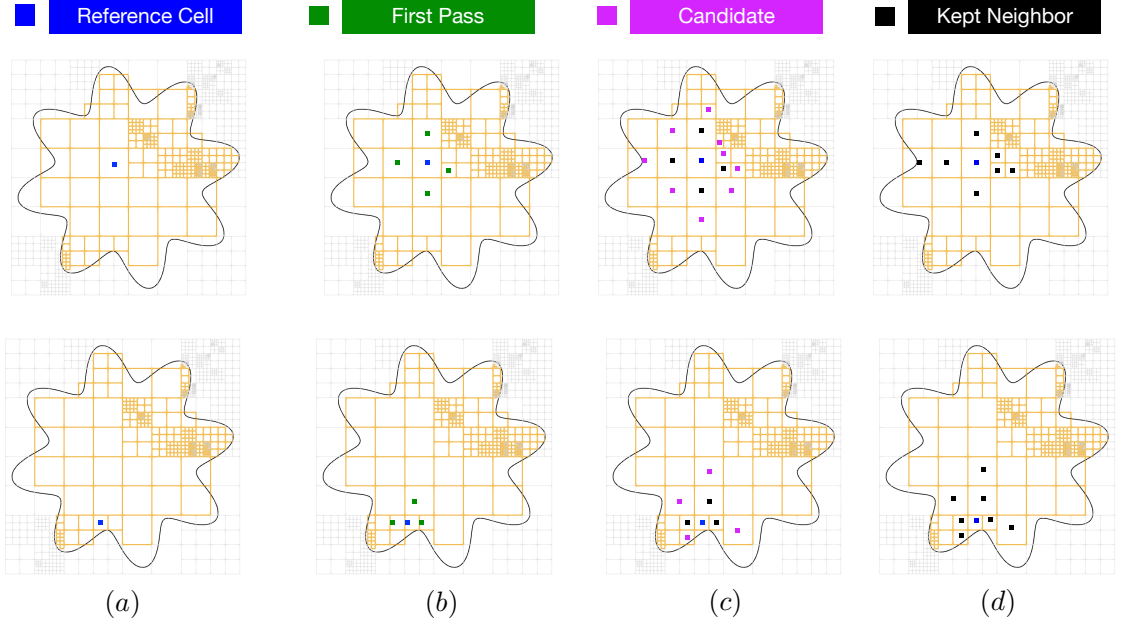


Figure 2.5: Illustration of neighborhood construction around a reference cell (in blue) for polynomial basis order $P = 4$. The process is carried out one layer at a time until enough neighbors are identified to satisfy the system size and independence conditions. When the reference cell lays next to the interface, the neighbors are chosen within Ω (bottom row).

2.4.1 Problem Setup

We consider the proposed elliptic problem (2.1) with varying α, μ values over two domains. The first domain is the unit square $\Omega = [-1, 1] \times [-1, 1]$. The second domain is the interior of an octofoil centered at $(0, 0)$ and defined by the zero contour of the following level set equation

$$\phi(x, y) = -\left[\frac{4}{5} + \frac{4}{25} \sin\left(8 \arctan\left(\frac{y}{x}\right)\right)\right] + \sqrt{x^2 + y^2}. \quad (2.18)$$

For both of these domains, we use the boundary condition formulation (2.2) and vary the weights β and γ to achieve the desired condition types. We also consider two types of mesh refinement: uniform and randomly generated (see Figure 2.6). Each of the representations will be used for the convergence analysis in this section.

We choose the exact solution to be $u_{\text{exact}}(x, y) = e^{x+y}$ and we set the forcing term $f(\mathbf{x})$ and the boundary condition $g(\mathbf{x})$ accordingly. For each of the representations, we vary the polynomial order of our basis from $P = 2$ to $P = 5$, and recursively split all grid cells to study the hp-convergence of the L^∞ -error.

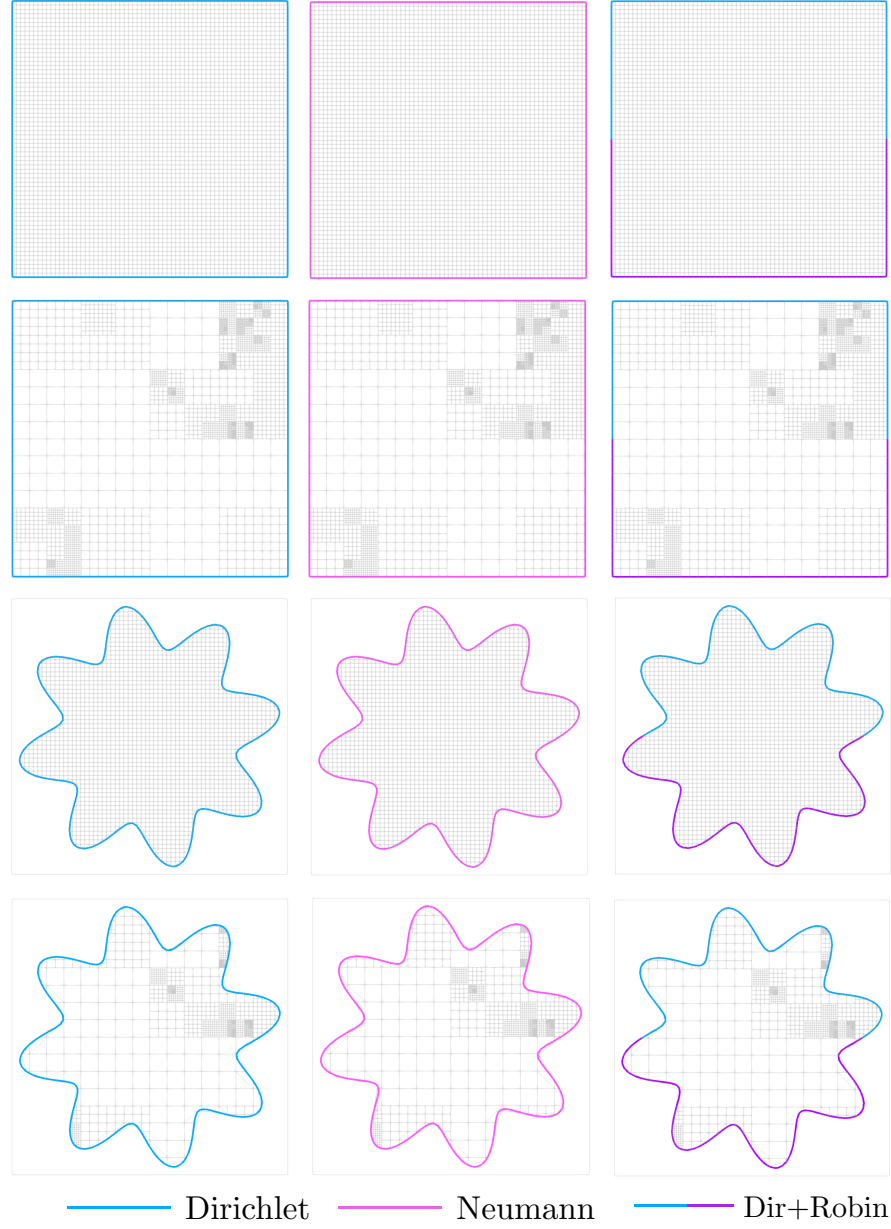


Figure 2.6: Grids and domains used for the convergence analysis - For each boundary condition type (Dirichlet, Neumann, Mixed Dirichlet Robin), enforced on a square (top rows) or the octofoil domain (bottom rows), we will consider both uniform (row 1 and 3) and adaptive grids (rows 2 and 4). This representation will be used to label all convergence figures hereafter. *Credit: this figure was created by Gbocho Terasaki.*

The local and global L^∞ -error are defined respectively as

$$||e_u(\mathbf{x}_i)|| = \left| u_{\text{exact}}(\mathbf{x}_i) - \sum_{q=1}^Q \alpha_i^q \phi_i^q(\mathbf{x}_i) \right|. \quad (2.19)$$

$$||e_u||_\infty = \max_{i=1..N} \left| u_{\text{exact}}(\mathbf{x}_i) - \sum_{q=1}^Q \alpha_i^q \phi_i^q(\mathbf{x}_i) \right|. \quad (2.20)$$

The Estimated Order of Convergence (EOC) will be computed from the global error, using power fitting curves.

Table 2.1: Convergence summary - Average estimated order of convergence for various boundary conditions, configurations, and polynomial orders $P = 2, \dots, 5$.

	Dirichlet	Neumann	Dir.+Robin	Vary $\frac{\alpha}{\mu}$	Degenerate	Overall Avg
P = 2	1.86	2.17	2.02	1.63	1.46	1.83
P = 3	1.96	2.20	2.07	2.16	1.91	2.06
P = 4	3.45	3.30	3.46	3.51	3.45	3.43
P = 5	3.66	4.53	3.89	4.46	3.60	4.03

2.4.2 Classical Boundary Conditions: Dirichlet, Neumann, and Robin

For our investigation of the classical boundary condition, we set $\alpha = \mu = 1$ and consider all twelve configurations depicted in Figure 2.6. For the specific case of the octofoil domain with Dirichlet boundary conditions ($\beta = 1$), using an adaptive refinement, the local error is illustrated in Figure 2.8. It is clear that as we increase the polynomial order and refine the grid, the local error converges.

The h-P convergence of the global error for each set of boundary conditions is represented in Figure 2.7. For all twelve of these configurations, we, again, see the error decaying to zero as the grid resolution and the polynomial order increase. The estimated order of convergence (EOC) (see Table 2.1 and Figure 2.7) are relatively consistent across all examples, ranging from $EOC \approx 2$ ($P = 2$) to $EOC \approx 4$ ($P = 5$).

2.4.3 Degenerate Cases

Here, we investigate degenerate cases where the PDE or boundary conditions may lead to a non-coercive weak formulation and thus potentially ill-posed problems, non-positive or even non-invertible systems. The LSQD method obtains the polynomial coefficient from an SPD system and does not rely on the weak formulation. Similarly to the Least Square Finite Element Method (LSFEM [58, 59]), the LSQD method constructs the discrete system by evaluating the PDE and the boundary conditions everywhere (formation of the matrix A), and then defining the Galerkin matrix by taking the scalar product of these evaluations ($A^T A$). As a result, it is not affected by the potential ill-posedness of the weak form, and the resulting linear system is guaranteed to be invertible.

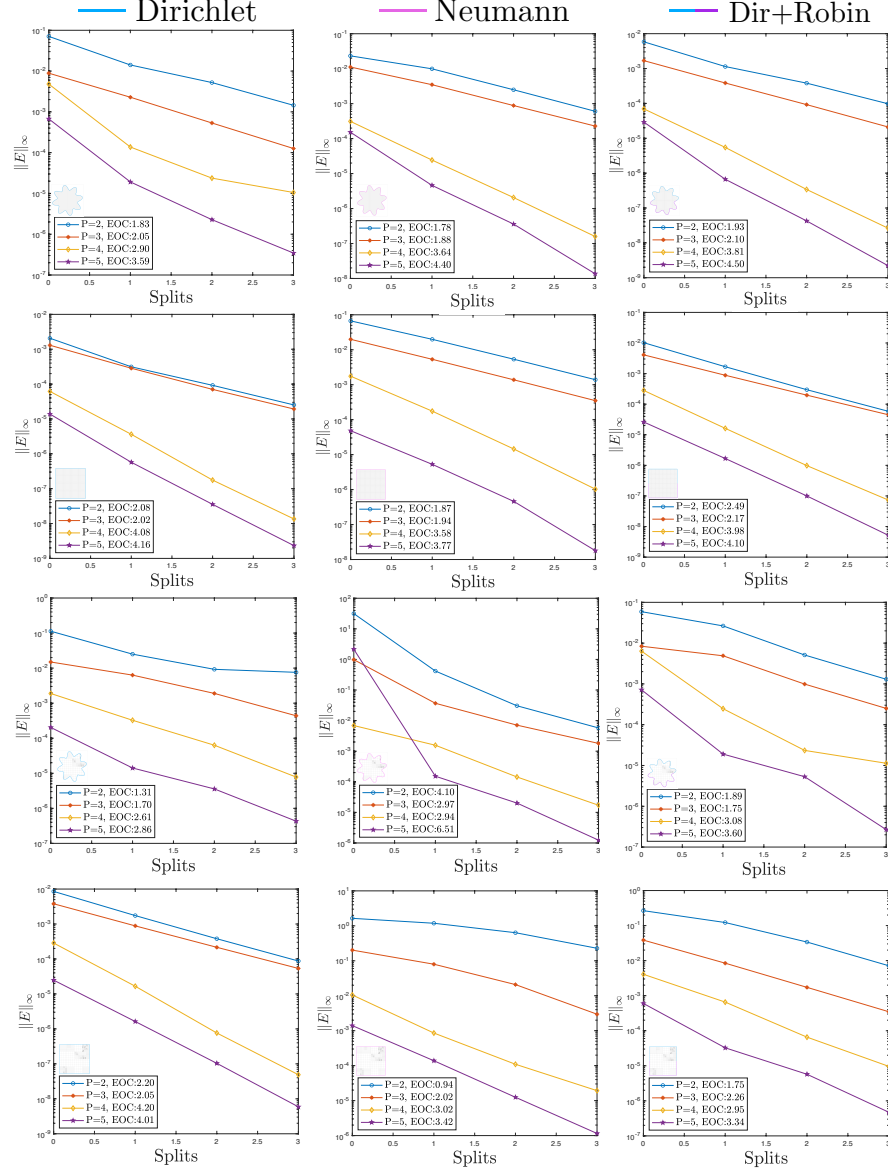


Figure 2.7: Convergence of L^∞ global error for three boundary condition types.

For the first case, we set $a = 1, \mu = 1$ and impose split boundary conditions of Dirichlet and Robin. The degeneration comes from the Robin boundary condition, for which we set $\beta = 1, \gamma = -1$. For the second case, we keep $a = 1$, but set $\mu = -100$ (*i.e.* Helmholtz's equation), and enforce a Dirichlet boundary condition. The convergence results are depicted in Figure 2.9. The method achieves h-P convergence for both degenerate cases and all grid configurations. The numerical error dips rapidly below reasonable error tolerances, and the EOC behaves normally, increasing with the polynomial order and split number. We note that results using exclusively traditional

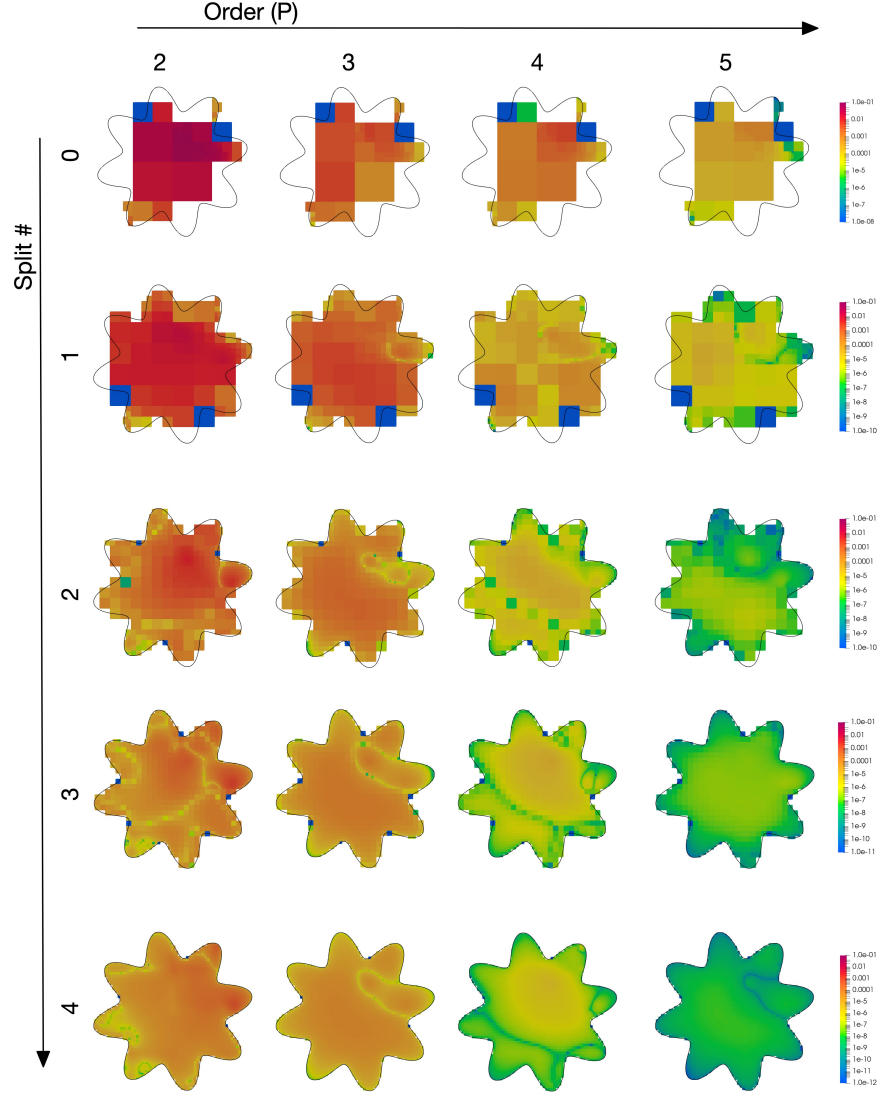


Figure 2.8: Local error convergence: Octofoil domain with Dirichlet boundary conditions on an adaptive grid. Error decays to zero both as polynomial order increases and as the grid is refined.

Robin boundary conditions ($\beta = 1$, $\gamma = 1$) on the entire domain achieved similar results.

2.4.4 Parameter Sensitivity Analysis

We focus here on Neumann boundary conditions. From (2.1), we note that when μ is significantly smaller than a , the problem resembles an interpolation problem, which is relatively straightforward to solve. On the other hand, if a is considerably smaller than

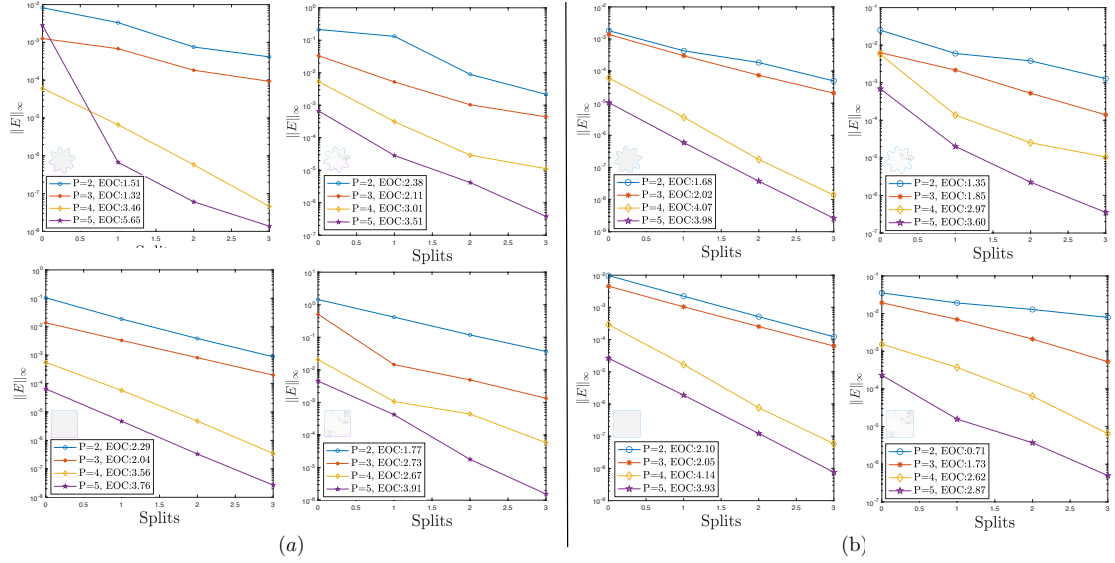


Figure 2.9: Degenerate cases: (a) Robin and Dirichlet mixed boundary conditions, (b) Dirichlet boundary condition with highly negative μ .

μ , the equation resembles an ill-posed Poisson problem.

To investigate how the solution is affected as the problem becomes close to being ill-posed, we select four different values of $\frac{\alpha}{\mu}$. The expectation is that as we increase the ratio, the problem should become easier to solve.

2.10 (a) displays convergence results for the octofoil interface with a uniform grid for $\frac{\alpha}{\mu} = 1, 10, 100, 1000$. We observe that for each polynomial basis order, P , the global error maintains a similar trend among the four tested ratios $\frac{\alpha}{\mu}$, respectively. This behavior corroborates the congruity of the Estimated Order of Convergence (EOC) between various $\frac{\alpha}{\mu}$ given order P . Clearly, the method performs as expected with the Neumann constraint across the various ratios; nonetheless, it is noteworthy that larger values of $\frac{\alpha}{\mu}$ allow for a significant decrease in the global error (≈ 2 orders of magnitude). In fact, when $\frac{\alpha}{\mu} \geq 100$, the method demonstrates accuracy comparable to, or surpassing, that which was achieved with Dirichlet boundary conditions.

2.4.5 Computational Time

It was established that lower $\frac{\alpha}{\mu}$ ratio values associated with a Neumann boundary condition create a more challenging computational problem (2.4.4). For this reason, we showcase the performance and efficiency of the LSQD method by exploring a variety of $\frac{\alpha}{\mu}$ ratios. fig. 2.10 (b) illustrates the maximum error obtained by our solver in comparison to the time it took to run serially on a Dell precision 3240. It is apparent that, in general, achieving a lower maximum error requires a longer runtime and a higher polynomial order. However, we also see that as the ratio $\frac{\alpha}{\mu}$ decreases, not only does it take more

time to solve the system, but the order of accuracy declines.

We also observe that the slope of the error versus computational time increases with polynomial order, indicating that in the asymptotic limit of highly resolved simulations (small error or large CPU time), higher-order polynomial computations become the most efficient. However, as shown in fig. 2.10 (b), this asymptotic regime is not yet reached in our results. In practice, it is imperative to decide which variable (accuracy or computational cost) takes precedence in a given setting. If minimizing error is the priority, and computational resources are available, then a higher-order polynomial basis is preferable. Conversely, if efficiency is more important or only a general solution is needed, a lower-order basis may suffice. fig. 2.10 (b) illustrates this trade-off: higher-order polynomials offer superior accuracy, but for moderate error tolerance, lower-order methods can be more cost-effective.

2.4.6 Comparison to Finite Volume Method

The Finite Volume Method (FVM), a well-established approach in scientific computing, excels at enforcing local conservation, handling discontinuities, and performing efficiently on simpler geometries with structured or unstructured meshes. However, FVM's reliance on mesh generation becomes a significant bottleneck when dealing with complex geometries, impacting both efficiency and accuracy, particularly with non-uniform grid refinement. The Least Squares Discretization (LSQD) method, in contrast, addresses this limitation by leveraging point cloud discretization, eliminating the need for a mesh, and offering a more straightforward implementation. As demonstrated in fig. 2.10 (b) and (c), while FVM may offer faster computation, LSQD consistently achieves lower errors, especially at higher polynomial orders, making it a compelling alternative when precision is paramount and complex geometries pose meshing challenges.

2.4.7 Error Estimator

We propose an alternative way to compute local error estimate as:

$$||e_u(\mathbf{x})||_{Est} = \max \left| \sum_{q=1}^Q \alpha_{l_1}^q \phi_{l_1}^q(\mathbf{x}) - \sum_{q=1}^Q \alpha_{l_2}^q \phi_{l_2}^q(\mathbf{x}) \right|, \quad l_1, l_2 \in L = (\text{neighboring cells}) \quad (2.21)$$

Obtaining analytical solutions to PDEs is only possible in the simplest cases, so it should be expected that the exact solution to the PDE (2.1) may not be available. We propose an original error estimator, built-in to the algorithm, as evaluating accuracy predicated on traditional methods tends to be infeasible. This error estimator acts as a prediction for the expected convergence. For a comparison between the theoretical prediction and practical results, refer to the Dirichlet case in Figure 2.11.

The function of our node-based error estimator is to compare local approximations from neighboring cells. The error is calculated around each node location $\mathbf{x}_i$ and is

determined by neighboring cells, more specifically, the current node's shared corners with surrounding cells. In one dimension, we can picture the error as the discontinuity between two local expansions around that node (see Figure 2.2). To visualize this process in two dimensional space, refer once again to Figure 2.11 (a). If we want to compute the estimate error around the reference node (in black), we first evaluate the local approximations at this node. Then, we use the local approximation functions constructed for the closest neighbor cells (blue), and do the evaluation at the considered reference node x_i . (note that the maximum amount of neighboring cells is four: left, right, top, and bottom). Finally, the error estimate is computed as the maximum difference between the local values obtained and the initial evaluation.

2.5 Conclusion

In this paper, we proposed a new paradigm for solving elliptic PDEs, one that would retain all of the advantages of meshless methods, while remaining accessible in theory and in practice. Least Squares Discretization Method (LSQD) does not require any quadrature rules, weak formulations, body-fitted grids, nor does it necessitate advanced mathematical theory. Nevertheless, it constructs h-P converging numerical solutions on non-trivial data structures with reasonable computational complexity. LSQD method easily handles complex geometries and is virtually impervious to ill-posed problems.

We demonstrated the feasibility in extending our method to a higher dimensional space, and its applicability to a broad range of problems, boundary conditions, and computational parameters (*e.g.* grid resolution, polynomial orders). For all cases, we consistently observe h-P convergence, with Estimated Order of Convergence (EOC) ranging from 1.83 to 4.53 (see table 2.1), and in practice the global error falls rapidly below any reasonable error tolerance. Moreover, when investigating degenerate cases (*i.e.* where the problem is potentially ill-posed as a consequence of the PDE coefficients or boundary conditions), the method observes kindred results (see ??), exemplifying the robustness of the LSQD method.

Finally, we proposed an *a posteriori* node-based error estimator to unveil the local solution agreement. From fig. 2.11 (c), we see that as the polynomial order increases and the grid is refined, the estimated and experimental errors display the same convergence pattern and similar spatial localization. This estimator cannot be used as a lower bound for actual error as seen in fig. 2.11 (c), but provides a general blueprint for where error is to be expected. Coupled with a spatial adaptivity capability, this error estimator could be used as an error controller.

From our computational time analysis (see 2.4.5), we observed steeper efficiency curves for higher polynomial orders, suggesting that using high-order representations might be advantageous, at least for very large systems. In practice, for small test problems like the ones considered here, these benefits may not manifest, and low polynomial orders seem to be the best strategy to get an accurate solution rapidly. In the examples presented here, computational times peaked around 10^3 seconds on a

single core, or ≈ 17 minutes for the largest systems ($\approx 1.3 \times 10^6$ degrees of freedom for 4 splits and $P=5$). Although we are willing to sacrifice computational time for ease of implementation, these values are significant and suggest that the method must be optimized to be of full practical interest.

