Swelling as a Stabilizing Mechanism During Ion Bombardment of Thin Films: An Analytical and Numerical Study

Jennifer M. Swenson
jswenson@smu.edu

Follow this and additional works at: https://scholar.smu.edu/hum_sci_mathematics_etds

Part of the Fluid Dynamics Commons, Nanoscience and Nanotechnology Commons, Partial Differential Equations Commons, and the Semiconductor and Optical Materials Commons

Recommended Citation
https://scholar.smu.edu/hum_sci_mathematics_etds/2

This Dissertation is brought to you for free and open access by the Mathematics at SMU Scholar. It has been accepted for inclusion in Mathematics Theses and Dissertations by an authorized administrator of SMU Scholar. For more information, please visit http://digitalrepository.smu.edu.
SWELLING AS A STABILIZING MECHANISM
DURING ION BOMBARDMENT OF THIN FILMS:
AN ANALYTICAL AND NUMERICAL STUDY

Approved by:

Dr. Scott Norris
Associate Professor of Mathematics

Dr. Vladimir Ajaev
Professor of Mathematics

Dr. Daniel Reynolds
Associate Professor of Mathematics

Dr. Paul Krueger
Professor of Mechanical Engineering
SWELLING AS A STABILIZING MECHANISM
DURING ION BOMBARDMENT OF THIN FILMS:
AN ANALYTICAL AND NUMERICAL STUDY

A Dissertation Presented to the Graduate Faculty of the
Dedman College
Southern Methodist University
in
Partial Fulfillment of the Requirements
for the degree of
Doctor of Philosophy
with a
Major in Computational and Applied Mathematics
by
Jennifer M. Swenson

B.S., 2013, Mathematics, Sam Houston State University
M.S., 2015, Computational and Applied Mathematics, Southern Methodist University

May 19, 2018
ACKNOWLEDGMENTS

Graduate school was never something that I thought I would (or could) pursue, so I am forever grateful for those who helped and encouraged me along the way.

Thank you to my outstanding advisor, Dr. Scott Norris, for creating an environment for me to grow, as both a researcher and a person. I appreciate you answering my endless questions, encouraging me during the disappointments, and taking a chance on me as your first student. We have consumed a lot of pizza and pastries to get to this point, but we made it.

Thank you to Dr. Daniel Reynolds, for numerical expertise, thoughtful conversations regarding boundary condition issues, and overall support while learning PETSc. A special thank you for sharing four classes worth of knowledge with me and for knowing that I am more than my performance anxiety.

Thank you to Dr. Vladimir Ajaev, for serving on my dissertation committee, helpful discussions over the swelling model, and for introducing me to fluid mechanics.

Thank you to Dr. Paul Krueger, for serving on my dissertation committee, beneficial discussions over the swelling equations, especially as it pertains to reviewer feedback, and for teaching one of my favorite classes, viscous flow.

Thank you to Dr. Joy Perkinson, for excellent experimental work, productive collaboration, and for letting me run a few ion bombardment/GISAXS experiments during beam time.

Thank you to the influential professors at Sam Houston State University. To Dr. Edward Swim, for being the driving force behind my pursuit of graduate school. To Dr. John Alford, for introducing me to applied mathematics. To Dr. David Pooley, for helping me enjoy physics again.

Thank you to my parents, Pete and Jeri-Lynn, for training me up in the way I should go, immeasurable support of my endeavors, and giving me a double dose of “math nerd”.
And to our dogs, for providing unending amounts of snuggles and “puppy therapy” over the years.

Thank you to my best friend, Mary, for being the sister I never had, ridiculously long phone calls, sharing the same sense of humor, and appreciating sarcasm.

Thank you to my Community Group and to my friends at Watermark, for continually pointing me to Jesus, speaking truth in love, and reminding me that my identity is not found in my research, but in Christ.

Thank you to Fuge camps, for changing my life: first as a student, then as a staffer. Thank you to the Fuge staffs I have served alongside, for encouraging me in the Lord both during camp and from afar, countless laughter, dance parties, and hammock villages.