To optimize LSQD, two avenues of investigation present themselves: reducing CPU time and minimizing numerical error. Given that the majority of computational time is spent solving the least squares system, future work will focus on accelerating the linear solver and reducing the condition number. Furthermore, LSQD offers numerous adjustable parameters, including evaluation point positions, the number of neighbors, basis function selection, and equation weighting. By carefully investigating the impact of these parameters on solution accuracy and system conditioning, the efficiency and robustness of LSQD can be further enhanced. While our initial presentation and analysis have been restricted to low-dimensional examples and a base implementation, the inherent accessibility and versatility of LSQD position it as a promising tool for computational scientists, with significant potential for further development and wider adoption.

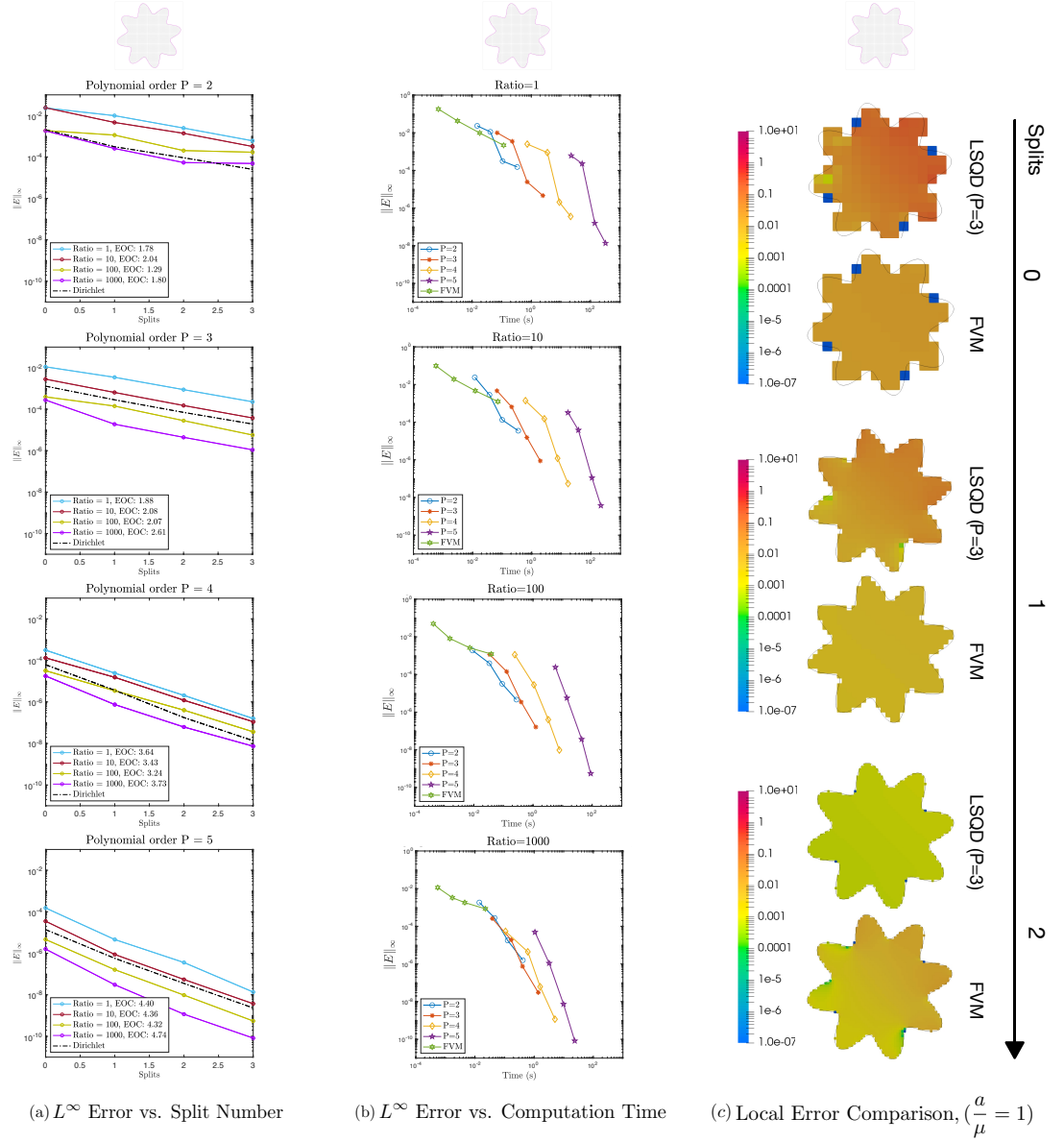


Figure 2.10: Neumann boundary condition on the octofoil domain with a uniform grid: impact of $\frac{a}{\mu}$ ratios. (a) Global error trend for increasing split number for each polynomial basis. (b) Impact of computational time on global error compared to FVM. (c) Illustration of LSQD vs FVM computational error for increasing grid refinement.

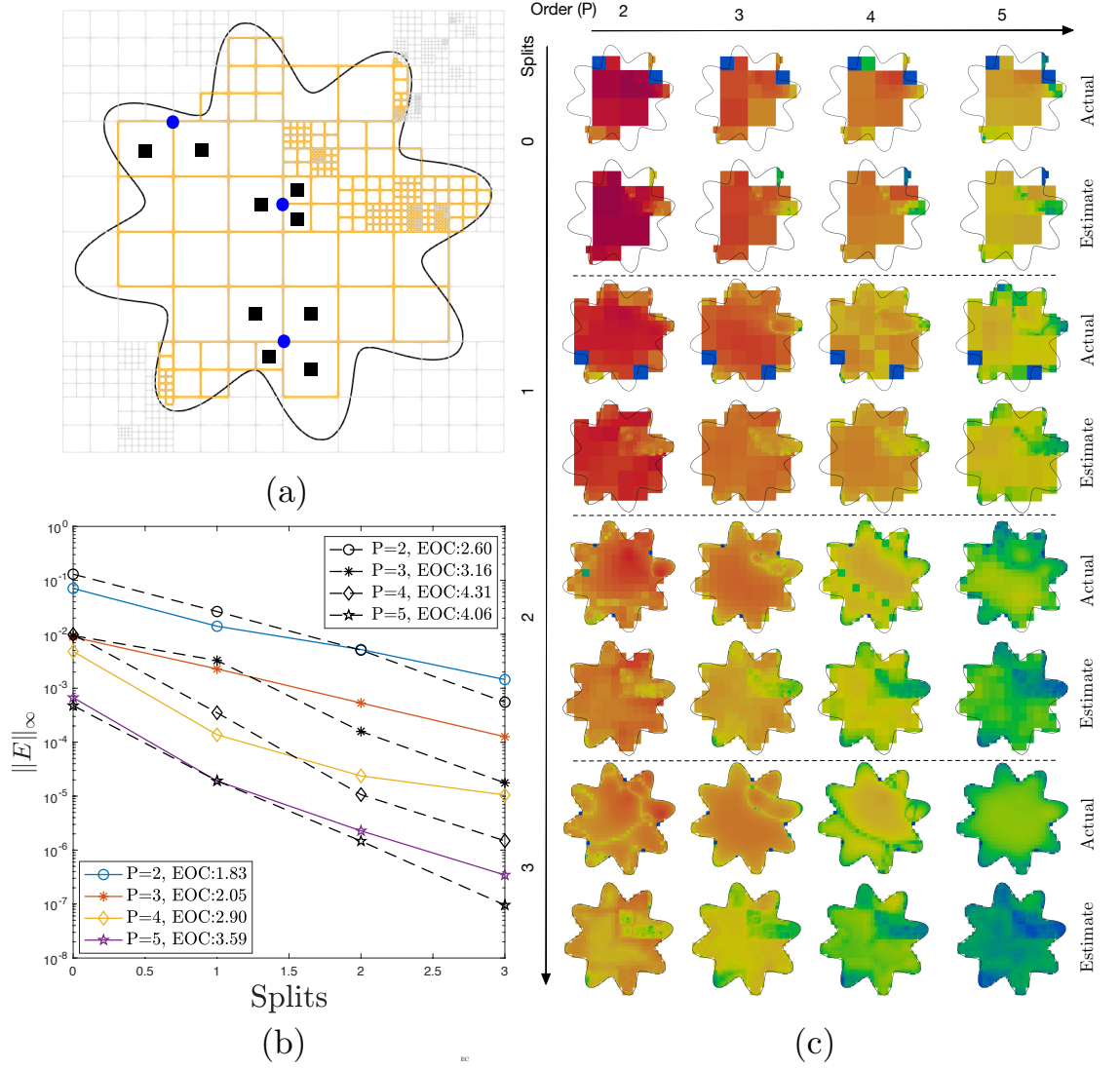


Figure 2.11: Error Estimator Analysis on an Adaptive Grid. (a) Three distinct node locations and what is considered their respective surrounding/neighbors cells. (b) Comparison between computational order of accuracy (colored lines) and predicted order of accuracy (dashed lines). (c) Illustration of computational and predicted error for increasing polynomial order and grid refinement. (Credit: Gbocho Terasaki helped create portions of this figure.)

Computational Modeling of Protein Conformational Changes - Application to the Opening SARS-CoV-2 Spike

3.1 Introduction

Proteins are polymers composed of amino acid chains that are folded into well-defined three-dimensional shapes. Their structure and shape are crucial for proper protein function. However, in many proteins the three-dimensional structure is not enough, and function must be facilitated by specific intra-protein motions. The SARS-CoV-2 spike (S) protein is one such protein [60, 61, 62, 63]. Naturally found as a homotrimer, the S-protein contains a buried receptor binding domain (RBD). The RBD has a high affinity to the human angiotensin-converting enzyme 2 (ACE2), and binding mediates viral entry to the cell. However, to facilitate ACE2 binding, the RBD must be exposed through a series of complex motions that occur in the unbound S protein trimer. [64, 61, 65, 66]. Recently, Zimmerman *et al.* [67] produced the first Markov state models simulation of the SARS-CoV-2 spike opening. This computational *tour de force* involved millions of citizen scientists collaborating through the Folding@home initiative [3], and produced an overall 0.1s of molecular trajectories. For a full characterization of the protein interaction, the molecular trajectory may not be enough.

One aspect that may provide some insight into the interactions and function of a protein is the electrostatic potential it generates. For the SARS-CoV-2 spike, according to a previous study, the affinity constant (derived from the electrostatic potential) for the RBD of SARS-CoV-2 to the ACE2 is 10 to 15 times greater than that of SARS-CoV, potentially contributing to its transmission efficiency [62]. The reason for the higher binding affinity was attributed to several mutations, most notably from the residue Val404, found in SARS-CoV, to the positively charged Lys417 in SARS-CoV-2. This mutation resulted in an intensified electrostatic potential

complementarity between the negatively charged ACE2 binding site and the now more positively charged RBD of SARS-CoV-2 [60, 62]. To understand the contribution of protein charge repositioning and also to help inform drug design strategies that leverage this distribution, it is imperative to know how the protein’s electrostatic potential changes as it deforms.

The Poisson-Boltzmann equation has long been recognized as the representation of choice to model the electric potential generated by proteins in solvents. It has drawn significant interest from the computational community since the pioneering calculations of Warwicker and Watson in the early 1980s [68], which has lead to the production of a broad variety of numerical solvers [69, 70, 71, 72, 73, 74, 75, 76] and open source software [77, 78, 79], built over the traditional spectrum of numerical methods. Such tools are, for example, employed in the context of drug development and discovery to calculate solvation free energies [80, 81, 82]. Using massively parallel architectures [83], these calculations can be carried out on considerably large proteins, such as the entire HIV-1 capsid (4, 884, 312 atoms, Protein database entry: 3J3Q).

In this work, we combine Markov state model simulations with partial differential equation modeling to obtain dynamical electric potential maps of deforming proteins. To illustrate this novel approach, we leverage the S protein opening trajectory created by the Folding@home initiative to examine and characterize the potential of the SARS-CoV-2 S protein during trimer opening. At each frame of the simulated trajectory, we reconstruct the protein surface and calculate the generated electric potential with the approach developed by Mirzadeh *et al.* [84, 85]. Using adaptive non-graded Octree grids and sharp discretizations, we efficiently produce high-fidelity solutions to the non-linear Poisson-Boltzmann equation. In the continuum model, all temporal variations are neglected, and as a consequence, all potential maps are independent, making their computation embarrassingly parallelizable. To verify the accuracy of our method, we construct practical *a posteriori* error estimates for the surface representation and electrostatic properties.

Our study of the electrostatic dynamics of the S protein opening reveals dramatic rearrangements of the electrostatic field during this process. These rearrangements act to localize a negatively charged field towards the interior of the S protein and expose a positive surface of its residue binding domain. This is in line with the negative charge of the target binding region on the ACE2 receptor and may aid the S protein in binding to its target with high affinity.

This project is arranged as follows: in section 3.2, we present the trajectory used for this study. Next, section 3.3 entails a brief description of the mathematical construction for the calculation of the potential field that is done on each frame in the trajectory. This section further describes the numerical method used along with a convergence study for computational validation. The dynamical electrostatic-geometric properties of the spike protein example are characterized in section 3.4, and conclusions are drawn in section 3.5.

The results presented here were done in collaboration with Selma Strango, Prof. Maxime Theillard, and Prof. Shahar Sukenik and this work was published in 2021 [4],

with the dissertation author as first author. The majority of the writing and figures that appear here are taken directly from the article. My specific writing contributions are the following: section 3.2, the convergence study in 3.3, the electrostatic potential results and analysis in 3.4, and the conclusion 3.5. The numerical method used here (presented in 3.3) was developed by Prof. Maxime Theillard.

3.2 Opening of the SARS-CoV-2 Spike Protein

The S protein exists primarily in the closed configuration, shielding the three identical RBDs in its core [67]. The Folding@home trajectory focuses on the opening of the unglycosylated, uncleaved spike configuration through the detachment of a single monomer from the spike's core, revealing its associated binding site. A similar (but not identical) monomeric-opening state has recently shown to be populated in $\approx 16\%$ of the unbound spike population in a recent cryoEM study [86].

3.2.1 Simulated Trajectory

The trajectory, from Zimmerman *et al.* [67] contains a 71-frame sequence that captures the most populated pathway of the spike protein's transition from its closed to open state. Each frame consists of a list of $N = 51,671$ atoms, represented by their positions $\mathbf{x}_{i=1..N}$, radius $r_{i=1..N}$ and fixed partial charge $z_{i=1..N}$. Throughout these frames, the protein undergoes continuous deformations, which shift the receptor-binding domain of one of the monomers from being hidden - while the spike is in its closed conformation - to being fully exposed once the spike opens. Figure 3.1*a* and 3.1*b* represent the protein's molecular structure in the open and closed configurations. Each one of its chains is depicted in a different color to illustrate the protein's trimeric structure. The revealed receptor-binding domain is located at the opening extremity of the red chain (see Figure 3.1*a*, 3.1*b*).

3.2.2 Relative Opening Measurement

To provide preliminary quantification of the spike opening, dozens of atom pairs on opposing sides of the spike were chosen, and the distance between them measured as the spike transitions from one conformation to another (i.e. as a function of frame) (Figure 3.1*a* and *b*). One of the atoms in the pair was chosen from the binding interface, the monomer depicting in red in Figure 3.1(*a* and *b*), while the second atom in the pair was chosen across the top of the spike opening. Although all pairs presented the same general trajectory, four of these atom pairs were arbitrarily selected as a subset for illustration.

The resulting relative distance between the atom pairs is measured in Angstroms ($\text{\AA}$), as shown in Figure 3.1*c*. The lines labeled 1, 2, 3, and 4 refer to the behavior of the four-pair subset, with their average behavior depicted in black. All four measurements

generally follow the same trend. This extension happens continuously outside of frames 50 to 55, which show abrupt variations.

3.2.3 Magnitude of the Conformation Change

To measure the total structural deformation, we compute the root-mean-square deviation (RMSD). It measures the average distance between atoms in the current position and some reference configuration, defined here as the initial closed configuration. Specifically

$$RMSD = \sqrt{\frac{1}{N} \sum_{i=1}^N |\mathbf{x}_i^n - \mathbf{x}_i^1|^2}, \quad (3.1)$$

where the superscripts n and 1 are for the current and initial states. Despite large variations at the initial and final stages, the RMSD evolution (depicted in Figure 3.1*d*), shares similar features with the evolution of the relative opening. These similarities suggest that the significant transformations that occur during the opening are recapitulated by the four distances in Figure 3.1*c*.

3.3 Mathematical Construction

The potential map at each iteration from the current protein structure is reconstructed using the non-linear Poisson-Boltzmann (PB) equation. Because of the way the trajectory has been obtained, all frames are independent. Thus, the potential maps are decoupled and can be computed separately.

A common way of portraying a molecular surface is by use of the van der Waals Surface depiction (see Figure 3.2*b*). Each atom in a molecule is depicted by a sphere with location $\mathbf{x}_{i=1\dots N}$ and radius $r_{i=1\dots N}$ that is defined by an isosurface on their electron density. The union of these spheres, depicted in gray in Figure 3.2, forms the van der Waals Surfaces (vdWS) of that molecule. The nature of the vdWS means that some regions might be identified as being exposed to the solvent, while in fact, their geometry makes them inaccessible to the solvent particles. As a consequence, we define the Solvent Accessible Surface (SAS [87]) of the protein, depicted as the blue outline in Figure 3.2 *b*; it is formed through the addition of the solvent's particle radius, r_P , to each $r_{i=1\dots N}$ resulting in a buffer around the vdWS. The Solvent Excluded Surface (SES) is preferable when discussing surface details of a molecule [88], as shown in Figure 3.2*b*. As the name suggests, the inner molecule region defined by this surface includes all locations the solvent cannot occupy, including the vdWS and the tiny crevasses on its exterior, shown as concave black triangles at the meeting of atoms A and B in Figure 3.2*b*.

To represent the biomolecule mathematically, we employ the level set method [89] and capture the SES location as the zero level set of an auxiliary field $\phi(\mathbf{x})$ defined over

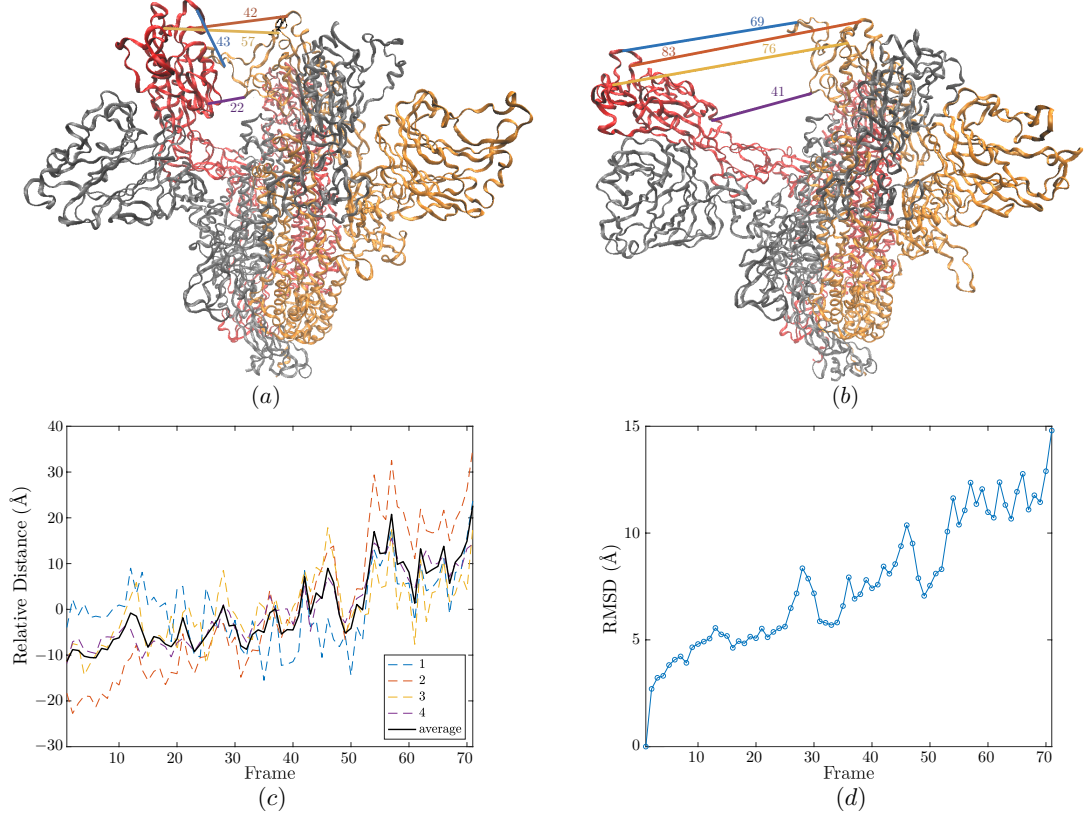


Figure 3.1: SARS-CoV-2 Spike protein visualized in its closed (a) and open (b) states. The four pairs of atoms, along with the separating distances, used to quantify the gap opening are represented with straight solid colored lines. (c) The relative distance (Å) between arbitrarily chosen atom pairs are represented as a function of time. One atom is selected on the receptor-binding domain (RBD), and the other opposite the RBD across the opening of the spike protein. This distance is relative to the average distance between each respective atom pair (1,2,3, and 4 in the graph), hence the term 'relative distance'. (d) The root mean squared deviation (RMSD) is calculated as a function of time with frame 1 (closed state) as the reference frame.

the domain of interest Ω . The solvent Ω^- , the Solvent Excluded Surface Γ , and the inside of the molecule Ω^+ are defined as

$$\Omega^- = \{\mathbf{x} \in \mathbb{R}^3 | \phi(\mathbf{x}) < 0\} \quad (3.2)$$

$$\Gamma = \{\mathbf{x} \in \mathbb{R}^3 | \phi(\mathbf{x}) = 0\} \quad (3.3)$$

$$\Omega^+ = \{\mathbf{x} \in \mathbb{R}^3 | \phi(\mathbf{x}) > 0\} \quad (3.4)$$

The normal $\mathbf{n}$ to the interface Γ is defined as pointing toward Ω^+ . It is calculated as the normalized level set gradient

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}. \quad (3.5)$$

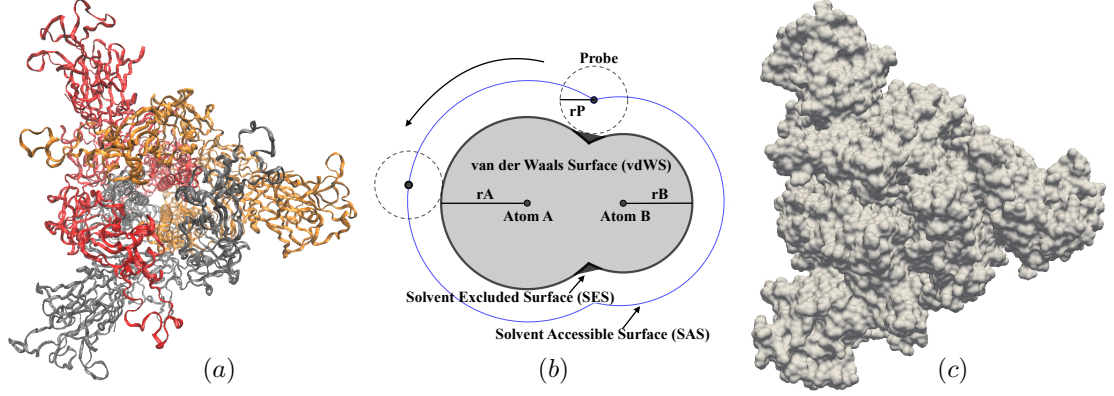


Figure 3.2: (a) Visualization of the top view of the Spike protein molecule (colored by monomer). Color denotes the three monomers (b) A cutaway diagram of a general molecular surface depicting the Van der Waals Surface (vdWS), solvent-excluded surface (SES), and solvent-accessible surface (SAS). (c) SES representation of the Spike protein oriented the same way as Figure (a) (top view).

The level set function is constructed following the approach proposed by Mirzadeh *et al.* in [85].

Refer to the above Table 3.1 for the full list of problems parameters and their non-dimensional values. The electrostatic potential, Ψ , around a biomolecule immersed in a binary z:z electrolyte solution can be described by the following non-linear Poisson-Boltzmann (PB) equation,

$$-\nabla \cdot (\epsilon \epsilon_0 \nabla \Psi) + 2c^b(\mathbf{x}) N_A e z \sinh\left(\frac{e\Psi}{k_B T}\right) = \sum_{i=1}^N q_i \delta(\mathbf{x} - \mathbf{x}_i), \quad \forall \mathbf{x} \in \Omega \setminus \Gamma. \quad (3.6)$$

The non-dimensional Poisson-Boltzmann equation has the form:

$$-\nabla \cdot (\epsilon \nabla \psi) + \kappa_D^2(\mathbf{x}) \sinh(\psi) = \sum_{i=1}^N \lambda_B z_i \delta(\mathbf{x} - \mathbf{x}_i), \quad \forall \mathbf{x} \in \Omega \setminus \Gamma. \quad (3.7)$$

The above non-dimensional PDE is completed with the following jump conditions:

$$[\psi]_\Gamma = 0, \quad [\epsilon \nabla \psi \cdot \mathbf{n}]_\Gamma = 0, \quad \forall \mathbf{x} \in \Gamma. \quad (3.8)$$

The non-dimensional potential ψ , is split into regular and singular parts: $\hat{\psi}$ and $\bar{\psi}$, respectively

$$\psi = \hat{\psi} + \bar{\psi}. \quad (3.9)$$

The singular part is itself split into two parts ψ^* and ψ_0

$$\bar{\psi}(\mathbf{x}) = \begin{cases} \psi^*(\mathbf{x}) + \psi_0(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega^+, \\ 0 & \text{if } \mathbf{x} \in \Omega^- \end{cases} \quad (3.10)$$

Parameter	Symbol	Definition	Value
Characteristic length	L	-	1
Characteristic potential (thermal voltage)	ψ_0	$\frac{k_B T}{e}$	$2.570 \times 10^{-2} V$
Vacuum permittivity	ϵ_0	-	$8.854 \times 10^{-12} Fm^{-1}$
Spike relative permittivity	ϵ^+	-	2
Solvent relative permittivity	ϵ^-	-	7.830×10
Boltzmann constant	k_B	-	$1.381 \times 10^{-23} JK^{-1}$
Avogadro number	N_A	-	$6.022 \times 10^{23} M^{-1}$
Elementary charge	e	-	$1.602 \times 10^{-19} C$
Temperature	T	-	$298.15 K$
Salt concentration	c^b	-	$1 \times 10.0^{-3} Mm^{-3}$
Probe radius	r_p	-	1.4
Valence of the background electrolyte	z	-	1
Non-dimensional inverse of Debbye length	κ_D	$\sqrt{\frac{2c^b N_A e^2 z L^2}{\epsilon_0 k_B T}}$	9.207×10^{-3}
Non-dimensional Bjerrum length	λ_B	$\frac{e^2}{k_B T \epsilon_0 L}$	7.039×10^3

Table 3.1: Problem parameters and non-dimensional numbers.

where ψ^* is the Coulombic potential due to singular charges,

$$\psi^*(\mathbf{x}) = \frac{\lambda_B}{4\pi\epsilon^+} \sum_{i=1}^N \frac{z_i}{|\mathbf{x} - \mathbf{x}_i|}, \quad (3.11)$$

and ψ_0 satisfies the following Poisson's problem

$$\nabla \psi_0 = 0, \quad \forall \mathbf{x} \in \Omega^+, \quad (3.12)$$

$$\psi_0 = -\psi^*, \quad \forall \mathbf{x} \in \Gamma. \quad (3.13)$$

Utilizing the decomposition shown above, the regular (non-singular) part of the solution is given by solving

$$-\nabla \cdot (\epsilon \nabla \hat{\psi}) + \kappa^2(\mathbf{x}) \sinh(\hat{\psi}) = 0, \quad \forall \mathbf{x} \in \Omega \setminus \Gamma. \quad (3.14)$$

3.3.1 Numerical Method

The numerical method, implemented on non-graded adaptive Octree grids, follows the general description given in [85], where each snapshot of the protein is treated independently. The surface and grid generation is done using the level set framework

developed by Min and Gibou [90]. The entire method is parallelized in a shared memory fashion using OpenMP [91, 92]. Calculations were performed on Octrees of maximum level ranging from 7 to 11.

Error Estimation

To determine the convergence of the overall method, we construct numerical error estimators. These estimators can be employed independently of the numerical approach, because they rely only on the decomposed PDE. The error on the interface representation, gradient of the solution (*i.e.* electric field), and solution itself can be estimated using the following metrics e_Γ, e_E, e_ψ

$$e_\Gamma = \frac{1}{|\Gamma|} \left| \int_\Gamma \nabla \psi^* \cdot \mathbf{n} - \sum_i^N \frac{\lambda z_i}{4\pi\epsilon^+} \right|, \quad (3.15)$$

$$e_E = \frac{1}{|\Gamma|} \left| \int_\Gamma \nabla \psi_0 \cdot \mathbf{n} \right|, \quad (3.16)$$

$$e_\psi = \frac{1}{|\Gamma|} \int_\Gamma |\psi^* + \psi_0|, \quad (3.17)$$

where $|\Gamma|$ is the surface area of Γ . Numerical errors for e_Γ can only come from calculating the normal or approximating the surface integral. When computing e_E , the main source of error would be in $\nabla \psi_0$. Finally, e_ψ is the average error in ψ_0 on the interface.

Convergence Study

Figure 3.3a depicts the spike protein's electrostatic potential at the initial frame (closed configuration) as the mesh is refined (denoted as $(\max_{\text{level}})$). The positive electrostatic potential is colored in blue, no charge in white, and the negative electrostatic potential is shown in red. For the least resolved simulations, ($\max_{\text{level}} = 7, 8$, it is clear that we do not get an accurate depiction of the electrostatic map, but we are able to, roughly, reconstruct the overall shape of our spike protein. As we increase the grid resolution, we obtain a more detailed representation of the protein and, similarly, we are able to clearly see variations in the electrostatic map.

In Figure 3.3b we see all three error metrics converge with increasing resolution (the same resolutions that were used in 3.3a). From these results, we are confident that the method is correctly implemented and that the most refined simulations accurately capture the S protein's electrostatic potential, thus, hereafter, our results are based on $\max_{\text{level}} = 11$.

3.4 Results

In this section, we discuss our observations of the spike protein's electrostatic potential over time in relation to the ACE2 receptor. The part of the spike containing

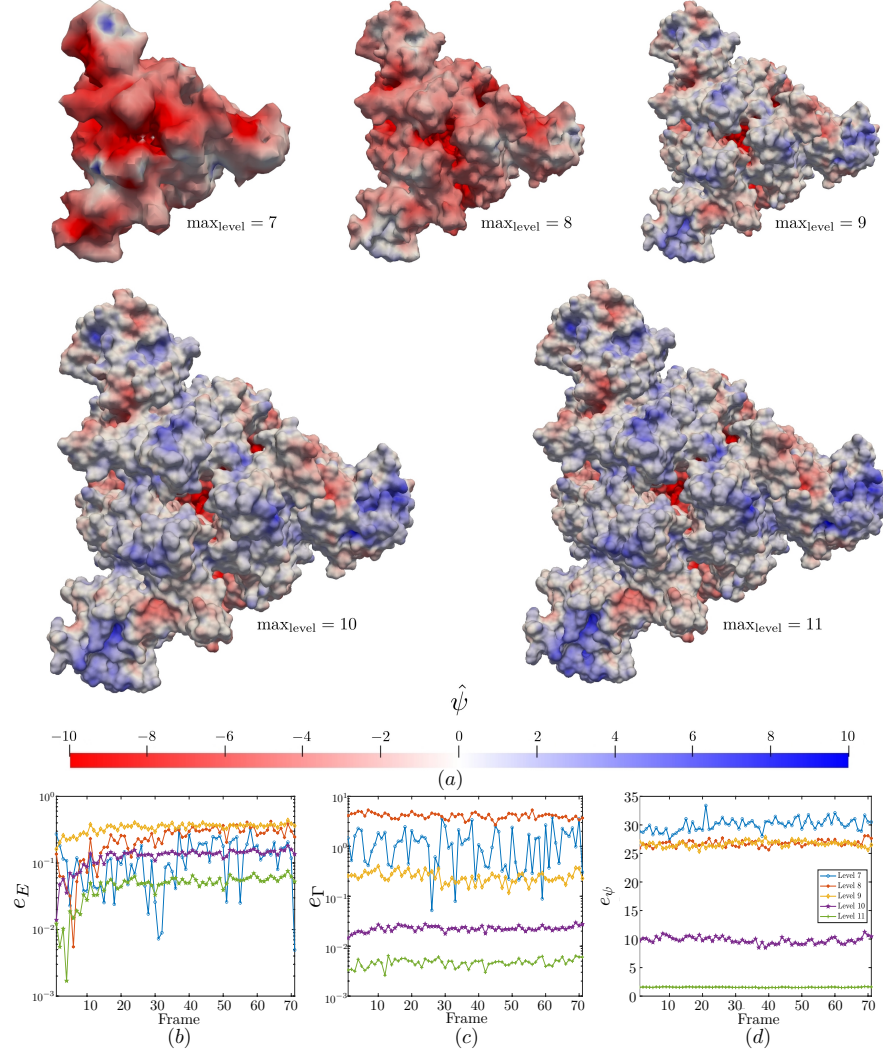


Figure 3.3: (a) Evolution of the spike's Solvent Excluded Surface and electrostatic potential as the maximal resolution increases. (top down view in the closed configuration (frame 1)). (b – d) Convergence of the interface representation (e_Γ), surface normal electric potential (e_E), and surface potential (e_ψ).

the three RBDs will be referred to as the top part, and the predominantly negatively charged portion as the bottom part. We see that the S protein remains predominantly in its closed conformation, hiding its receptor-binding domains (RBDs). This configuration would impede its binding to the ACE2 receptor. So, in order to bind with ACE2, the S protein transforms into its open conformation, revealing its binding interface.

From the simulations seen in Figure 3.4a, the top part of the spike protein is shown to be mostly positively charged, aligning with the fact that the ACE2 binding site (Figure 3.5) is negatively charged. However, we notice that the top of the spike also uncovers

a negatively charged core, χ . So, as the spike opens, it is possible that it would repulse the ACE2 receptor. Thus, the goal here is to characterize the geometrical and electric properties of χ , the cloud $\mathcal{C}$ that it creates, and the binding site $\mathcal{B}$.

3.4.1 Spike Protein Core

The negatively charged core χ , illustrated in Figure 3.4*b*, is the area of the SES at the top of the protein where the electric potential is more negative than the threshold value $c_\chi = -5$, this portion is colored in red. From figure 3.4 *d* and *e*, we are able to see the relative size and potential variations over the molecular trajectory. It is apparent from Figure 3.4, that while area shrinks ($\approx 50\%$), the average potential variation remains small ($\approx 10\%$). For both of these cases, the largest variations occur, roughly, in the first 10 iterations.

3.4.2 Electric Cloud

The negatively charged electric cloud, $\mathcal{C}$, generated by the core χ , is defined as the region in the upper part of the solvent where the potential is below the threshold value $c_c = -2$ (see Figure 3.4*c* - colored in pink). The results in Figure 3.4*d* and *e* indicate $\mathcal{C}$ expands by $\approx 42\%$ as the spike opens, while its average potential decreases in magnitude by 30%. We interpreted this phenomenon as a dissolution of the negative charges caused by the spike opening. When the spike opened to remove one its monomers between the core and solvent, this may have diluted the surface charges in the solvent; thus the core and its charge contracted.

3.4.3 Electric Field

Figure 3.6 illustrates streamlines for the spike's electric far field. In the (top image) closed configuration, the majority of streamlines emerging from the bottom part of the spike are pointing away from the spike. Only a few of the streamlines return to the protein. In the upper portion of the protein, the streamlines are predominantly closed, suggesting that in this configuration the spike protein may not be able to attract opposite charges at long range; so in this conformation, the spike has a limited attracting potential. As the spike opens (bottom image), the streamlines at the bottom remain unchanged from the closed configuration. However, in the top of the protein we now see a rearrangement and the streamlines are predominantly open. The streamlines now reach further outward, potentially attracting the ACE2 receptor.

Figure ??, once again, depicts the entire electric field structure; additionally, it presents the region, colored in blue, in the solvent where the electric field points away from the protein, or equivalently where negative charges would be drawn to the spike. In the closed (left) configuration, this region is predominantly located above the spike, while in the open configuration it is split into three sub-regions, each centered around one of the RBDs. In the open (right) configuration, we observe the sub-region centered

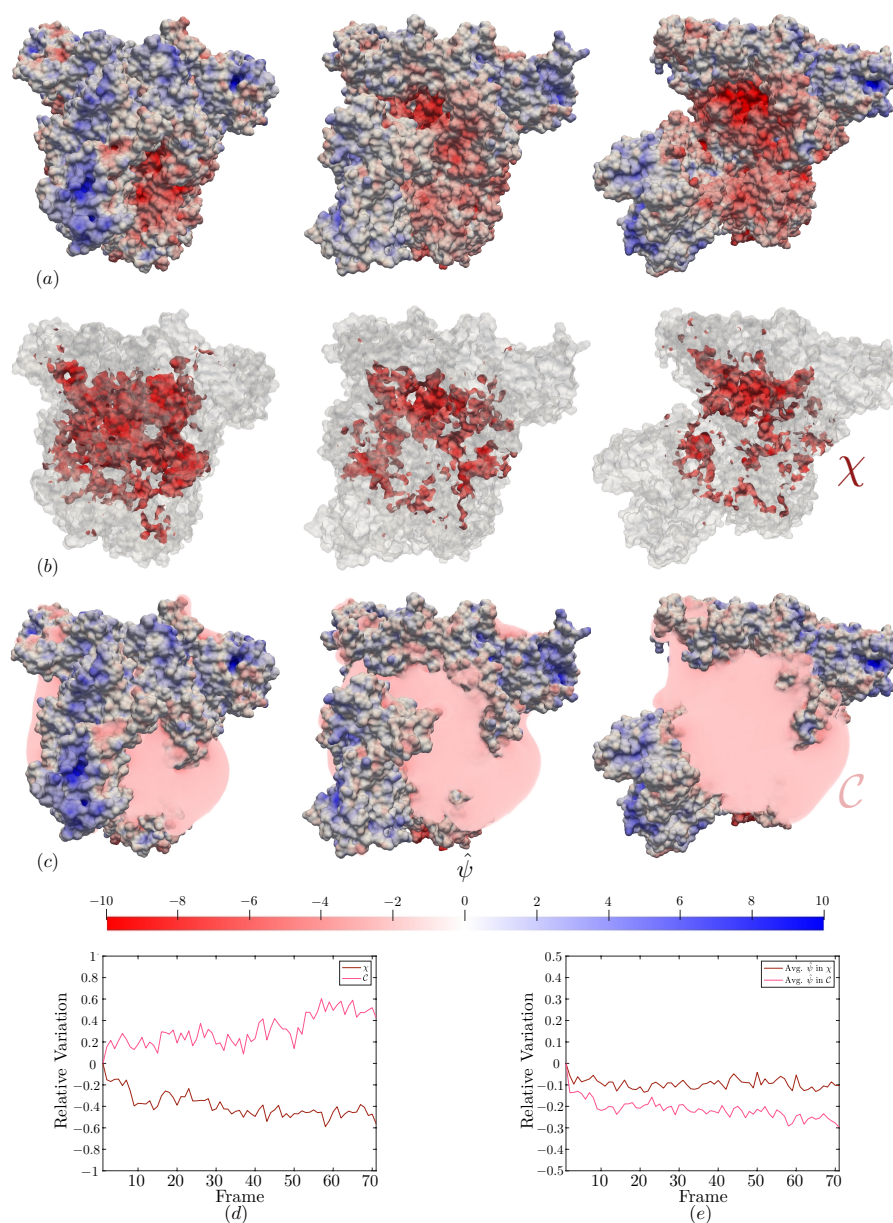


Figure 3.4: (a) Potential map of the SARS-Cov-2 spike open (frame 1), intermediate (frame 35), and close configuration (frame 71). (b) Evolution of the negatively charged core as the spike opens. (c) Repulsive electric cloud generated by the negatively charged core. (d – e) Characterization of the negatively charged core χ and repulsive electric cloud C : relative size (d) and potential (e) variations over the entire molecular trajectory.

around the now visible RBD. This portion represents 45% of the total region colored in blue and contains 34% of its total electric potential energy.

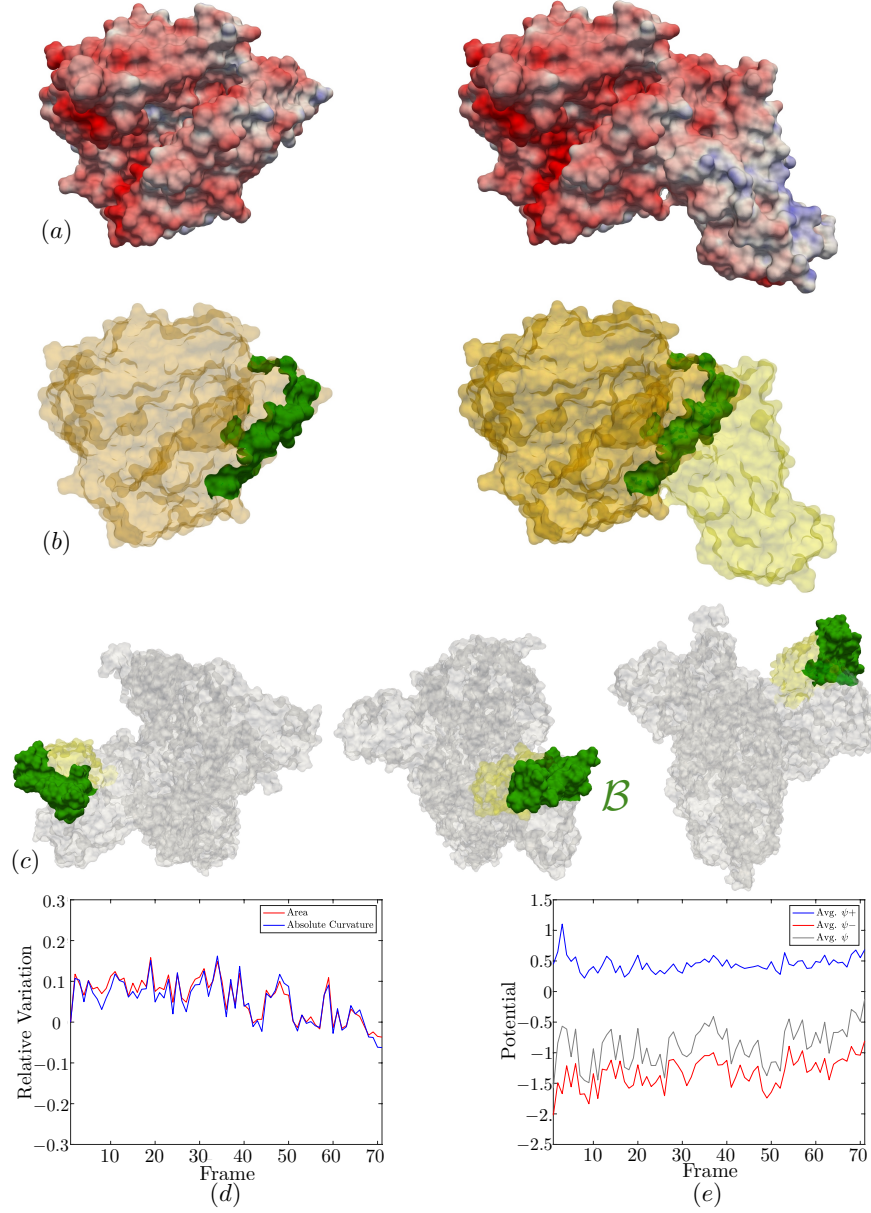


Figure 3.5: (a) Portion of the ACE2 receptor displaying surface charge (left). ACE2 and RBD of the spike protein connected (right). (b) Binding site of ACE2 highlighted in green (left). Binding site of ACE2 highlighted within the ACE2 (orange) and RBD (yellow) connection (right). (c) SARS-Cov-2 spike in open conformation with RBD (yellow) and binding site, $\mathcal{B}$ (green) highlighted. (d – e) Characteristics of the binding site, $\mathcal{B}$, of the spike protein. (d) Relative variance of the area and absolute curvature along with average potential. (e) average positive and negative potential.

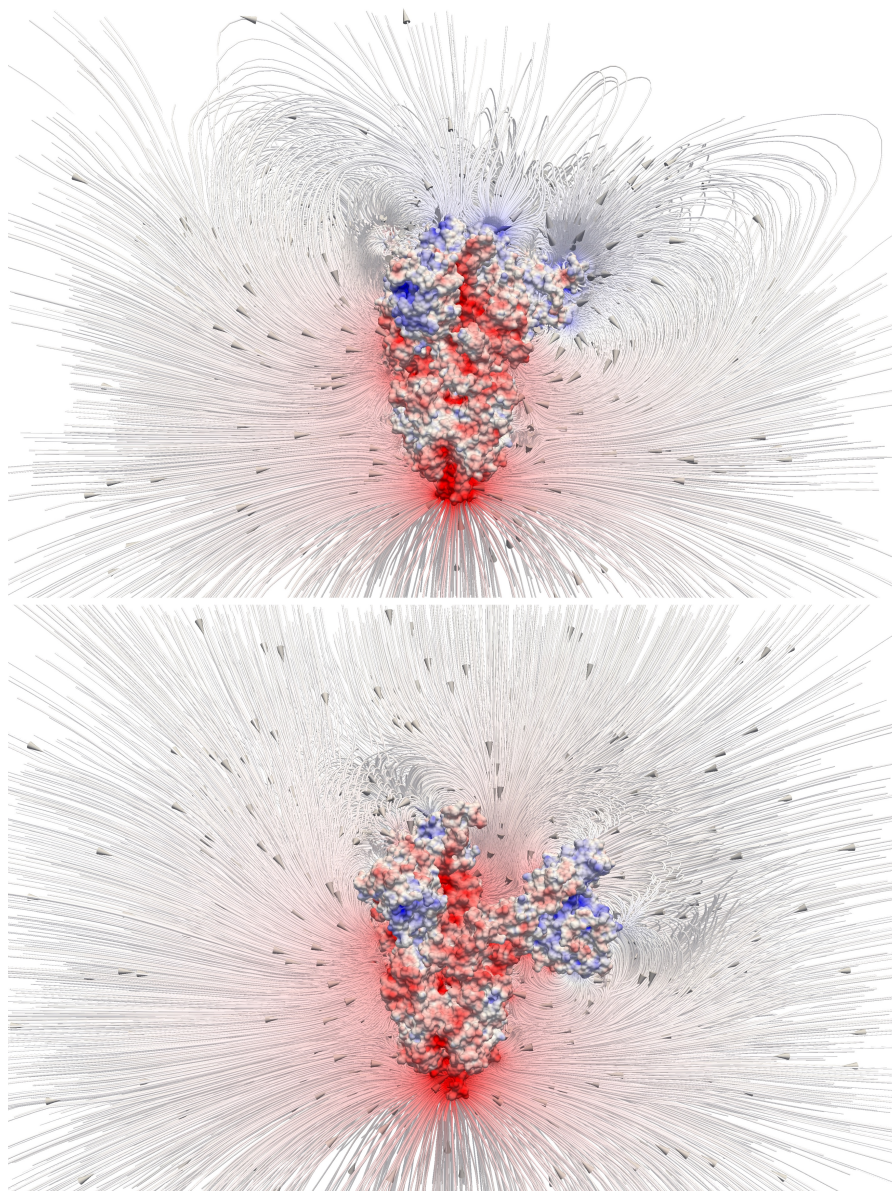


Figure 3.6: Streamlines and direction of the electric field in the close (top) and open (bottom) configurations. As the spike opens, the streamlines emanating from the upper part of the molecule are observed to open as well. This effect is particularly noticeable for the lines emerging from the RBD domain (opening extremity on the right).

3.5 Conclusion

The principal objective was to reveal the impact of the SARS-CoV-2 spike protein's transition from its open to closed conformation. We examined the entire protein's electrostatic potential dynamics, focusing particularly on its receptor binding domains. In the Folding@home trajectory we analyzed, one of the three monomers detaches

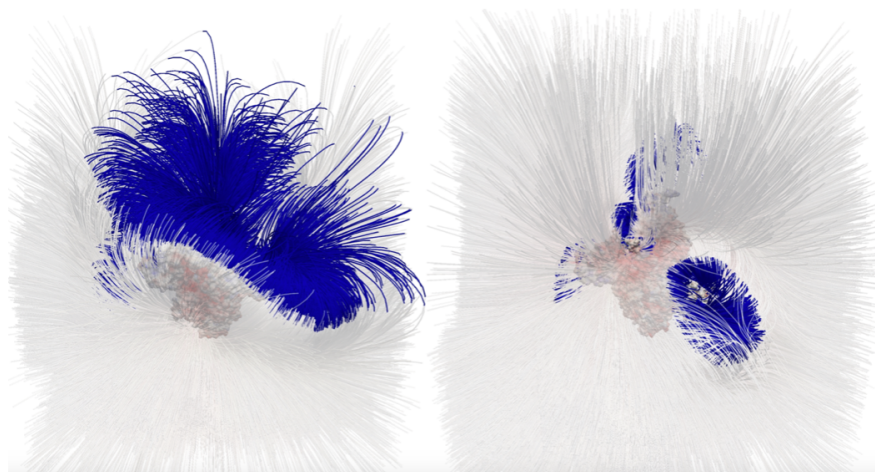


Figure 3.7: Streamlines of the electric field ($\mathbf{E} = -\nabla\hat{\psi}$) in the close (left) and open (right) configurations over the entire computational domain. The portion of the streamlines where the electric field is pointing away from the protein (*i.e.* $\nabla\psi \cdot \mathbf{x} = -\mathbf{E} \cdot \mathbf{x} < 0$), and therefore negative charges would be drawn to the it, is depicted in blue.

from the core of the complex and becomes visible to the surrounding environment, consistent with recent cryo-EM derived structures [86]. Our results show that despite the dramatic molecular displacement engendered by this strategic re-positioning, the geometric and electric properties of the RBD itself remained largely unaltered. However, changes emanating from the exposure of the core of the S protein caused a change in the electric fields surrounding the RBD.

From our continuum analysis, both of the surface potential and the volumetric potential in the vicinity of the binding region pointed to the existence of an inner negatively charged core on the surface of the spike, which was revealed to the solvent as the spike opened. This negatively charged surface shrank as the spike opened, inducing a negatively charged region between the exposed RBD and the two hidden ones. Initially, the emergence of this electric cloud was puzzling; as the binding site of the ACE2 receptor itself is almost entirely negatively charged – we expected this cloud to repulse the receptor. However, this cloud did not envelop the spike’s binding site, $\mathcal{B}$ – it became more positively charged. The electric field, which we observe uncoiling above the protein, exhibits a dramatic rearrangement. This is correlated to the emergence of the negatively charged cloud. The transition from the entire top part of the spike attracting negatively charged structures (*e.g.* ACE2 binding site), to attracting a specific RBD can be seen as strategic.

At the computational level, we developed a novel framework, combining molecular dynamics simulations with a continuum physics numerical solver to produce high-resolution multiscale dynamical potential maps of deforming proteins and their surrounding environment. We hoped that the approach presented here could provide

the scientific community with invaluable information, which could potentially be used to elucidate how changes to physical properties such as electrostatics translate into function.

Spatially Resolved Multi-Generation Modeling of Prion Aggregate Inheritance in Budding Yeast

4.1 Introduction

Protein aggregation is a hallmark of many cellular stress responses and is strongly implicated in neurodegenerative disease. Misfolded proteins can disrupt normal cellular function, interfere with intracellular transport, and trigger toxic stress pathways [93, 94]. In some cases, misfolded proteins can also act as templates that convert normally folded proteins into aggregated states, thereby propagating pathological conformations over time [95]. One important class of such proteins is prions, which are transmissible and associated with fatal neurodegenerative mammalian diseases including sheep scrapie, bovine spongiform encephalopathy, and Creutzfeldt–Jakob disease [96].

The yeast *Saccharomyces cerevisiae* provides a tractable and ethically advantageous system for studying protein aggregation and prion propagation [97, 98, 99, 100, 101]. In contrast to mammalian systems, prion-like protein aggregation in yeast can be studied experimentally without causing harm to the organism, while still preserving many of the essential features of aggregate inheritance and persistence. In fact, *Saccharomyces cerevisiae* contains several harmless phenotypes that we now know to be caused by prion proteins [102]. Experimental studies have shown that aggregated proteins in dividing yeast cells are often inherited asymmetrically, with aggregates preferentially retained in the mother cell [103]. This asymmetry has important physiological consequences, including replicative aging, impaired stress response, and reduced cellular fitness [104, 105]. Experimental work has established that this asymmetric retention is not driven by active transport but is instead consistent with diffusion-limited partitioning within the confining geometry of the dividing cell [5], suggesting that spatial and geometric effects are central to understanding aggregate inheritance.