“And whatever you do, whether in word or deed, do it all in the name of the Lord Jesus, giving thanks to God the Father through Him.” Colossians 3:17
Irradiation of semiconductor surfaces often leads to the spontaneous formation of rippled structures at certain irradiation angles. However, at high enough energies, these structures are observed to vanish for all angles, despite the absence of any identified, universally-stabilizing physical mechanisms in operation. Here, we examine the effect on pattern formation of radiation-induced swelling, which has been excluded from prior treatments of stress in irradiated films. After developing a suitable continuum model, we perform a linear stability analysis to determine its effect on stability. Under appropriate simplifying assumptions, we find swelling indeed to be stabilizing at wavenumbers typical of experimental observations. We relax simplifications and perform a numerical linear stability analysis on the nonlinear model, where stability regimes result from our parameter sweeps. Therefore, this mechanism may account for the vanishing ripples observed at high energies.

Grazing-Incidence Small-Angle X-ray Scattering (GISAXS) is an experimental technique that can be used during ion bombardment to study nanostructures. However, in the nonlinear regime, the GISAXS profile no longer equals the structure factor and therefore cannot be directly compared to mathematical calculations. Here, we compare nonlinear regime experiments to nonlinear simulations and simulated GISAXS. After numerically simulating an appropriate PDE, we simulate GISAXS scattering across the simulated surfaces. We discover that the sawtooth and chevron structures found in experiments also appear in our simulated surfaces. Additionally, asymmetry and harmonic peaks observed in experimental GISAXS
profiles are also visible in our simulated GISAXS. Therefore, this framework may establish a useful way of interpreting late-stage GISAXS data.
# TABLE OF CONTENTS

LIST OF FIGURES ........................................................................................................... x

CHAPTER

1. Introduction .................................................................................................................. 1

2. The Effect of Swelling on Ion-Induced Pattern Formation: Model and Linearization .................................................. 14
   2.1. Introduction ............................................................................................................ 14
   2.2. Model and Non-Dimensionalization ................................................................. 15
      2.2.1. Model ........................................................................................................... 15
      2.2.2. Conversion to a Moving Frame and Non-Dimensionalization .......... 20
   2.3. Analysis .................................................................................................................. 22
      2.3.1. Steady State Solution ................................................................................... 22
      2.3.2. Linear Stability Analysis ........................................................................... 24
      2.3.3. Small-A limit ............................................................................................... 26
      2.3.4. Solution ........................................................................................................ 28
   2.4. Discussion .............................................................................................................. 30
   2.5. Conclusions .......................................................................................................... 37

3. The Effect of Swelling on Ion-Induced Pattern Formation: Numerical Implementation .................................................. 39
   3.1. Introduction ........................................................................................................... 39
   3.2. Small-A Equations .............................................................................................. 40
      3.2.1. Discretizations of the Small-A Equations ................................................. 41
      3.2.2. Creating the Vector for the System of Small-A Equations ................. 43
      3.2.3. Small-A Equations Results ..................................................................... 46
   3.3. Original Linearized Equations .......................................................................... 50
3.3.1. Discretizations of the Linearized Equations ......................... 51
3.3.2. Creating the Vector for the System of Linearized Equations ....... 52
3.3.3. Linearized Equations Results with Small-A Value ................... 54
3.3.4. Linearized Equations Results with Increasing A ...................... 56

3.4. Discussion and Conclusion ............................................. 58

4. Surface Dynamics and GISAXS Spectra in the Nonlinear Regime:
  Comparing Experiments and Simulations ................................. 61
4.1. Introduction ........................................................... 61
4.2. Nonlinear GISAXS Results ............................................ 63
4.3. Model: aKS Equation ................................................... 63
4.4. Simulation ............................................................. 66
  4.4.1. Time-Stepping ..................................................... 67
  4.4.2. Spatial Discretization ............................................. 68
  4.4.3. PETSc Implementation ............................................. 71
4.5. Simulation Results ..................................................... 72
4.6. Simulated GISAXS ...................................................... 78
4.7. Conclusion ............................................................ 80