Mathematical modeling has played an important role in clarifying the mechanisms

of prion propagation in yeast. The dominant framework has been the nucleated polymerization model (NPM) and its extensions, which describe aggregate dynamics through systems of ordinary differential equations (ODEs) treating each cell as a well-mixed compartment [106, 107]. These models have been successful in capturing coarse aggregation kinetics, relating molecular parameters to observable phenotypes, and studying the role of chaperones such as Hsp104 in aggregate fragmentation [107]. More recently, impulsive differential equation frameworks have extended this approach to account for asymmetric division and multi-generational propagation at the colony level, successfully reconciling previously conflicting experimental observations on the *[PSI+]* prion [108]. Stochastic multi-scale models have further enabled the tracking of discrete aggregate numbers across cell lineages using Gillespie-type simulations [109, 110, 111]. Despite their successes, all of these approaches treat the intracellular environment as spatially homogeneous. While some frameworks encode asymmetric partitioning at division as a prescribed parameter [108, 110], none derive this asymmetry from the underlying spatial and diffusive dynamics that experiments suggest are its physical origin [5].

In this chapter, we address the next natural step by extending the framework of Heydari *et al.* [6], who demonstrated that geometry alone can produce asymmetric aggregate partitioning in a single mother–daughter division event, to simulate explicit multi-generation yeast cell lineages. Banwarth-Kuhn and Sindi [112] identify the requirements for such a framework explicitly: it must incorporate asymmetric protein distribution at the time of division, simulate prion dynamics within each individual cell using spatially resolved models of intracellular dynamics, and track protein concentration through different cell lineages to investigate the impact of lineage history on aggregate persistence. While the governing equations are fully spatially resolved, lineage-level analysis is performed using cell-integrated aggregate quantities, as the spatial fields serve primarily to produce physically consistent partitioning at each division event rather than to characterize intracellular spatial patterns across generations. The present work satisfies the first two of these requirements directly, and the parallelized pipeline developed here is in principle extensible to the third—varied cell cycle lengths between generations—which we leave as a direction for future work.

The model incorporates diffusion of soluble and aggregated protein species, reversible aggregation reactions, and a logistic growth term reflecting the increasing protein content of growing cells, with diffusion coefficients calibrated directly from fluorescence loss in photobleaching (FLIP) experimental data [113]. Cell geometry and division are represented using a level-set-based description of the evolving mother–daughter domain, and the governing equations are discretized using a mass-conservative finite volume method on adaptive octree grids. To simulate full binary lineage trees to generation five and beyond, we developed a parallelized pipeline that automates the inter-generational handoff and distributes simulations across a high-performance computing (HPC) cluster, a computational investment necessitated by the cost of three-dimensional, time-dependent PDE solves that has no analogue in the sequential execution of well-mixed or stochastic lineage models. This combination

of spatial resolution, biologically informed parameters, and lineage-scale computation represents a qualitatively new mode of investigation for yeast prion dynamics.

This framework allows us to simulate protein aggregation and inheritance over multiple generations and to study how aggregate distributions evolve across explicit cell lineages. In particular, we investigate how model parameters govern the balance between aggregate dilution and long-term retention, and we analyze the resulting generation-dependent patterns in aggregate burden. Beyond reproducing asymmetric inheritance, the model provides a computational setting for identifying regimes in which prion burden persists across generations and regimes in which it is effectively suppressed.

The remainder of this chapter is organized as follows. Section 4.2 introduces the governing reaction–diffusion equations and the level-set representation of the dividing yeast geometry. Section 4.3 briefly reviews the conservative finite volume method used for all simulations. Section 4.4 describes the simulation setup, including biologically motivated parameter choices and the calibration of diffusion and growth rates. Preliminary multi-generation results establishing the key qualitative features of the model are presented in Section 4.5. Section 4.6 describes the binary lineage representation and HPC parallelization strategy. Full lineage simulation results across a structured parameter sweep are presented in Section 4.7. Finally, Section 4.8 discusses the biological interpretation, limitations, and directions for future work.

4.2 Mathematical Modeling

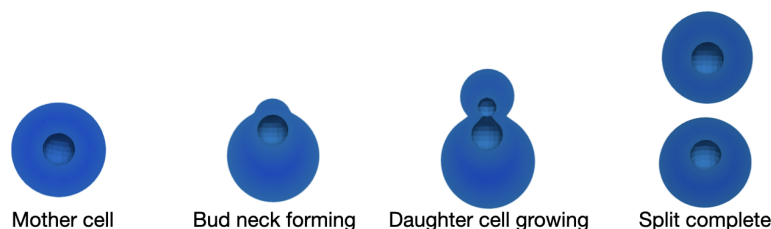


Figure 4.1: Three-dimensional simulation of a dividing yeast cell with a splitting nucleus. As time progresses, a daughter cell buds from the mother cell, grows, and ultimately detaches.

Mathematical modeling provides a useful framework for investigating intracellular protein dynamics, but it necessarily requires simplifying assumptions about both the cell and its environment. In the present work, we model protein aggregation inside a deforming yeast cell whose geometry evolves throughout the budding process. Unlike well-mixed ordinary differential equation models, a spatially resolved formulation can represent intracellular heterogeneity, diffusion-driven redistribution, and the geometric

effects of asymmetric division. These features are essential for studying how aggregates are retained or diluted across generations.

The biological process of interest involves two protein species: a monomer A and an aggregate species B , which is modeled here as a dimer. We assume that two monomers can combine to form a dimer, while the dimer can dissociate into two monomers, according to the reaction scheme



The corresponding concentrations of the monomer and aggregate are denoted by ψ_A and ψ_B , respectively.

4.2.1 Governing Equations

Let $\Omega(t) \subset \mathbb{R}^3$ denote the time-dependent intracellular domain of a dividing yeast cell. We assume that both species diffuse throughout the cell with diffusion coefficients D_A and D_B , respectively. The dimerization and degradation rates are denoted by γ_{AB} and γ_{BA} . The reaction-diffusion system governing the concentrations ψ_A and ψ_B is therefore

$$\frac{\partial \psi_A}{\partial t} - D_A \Delta \psi_A = 2\gamma_{BA}\psi_B - \gamma_{AB}\psi_A^2 \quad \forall \mathbf{x} \in \Omega(t), \quad (4.2)$$

$$\frac{\partial \psi_B}{\partial t} - D_B \Delta \psi_B = \frac{1}{2}\gamma_{AB}\psi_A^2 - \gamma_{BA}\psi_B \quad \forall \mathbf{x} \in \Omega(t). \quad (4.3)$$

The cell membrane is assumed to be hermetic, so both species satisfy no-flux boundary conditions on the cell boundary $\partial\Omega(t)$:

$$\nabla \psi_A \cdot \mathbf{n} = 0 \quad \forall \mathbf{x} \in \partial\Omega(t), \quad (4.4)$$

$$\nabla \psi_B \cdot \mathbf{n} = 0 \quad \forall \mathbf{x} \in \partial\Omega(t), \quad (4.5)$$

where $\mathbf{n}$ denotes the outward unit normal to $\partial\Omega(t)$. These boundary conditions enforce conservation of total protein mass in the absence of reactions and growth.

Diffusion System with a Nonlinear Source

To better understand the effect of cellular growth independently of aggregation, we first consider a limiting case in which the reaction terms are removed by setting $\gamma_{AB} = \gamma_{BA} = 0$. In this case, the aggregate concentration remains identically zero, and the monomer concentration evolves under diffusion and a nonlinear source term:

$$\frac{\partial \psi_A}{\partial t} - D_A \Delta \psi_A = r\psi_A(1 - \psi_A) \quad \forall \mathbf{x} \in \Omega(t), \quad (4.6)$$

$$\frac{\partial \psi_B}{\partial t} - D_B \Delta \psi_B = 0 \quad \forall \mathbf{x} \in \Omega(t). \quad (4.7)$$

Here, r denotes a growth rate for the monomer concentration. This term is introduced to reflect the fact that protein content may increase as the cell grows. The diffusion-only system serves as a useful baseline for calibrating and interpreting the effect of the nonlinear growth term in the absence of aggregation.

Reaction–Diffusion System with a Nonlinear Source

To incorporate both protein growth and aggregation, we augment the reaction–diffusion system with the same nonlinear source term for the monomer concentration:

$$\frac{\partial \psi_A}{\partial t} - D_A \Delta \psi_A = 2\gamma_{BA}\psi_B - \gamma_{AB}\psi_A^2 + r\psi_A(1 - \psi_A) \quad \forall \mathbf{x} \in \Omega(t), \quad (4.8)$$

$$\frac{\partial \psi_B}{\partial t} - D_B \Delta \psi_B = \frac{1}{2}\gamma_{AB}\psi_A^2 - \gamma_{BA}\psi_B \quad \forall \mathbf{x} \in \Omega(t). \quad (4.9)$$

This reaction–diffusion–logistic system models the interplay between intracellular diffusion, reversible aggregation, and protein production during cell growth. In particular, it allows us to study whether aggregate burden is amplified, retained, or diluted as the yeast cell grows and divides.

4.2.2 Level Set Representation

Following the level set framework developed in [6], we represent the interior of the dividing yeast cell as a time-dependent domain $\Omega(t)$. This domain is constructed from multiple subdomains, each associated with a level set function. The full geometry is obtained by taking unions and intersections of these subdomains, which correspond to min and max operations on the associated level set functions.

We denote the mother cell level set function by ϕ_m and the daughter cell level set function by ϕ_d . Similarly, the mother and daughter nuclei are represented by ϕ_{mn} and ϕ_{dn} , respectively. In this chapter, we consider the biologically relevant case of a splitting nucleus, in which the daughter nucleus gradually detaches from the mother nucleus during budding. The daughter nucleus is described by

$$\phi_{dn}(\mathbf{x}, t) = |\mathbf{x} - \mathbf{x}_{dn}(t)| - r_{mn}(t) - r_{dn}(t). \quad (4.10)$$

The full intracellular domain is then given by

$$\Omega(t) = (\Omega_m \cup \Omega_d(t)) \cap (\Omega_{mn}(t) \cup \Omega_{dn}(t))^c, \quad (4.11)$$

which is equivalent, at the level set level, to

$$\max(\min(\phi_m, \phi_d), -\min(\phi_{mn}, \phi_{dn})). \quad (4.12)$$

This representation provides a flexible way to describe the evolving mother–daughter geometry while preserving a sharp description of the cell boundary and nucleus.

4.3 Numerical Method

This section provides a brief overview of the conservative finite volume method used to solve the reaction–diffusion systems introduced in Section 4.2. The numerical framework employed here follows the approach developed by Heydari *et al.* [6], which was designed specifically for reaction–diffusion systems evolving on deforming domains represented using level set geometry. A key feature of this method is that it preserves total mass during the simulation, even as the computational domain evolves in time.

Traditional finite volume formulations applied to time-dependent problems may fail to preserve mass exactly when the domain changes between time steps. In such cases, the spatial integrals defining the finite volume discretization are evaluated on control volumes defined at time t^{n+1} , while the integrands correspond to quantities evaluated at time t^n . This mismatch can introduce small but cumulative mass errors over long simulations. To address this issue, the conservative formulation proposed in [6] integrates the governing equations over a space–time control volume, ensuring that mass is preserved across time steps.

4.3.1 Finite Volume Discretization

To illustrate the discretization procedure, consider the simplified case of the diffusion system with a nonlinear source term given in equation (4.6), while temporarily ignoring the second equation (4.7). The governing equation is

$$\frac{\partial \psi_A}{\partial t} - D_A \Delta \psi_A = r \psi_A (1 - \psi_A). \quad (4.13)$$

We begin by discretizing the equation in time using a semi-implicit scheme. Let superscripts n and $n + 1$ denote quantities evaluated at times t^n and t^{n+1} , respectively. The semi-discrete formulation becomes

$$\frac{\psi_A^{n+1} - \psi_A^n}{\Delta t} - D_A \Delta \psi_A^{n+1} = r \psi_A^{n+1} - r (\psi_A^n)^2. \quad (4.14)$$

Here the logistic growth term $r \psi_A (1 - \psi_A)$ is treated in a semi-implicit fashion: the linear component $r \psi_A$ is evaluated at t^{n+1} and absorbed into the left-hand side of the resulting linear system, while the nonlinear component $-r \psi_A^2$ is evaluated explicitly at t^n and contributes to the right-hand side. Rearranging these terms yields

$$(1 - \Delta t r) \psi_A^{n+1} - \Delta t D_A \Delta \psi_A^{n+1} = \psi_A^n - \Delta t r (\psi_A^n)^2. \quad (4.15)$$

To construct the finite volume formulation, we integrate this equation over the control volume associated with grid cell C_i . Let

$$V_i^{n+1} = C_i \cap \Omega^{n+1}$$

denote the portion of the control volume lying inside the computational domain at time t^{n+1} . Integrating over V_i^{n+1} yields

$$\int_{V_i^{n+1}} (1 - \Delta tr) \psi_A^{n+1} - \int_{V_i^{n+1}} \Delta t D_A \Delta \psi_A^{n+1} = \int_{V_i^{n+1}} \psi_A^n - \Delta tr (\psi_A^n)^2. \quad (4.16)$$

The diffusion term can then be converted to a surface integral using the divergence theorem:

$$\int_{V_i^{n+1}} (1 - \Delta tr) \psi_A^{n+1} - \Delta t D_A \int_{\partial V_i^{n+1}} \nabla \psi_A^{n+1} \cdot \mathbf{n} = \int_{V_i^{n+1}} \psi_A^n - \Delta tr (\psi_A^n)^2. \quad (4.17)$$

In the traditional finite volume formulation, this step would complete the discretization. However, because the domain evolves between time steps, the control volumes V_i^n and V_i^{n+1} may differ. As a result, the right-hand side of the equation does not represent the total mass contained in the control volume at time t^n .

To address this issue, the conservative formulation described in [6] integrates the governing equations over a space–time control volume rather than over purely spatial volumes. This approach guarantees that the numerical method preserves total mass even as the domain geometry evolves due to cell growth and division.

4.3.2 Geometry Representation and Grid Structure

The computational domain $\Omega(t)$ is represented using level set functions as described in Section 4.2. The numerical method operates on adaptive octree grids, which allow efficient resolution of the evolving geometry while concentrating computational effort near the cell boundary and other regions of interest.

At each time step, the level set functions define the intersection between the computational grid and the interior of the cell. The finite volume integrals are then evaluated over the resulting cut cells. This approach allows the evolving mother–daughter geometry to be captured without explicitly remeshing the domain at every time step.

The resulting discretization produces a sparse linear system for the unknown concentrations at each time step. The nonlinear reaction terms are treated explicitly, while the diffusive terms are treated implicitly, resulting in a stable semi-implicit scheme suitable for long-time simulations.

This numerical framework enables the simulation of reaction–diffusion processes on deforming domains while preserving the physical constraint of mass conservation. In the next section, we describe the parameter choices and simulation setup used to model prion aggregation during yeast cell division.

4.4 Simulation Setup and Parameter Calibration

4.4.1 Simulation Parameters

We describe the key parameters used to configure the cell division simulations. Byrne *et al.* [114] found the cell reproduction time to be 1.46 hours on average (for the strains YJW512 [PSI +]). Given this information, we take the cell division time to be $T = 90$ min, meaning that at this time the budding process is complete; however, to allow the daughter cell to grow to its mature size, the simulation final time is taken to be $T = 190$ min. Next, we chose the initial mother cell volume to be $37\mu\text{m}^3$ [6]; from [115], we have that the volume of the daughter cell must be two-thirds of the volume of the mother cell when budding is complete. Similarly, for the nucleus size, we use the parameters from [6].

In all of the simulations, we use constant initial concentration fields with random spatial noise. These initial profiles are re-scaled so that the initial total mass is identical across all examples shown here. For multiple generation simulations, the initial concentration field is based on the previous generation's final concentration profile. Although we use the minute and micrometer as our characteristic time and length scales, all concentrations are non-dimensionalized and all quantities will be referred to in their dimensionless form.

The principal simulation parameters are summarized below.

Parameter	Value
Budding Time	90 min
Simulation Time (T)	190 min
Mother Radius (r_m)	$2.01\ \mu\text{m}$
Daughter Radius After Split (r_d)	$1.81\ \mu\text{m}$
Nucleus Radius (r_n)	$0.89\ \mu\text{m}$
Characteristic concentration (ψ_0)	$713\ \mu\text{m}^{-3}$

4.4.2 Diffusion Coefficients (FLIP fitting)

To obtain biologically informed diffusion coefficients, we use fluorescence loss in photobleaching (FLIP) data reported in [113]. This data is specific to different conformations of the yeast prion protein, illustrating relative fluorescence intensity as a function of time post-bleaching (see Fig 4.2). Derdowski *et al.* establish that prion proteins adopt a variety of conformations or strains, which create phenotypes of distinct severity and stability. They focused on the $[\text{PSI}+]^{\text{Weak}}$ and $[\text{PSI}+]^{\text{Strong}}$ conformations of the yeast prion protein Sup35 in addition to two non-prion proteins (*psi-* and diffuse); the transfer of Sup35 to daughter cells was monitored.

Figure 4.2 shows that fluorescence intensity decreases over time for all strains considered. The principal distinction is that the prion strains exhibit slower fluorescence loss than the non-prion strains, consistent with reduced effective mobility.

This is in line with how we would like to simulate the diffusions D_A, D_B , as D_A would diffuse faster than the simulated aggregate with diffusion D_B .

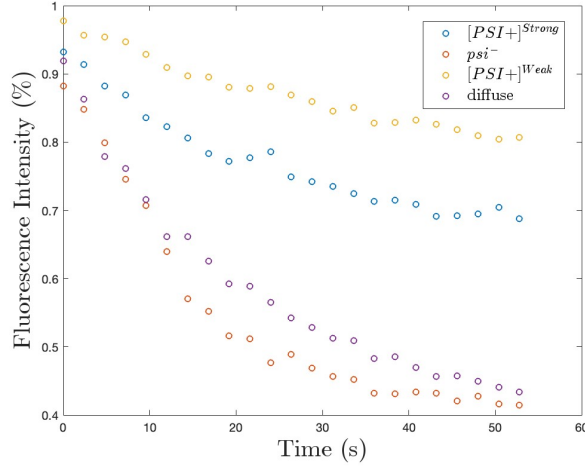


Figure 4.2: FLIP data from [113]. Two conformations of the yeast prion protein ($[PSI+]^{Weak}$ and $[PSI+]^{Strong}$) are depicted in yellow and blue, respectively. The remaining data ($psi-$ and diffuse) are non-prion strains and are seen to diffuse into the daughter cell more easily.

Model Fitting

From figure 4.2, it is clear that the data follows the same general trend for each of the strains. Rather than attempting to reproduce the full FLIP experiment in the deforming-domain PDE framework, we use a simple phenomenological decay model to estimate effective diffusion coefficients (D_A, D_B) from the observed intensity curves.

Our proposed model has the following form:

$$I(t) = (I_f - I_0)e^{-\frac{L^2}{Dt}} + I_0, \quad (4.18)$$

where I_0, I_f are the initial and final experimental fluorescence intensities, respectively. L^2 is the squared length of the cell and D is the diffusion.

Our goal is not to interpolate the experimental curves exactly, but to extract effective diffusivities that reproduce their dominant time scales and relative ordering. For each of the prion and non-prion strains — $[PSI+]^{Weak}$, $[PSI+]^{Strong}$, $psi-$, and diffuse — figure 4.3 depicts the L^∞ error for various diffusion inputs. It is apparent that the diffusion values which minimize the error for the prion strains ($[PSI+]^{Weak}$, $[PSI+]^{Strong}$) are relatively close to each other, as is the proximity of the diffusions which minimize either of the non-prion strains ($psi-$, diffuse).

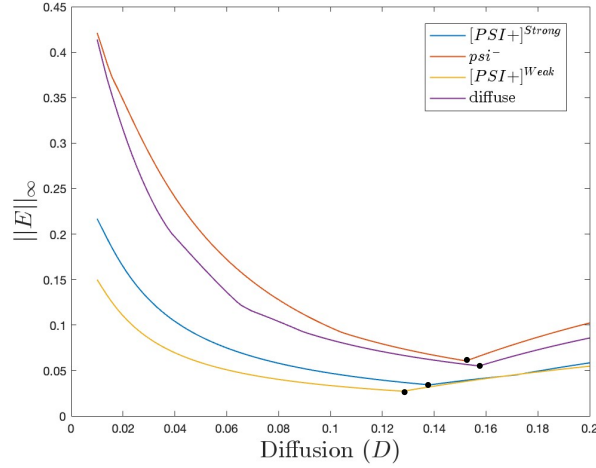


Figure 4.3: Fitting experimental data. The data provided is used to fit the decay model for all prion and non-prion strains. The diffusion which minimizes the error ($\|L\|_\infty$) between the model and experimental data, for each respective case, is emphasized as a black point.

Now that the diffusion coefficients are determined, Figure 4.4 shows how well the model reproduces the experimental data. For each strain, we substitute the estimated diffusion coefficient into equation 4.18 and compare the predicted intensity curve to the observed fluorescence loss. The model captures the general shape of the experimental curves across all four strains; the dominant timescale of fluorescence loss is well reproduced, and the relative ordering between strains is preserved: prion strains exhibit lower fluorescence loss than non-prion strains, consistent with reduced effective mobility. However, beginning at approximately $s = 30$ seconds, the predicted curves diverge from the data in all cases, underestimating the rate of fluorescence loss at later times.

This discrepancy is expected given the simplicity of the phenomenological decay model — the full FLIP experiment involves a complex bleaching geometry that is not explicitly represented here. Nonetheless, the fit is sufficient for our purposes: we require biologically motivated estimates of the relative mobility of soluble and aggregated Sup35, not a precise reproduction of the experimental protocol. The correct ordering between species ($D_A > D_B$) is clearly supported by the data, and this is the key property our model requires.

For the remainder of this chapter, we utilize the non-dimensionalized version of the diffusion coefficients determined from the FLIP data. We take D_A to be the diffusion coefficient fitted to the psi^- strain, and D_B to be fitted to the $[PSI+]^{Strong}$ strain, reflecting the expected mobility difference between soluble and aggregated species.

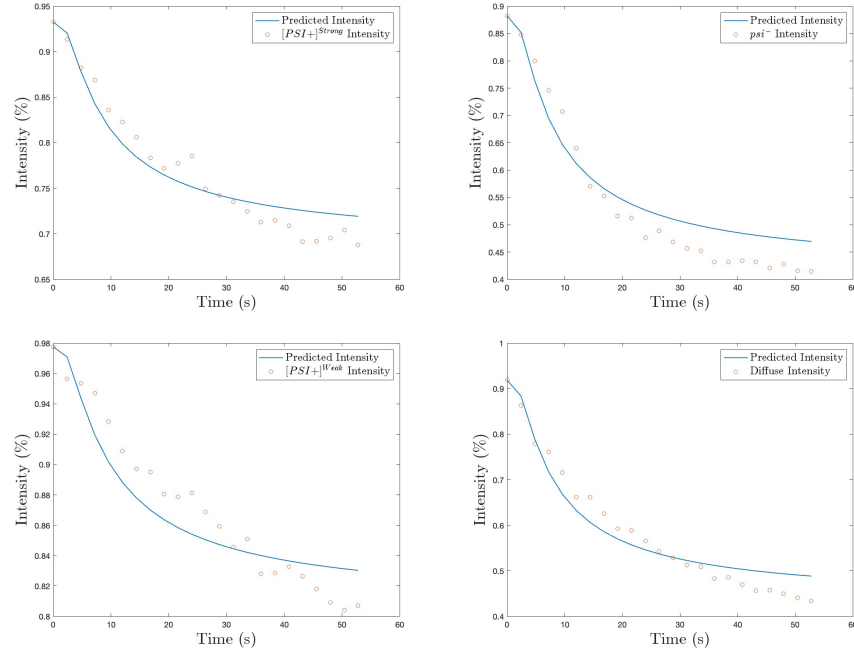


Figure 4.4: Model Fit: Comparison of how the experimental data fits the chosen model for each of the strains. Here, the respective optimal diffusion coefficients are used for the predicted intensity, meaning that 4.18 now has a specific value of D for each strain.

4.4.3 Growth Rate Determination

In this section, we vary the growth rate r for values in the range $(0, 1)$, while keeping the diffusion rates constant. The fraction of final mass for all considered growth rates are presented (see Fig. 4.5).

We measure the amount of transmitted material by computing the fraction of total mass transferred to the daughter cell over time:

$$F_A(t) = \frac{\int_{\Omega_D(t)} \psi_A}{\int_{\Omega_D(t) \cup \Omega_M(t)} \psi_A} = \frac{\int_{\Omega_D(t)} \psi_A}{M_T^0} \quad (4.19)$$

where $\Omega_D(t)$ and $\Omega_M(t)$ denote the inside of the daughter and mother cells, respectively, at time t , and M_T^0 is the initial total mass.

Figure 4.5 depicts the fraction of the mass that is transferred to the daughter cell for different growth rates, r . Recall, the duration of these simulations is 190 minutes, a little over double the amount of time it takes for the budding process to end (90 minutes); this is done in order to allow the daughter cell to grow to her full capacity and become prepared to split. Note that the daughter cell splits off from the mother cell at timestep = 129, or equivalently $t = 70$ minutes. By the end of the split process (timestep = 165,

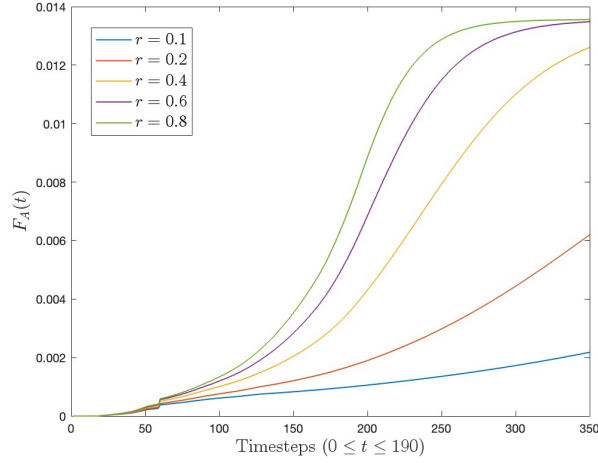


Figure 4.5: Fraction of monomer mass in the daughter cell, $F_A(t)$, for growth rates $r \in \{0.1, 0.2, 0.4, 0.6, 0.8\}$, with diffusion coefficients fixed at their FLIP-fitted values. Lower growth rates fail to reach a steady state within the 190-minute simulation window, indicating that the monomer concentration in the daughter cell has not yet reached a steady state.. At $r = 0.8$, F_A plateaus by the end of the simulation, satisfying the steady-state criterion used to select this value for all preliminary simulations.

or equivalently $t = 90$ minutes), the daughter cell is roughly two-thirds of the volume of the mother cell. Observing the results, specifically for the rates $r = 0.1, 0.2, 0.4$, we see that the fraction of the mass transferred to the daughter cell (F_A) is not reaching a steady state, meaning that if the simulation were allowed to continue, the daughter-cell monomer content has not yet reached its post-division plateau by the end of the simulation window. Technically, no mass is being transferred from the mother cell after the splitting is complete; instead, the figure depicts the growth rate at play, the monomer A continues to increase in mass until the daughter is ready to split.

Let us illustrate what the split process would look like for a growth rate that does not result in a steady state by the end of the simulation (see Fig. 4.6), in a diffusion only system. As in figure 4.5, here we cannot say with certainty that the mass of the daughter cell is done increasing. Because in this example we consider a diffusion only system, by the end of the simulation, we expect that the mother cell and daughter cell are identical. However, at $t_s = 350$ ($t = 190$), the daughter cell is not illustrated as having the same concentration map as the mother cell (*i.e.* the color for the concentration maps is not identical — the daughter cell is a lighter color than the mother cell, meaning it is not yet done growing in ψ_A mass).

From this point on, if we use the system containing the logistic growth rate we will use the growth rate $r = 0.8$.

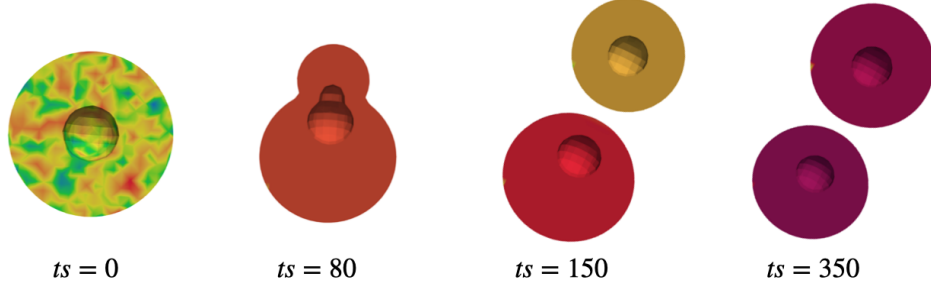


Figure 4.6: Time progression of the cell geometry for the growth rate $r = 0.4$, for a system with no reaction, only diffusion and logistic growth. The timestep is defined as ts , they are, respectively, $t = 0, 20, 81, 190$ minutes.

4.5 Preliminary Multi-Generation Results

4.5.1 Inter-Generational Initialization

At each division event, the spatially resolved concentration fields ψ_A and ψ_B are computed throughout the full budding process. Upon completion of division, the spatially averaged concentrations within the mother and daughter domains are computed and used as uniform initial conditions for the subsequent generation:

$$\bar{\psi}_k^{(n+1)} = \frac{1}{|\Omega_k|} \int_{\Omega_k} \psi_k^{(n)}(T) d\mathbf{x}, \quad k \in \{A, B\}, \quad (4.20)$$

where Ω_k denotes the respective post-division cell domain and T is the final simulation time. Random spatial noise of small amplitude is then superimposed on this uniform field, consistent with the initialization procedure used at generation zero (see Section 4.4).

This mean-field handoff is a deliberate modeling choice that preserves the biologically meaningful quantity (the inter-generational asymmetry in aggregate burden between mother and daughter), while treating each newly initialized cell as spatially uniform at birth. Crucially, the averaging is not performed immediately at the moment of cytokinesis, but rather after a full post-division growth phase during which the PDE has continued to evolve the concentration fields within each cell. The concentration fields plotted on the full computational mesh in Figure 4.8 show negligible spatial variation within individual cells by the end of each simulation ($T = 190$ min), consistent with the reaction-free validation in Figure 4.7, and lending direct support to the uniform initialization assumption for the subsequent generation. This approach also avoids the nontrivial numerical challenge of interpolating a full three-dimensional concentration field onto a freshly constructed mesh at each generation, which we leave as a direction for future work.

4.5.2 Reaction-Free Validation

Before introducing aggregation, we first verify the behavior of the diffusion–growth system in isolation. When $\gamma_{AB} = \gamma_{BA} = 0$, no aggregation occurs and $\psi_B \equiv 0$; the monomer ψ_A evolves purely under diffusion and logistic growth. In this case, both the mother and daughter cell should reach the same average concentration once the daughter has grown to maturity, since no asymmetric mechanism is present. Figure 4.7 confirms this expectation over two generations for $r = 0.8$: the concentration fields within each cell are spatially uniform and symmetric across the lineage by the end of each simulation. This serves as a validation of both the growth rate selection and the inter-generational initialization procedure.

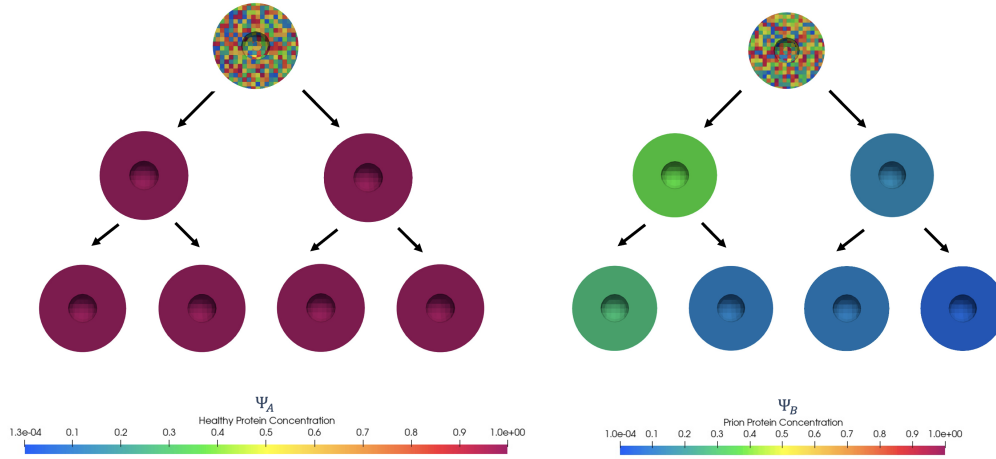


Figure 4.7: Two-generation lineage tree for the diffusion–growth system ($\gamma_{AB} = \gamma_{BA} = 0$, $r = 0.8$). As expected in the absence of aggregation, mother and daughter cells reach identical average concentrations once the daughter has grown to maturity.

4.5.3 Reaction-Diffusion System with a Nonlinear Source

With aggregation reinstated, the full reaction–diffusion–logistic system produces a qualitatively different picture from the reaction-free case. Figure 4.8 shows a three-generation lineage tree using the diffusion coefficients fitted to the FLIP data in Section 4.4.2, which are held fixed throughout all subsequent simulations. A clear asymmetry emerges immediately: the mother cell at each division preferentially retains aggregate, while daughter cells receive a diminished share. By the second generation this effect is already pronounced: mother-lineage cells retain roughly twice the aggregate concentration of their daughter-lineage counterparts, and the disparity deepens with each successive division. By the third generation, the rightmost daughter cells carry very little aggregate burden relative to the ancestral gen 0 mother.

This behavior is a direct consequence of the low, similar diffusivities of ψ_A and ψ_B under the FLIP-fitted parameters: because neither species redistributes rapidly within a cell, the aggregate that is partitioned to the mother at division remains concentrated there, with little opportunity for equilibration before the next division event. The result is a lineage in which replicative age and aggregate burden are tightly coupled from the very first generation.

These simulations establish the key qualitative feature of the model and motivate the more systematic investigation that follows. However, constructing each generation manually (*i.e.* running a simulation, extracting averaged concentrations, and initializing the next by hand) limits the tractable tree depth to two or three generations and precludes any structured exploration of the reaction rate parameters γ_{AB} , γ_{BA} , and the growth rate r . To address this, we developed a parallelized lineage construction framework that automates the inter-generational handoff and distributes simulations across an HPC cluster, enabling full binary lineage trees to generation four and beyond across a structured parameter sweep.

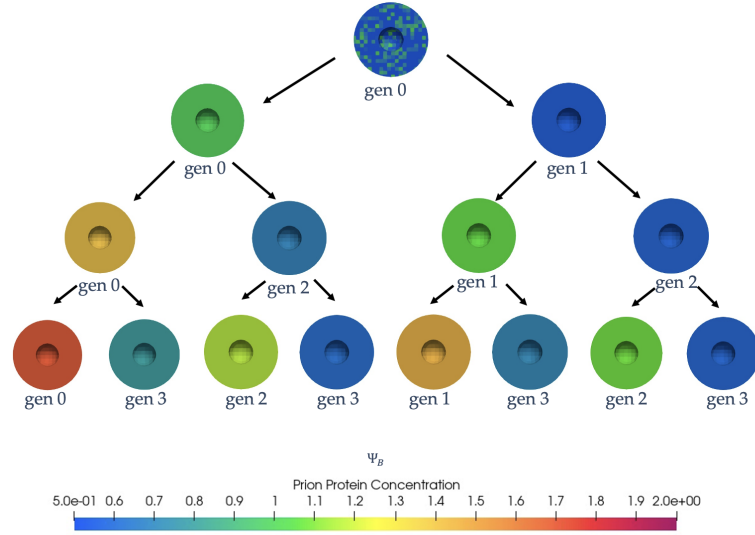


Figure 4.8: Three-generation lineage tree for the reaction–diffusion–logistic system. Diffusion coefficients are taken from the FLIP data fit in Section 4.4.2.

4.6 Lineage Construction and HPC Implementation

4.6.1 Binary Lineage Representation

Each cell division produces two daughter cells: a mother cell, which continues dividing, and a newly born daughter cell, which itself matures and divides in subsequent generations. This process naturally gives rise to a binary tree structure, in

which each node represents a single cell and each edge represents a division event. We index cells using a binary naming convention: the root cell (generation 0) is assigned the label 0, and at each division, the mother and daughter cells inherit labels $c0$ and $c1$ respectively, where c denotes the parent label. The generation number of a cell is defined by the position of the last 1 in its binary label: a cell whose label consists entirely of zeros (e.g., 00000) is a generation-0 cell, while a cell with label 0101 belongs to generation 3, and 00001 belongs to generation 4. This convention reflects the replicative age of the cell: each appended 1 marks a division event in which the cell was born as a new daughter.

For a lineage tree of depth G , the total number of simulations required is

$$N_{\text{sims}} = 2^{G+1} - 1, \quad (4.21)$$

growing exponentially with generation depth. At generation $G = 5$, for example, this corresponds to 63 independent simulations, each requiring a full three-dimensional PDE solve over 190 minutes of simulated time. Sequential execution of a lineage tree of this size is computationally prohibitive, motivating the parallelized framework described below.

4.6.2 Parallelization Strategy

The binary tree structure of the lineage admits a natural parallelization strategy: since each cell's simulation depends only on the final concentration fields of its parent, sibling cells and cells across different branches of the tree are mutually independent and can be simulated concurrently. We exploit this structure by submitting each simulation as an independent job on a high-performance computing (HPC) cluster, with job dependencies configured so that a child job is submitted automatically upon successful completion of its parent.

Concretely, upon completion of the simulation for cell c , the spatially averaged concentrations $\bar{\psi}_A$ and $\bar{\psi}_B$ are extracted from its final concentration fields and written to disk. Two child jobs — one for the mother cell and one for the daughter cell — are then submitted to the cluster queue, each initialized with the appropriate inherited concentrations. Beyond this parent-child dependency, no ordering is imposed on the execution of jobs across the tree. As a result, the progression of simulations through the lineage is determined by cluster node availability: different branches may advance at different rates, and simulations from different generation levels may run concurrently. This asynchronous execution is valid precisely because no two nodes in the tree share a dependency, except through their common ancestor.

The practical effect is that wall-clock time is governed by the depth of the longest completed branch and the availability of cluster resources, rather than by the total number of simulations. This represents a substantial reduction in runtime compared to sequential execution of the full lineage tree.

4.7 Lineage Simulation Results

4.7.1 Full Lineage Trajectories

Figure 4.9 shows the mean aggregate concentration ψ_B as a function of time for every cell in the simulated lineage, across the parameter sweep $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with growth rate $r = 0.8$ held fixed throughout. Each curve corresponds to a single simulated cell, colored by generation index, with earlier generations shown in darker purple and later generations in progressively lighter yellow-green tones. Time runs continuously from each cell's initialization through the end of its 190-minute simulation window.

Within each individual trajectory, three phases are visible. During the pre-division growth phase, ψ_B increases as dimerization converts monomer into aggregate faster than division dilutes it. At the moment of cytokinesis, each cell inherits a mean concentration derived from its parent, producing a sharp downward step in the daughter trajectory while the mother retains the majority of the burden. Following division, both cells enter a recovery phase in which ψ_B climbs again to a local quasi-steady state before the next division event. This growth–drop–recovery pattern repeats at every generation and is qualitatively consistent across all three degradation regimes.

The more striking feature of these graphs is the systematic separation of the trajectories by lineage history. Rather than randomly overlapping, the curves are ordered by replicative age: older-lineage cells sit persistently higher in ψ_B at every comparable time point, while cells born as daughters in later generations begin lower and remain lower throughout their simulation window, even after the post-division recovery. This ordering reflects the cumulative effect of asymmetric inheritance — each division preferentially retains aggregate in the mother, so cells with longer replicative histories carry a compounding burden that no single division event can erase. Aggregate burden and replicative age are therefore closely correlated across the lineage.

Increasing γ_{BA} shifts the overall level of ψ_B downward across all rows, as stronger degradation converts aggregate back to monomer more efficiently, and the spread between the oldest and youngest trajectories narrows. Increasing γ_{AB} shifts the overall level upward across all columns, as stronger dimerization drives more monomer into aggregate form. In both cases the qualitative structure — growth–drop–recovery within each cell and monotone generational ordering across the lineage — is preserved. This confirms that lineage memory is a robust feature of the model governed primarily by the asymmetry of division, and that the biochemical parameters modulate the overall burden level without disrupting the inheritance dynamics.

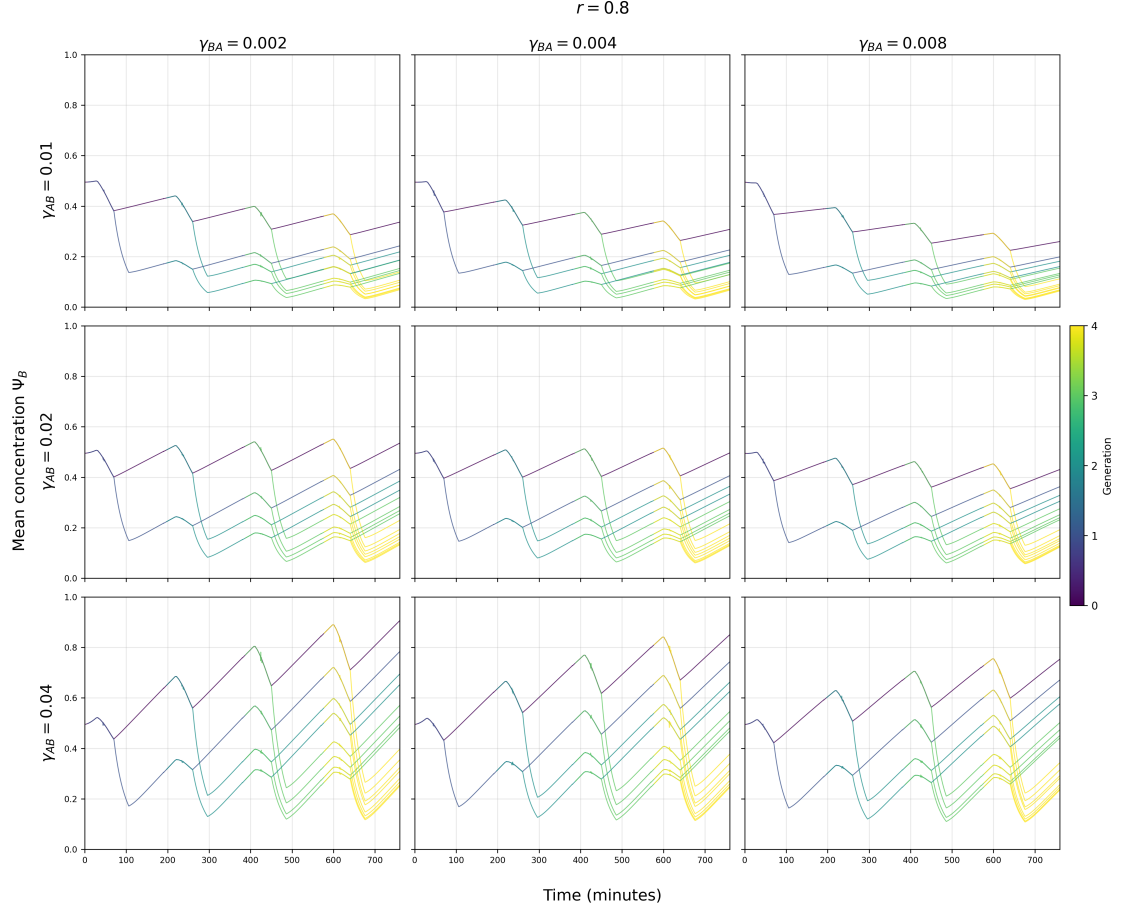


Figure 4.9: Mean aggregate concentration ψ_B over time for all cells in a five-generation lineage tree, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.8$ fixed. Each curve corresponds to a single simulated cell; color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). Sharp downward steps mark division events, after which both mother and daughter enter a recovery phase. Trajectories are ordered by replicative age across all parameter combinations, with older-lineage cells carrying persistently higher aggregate burden. Corresponding figures for $r = 0.5$ and $r = 0.2$ are provided in Appendix A.

4.7.2 Aggregate Mass Fraction

To complement the trajectory analysis of Section 4.7.1, we examine the aggregate mass fraction

$$F_B(t) = \frac{2\psi_B}{\psi_A + 2\psi_B}, \quad (4.22)$$

which measures the proportion of total protein mass committed to the aggregate state at each point in time. Figure 4.10 shows $F_B(t)$ for all cells in the five-generation lineage under the same three degradation regimes as in Section 4.7.1.

The overall structure mirrors that of the raw trajectories: curves are ordered by replicative age, with older-lineage cells maintaining a persistently higher aggregate mass fraction while later-generation cells are tracking progressively lower. Sharp downward steps at each division event reflect the asymmetric partitioning of aggregate to the mother, consistent with the behavior observed in ψ_B directly. The small but sharp upward spikes visible at division are a geometric artifact of cytokinesis rather than a physical increase in aggregate: at the moment a daughter cell detaches, she is small relative to her eventual mature size, so the aggregate she carries represents a transiently elevated fraction of her total protein mass. As the daughter grows and acquires additional monomer through logistic production, F_B falls back toward its pre-division trajectory. This pattern is consistent across all nine illustrations.

The effect of γ_{BA} is visible across the columns of Figure 4.10: as degradation rate increases, the long-term F_B values shift downward across all cells and all generations, reflecting the progressive conversion of aggregate back to monomer. The effect of γ_{AB} is visible across the rows: higher dimerization raises the overall level of F_B throughout the lineage, as stronger aggregation drives a larger fraction of total protein into the aggregate state. In both cases the generational hierarchy is preserved (*i.e.*, older cells consistently maintain a higher aggregate fraction than younger ones at every comparable time point), confirming that neither parameter disrupts the lineage memory effect. Rather, γ_{BA} and γ_{AB} together control the overall level of aggregate burden, while the geometry of division determines its generational structure.

4.7.3 Generation-Level Statistics

To summarize the distribution of aggregate burden across the lineage more compactly, Figures 4.11, 4.12, and 4.13 shows box plots of the final mean aggregate concentration ψ_B grouped by generation index, across a structured sweep of both the dimerization rate $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and the degradation rate $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r \in \{0.8, 0.5, 0.2\}$ (figures) held fixed throughout. Each panel summarizes the distribution of final ψ_B values across all cells belonging to a given generation; generation 0 contains only the single root cell and therefore appears as a point rather than a box.

The most prominent feature across all panels is the monotone decrease in median ψ_B with generation index. Cells at generation 0 consistently carry the highest

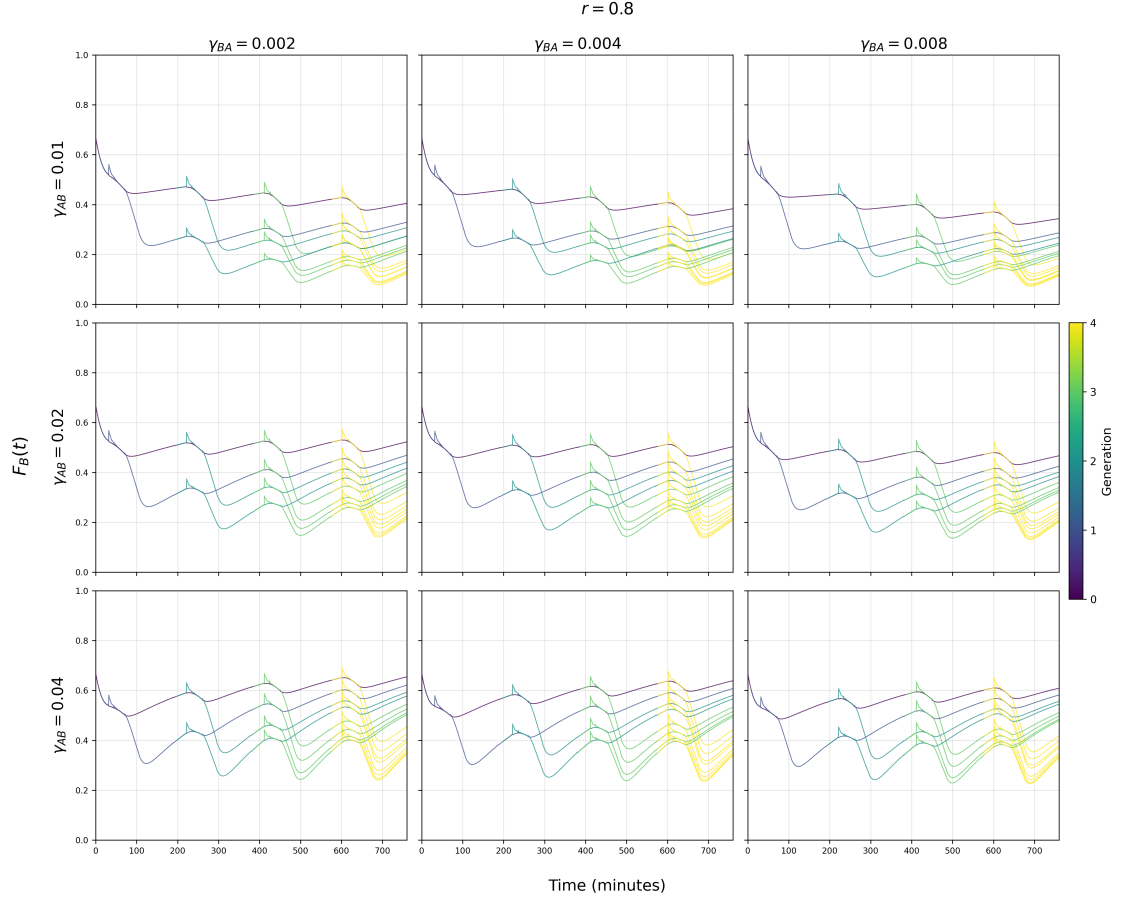


Figure 4.10: Aggregate mass fraction $F_B(t) = 2\psi_B/(\psi_A + 2\psi_B)$ over time for all cells in a five-generation lineage, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.8$ fixed. Color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). At each division event ($t = 70$), the newly detached daughter cell exhibits a transient upward spike in F_B reflecting her small initial volume; as she grows and acquires additional monomer the fraction falls to its post-division level. The mother experiences a modest dilution at division but remains at a substantially higher F_B than the daughter, reflecting the asymmetric partitioning of aggregate at cytokinesis. Increasing γ_{BA} shifts the long-term balance from aggregate retention toward dilution across all rows, while increasing γ_{AB} raises the overall level of F_B across all columns. Generational ordering persists across all nine parameter combinations. Corresponding figures for $r = 0.5$ and $r = 0.2$ are provided in Appendix A.

aggregate burden, and the median falls at each successive generation, reaching its lowest values at generation 4. This pattern directly reflects the cumulative effect of asymmetric inheritance: each division preferentially retains aggregate in the mother-lineage cell, so cells with shorter replicative histories begin each generation with less inherited burden and accumulate less over their simulation window. The generational ordering is preserved across every parameter combination shown, confirming that lineage memory is a robust structural feature of the model rather than an artifact of any particular parameter choice.

The effect of varying γ_{BA} is visible across the columns within each figure: increasing degradation shifts the median downward at every generation and produces a modest reduction in variance, but leaves the shape of the generational separation intact. The effect of varying γ_{AB} is visible across the rows: higher dimerization shifts the entire distribution upward, as more monomer is converted to aggregate throughout the lineage, but the monotone generational ordering is preserved. The effect of varying r is visible across the three figures: at low growth rate ($r = 0.2$, Figure 4.13) the values of ψ_B are substantially lower than at high growth rate ($r = 0.8$, Figure 4.11), consistent with the heatmap results of Section 4.7.5, but the monotone decrease with generation index and the generational separation are preserved across all three growth rates. Together these observations confirm that γ_{BA} controls the overall level of aggregate burden, γ_{AB} controls the rate at which that burden accumulates, and r controls the scale of protein production, while none of these parameters disrupts the inheritance structure imposed by the geometry of division.

Notably, the effect of γ_{BA} on the final aggregate distributions is substantially weaker than that of either γ_{AB} or r . Across columns — as degradation increases from 0.002 to 0.008 — the median concentrations shift only modestly, whereas moving across rows or across figures produces comparatively dramatic changes. This observation suggests a clear parameter hierarchy: dimerization and synthesis are the dominant levers governing aggregate burden across the lineage, while degradation plays a secondary modulatory role. This hierarchy is consistent with findings by Christiano et al. [116], who demonstrated through proteome-wide turnover measurements in *S. cerevisiae* that the abundance of stable proteins is governed primarily by synthesis and dilution rather than active degradation. Our results extend this principle from steady-state protein abundance to lineage-level aggregate fate — degradation is a weak lever not only on protein levels but on the generational structure of aggregate inheritance.