5. Stress-Free Strain Rates from the BCA:
  Calculating Deviatoric Strain ............................................ 82
5.1. Introduction ........................................................... 82
5.2. Displacement Fields .................................................... 84
  5.2.1. PyCraters and TRI3DST ........................................... 84
  5.2.2. Calculate the Displacement Fields ............................... 84
5.3. Strain Tensor .......................................................... 85
  5.3.1. Smooth Statistical Noise ........................................... 85
  5.3.2. Calculate the Strain Field ....................................... 86
5.3.3. Zeroth Moment ................................................................. 89
5.3.4. Deviatoric component ....................................................... 90
5.4. Result Comparison ............................................................. 91
5.4.1. Some Observations ......................................................... 93
5.5. Conclusion ................................................................. 94

APPENDIX

A. Derivation of Kinematic Condition ....................................... 96

BIBLIOGRAPHY ................................................................. 98
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Here are examples of rippled patterns found in sand dunes and the ocean floor. Ripples in sand dunes form from the wind moving across the surface, while ripples in the ocean floor form from the current moving along the sand. The original images are from Skeeze [38] and Harwood [16] respectively and are edited with arrows to emphasize the direction of the wind or water across the surfaces. ..................................</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Rippled patterns can also be found on a much smaller scale due to ion bombardment. Here, the ripples form in the same direction as the ion beam. This figure is from work by Chan and Chason [10] (reprinted from [10], with the permission of AIP Publishing), combining their schematic with an image from work by Mayer, Chason, and Howard [21] (reprinted from [21], with the permission of AIP Publishing). ..................</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>Our schematic model of ion bombardment, for the case of normal incidence irradiation. Here ( h_0 ) is the film depth. As the bombardment process continues, ions are sputtered away from the surface, causing the top layer ( h_2 ) to recede. Note that the ion beam is perpendicular to the surface of the material and that as ( h_2 ) recedes and moves down into the amorphous layer, ( h_1 ) will also move down into the crystalline substrate at the same rate. ........................................................</td>
<td>4</td>
</tr>
<tr>
<td>1.4</td>
<td>Here is an example of some experimental observations. The resulting surface morphology can be determined from any combination of ion beam angle and ion beam energy within the given regime. For ( \theta &lt; 50 ), surfaces will be stable and produce no patterns. For ( \theta &gt; 50 ), surfaces will be unstable and produce rippled structures - either in the same direction as the ion beam (parallel mode) for ( 50 &lt; \theta &lt; 80 ) or perpendicular to the beam (perpendicular mode) for ( \theta &gt; 80 ). Mathematical models largely reflect this structure. This image is from work by Madi and Aziz [18] (reprinted from [18], with permission from Elsevier). ..................</td>
<td>6</td>
</tr>
</tbody>
</table>
Here is an example of some experimental observations that are not well understood. For high ion beam energies, ripples vanish for all incidence angles. However, according to existing mathematical models, mechanisms should produce patterns at some $\theta$. This disagreement suggests that current models may be missing an unconditionally stable mechanism. This image is from Hofssäss, Bobes, and Zhang’s work [17] (reprinted from [17], with the permission of AIP Publishing).

A possible missing mechanism. At higher energies, cavities are created as the ions embed into the amorphous layer of the material. The cavities remain in the amorphous layer and grow over time until they are sputtered away. Existing mathematical models neglect this expansion. We are interested in the effect of this mechanism. These images are from work by Chini, Okuyama, Tanemura, and Nordlund [11] (reprinted figures with permission from [11] by the American Physical Society).

A schematic of GISAXS experimentation. The advantage of GISAXS is that one can obtain “real time” results of surface behavior during ion bombardment, but the cost is that GISAXS only provides the structure factor. This schematic is from Yamamoto’s work [41].

A comparison of GISAXS data and mathematical models in the linear regime, where the data fits experimentation. These images are from work by Norris, Perkinson, Mokhtarzadeh, Anzenberg, Aziz, and Ludwig [28] (licensed under CC BY 4.0 [9]).

A schematic illustration of our problem geometry, for the case of normal incidence irradiation. Here $h_0$ is the film depth (typically twice the characteristic distance that the ions are embedded into the material). As the bombardment process continues, ions are sputtered away from the surface, causing the top layer $h_2$ to recede at speed $V$. Note that the ion beam is perpendicular to the surface of the material and that as $h_2$ recedes and moves down into the amorphous layer, $h_1$ will also move down into the crystalline substrate at the same rate.

Plots of the steady state solutions, Eqs. (2.33)-(2.36), where $A \vec{z} \in [0, 10]$ is an arbitrary domain chosen to illustrate qualitative behavior. The density (a) and pressure (b) of the material decrease as the material moves toward the surface. (c) In the moving frame of reference, the velocity is constant when material first enters the amorphous layer, then increases under continual swelling as material moves toward the surface. (d) The age of the material is zero and increases until parcels reach the surface and are sputtered away.
2.3 A plot of the dimensionless growth rate $\Sigma_1$ given in Eq. 2.68, for the case $\bar{h}_{11} = \bar{h}_{21}$ (normal incidence). For $Q \lesssim 1.5$, the negative sign of $\Sigma_1$ indicates that swelling is stabilizing. For $Q \gtrsim 1.5$, the positive sign of $\Sigma_1$ indicates that it becomes destabilizing. Though not pictured here, $\Sigma_1$ saturates at a value of 1 as $Q \to \infty$.

2.4 Visualizations of the relative flow field $v_{11}(\bar{x}, \bar{z}; Q) - v_{11}(\bar{x}, \bar{z}; 0)$ for $Q = \{0.7, 1.5, 2.0\}$. (a) For $Q = 0.7$, the availability of lateral expansion at the hilltops induces flow from hilltops into valleys, which is stabilizing. (b) For $Q = 1.5$, this former mechanism is essentially offset by shear forces pulling material originating in valleys uphill. (c) For $Q = 2.0$, shear forces have become stronger than the lateral hilltop expansion mechanism, and the result is destabilizing.

2.5 A 2D heatmap plot of $\Sigma_1(Q; \theta)$. Blue indicates stability, red indicates instability, and grey indicates neutral stability. The dashed line is a guide to the eye, and separates a domain of unconditional stability on the left, from a domain of alternating stability and instability on the right. Notably, long-wavelength perturbations are stabilized at all angles, and the range of stable wavenumbers grows with increasing $\theta$.

2.6 A plot of film depth divided by sputter yield (which is proportional to the dwell time in Eq. (2.72)) for different ions and energies. The minimum of values align well with the range 250 eV – 2 keV in which the majority of studies on ripple formation have been performed.

3.1 Visualizations of the analytical and numerical solutions of the small-A equations, with $h_{11}=0.1$, $h_{21}=0.1$, and $M_z=10001$. We sweep over $Q$ values from 0.0 to 5.0, at increments of 0.05. The results appear to agree and are indiscernible from one another in this figure, with stability regimes (where $\Sigma_1 < 0$) for $Q \leq 1.5$, which is within experimental parameters.

3.2 The convergence plot for our numerical small-A equations, with solutions compared to our analytical small-A solutions. Parameter values are chosen to be $A=0.01$, $Q=1.0$, $h_{11}=0.1$, and $h_{21}=0.1$, with various step sizes, $\Delta M_z=\{0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.002, 0.001, 0.0005, 0.0002\}$. The convergence rate is of order 2, which is consistent with our second-order upwinding scheme.

3.3 Visualizations of the analytical and numerical solutions of the small-A equations, along with numerical solutions of the linearized equations, with a small value for $A$: $A=0.00001$. Additionally, we choose $h_{11}=0.1$, $h_{21}=0.1$, $M_z=10001$ and sweep over $Q$ values from 0.0 to 5.0, at increments of 0.05. The results between all three methods agree, with stability regimes ($\Sigma/A < 0$) for $Q \leq 1.5$, which is within experimental parameters.
3.4 The convergence plot for our numerical linearized equations, with chosen parameter values $A=0.01$, $Q=1.0$, $h_{11}=0.1$, and $h_{21}=0.1$, with various step sizes, $dMz=\{.2, .1, .05, .02, .01, .005, .002, .001, .0005\}$. The error is calculated by comparing these solutions to results when the step size is smaller, $dMz=0.0002$. The convergence rate is of order 2, which is consistent with our second-order upwinding scheme.

3.5 We plot solutions for $\sigma$ when increasing $A$ in the linearized equations, with parameter values $h_{11}=0.1$, $h_{21}=0.1$, and $Mz=501$. We sweep over $Q$ values from 0.0 to 5.0 and find that stability regimes exist for each value of $A$. This suggests that swelling has an effect on stability for various expansion rates.