A secondary feature worth noting is the widening of the interquartile range in later generations. At generation 4, the spread of ψ_B values across cells is larger than in earlier generations, reflecting the divergence of replicative histories within the lineage tree: by generation 4, some cells belong to heavily mother-biased branches that have accumulated aggregate across many divisions, while others belong to daughter-biased branches that have been repeatedly initialized with lower inherited concentrations. This within-generation variance is a direct attribute of lineage memory: cells at the same generation can carry substantially different aggregate burdens depending on their

replicative history, reflecting the compounding effect of asymmetric inheritance across successive divisions.

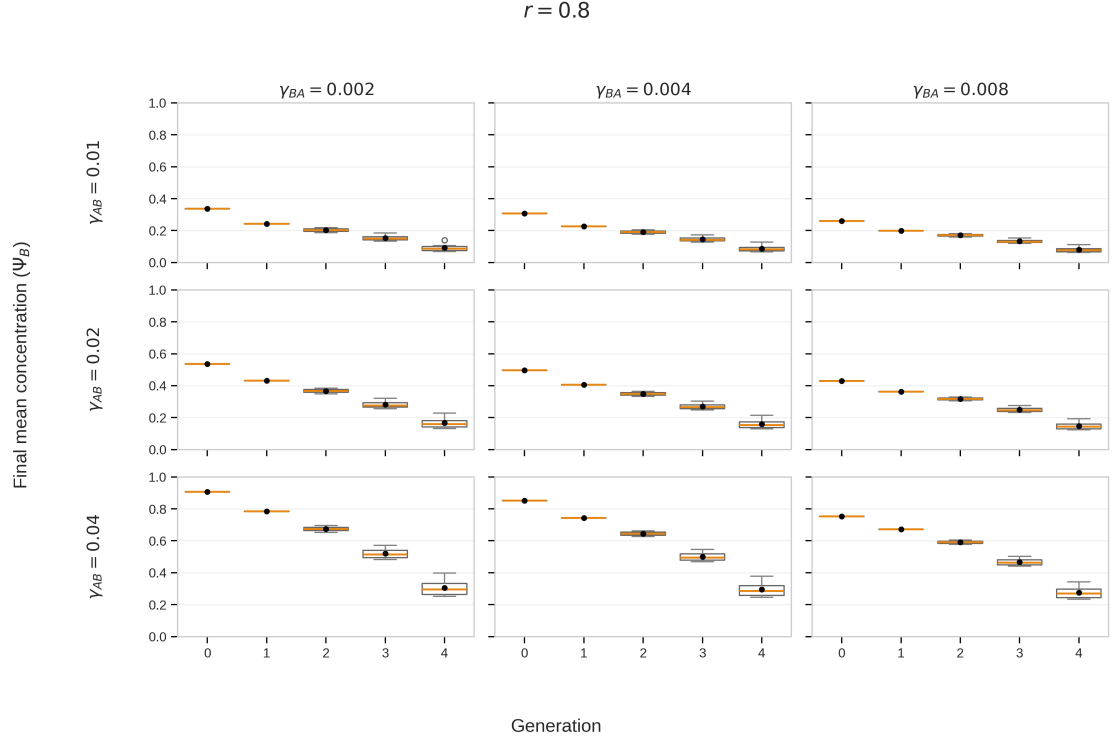


Figure 4.11: Box plots of final mean aggregate concentration ψ_B grouped by generation index, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.8$ fixed. Orange lines indicate the median; filled circles indicate the mean; open circles indicate outliers. Generation 0 contains only the single root cell and appears as a point. Median ψ_B decreases monotonically with generation index across all parameter combinations, confirming that lineage memory is robust to variation in both dimerization and degradation rates. Increasing γ_{BA} lowers overall burden while increasing γ_{AB} raises it; neither parameter disrupts the generational ordering.

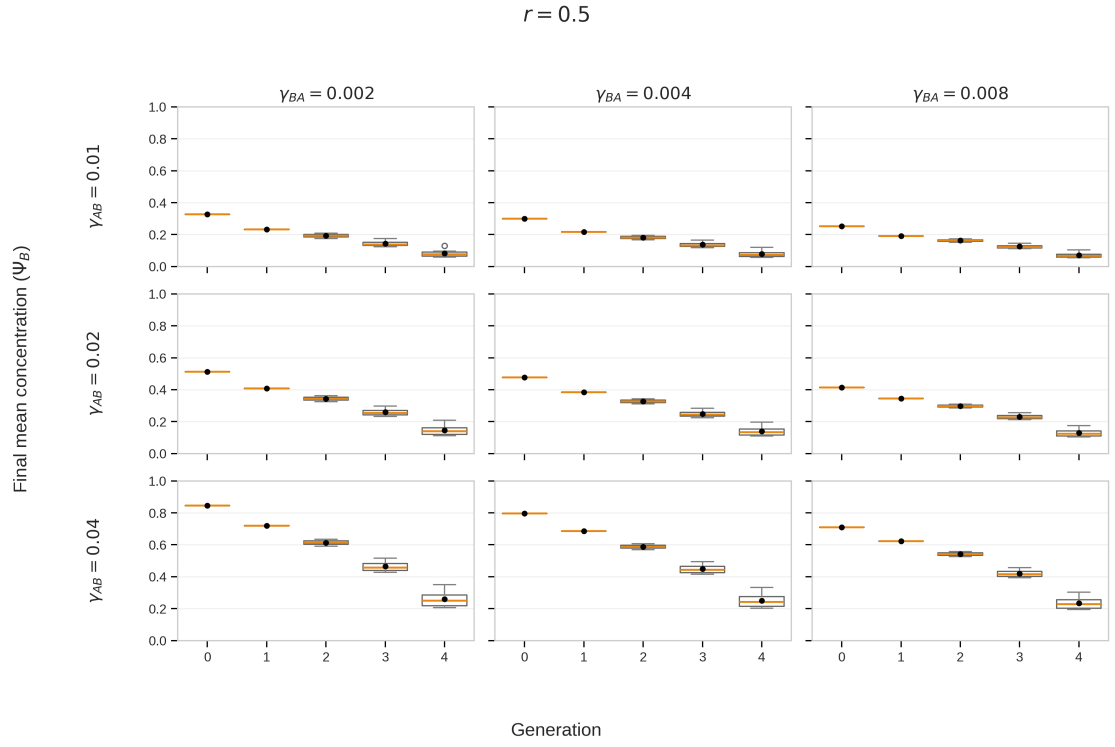


Figure 4.12: Box plots of final mean aggregate concentration ψ_B grouped by generation index, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.5$ fixed.

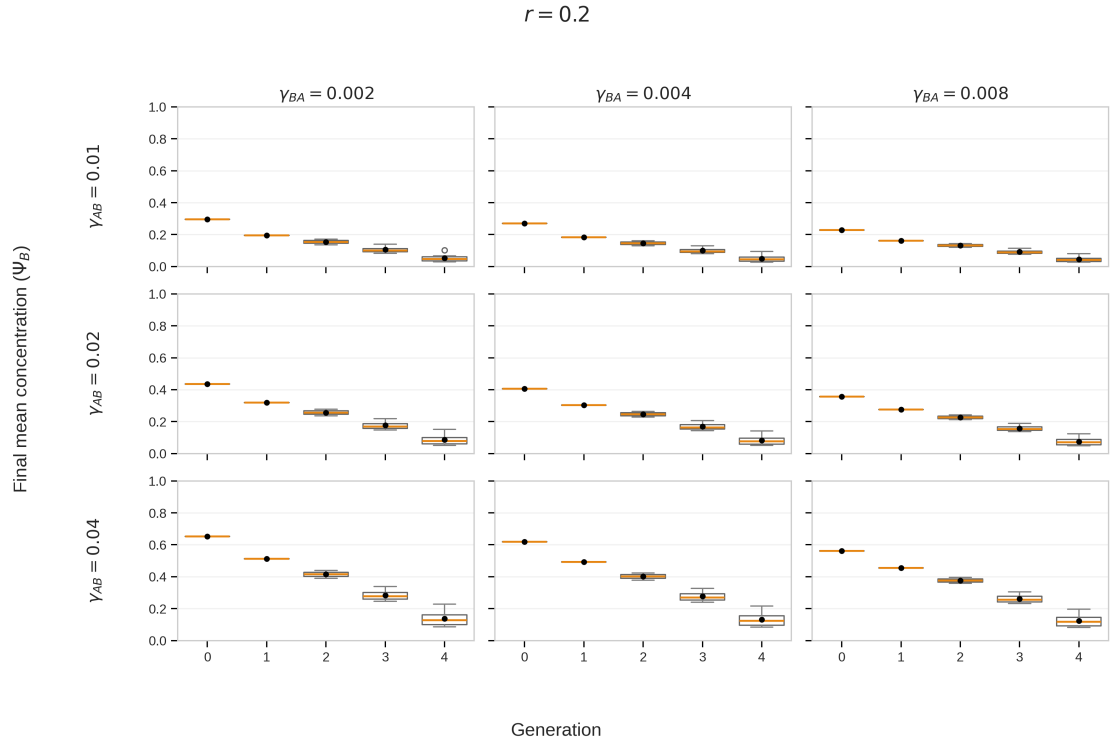


Figure 4.13: Box plots of final mean aggregate concentration ψ_B grouped by generation index, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.2$ fixed.

4.7.4 Geometric Decay of Aggregate Burden Across Generations

The trajectory and box plot results of Sections 4.7.1-4.7.3 establish that median ψ_B decreases monotonically with generation index across all 27 parameter combinations. A natural question is whether this decrease follows a quantifiable pattern — one that could be used to predict aggregate burden at generations beyond those simulated. To address this we fit a geometric decay model to the final-time concentration values across the lineage.

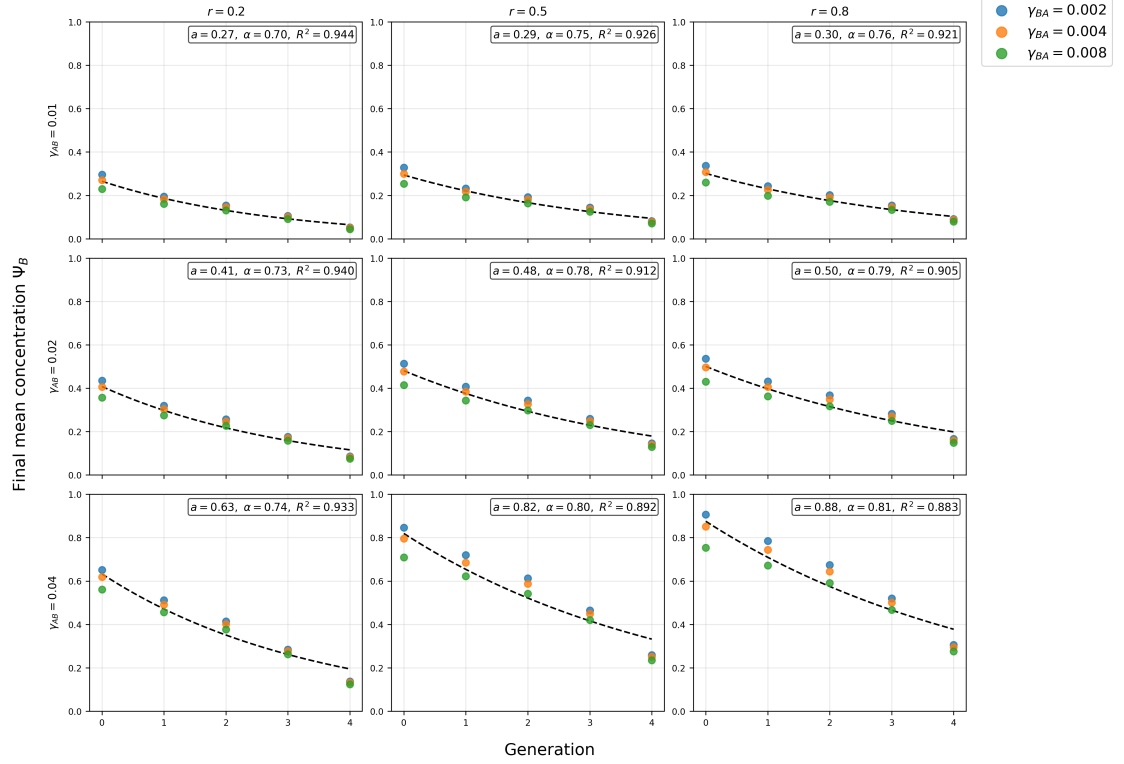


Figure 4.14: Geometric decay model $\psi_B(g) = a \cdot \alpha^g$ fitted to final-time aggregate concentrations across all 27 parameter combinations, organized as a 3×3 grid with γ_{AB} varying across rows and r varying across columns. Colored scatter points show generation means for each γ_{BA} value. Dashed lines show the fitted decay curve with annotated parameters a , α , and R^2 .

We propose the model

$$\psi_B(g) = a \cdot \alpha^g, \quad (4.23)$$

where $a > 0$ is the predicted aggregate burden at generation 0 and $\alpha \in (0, 1)$ is the per-generation retention factor — the fraction of aggregate burden carried over from one generation to the next on average. The parameters a and α are determined by minimizing

the sum of squared residuals:

$$\min_{a, \alpha} \sum_{i=1}^N \left(\psi_B^{(i)} - a \cdot \alpha^{g_i} \right)^2 \quad (4.24)$$

over the generation means pooled across all three γ_{BA} values for each fixed (γ_{AB}, r) combination, where $N = 15$ (5 generations $\times$ 3 γ_{BA} values) and g_i denotes the generation index of the i -th data point.

Figure 4.14 shows the fitted geometric decay curves alongside the generation means for all 27 parameter combinations, organized as a 3×3 grid with γ_{AB} varying across rows and r varying across columns. The γ_{BA} conditions are shown as colored scatter points within each panel.

Two observations stand out. First, the scatter points cluster tightly within each panel regardless of γ_{BA} , confirming that degradation rate has relatively little effect on the generational ordering of aggregate burden — consistent with the box plot results of Section 4.7.3. Second, the geometric decay model fits the data well across all nine panels, with R^2 values consistently above 0.88, indicating that the per-generation dilution of aggregate burden follows a clean multiplicative pattern.

The fitted retention factor α ranges from approximately 0.70 to 0.81 across all parameter combinations, with higher values of γ_{AB} and r associated with larger α , reflecting the fact that stronger dimerization and faster monomer production both partially offset the dilution of aggregate at each division. Importantly, $\alpha < 1$ in all cases, confirming that aggregate burden decreases with each successive generation under all parameter combinations investigated.

The geometric decay model enables a direct prediction of the generation at which aggregate burden in the youngest generations drop below any prescribed threshold ϵ . Setting $\psi_B(g^*) = \epsilon$ and solving for g^* gives:

$$g^* = \frac{\log(\epsilon/a)}{\log(\alpha)} \quad (4.25)$$

For example, taking $\epsilon = 0.1 \cdot a$ — the generation at which the youngest-daughter cells carry 10% of the founder's end-time burden — and using the fitted parameters for $\gamma_{AB} = 0.01$, $\gamma_{BA} = 0.008$, $r = 0.2$ ($a \approx 0.27$, $\alpha \approx 0.70$) — the case for which the model predicts the most rapid dilution — gives $g^* \approx 6.5$ generations to reach 10% of the founder's burden. To validate this prediction beyond the five-generation parameter sweep, we ran a deeper simulation to generation depth $G = 7$.

Figure 4.15 shows the full trajectory of all cells in the eight-generation lineage tree. The founder-line cell maintains a persistently high aggregate burden across all eight generations, while the youngest-daughter trajectories converge toward near-zero by generations 6–7 — consistent with the prediction of $g^* \approx 6.5$. The deeper simulation reveals that by generation 7 the youngest-daughter trajectories carry approximately 13% of the founder's burden: within 3 percentage points of the model prediction.

The slight overestimate — 10% predicted versus 13% observed — is expected given that the geometric decay model is fitted to generation means pooled across all

γ_{BA} values, whereas the deeper simulation uses a single fixed parameter combination. The agreement is nonetheless remarkably close given the simplicity of the two-parameter model. Together these results demonstrate that the geometric decay model provides a compact, interpretable, and predictive summary of aggregate dilution across the lineage — capturing the essential structure of lineage memory in two parameters whose biological interpretation is transparent and whose values are robust across the full parameter sweep.

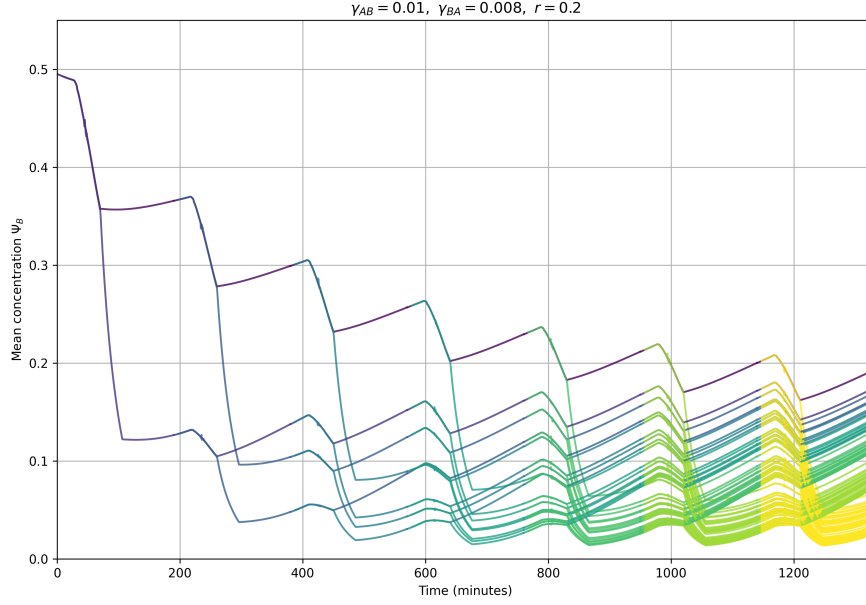


Figure 4.15: Full lineage trajectories for $\gamma_{AB} = 0.01$, $\gamma_{BA} = 0.008$, $r = 0.2$ extended to generation depth $G = 8$. Color encodes generation index from earliest (purple) to latest (yellow-green). The founder-line cell maintains persistently high aggregate burden while the youngest-daughter trajectories converge toward near-zero by generations 6–7, consistent with the geometric decay model prediction of $g^* \approx 6.5$.

4.7.5 Lineage Dilution Across the Parameter Space

The results of Sections 4.7.1–4.7.4 establish that asymmetric inheritance and generational stratification are robust features of the model under a fixed parameter set. A natural follow-up question is how the degree of stratification depends on the biochemical parameters γ_{AB} , γ_{BA} , and r . In particular, we ask: across our full parameter space, how much does the aggregate burden of the youngest-generation cells diverge from that of the founder-line cell?

To quantify this, we introduce the dilution ratio $\mathcal{R}$, defined as the mean aggregate concentration of the maximally young cells at the final generation relative to that of the

founder-line cell:

$$\mathcal{R} = \frac{1}{|\mathcal{L}_G|} \frac{\sum_{c \in \mathcal{L}_G} \psi_B^{(c)}(T)}{\psi_B^{(0)}(T)}, \quad (4.26)$$

where G denotes the maximum generation depth of the lineage tree, $\mathcal{L}_G$ denotes the set of cells at generation G whose replicative age is maximal (*i.e.* whose binary label ends in 1), $|\mathcal{L}_G|$ is the number of such cells, $\psi_B^{(c)}(T)$ is the mean aggregate concentration in cell c at the end of its simulation window, and $\psi_B^{(0)}(T)$ is the corresponding quantity for the founder-line cell 0, which has been the mother-lineage cell at every division event. Values of $\mathcal{R}$ close to 1 indicate little stratification across the lineage, while values close to 0 indicate strong dilution in the youngest-generation cells relative to the founder.

This metric is useful for two reasons. First, it captures the full extent of lineage memory in a single scalar: a value of $\mathcal{R}$ close to 1 indicates that the youngest generation of cells carry nearly as much aggregate as the founder, meaning parameters wash out the asymmetry produced by division geometry; a value close to 0 indicates that the youngest cells are almost completely diluted relative to the founder, meaning lineage memory is strong and the geometric asymmetry compounds dramatically across generations. Second, by averaging over all cells in $\mathcal{L}_G$ rather than selecting a single cell, $\mathcal{R}$ is robust to the stochastic variation in initialization noise that affects individual simulations. Together these properties make $\mathcal{R}$ a compact and interpretable summary of how biochemical parameters modulate the strength of lineage memory across the full 27-case parameter sweep.

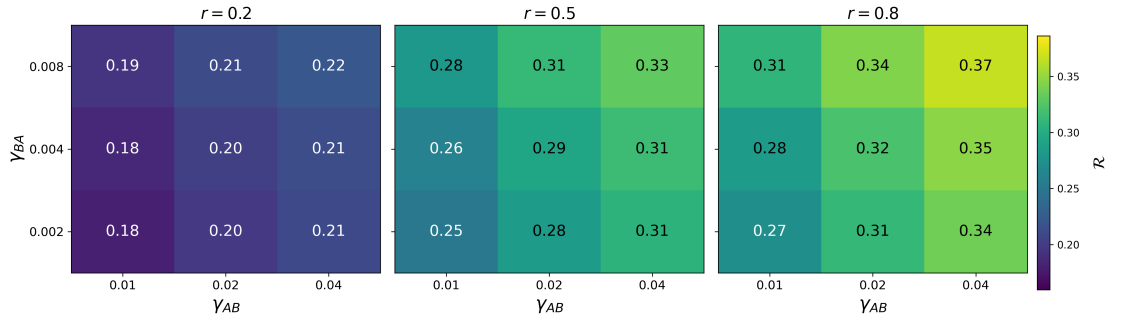


Figure 4.16: Heatmap of the dilution ratio $\mathcal{R}$ across the full parameter sweep, for $r \in \{0.2, 0.5, 0.8\}$ (panels, left to right), $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (x-axis), and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (y-axis). Values of $\mathcal{R}$ close to 1 indicate little divergence across the lineage; values close to 0 indicate strong dilution in the youngest-generation cells relative to the founder.

Figure 4.16 shows $\mathcal{R}$ across all 27 parameter combinations. The most immediate observation is that $\mathcal{R}$ remains well below 1 across the entire parameter space, with a

maximum value of 0.37, confirming that the final-generation cells never approach the aggregate burden of the founder-line cell under any parameter combination investigated. The geometric asymmetry of division produces robust and persistent lineage memory regardless of the biochemical regime.

Within this overall picture, $\mathcal{R}$ increases with all three parameters, though for distinct reasons. The increase with r reflects the role of protein production during cell growth: at low growth rate ($r = 0.2$) values range from 0.18 to 0.22, while at high growth rate ($r = 0.8$) they range from 0.27 to 0.37, suggesting that higher monomer production partially offsets the dilution of aggregate in final-generation cells, narrowing the gap relative to the founder. The increase with γ_{AB} reflects higher overall aggregate accumulation: stronger dimerization drives more monomer into aggregate form across all cells, and the final-generation cells accumulate more burden relative to the founder.

The increase with $\mathcal{R}$ with γ_{BA} is more subtle. Degradation reduces aggregate burden in all cells, but its absolute effect scales with the concentration present — cells carrying more aggregate lose more to degradation. Since the founder consistently carries the highest concentration in the tree, it experiences the largest absolute reduction, while the final-generation cells, carrying less aggregate to begin with, are reduced by a smaller absolute amount. The net effect is that the gap between the founder and the final-generation cells narrows as γ_{BA} increases, producing a higher $\mathcal{R}$. This should not be interpreted as degradation weakening lineage memory — the generational ordering is preserved, as shown in the preceding sections — but rather as a compression of the overall range of aggregate values across the lineage.

4.8 Discussion

This chapter developed a spatially resolved reaction–diffusion framework for simulating prion aggregate dynamics across explicit multi-generation yeast cell lineages. By extending the single-division framework of Heydari *et al.* [6] to a full binary lineage tree and automating the inter-generational handoff via a parallelized HPC pipeline, we produced — to our knowledge — the first spatially resolved, multi-generation computational study of prion aggregate inheritance in budding yeast. The results consistently demonstrate that asymmetric aggregate inheritance is a robust, geometrically driven phenomenon that persists across a wide range of biochemical parameters.

Biological Interpretation of Key Findings

The most prominent finding across all simulations is the systematic stratification of aggregate burden by replicative age. Cells with longer replicative histories carry persistently higher aggregate concentrations than cells born as daughters in later generations, and this ordering is maintained across all parameter regimes investigated. Each division event preferentially retains aggregate in the mother-lineage cell, and because the partitioning is asymmetric at every generation, the burden compounds over time in the aging lineage while daughter lineage cells are repeatedly diluted. The model therefore recapitulates the well-established link between replicative aging and protein aggregate accumulation in yeast, and provides a spatially grounded mechanistic account of how that link arises from diffusion and geometry rather than from any actively regulated transport process. Importantly, this asymmetry is not encoded as a modeling assumption — it emerges from the same physical mechanism identified experimentally by Zhou *et al.* [5]: slow diffusion through the narrow bud neck geometry is sufficient to produce the observed retention bias without invoking active transport.

The dilution ratio $\mathcal{R}$, defined as the mean final concentration of the youngest-generation cells relative to the founder, satisfies $\mathcal{R} \leq 0.37$ across all 27 parameter combinations tested. This confirms that the youngest lineage cells never approach the aggregate burden of the founder under any biochemical regime investigated. Furthermore, since $\mathcal{R}$ is computed at split 4 and the geometric decay model predicts further reduction at deeper generations, this result is conservative — the true degree of stratification at greater lineage depths is likely stronger than reported here.

Parameter Sensitivity

The structured parameter sweep across dimerization rate γ_{AB} and degradation rate γ_{BA} reveals a clear separation of roles. Increasing γ_{BA} lowers the overall level of aggregate burden throughout the lineage, compressing trajectories downward and

shifting the long-term mass fraction toward lower values, but leaves the generational ordering and the shape of the organization intact. Increasing γ_{AB} shifts the entire distribution upward, as faster dimerization converts more monomer into aggregate throughout the lineage, but again without disrupting the ordering. Together these observations indicate that the inheritance structure — the monotone stratification of aggregate burden by replicative age — is determined primarily by the geometry of division and the diffusion coefficients, while the biochemical parameters control the overall scale of aggregate accumulation within that structure.

Notably, the effect of γ_{BA} on the final aggregate distributions is substantially weaker than that of either γ_{AB} or r . Across columns of the parameter sweep — as degradation triples from 0.002 to 0.008 — the median concentrations shift only modestly, whereas varying γ_{AB} or r produces comparatively dramatic changes. This observation suggests a clear parameter hierarchy: dimerization and synthesis are the dominant levers governing aggregate burden across the lineage, while degradation plays a secondary modulatory role. This hierarchy is consistent with findings by Christiano et al. [116], who demonstrated through proteome-wide turnover measurements in *S. cerevisiae* that stable protein levels are governed primarily by synthesis and dilution rather than active degradation. Our results extend this principle from steady-state protein abundance to lineage-level aggregate fate.