4.1 Here are AFM images of saturated patterns with ion beam angles of 55° (top left), 70° (top right), 75° (bottom left), and 85° (bottom right). Brightness in this image represents structure height. As you scan the image vertically, from bottom to top, the brightness increases, then suddenly becomes darker, then increases again. This corresponds to sawtooth structures in the X direction. Additionally, we observe sharp, chevron shapes near the peaks. We note that in these images, ions are aimed in the vertical direction. In models, the ion direction is traditionally called the ‘x’ direction. The original images are from Perkinson and Aziz (private communication), with emphasis added to show chevrons and the direction of the ions.

4.2 (top) GISAXS analysis in the $x$-direction of the resulting patterns with an ion beam angle of 70° shows asymmetrical scattering. (bottom) GISAXS analysis in the $y$-direction of the resulting patterns with an ion beam angle of 75° shows a “smeared out” final scattering profile. These lines represent times during the ion bombardment process, starting with purple and ending with yellow. These images are from Perkinson and Aziz (private communication).

4.3 We present surface patterns of the 2D higher-order aKS with parameters $\beta_1 = -1$, $\beta_2 = 0$, $\gamma_1 = -1$, $\gamma_2 = -1$, $\gamma_3 = -1$, $\delta_1 = -0.5$, $\delta_2 = -0.5$, $\epsilon_1 = 0.05$, and $\epsilon_2 = 0$. We show one quarter of the simulated domain to better highlight the surface features. Starting with a random plot between 0 and 0.01, pattern formations are already visible by $t = 50$ (top left) and form ripples by $t = 500$ (top right). Sawtooth structures are present by $t = 1000$ (bottom left) and become increasingly chevron-shaped by $t = 3000$ (bottom right).
4.4 Here is the convergence plot for our higher-order aKS equations, with chosen parameters values $h_x = 0.1$, $h_y = 0.1$, with a final time of 100, and the same values from our higher-order aKS equation: $\beta_1 = -1$, $\beta_2 = 0$, $\gamma_1 = -1$, $\gamma_2 = -1$, $\gamma_3 = -1$, $\delta_1 = -0.5$, $\delta_2 = -0.5$, $\epsilon_1 = 0.05$, and $\epsilon_2 = 0$. The error is calculated by comparing solutions with $t = \{2, 5, 10, 12, 15, 20\}$, where $dt = 1/t$ and therefore range from 0.5 to 0.05 to results when the time step is smaller, $dt = 0.01$. The convergence rate is of order 2, which is consistent with the Crank-Nicolson time stepping scheme.

4.5 Here is the convergence plot for our higher-order aKS equations, with a chosen time step of $dt = 0.1$, with a final time of 100, and the same values from our higher-order aKS equation: $\beta_1 = -1$, $\beta_2 = 0$, $\gamma_1 = -1$, $\gamma_2 = -1$, $\gamma_3 = -1$, $\delta_1 = -0.5$, $\delta_2 = -0.5$, $\epsilon_1 = 0.05$, and $\epsilon_2 = 0$. The error is calculated by comparing solutions with $h_x = \{2, 5, 10, 12, 15, 20, 50, 100, 200, 500\}$, where $dh_x = 1/h_x$ and therefore range from 0.5 to 0.002 to results when the time step is smaller, $dh_x = 0.0001$. The convergence rate is of order 2, which is consistent with our chosen second-order discretizations.

4.6 Visualizations of the corresponding GISAXS progression plots for the previously shown simulation of the higher-order aKS equation in 2D. The profiles begin with a low scattering intensity at high wavenumbers, which suggests an absence of structures. Slight asymmetry is visible in the X direction at $t = 500$ (near $q_x = 0.8$), at $t = 1000$ (near $q_x = -0.7$), and most noticeably at $t = 3000$ (near $q_x = -2.2$). In our simulated GISAXS spectra, we observe asymmetry (which corresponds to the emergence of sawtooths), harmonic peaks, and “smeared out” scattering patterns at late times.