Geometric Decay of Aggregate Burden Across Generations

A compact summary of the generational decay pattern is provided by the geometric decay model $\psi_B(g) = a \cdot \alpha^g$, fitted to the final-time aggregate concentrations grouped by generation index across the parameter sweep. The per-generation retention factor $\alpha \in [0.70, 0.81]$ is remarkably stable across all 27 parameter combinations tested, confirming that the rate of generational decay is governed primarily by the geometry and diffusion coefficients rather than the biochemical reaction rates. The parameter a , by contrast, varies substantially with γ_{AB} and r , reflecting the dependence of overall aggregate scale on these parameters. Together, a and α provide a predictive description of how aggregate burden distributes across a lineage tree. Setting a threshold of $\epsilon = 0.1 \times a$ and solving for the generation at which the youngest lineage cells reach this threshold yields $g^* = \log(0.1)/\log(\alpha) \approx 6.5$ for the most favorable parameter combination ($\alpha \approx 0.70$). Validation via simulation to split 7 confirms that the youngest cells carry approximately 13% of the founder’s burden at that depth — slightly above the predicted threshold, consistent with ongoing monomer synthesis providing a floor that pure geometric dilution does not capture.

Parameter Calibration

Diffusion coefficients were calibrated by fitting an exponential decay model to FLIP experimental data from Derdowski et al. [113]. We acknowledge that this data was collected in a somewhat different experimental context than the simulations

presented here; nonetheless, the fitted values provide biologically motivated estimates that are grounded in direct measurements of Sup35 mobility in yeast cells. The monomer diffusivity D_A was taken from the psi^- strain and the aggregate diffusivity D_B from the $[PSI+]^{Strong}$ strain, reflecting the expectation that aggregated species diffuse more slowly than soluble monomers. The growth rate $r = 0.8$ was selected as the baseline value on the basis that it produces a stable steady-state mass fraction in the daughter cell by the end of the simulation window, validated by the reaction-free system in which mother and daughter reach identical monomer concentrations at maturity. This choice ensures that the inter-generational handoff captures the daughter cell's true post-division equilibrium concentration rather than a transient mid-growth value. The sensitivity of results to this choice is assessed through the structured parameter sweep, which includes $r \in \{0.2, 0.5, 0.8\}$.

Limitations and Future Directions

Several modeling simplifications merit consideration. Most notably, at each division event the full spatial concentration profile is collapsed to a scalar mean, which is then used to initialize the subsequent generation. While this preserves inter-generational differences in aggregate burden, it assumes that spatial gradients within each cell have equilibrated by the end of the simulation window. This assumption is supported by the concentration fields plotted on the full computational mesh, which show negligible spatial variation within individual cells by $T = 190$ min (Figure 4.8), suggesting that the mean-field handoff does not discard meaningful spatial information in practice. Nonetheless, a natural extension of this work would carry the full three-dimensional concentration field forward across division events, requiring careful interpolation onto the geometry of the nascent daughter cell.

The model also assumes uniform cell cycle lengths across all generations. As noted by Banwarth-Kuhn and Sindi [112], varied cell cycle lengths are an important feature of a complete multi-generation framework, since the growth phase of a newly born daughter cell influences how inherited aggregate is diluted before the next division. The parallelized pipeline developed here is, in principle, extendable to this case, since each cell's simulation time is a configurable parameter, and we leave this as a direction for future work.

More broadly, the reaction–diffusion–logistic model employed here is a deliberate simplification of the full complexity of prion dynamics. The two-species dimerization scheme does not capture aggregate size distributions, fragmentation by chaperones such as Hsp104, or the nucleation and stochastic loss dynamics that govern prion appearance and clearance at low propagon numbers [114]. Extensions that incorporate these processes — for example, coupling the spatial PDE framework to a nucleated polymerization model for the reaction terms — would substantially increase the biological realism of the model. Additional directions include incorporating more realistic cell geometries, modeling the effect of guanidine hydrochloride curing on aggregate clearance across generations, and extending the framework to account for

additional intracellular structures such as the endoplasmic reticulum and mitochondria.

Despite these simplifications, the framework succeeds in its primary aim: demonstrating that spatially resolved, geometrically driven asymmetric inheritance produces robust lineage memory across multiple generations, and providing a computational platform for systematically exploring how biochemical parameters regulate that memory. The combination of spatial resolution, biologically informed parameters, and lineage-scale computation developed here opens a new mode of investigation for yeast prion dynamics that complements and extends existing well-mixed and stochastic approaches.

Conclusion and future work

5.1 Summary of Contributions

This dissertation has developed and applied computational methods for solving partial differential equations on complex, irregular, and continuously deforming geometries. The unifying conviction across all three projects is that spatial structure is not a peripheral concern but is frequently the mechanism through which the underlying physics and biology operate. Each project approaches this conviction from a different angle — one methodological and two applied — and together they demonstrate the breadth and depth of what spatially resolved PDE computation can reveal when brought to bear on problems where geometry matters.

5.1.1 A New Meshless Method for Elliptic PDEs (LSQD)

The first contribution of this dissertation is the Least Squares Discretization (LSQD) method, a meshless framework for solving elliptic PDEs on scattered point distributions with arbitrary geometry and boundary conditions. LSQD requires no quadrature rules, no weak formulation, and no body-fitted mesh — the PDE, boundary conditions, and continuity conditions are enforced pointwise and assembled into an overdetermined linear system whose least squares solution defines the approximation. The method achieves h-P convergence on both uniform and highly adaptive non-graded quadtree grids, with estimated orders of convergence ranging from 1.83 to 4.53 across a broad range of problem configurations. It handles Dirichlet, Neumann, and Robin boundary conditions, remains robust in degenerate cases where traditional weak formulations may be ill-posed, and includes a built-in *a posteriori* error estimator that enables local error assessment without access to an analytical solution.

The principal strength of LSQD lies in its accessibility. The method requires minimal mathematical background to implement, scales with the number of grid points, and produces accurate solutions on complex geometries — including non-convex domains such as the octofoil — without the overhead of mesh generation

or quadrature rule construction. A comparison with the finite volume method demonstrates that LSQD consistently achieves lower errors at higher polynomial orders, making it a compelling alternative when precision is paramount and complex geometries pose meshing challenges.

5.1.2 Electrostatic Dynamics of the SARS-CoV-2 Spike Protein

The second contribution applies spatially resolved PDE computation to characterize how the electrostatic surface potential of the SARS-CoV-2 spike protein evolves as it transitions from its closed to open conformation. Combining Markov state model molecular dynamics trajectories from the Folding@home initiative with a high-resolution numerical solver for the nonlinear Poisson-Boltzmann equation — implemented on adaptive octree grids with a sharp level set representation of the protein surface — we produced dynamical electrostatic potential maps across 71 frames of the opening trajectory.

The results reveal dramatic rearrangements of the electric field during spike opening. Most notably, a negatively charged core is exposed as the spike opens, while the receptor binding domain (RBD) becomes more positively charged and the electric field transitions from a globally attracting configuration to one focused specifically around the exposed binding domain. These findings suggest that the geometric reorganization of the spike protein during opening may actively facilitate selective binding to the negatively charged ACE2 receptor — a conclusion that would not have been accessible without spatially resolving the electrostatic field on the deforming protein surface. This work was published in the *Journal of Computational Physics* in 2021 [4].

5.1.3 Multi-Generation Prion Aggregate Inheritance in Budding Yeast

The third contribution develops the first spatially resolved, multi-generation computational model of prion aggregate inheritance in budding yeast. Building on the finite volume framework of Heydari *et al.* [6], which demonstrated that cell geometry alone produces asymmetric partitioning in a single division event, we extend this approach to simulate full binary lineage trees across multiple generations. The model incorporates diffusion of monomer and aggregate species, reversible dimerization, and logistic growth, with diffusion coefficients calibrated from fluorescence loss in photobleaching (FLIP) experimental data. A parallelized HPC pipeline automates the inter-generational handoff and enables a structured sweep across 27 biochemical parameter combinations.

The simulations demonstrate several principal findings. Aggregate burden is stratified by replicative age across all 27 parameter combinations — the monotone decrease in median ψ_B with generation index is preserved regardless of dimerization rate, degradation rate, or growth rate, confirming that lineage memory is a robust

geometric phenomenon driven primarily by the asymmetry of division rather than by the specific biochemical regime. The deterministic nature of the PDE framework confirms that aggregate partitioning at division is governed entirely by the physics of diffusion and geometry rather than stochastic fluctuations. The per-generation dilution of aggregate burden follows a geometric decay model $\psi_B(g) = a \cdot \alpha^g$, with the retention factor α ranging from approximately 0.70 to 0.81 across all parameter combinations — providing a compact, interpretable, and predictive summary of lineage memory dynamics. A deeper simulation to eight generations under the parameter combination most likely to exhibit bifurcation behavior confirms the geometric decay model’s predictive power: the youngest-daughter trajectories carry approximately 13% of the founder’s burden by generation 7, within 3 percentage points of the model’s prediction of 10%.

5.2 Limitations

Each project carries limitations that motivate future work.

The LSQD method, while accessible and convergent, incurs significant computational cost for large systems — computational times peaked around 10^3 seconds on a single core for the largest test problems. The method has been demonstrated in two dimensions and requires further optimization — including accelerated linear solvers, reduced condition numbers, and systematic investigation of evaluation point placement and basis function selection — before it can be of full practical interest at scale. Extension to three dimensions and time-dependent problems remains an important direction for future work.

The SARS-CoV-2 electrostatics study is limited by the resolution of the molecular dynamics trajectories and the assumptions of the Poisson-Boltzmann framework, which treats the solvent as a continuum dielectric and does not capture ion-specific effects or molecular crowding. The study also considers only the opening trajectory of the spike protein and does not model the full binding event with ACE2.

The prion aggregation model makes several simplifying assumptions. The mean-field handoff between generations — initializing the next generation from the spatially averaged final concentration of the parent — is justified by the observation that concentration fields equilibrate spatially by the end of each simulation window, but introduces an approximation that would benefit from quantitative validation. The model also assumes uniform cell cycle lengths and does not incorporate organelle tethering mechanisms identified in subsequent experimental work [101] as a secondary contributor to asymmetric inheritance. The deterministic PDE framework captures the mean behavior of the system but does not account for the molecular-level stochastic fluctuations that contribute to cell-to-cell variability in aggregate burden observed experimentally [117], nor does it incorporate the fragmentation dynamics of Hsp104 that are thought to play a role in prion curing and propagation.

5.3 Future Directions

The three projects of this dissertation open several natural directions for future work.

For LSQD, the most immediate priority is computational optimization — accelerating the linear solver, reducing the condition number through improved preconditioning, and systematically investigating the impact of adjustable parameters on solution accuracy and efficiency. Extension to three dimensions and to time-dependent PDEs would significantly broaden the method’s applicability and position it as a practical alternative to mesh-based solvers for problems involving complex or deforming geometries.

For the SARS-CoV-2 study, a natural extension would incorporate the ACE2 receptor explicitly into the electrostatic model, enabling direct simulation of the binding event and quantitative prediction of binding affinity as a function of spike conformation. The framework developed here — nonlinear Poisson-Boltzmann on adaptive octree grids with level set geometry tracking — is general enough to be applied to other viral proteins and conformational transitions of biological interest.

For the prion aggregation model, three directions are most immediate. First, replacing the mean-field handoff with a full spatial initialization — passing the complete concentration field rather than its spatial average — would remove the principal approximation in the current framework and enable a more faithful representation of spatial heterogeneity across generations. Second, incorporating stochastic fluctuations into the model — either through a stochastic PDE framework or a hybrid PDE-particle approach — would enable the model to capture prion curing events and cell-to-cell variability that the current deterministic framework cannot reproduce. Third, exploring the bifurcation behavior observed in the eight-generation simulation — identifying the parameter threshold at which youngest-daughter lineages effectively clear their aggregate burden — would connect the computational results directly to the experimentally observed prion sectoring phenotype and provide a mechanistic prediction of the conditions under which prion loss occurs.

More broadly, the mathematical framework developed in Chapter 4 — reaction-diffusion PDEs on deforming geometries with asymmetric inheritance across a binary lineage tree — is general enough to be extended to other aggregating proteins and other cell types. In mammalian cells, where prion-like propagation of tau, alpha-synuclein, and TDP-43 aggregates is increasingly recognized as a driver of neurodegeneration, a spatially resolved model of aggregate inheritance across cell divisions could provide mechanistic insight into disease progression that well-mixed models cannot capture. The connection between the geometric asymmetry of division and the long-term fate of protein aggregates — explored here in the tractable setting of budding yeast — may ultimately prove relevant to understanding how neurodegenerative diseases spread through tissue over time.

5.4 Closing Remarks

The central message of this dissertation is simple: geometry matters. A protein that changes shape as it opens, a cell that partitions its contents asymmetrically as it divides — in both cases, the spatial structure of the problem is inseparable from its scientific content. The numerical methods and computational frameworks developed here are designed to take that structure seriously, to resolve it faithfully, and to ask what it reveals about the biology that would remain hidden in a well-mixed or geometry-free approximation. The answers — a shift in electrostatic focus that may facilitate viral binding, a lineage memory that compounds deterministically across generations — suggest that this approach is not merely technically interesting but scientifically necessary.

Bibliography

- [1] Randall J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. Society for Industrial and Applied Mathematics, Jan. 2007.
- [2] Anna Kucherova et al. *Least Squares Discretizations (LSQD): a robust and versatile meshless paradigm for solving elliptic PDEs*. 2024.
- [3] *Folding @ home*. <http://foldingathome.org>.
- [4] A Kucherova et al. “Computational modeling of protein conformational changes - Application to the opening SARS-CoV-2 spike”. In: *Journal of Computational Physics* 444 (Nov. 2021).
- [5] Chuankai Zhou et al. “Motility and Segregation of Hsp104-Associated Protein Aggregates in Budding Yeast”. In: *Cell* 147.5 (Nov. 2011), pp. 1186–1196.
- [6] A. Ali Heydari, Suzanne S. Sindi, and Maxime Theillard. “Conservative finite volume method on deforming geometries: The case of protein aggregation in dividing yeast cells”. In: *Journal of Computational Physics* 448 (2022), p. 110755.
- [7] R. A. Gingold and J. J. Monaghan. “Smoothed particle hydrodynamics: theory and application to non-spherical stars”. In: *Monthly Notices of the Royal Astronomical Society* 181.3 (Dec. 1977), pp. 375–389.
- [8] L. B. Lucy. “A numerical approach to the testing of the fission hypothesis.” In: 82 (Dec. 1977), pp. 1013–1024.
- [9] S. N. Atluri and T. Zhu. “A new Meshless Local Petrov-Galerkin (MLPG) approach in computational mechanics”. In: *Computational Mechanics* 22.2 (Aug. 1998), pp. 117–127.
- [10] S. N. Atluri and T.-L. Zhu. “The meshless local Petrov-Galerkin (MLPG) approach for solving problems in elasto-statics”. In: *Computational Mechanics* 25.2 (Mar. 2000), pp. 169–179.

- [11] T. Belytschko et al. “Meshless methods: An overview and recent developments”. In: *Computer Methods in Applied Mechanics and Engineering* 139.1 (1996), pp. 3–47.
- [12] T. Belytschko, Y. Y. Lu, and L. Gu. “Element-free Galerkin methods”. In: *International Journal for Numerical Methods in Engineering* 37.2 (1994), pp. 229–256.
- [13] E.J. Kansa. “Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations”. In: *Computers Mathematics with Applications* 19.8 (1990), pp. 147–161.
- [14] Wing Kam Liu, Sukky Jun, and Yi Fei Zhang. “Reproducing kernel particle methods”. In: *International Journal for Numerical Methods in Fluids* 20.8-9 (1995), pp. 1081–1106.
- [15] Farrukh Mazhar et al. “On the meshfree particle methods for fluid-structure interaction problems”. In: *Engineering Analysis with Boundary Elements* 124 (2021), pp. 14–40.
- [16] Manish Kumar et al. “A review to explore different meshless methods in various Structural problems”. In: *IOP Conference Series: Materials Science and Engineering* 1116.1 (Apr. 2021), p. 012119.
- [17] Bin He et al. “A leap-frog meshless method with radial basis functions for simulating electromagnetic wave splitter and rotator”. In: *Engineering Analysis with Boundary Elements* 118 (2020), pp. 225–242.
- [18] Grand Joldes et al. “Suite of meshless algorithms for accurate computation of soft tissue deformation for surgical simulation”. In: *Medical Image Analysis* 56 (2019), pp. 152–171.
- [19] Siddhant Kumar, Kostas Danas, and Dennis M. Kochmann. “Enhanced local maximum-entropy approximation for stable meshfree simulations”. In: *Computer Methods in Applied Mechanics and Engineering* 344 (2019), pp. 858–886.
- [20] T. Zhu, J.-D. Zhang, and S. N. Atluri. “A local boundary integral equation (LBIE) method in computational mechanics, and a meshless discretization approach”. In: *Computational Mechanics* 21.3 (Apr. 1998), pp. 223–235.
- [21] T. Belytschko, D. Organ, and C. Gerlach. “Element-free galerkin methods for dynamic fracture in concrete”. In: *Computer Methods in Applied Mechanics and Engineering* 187.3 (2000), pp. 385–399.
- [22] C. Armando Duarte and J. Tinsley Oden. “H-p clouds—an h-p meshless method”. In: *Numerical Methods for Partial Differential Equations* 12.6 (1996), pp. 673–705.

- [23] N. Sukumar, B. Moran, and T. Belytschko. “The natural element method in solid mechanics”. In: *International Journal for Numerical Methods in Engineering* 43.5 (1998), pp. 839–887.
- [24] David González, Elías Cueto, and Manuel Doblaré. *Towards an Isogeometric Meshless Natural Element Method*. Ed. by A. J. M. Ferreira et al. Dordrecht: Springer Netherlands, 2009, pp. 237–257.
- [25] J. Yvonnet et al. “A new extension of the natural element method for non-convex and discontinuous problems: the constrained natural element method (C-NEM)”. In: *International Journal for Numerical Methods in Engineering* 60.8 (2004), pp. 1451–1474.
- [26] Arthur Guittet, Maxime Theillard, and Frédéric Gibou. “A stable projection method for the incompressible Navier-Stokes equations on arbitrary geometries and adaptive Quad/Octrees”. In: *J. Comput. Phys.* 292 (2015), pp. 215–238.
- [27] Qiang Du, Max D. Gunzburger, and Lili Ju. “Voronoi-based finite volume methods, optimal Voronoi meshes, and PDEs on the sphere”. In: *Computer Methods in Applied Mechanics and Engineering* 192.35 (2003), pp. 3933–3957.
- [28] I. Babuska and J. M. Melenk. “THE PARTITION OF UNITY METHOD”. In: *International Journal for Numerical Methods in Engineering* 40.4 (1997), pp. 727–758.
- [29] T. Liszka and J. Orkisz. “The finite difference method at arbitrary irregular grids and its application in applied mechanics”. In: *Computers & Structures* 11.1 (1980). Special Issue-Computational Methods in Nonlinear Mechanics, pp. 83–95.
- [30] E. Onate et al. “A FINITE POINT METHOD IN COMPUTATIONAL MECHANICS. APPLICATIONS TO CONVECTIVE TRANSPORT AND FLUID FLOW”. In: *International Journal for Numerical Methods in Engineering* 39.22 (1996), pp. 3839–3866.
- [31] Richard Franke. “Scattered data interpolation: tests of some methods”. In: *Mathematics of Computation* 38 (1982), pp. 181–200.
- [32] Yong Duan. “A note on the meshless method using radial basis functions”. In: *Computers Mathematics with Applications* 55.1 (2008), pp. 66–75.
- [33] E.J. Kansa and Y.C. Hon. “Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations”. In: *Computers Mathematics with Applications* 39.7 (2000), pp. 123–137.
- [34] John C. Mairhuber. “On Haar’s Theorem Concerning Chebychev Approximation Problems Having Unique Solutions”. In: *Proceedings of the American Mathematical Society* 7.4 (1956), pp. 609–615.

- [35] Philip C. Curtis. “ n -parameter families and best approximation.” In: *Pacific J. Math.* 9.4 (1959), pp. 1013–1027.
- [36] Charles A. Micchelli. “Interpolation of scattered data: Distance matrices and conditionally positive definite functions”. In: *Constructive Approximation* 2.1 (1986), pp. 11–22.
- [37] M. J. Powell. *Five Lectures on Radial Basis Functions*. Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby: Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2005.
- [38] Cécile Piret. “The orthogonal gradients method: A radial basis functions method for solving partial differential equations on arbitrary surfaces”. In: *Journal of Computational Physics* 231.14 (2012), pp. 4662–4675.
- [39] Edward J. Fuselier and Grady B. Wright. “A High-Order Kernel Method for Diffusion and Reaction-Diffusion Equations on Surfaces”. In: *Journal of Scientific Computing* 56 (2013), pp. 535–565.
- [40] G. B. Wright, N. Flyer, and D. A. Yuen. “A hybrid radial basis function–pseudospectral method for thermal convection in a 3-D spherical shell”. In: *Geochemistry, Geophysics, Geosystems* 11.7 (2010).
- [41] B. Sarler. “A Radial Basis Function Collocation Approach in Computational Fluid Dynamics”. In: *Computer Modeling in Engineering & Sciences* 7.2 (2005), pp. 185–194.
- [42] Nam Mai-Duy and Thanh Tran-Cong. “Numerical solution of Navier–Stokes equations using multiquadric radial basis function networks”. In: *International Journal for Numerical Methods in Fluids* 37.1 (2001), pp. 65–86.
- [43] Bozidar Sarler, Janez Perko, and Ching-Shyang Chen. “Radial basis function collocation method solution of natural convection in porous media”. In: *International Journal of Numerical Methods for Heat & Fluid Flow* 14 (Mar. 2004), pp. 187–212.
- [44] I Kovacevic, A Poredos, and Bozidar Sarler. “Solving the Stefan problem with the radial basis function collocation method”. In: *Numerical Heat Transfer: Part B: Fundamentals* 44 (Dec. 2003), pp. 575–599.
- [45] Nam Mai-Duy and Thanh Tran-Cong. “Mesh-free radial basis function network methods with domain decomposition for approximation of functions and numerical solution of Poisson’s equations”. In: *Engineering Analysis with Boundary Elements* 26.2 (2002), pp. 133–156.
- [46] Rolland L. Hardy. “Multiquadric equations of topography and other irregular surfaces”. In: *Journal of Geophysical Research (1896-1977)* 76.8 (1971), pp. 1905–1915.

- [47] B. Sarler and R. Vertnik. “Meshfree explicit local radial basis function collocation method for diffusion problems”. In: *Computers Mathematics with Applications* 51.8 (2006). Radial Basis Functions and Related Multivariate Meshfree Approximation Methods: Theory and Applications, pp. 1269–1282.
- [48] K. C. Chung. “A GENERALIZED FINITE-DIFFERENCE METHOD FOR HEAT TRANSFER PROBLEMS OF IRREGULAR GEOMETRIES”. In: *Numerical Heat Transfer* 4.3 (1981), pp. 345–357.
- [49] Bengt Fornberg and Natasha Flyer. “Solving PDEs with radial basis functions”. In: *Acta Numerica* 24 (2015), pp. 215–258.
- [50] C. Shu et al. “An upwind local RBF-DQ method for simulation of inviscid compressible flows”. In: *Computer Methods in Applied Mechanics and Engineering* 194.18 (2005), pp. 2001–2017.
- [51] Soheil Soleimani et al. “Local RBF-DQ method for two-dimensional transient heat conduction problems”. In: *International Communications in Heat and Mass Transfer* 37.9 (2010), pp. 1411–1418.
- [52] Y.Y. Lu, T. Belytschko, and M. Tabbara. “Element-free Galerkin method for wave propagation and dynamic fracture”. In: *Computer Methods in Applied Mechanics and Engineering* 126.1 (1995), pp. 131–153.
- [53] Vinh Phu Nguyen et al. “Meshless methods: A review and computer implementation aspects”. In: *Mathematics and Computers in Simulation* 79.3 (2008), pp. 763–813.
- [54] Vivek G. Patel and Nikunj V. Rachchh. “Meshless method – Review on recent developments”. In: *Materials Today: Proceedings* 26 (2020). 10th International Conference of Materials Processing and Characterization, pp. 1598–1603.
- [55] Antonio Eirís et al. “MLS-SPH-ALE: A Review of Meshless-FV Methods and a Unifying Formulation for Particle Discretizations”. In: *Archives of Computational Methods in Engineering* 30.8 (2023), pp. 4959–4981.
- [56] Christopher C. Paige and Michael A. Saunders. “LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares”. In: *ACM Trans. Math. Softw.* 8.1 (Mar. 1982), pp. 43–71.
- [57] Michael A. Saunders. “Solution of sparse rectangular systems using LSQR and CRAIG”. In: *BIT Numerical Mathematics* 35.4 (1995), pp. 588–604.
- [58] Bo-Nan Jiang and Louis A. Povinelli. “Least-squares finite element method for fluid dynamics”. In: *Computer Methods in Applied Mechanics and Engineering* 81.1 (1990), pp. 13–37.
- [59] Bo-nan Jiang. “Basis of LSFEM”. In: *The Least-Squares Finite Element Method: Theory and Applications in Computational Fluid Dynamics and Electromagnetics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 47–79.

- [60] K. Hassanzadeh et al. "Considerations around the SARS-CoV-2 Spike Protein with Particular Attention to COVID-19 Brain Infection and Neurological Symptoms". eng. In: *ACS chemical neuroscience* 11.15 (Aug. 2020). PMC7374936[pmcid], pp. 2361–2369.
- [61] J. Lan et al. "Structure of the SARS-CoV-2 spike receptor-binding domain bound to the ACE2 receptor". In: *Nature* 581.7807 (May 2020), pp. 215–220.
- [62] Y. Wang, M. Liu, and J. Gao. "Enhanced receptor binding of SARS-CoV-2 through networks of hydrogen-bonding and hydrophobic interactions". In: *Proceedings of the National Academy of Sciences* 117.25 (2020), pp. 13967–13974.
- [63] D. Wrapp et al. "Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation". In: *Science* 367.6483 (2020), pp. 1260–1263.
- [64] M. Amin, M. K. Sorour, and A. Kasry. "Comparing the Binding Interactions in the Receptor Binding Domains of SARS-CoV-2 and SARS-CoV". In: *The Journal of Physical Chemistry Letters* 11.12 (2020). PMID: 32478523, pp. 4897–4900.
- [65] M. Letko, A. Marzi, and V. Munster. "Functional assessment of cell entry and receptor usage for SARS-CoV-2 and other lineage B betacoronaviruses". In: *Nature Microbiology* 5.4 (2020), pp. 562–569.
- [66] W. Tai et al. "Characterization of the receptor-binding domain (RBD) of 2019 novel coronavirus: implication for development of RBD protein as a viral attachment inhibitor and vaccine". In: *Cellular & Molecular Immunology* 17.6 (June 2020), pp. 613–620.
- [67] M. I. Zimmerman et al. "Citizen Scientists Create an Exascale Computer to Combat COVID-19". In: *bioRxiv* (2020).
- [68] J. Warwicker and H. C. Watson. "Calculation of the electric potential in the active site cleft due to alpha-helix dipoles". In: *Journal of Molecular Biology* 157.4 (1982), pp. 671–679.
- [69] N. A. Baker, D. Bashford, and D. A. Case. "Implicit solvent electrostatics in biomolecular simulation". In: *New algorithms for macromolecular simulation* (2006), pp. 263–295.
- [70] N. A. Baker. "Improving implicit solvent simulations: a Poisson-centric view." In: *Current opinion in structural biology* 15.2 (Apr. 2005), pp. 137–43.
- [71] A. H. Boschitsch and M. O. Fenley. "A Fast and Robust Poisson–Boltzmann Solver Based on Adaptive Cartesian Grids". In: *Journal of Chemical Theory and Computation* 7.5 (2011), pp. 1524–1540.
- [72] J. Chen, C. L. Brooks, and J. Khandogin. "Recent advances in implicit solvent-based methods for biomolecular simulations." In: *Current opinion in structural biology* 18.2 (Apr. 2008), pp. 140–8.

- [73] W. Geng, S. Yu, and G.W. Wei. "Treatment of charge singularities in implicit solvent models". In: *J. Chem. Phys* 127(11) (2007), p. 114106.
- [74] C. Li et al. "Progress in developing Poisson-Boltzmann equation solvers". eng. In: *Molecular based mathematical biology* 1 (Mar. 2013). PMC3816640[pmcid], 10.2478/mlbmb-2013-0002.
- [75] P. Koehl. "Electrostatics calculations : latest methodological advances". In: *Current Opinion in Structural Biology* (2006), pp. 142–151.
- [76] B.Z. Lu et al. "Recent Progress in Numerical Methods for the Poisson Boltzmann Equation in Biophysical Applications". In: *Communications in computational physics* 3 (5) (2008), pp. 973–1009.
- [77] L. E. Felberg et al. "PB-AM: An open-source, fully analytical linear poisson-boltzmann solver". In: *Journal of computational chemistry* 38.15 (June 2017), pp. 1275–1282.
- [78] E. Jurrus et al. "Improvements to the APBS biomolecular solvation software suite". In: *Protein Science* 27.1 (2018), pp. 112–128.
- [79] W. Rocchia, E. Alexov, and B. Honig. "Extending the Applicability of the Nonlinear Poisson-Boltzmann Equation: Multiple Dielectric Constants and Multivalent Ions". In: *The Journal of Physical Chemistry B* 105.28 (July 2001), pp. 6507–6514.
- [80] S. Decherchi et al. "Implicit solvent methods for free energy estimation". eng. In: *European journal of medicinal chemistry* 91 (Feb. 2015). S0223-5234(14)00799-5[PII], pp. 27–42.
- [81] R. C. Harris, T. Mackoy, and M. O. Fenley. "Problems of robustness in Poisson-Boltzmann binding free energies". eng. In: *Journal of chemical theory and computation* 11.2 (Feb. 2015). PMC4610304[pmcid], pp. 705–712.
- [82] D. D. Nguyen, B. Wang, and G.-W. Wei. "Accurate, robust, and reliable calculations of Poisson-Boltzmann binding energies". eng. In: *Journal of computational chemistry* 38.13 (May 2017). PMC5844473[pmcid], pp. 941–948.
- [83] R. Egan and F. Gibou. "Fast and scalable algorithms for constructing Solvent-Excluded Surfaces of large biomolecules". In: *Journal of Computational Physics* 374 (2018), pp. 91–120.
- [84] M. Mirzadeh, M. Theillard, and F. Gibou. "A second-order discretization of the nonlinear PoissonBoltzmann equation over irregular geometries using non-graded adaptive Cartesian grids". In: *Journal of Computational Physics* 230 (2010), pp. 2125–2140.
- [85] M. Mirzadeh et al. "An Adaptive, Finite Difference Solver for the Nonlinear Poisson-Boltzmann Equation with Applications to Biomolecular Computations". In: *Comm. Comput. Phys.* 13 (01 2013), pp. 150–173.

- [86] D. Benton et al. “Receptor binding and priming of the spike protein of SARS-CoV-2 for membrane fusion”. en. In: *Nature* (Sept. 2020).
- [87] B. Lee and F.M. Richards. “The interpretation of protein structures: estimation of static accessibility”. In: *Journal of molecular biology* 55.3 (1971), pp. 379–400.
- [88] Q. Pan and X.-C. Tai. “Model the Solvent-Excluded Surface of 3D Protein Molecular Structures Using Geometric PDE-Based Level-Set Method”. In: *Communications in Computational Physics* 6 (2009), pp. 777–792.
- [89] S. Osher and J. Sethian. “Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations”. In: *J. Comput. Phys.* 79 (1988), pp. 12–49.
- [90] C. Min and F. Gibou. “A Second Order Accurate Level Set Method on Non-Graded Adaptive Cartesian Grids”. In: *J. Comput. Phys.* 225 (2007), pp. 300–321.
- [91] L. Dagum and R. Menon. “OpenMP: An Industry-Standard API for Shared-Memory Programming”. In: *IEEE Comput. Sci. Eng.* 5.1 (Jan. 1998), pp. 46–55.
- [92] OpenMP Architecture Review Board. *OpenMP Application Program Interface Version 5.0*. May 2018.
- [93] Giampaolo Merlini et al. In: 39.11 (2001), pp. 1065–1075.
- [94] Jun-Hao Wen et al. “Cellular Protein Aggregates: Formation, Biological Effects, and Ways of Elimination”. In: *International Journal of Molecular Sciences* 24.10 (2023).
- [95] Jason K. Davis and Suzanne S. Sindi. “A mathematical model of the dynamics of prion aggregates with chaperone-mediated fragmentation”. In: *Journal of Mathematical Biology* 72.6 (2016), pp. 1555–1578.
- [96] Susan W Liebman and Yury O Chernoff. “Prions in Yeast”. In: *Genetics* 191.4 (Aug. 2012), pp. 1041–1072.
- [97] Reed B. Wickner et al. “Yeast Prions: Structure, Biology, and Prion-Handling Systems”. In: *Microbiology and Molecular Biology Reviews* 79.1 (2015), pp. 1–17.
- [98] Reed B. Wickner et al. “Yeast Prions”. In: *Prion* 1.2 (2007). PMID: 19164928, pp. 94–100.
- [99] Yury O. Chernoff. “Stress and prions: Lessons from the yeast model”. In: *FEBS Letters* 581.19 (May 2007), pp. 3695–3701.
- [100] Yury O. Chernoff et al. “Chapter Seven - Application of yeast to studying amyloid and prion diseases”. In: ed. by Dhavendra Kumar. Vol. 105. *Advances in Genetics*. Academic Press, 2020, pp. 293–380.

- [101] Chuankai Zhou et al. “Organelle-Based Aggregation and Retention of Damaged Proteins in Asymmetrically Dividing Cells”. In: *Cell* 159.3 (Oct. 2014), pp. 530–542.
- [102] Mick F. Tuite and Tricia R. Serio. “The prion hypothesis: from biological anomaly to basic regulatory mechanism”. In: *Nature Reviews Molecular Cell Biology* 11.12 (Nov. 2010), pp. 823–833.
- [103] Beidong Liu et al. “The Polarisome Is Required for Segregation and Retrograde Transport of Protein Aggregates”. In: *Cell* 140.2 (2010), pp. 257–267.
- [104] Juha Saarikangas and Yves Barral. “Protein aggregation as a mechanism of adaptive cellular responses”. In: *Current Genetics* 62.4 (2016), pp. 711–724.
- [105] Jens Tyedmers, Axel Mogk, and Bernd Bukau. “Cellular strategies for controlling protein aggregation”. In: *Nature reviews. Molecular cell biology* 11 (Oct. 2010), pp. 777–88.
- [106] Mikahl Banwarth-Kuhn and Suzanne Sindi. “How and why to build a mathematical model: A case study using prion aggregation”. In: *Journal of Biological Chemistry* 295.15 (Apr. 2020), pp. 5022–5035.
- [107] Jason K. Davis and Suzanne S. Sindi. “A mathematical model of the dynamics of prion aggregates with chaperone-mediated fragmentation”. In: *Journal of Mathematical Biology* 72.6 (Aug. 2015), pp. 1555–1578.
- [108] Paul Lemarre, Laurent Pujo-Menjouet, and Suzanne S. Sindi. “A unifying model for the propagation of prion proteins in yeast brings insight into the [PSI⁺] prion”. In: *PLOS Computational Biology* 16.5 (May 2020). Ed. by Philip K Maini, e1007647.
- [109] K. C. Kunes, D. L. Cox, and R. R. P. Singh. “One-dimensional model of yeast prion aggregation”. In: *Physical Review E* 72.5 (Nov. 2005).
- [110] Motomasa Tanaka et al. “The physical basis of how prion conformations determine strain phenotypes”. In: *Nature* 442.7102 (June 2006), pp. 585–589.
- [111] Saul Acevedo and Alexander J. Stewart. “Eco-evolutionary trade-offs in the dynamics of prion strain competition”. In: *Proceedings of the Royal Society B: Biological Sciences* 290.2002 (July 2023).
- [112] Mikahl Banwarth-Kuhn and Suzanne Sindi. “Multi-Scale Mathematical Modeling of Prion Aggregate Dynamics and Phenotypes in Yeast Colonies”. In: *Apolipoproteins, Triglycerides and Cholesterol*. IntechOpen, Nov. 2019.
- [113] Aaron Derdowski et al. “A Size Threshold Limits Prion Transmission and Establishes Phenotypic Diversity”. In: *Science* 330.6004 (2010), pp. 680–683.
- [114] Lee J. Byrne et al. “The Number and Transmission of [PSI⁺] Prion Seeds (Propagons) in the Yeast *Saccharomyces cerevisiae*”. In: *PLOS ONE* 4.3 (Mar. 2009), pp. 1–10.

- [115] C B Tyson, P G Lord, and A E Wheals. “Dependency of size of *Saccharomyces cerevisiae* cells on growth rate.” In: *Journal of Bacteriology* 138.1 (1979), pp. 92–98.
- [116] Romain Christiano et al. “Global Proteome Turnover Analyses of the Yeasts *S.cerevisiae* and *S.pombe*”. In: *Cell Reports* 9.5 (Dec. 2014), pp. 1959–1965.
- [117] Juliet Ansel et al. “Cell-to-Cell Stochastic Variation in Gene Expression Is a Complex Genetic Trait”. In: *PLoS Genetics* 4.4 (Apr. 2008). Ed. by Jonathan Flint, e1000049.

Appendix **A**

Appendix for Chapter 4

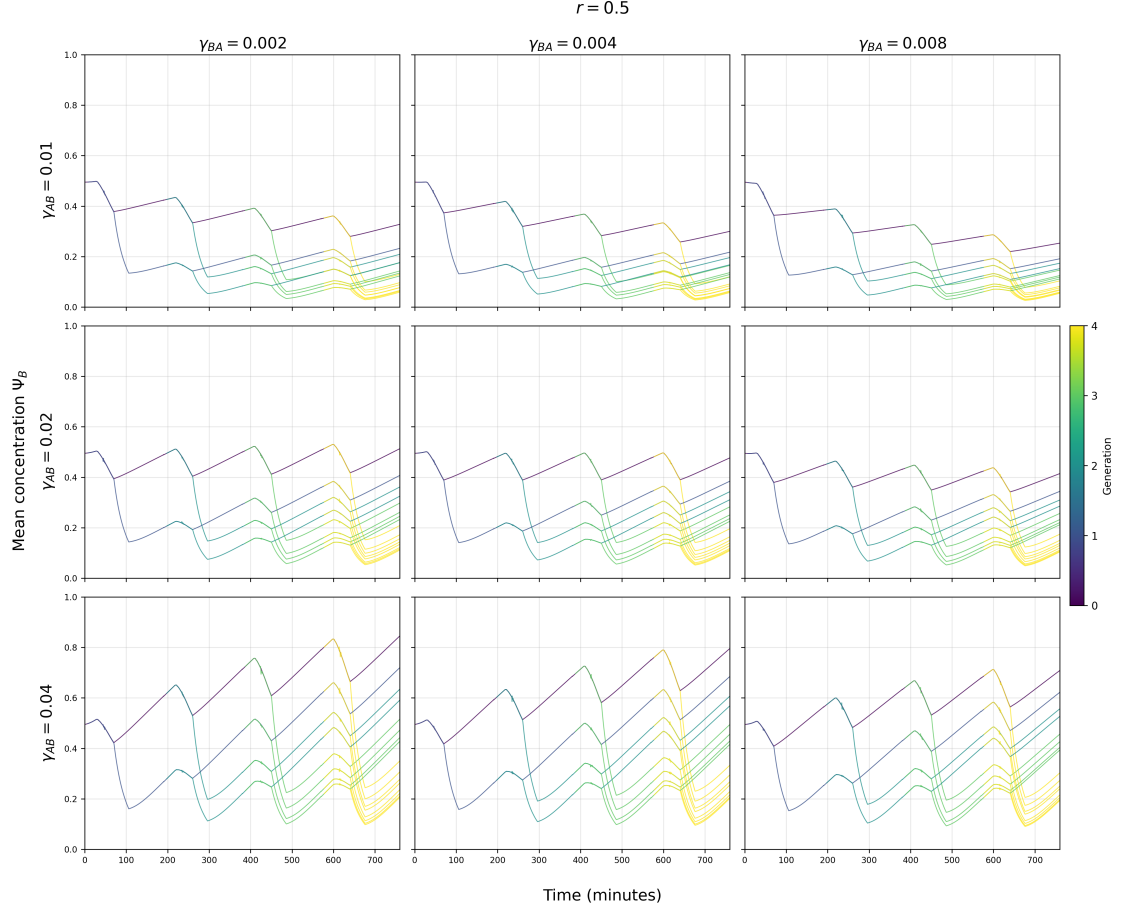


Figure A.1: Mean aggregate concentration ψ_B over time for all cells in a five-generation lineage tree, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.5$ fixed. Each curve corresponds to a single simulated cell; color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). Sharp downward steps mark division events, after which both mother and daughter enter a recovery phase. Trajectories are ordered by replicative age across all parameter combinations, with older-lineage cells carrying persistently higher aggregate burden.

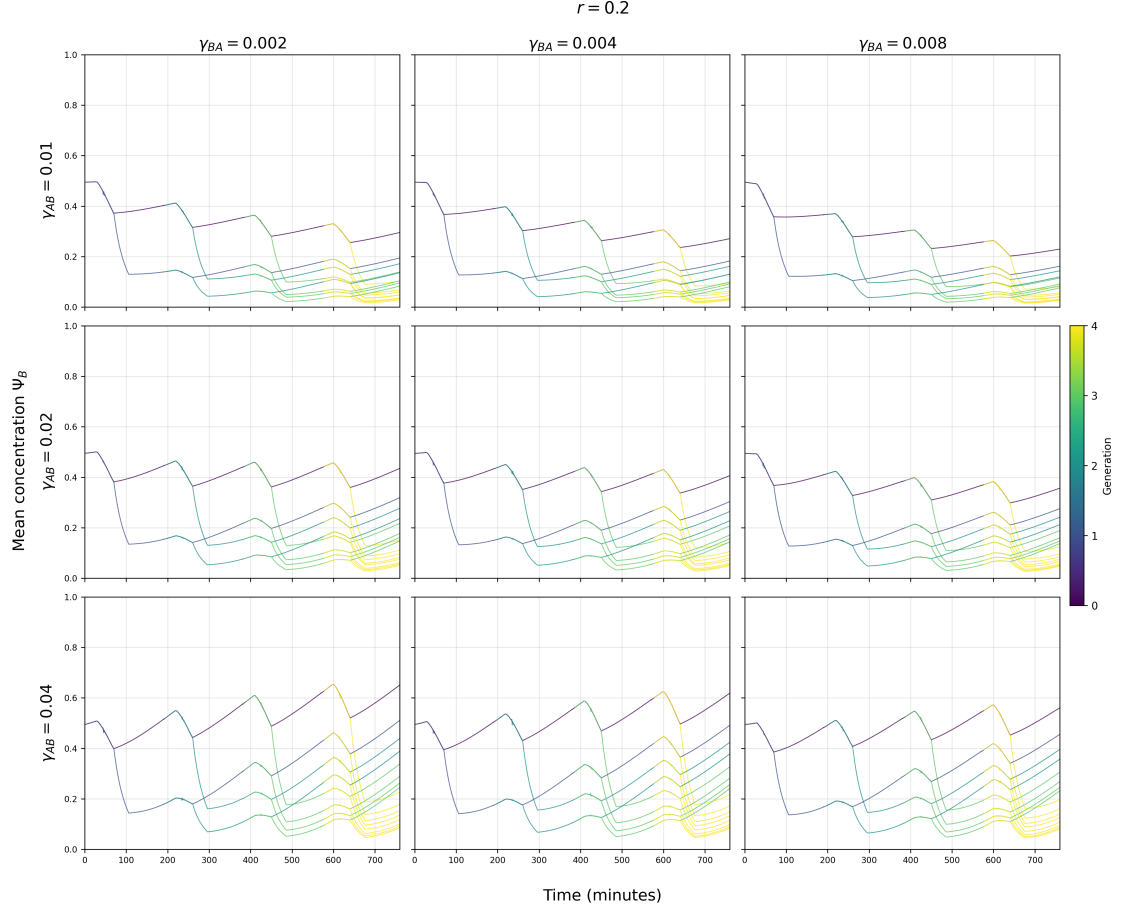


Figure A.2: Mean aggregate concentration ψ_B over time for all cells in a five-generation lineage tree, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.2$ fixed. Each curve corresponds to a single simulated cell; color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). Sharp downward steps mark division events, after which both mother and daughter enter a recovery phase. Trajectories are ordered by replicative age across all parameter combinations, with older-lineage cells carrying persistently higher aggregate burden.

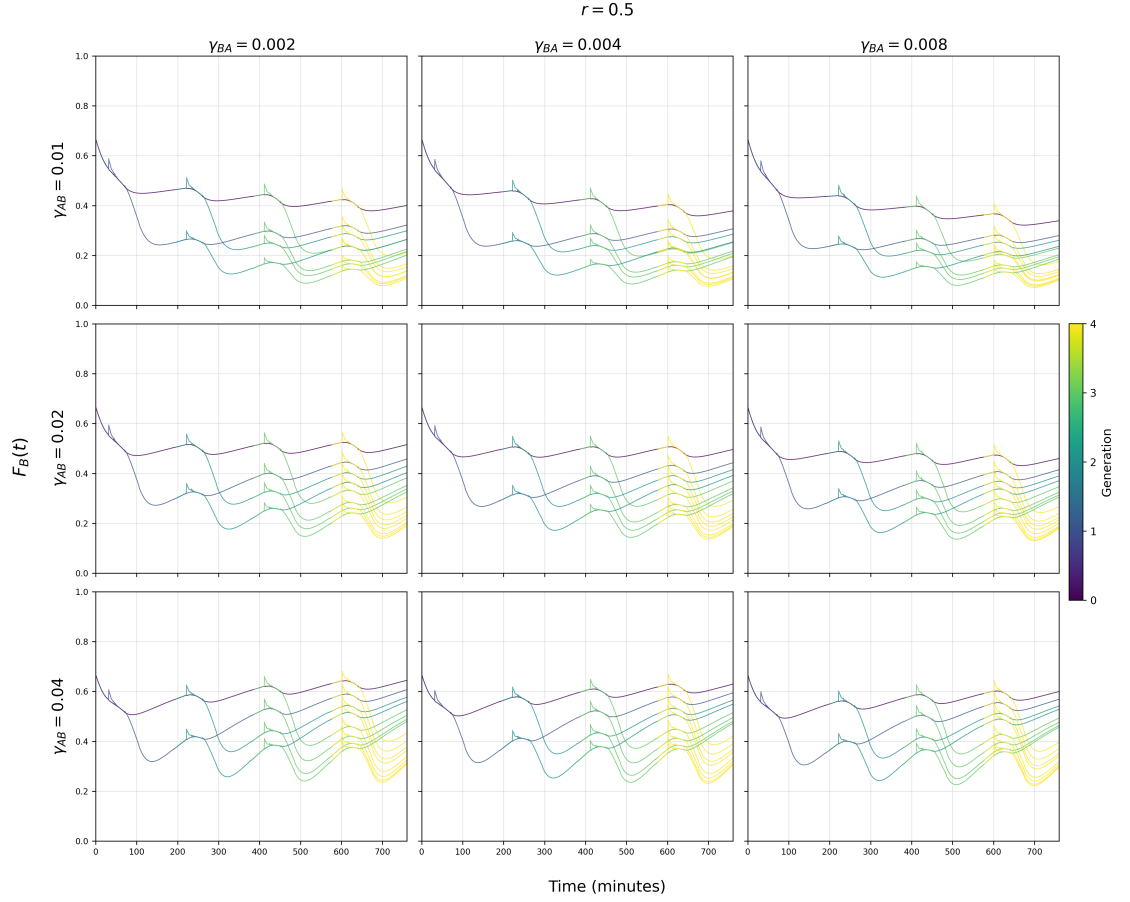


Figure A.3: Aggregate mass fraction $F_B(t) = 2\psi_B/(\psi_A + 2\psi_B)$ over time for all cells in a five-generation lineage, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.5$ fixed. Color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). At each division event ($t = 70$), the newly detached daughter cell exhibits a transient upward spike in F_B reflecting her small initial volume; as she grows and acquires additional monomer the fraction falls to its post-division level. The mother experiences a modest dilution at division but remains at a substantially higher F_B than the daughter, reflecting the asymmetric partitioning of aggregate at cytokinesis. Increasing γ_{BA} shifts the long-term balance from aggregate retention toward dilution across all rows, while increasing γ_{AB} raises the overall level of F_B across all columns. Generational ordering persists across all nine parameter combinations.

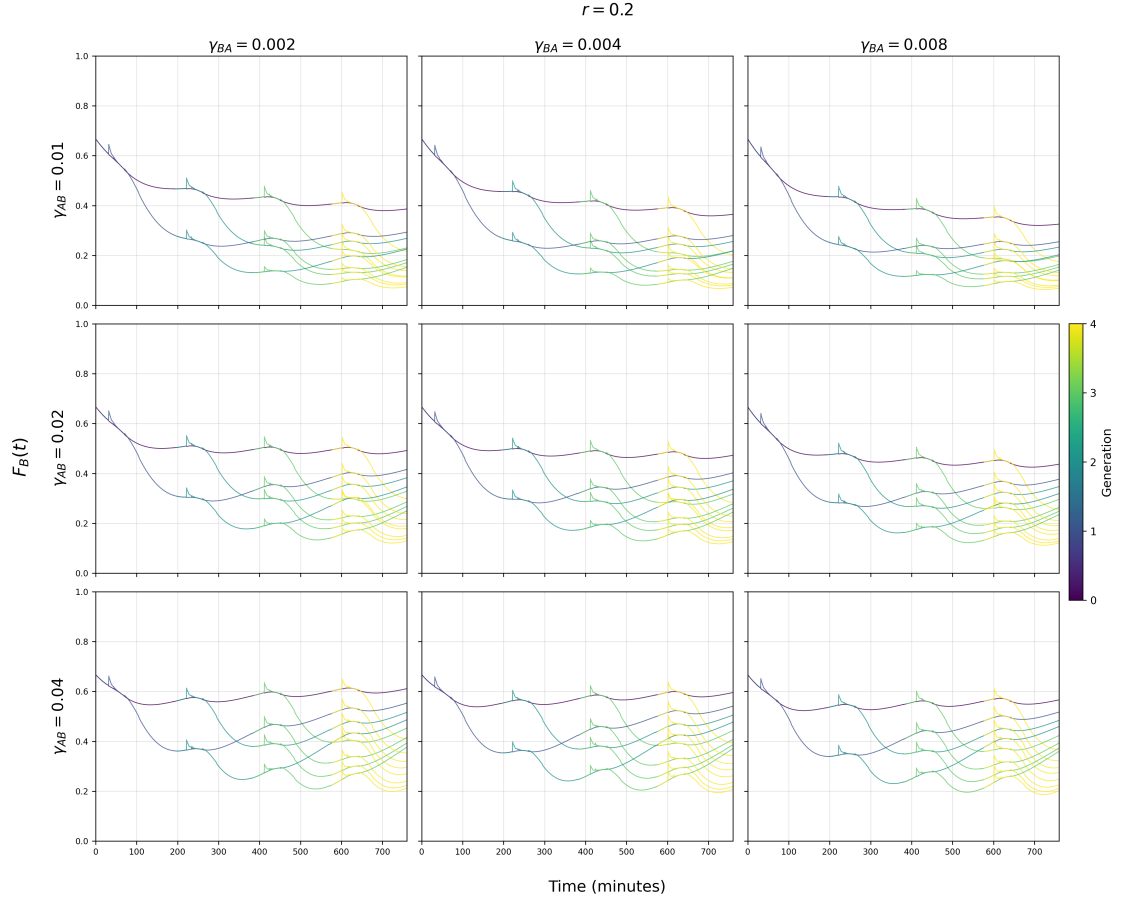


Figure A.4: Aggregate mass fraction $F_B(t) = 2\psi_B/(\psi_A + 2\psi_B)$ over time for all cells in a five-generation lineage, for $\gamma_{AB} \in \{0.01, 0.02, 0.04\}$ (rows) and $\gamma_{BA} \in \{0.002, 0.004, 0.008\}$ (columns), with $r = 0.2$ fixed. Color indicates generation index from earliest (purple, generation 0) to latest (yellow-green, generation 4). At each division event ($t = 70$), the newly detached daughter cell exhibits a transient upward spike in F_B reflecting her small initial volume; as she grows and acquires additional monomer the fraction falls to its post-division level. The mother experiences a modest dilution at division but remains at a substantially higher F_B than the daughter, reflecting the asymmetric partitioning of aggregate at cytokinesis. Increasing γ_{BA} shifts the long-term balance from aggregate retention toward dilution across all rows, while increasing γ_{AB} raises the overall level of F_B across all columns. Generational ordering persists across all nine parameter combinations.