5.1 The displacement field for $\theta = \{0, 10, 20, 30, 40, 50, 60, 70, 80\}$, where we see the correlation between atomistic movement and ion beam incidence angle.

5.2 The components of the strain tensor for $\theta = \{0, 10, 20, 30\}$.

5.3 The components of the strain tensor for $\theta = \{40, 50, 60, 70, 80\}$.

5.4 Zeroth moments for both the strain tensor and deviatoric strain tensor for incidence angles of $\theta = \{0, 10, 20, 30, 40, 50, 60, 70, 80\}$.

5.5 Strain tensor results from Norris’ $D(\theta)$ in [27].

A.1 A schematic illustration of a patch of surface under ion bombardment, used in the derivation of the modified kinematic condition.
This is dedicated to my parents, for their endless love and support.
Chapter 1
Introduction

When a material is repeatedly bombarded by an object, the surface may become deformed into patterned structures. These patterned surfaces have been observed in natural materials. Rippled structures can form on the surface of sand dunes as a result of wind moving across the surface. These patterns can also form along the ocean floor as the current moves across the sand. Examples of these structures are presented in Figure 1.1 from Skeeze [38] and Harwood [16] respectively. In this dissertation, we investigate pattern formation at a much smaller scale.

Surface deformation is also known to occur on the microscopic level due to ion bombardment. An example of the resulting patterns is seen in Figure 1.2 from work by Chan and Chason [10], where ripples form across the surface in the direction of the ion beam. Ion bombardment is the process of ionizing a gas and rapidly accelerating the resulting ions at a target material. In a technique known as focused ion beam (FIB), ionization occurs from an electric field and electrostatic lenses are used to focus the beam. Our schematic model of ion bombardment is presented in Figure 1.3. When ions hit the target, they can ricochet off of the surface or embed into the material. Impacts from this ion and target collision can result in the ejection of atomistic particles from the surface, therefore eroding the material. This removal of target material during ion irradiation is called sputtering. Additional surface deformation can occur when the incoming ions implant or hit the atoms in the surface layer, leading to a redistribution of the remaining atoms. The combination of ions impacting surface atoms and particles sputtering away from the target can deform the surface into organized patterns.

Spontaneous nanoscale pattern formation due to ion bombardment has been observed on the surfaces of amorphous or amorphized materials since the 1960s [25]. These patterns
Figure 1.1. Here are examples of rippled patterns found in sand dunes and the ocean floor. Ripples in sand dunes form from the wind moving across the surface, while ripples in the ocean floor form from the current moving along the sand. The original images are from Skeeze [38] and Harwood [16] respectively and are edited with arrows to emphasize the direction of the wind or water across the surfaces.
can take the form of ripples or arrays of dots [42], at sizes as small as 7nm [40], and are dependent on parameters such as the angle of incidence of the ion beam, the ion energy, and in some cases, the ion flux [10, 13]. Studying such structures is of interest because ion beams are already ubiquitous throughout the semiconductor industry. Therefore, the ability to harness spontaneous, “bottom-up” pattern forming processes would open a new avenue for material surface processing using established, inexpensive technology.

During ion beam irradiation, the semiconductor surface evolves under the influence of several distinct physical mechanisms. Most obviously, irradiation causes some material to be sputtered away from the surface, an effect which was shown to be destabilizing in early, seminal work [4, 35, 36]. It has also gradually been realized that the atoms remaining on the surface, but redistributed to new locations, play an important role, and can stabilize or destabilize surfaces depending on irradiation geometry [6, 14, 23, 30, 32]. More recently, it has also been shown that ion impacts create significant stress within the material [19], and that the material response to these stresses can, itself, contribute to pattern formation [7, 8, 22, 29].
Figure 1.3. Our schematic model of ion bombardment, for the case of normal incidence irradiation. Here $h_0$ is the film depth. As the bombardment process continues, ions are sputtered away from the surface, causing the top layer $h_2$ to recede. Note that the ion beam is perpendicular to the surface of the material and that as $h_2$ recedes and moves down into the amorphous layer, $h_1$ will also move down into the crystalline substrate at the same rate.
Since we seek to understand the presence and nature of these spontaneous pattern formations, for reasons mentioned above, we create models to recreate experimental outcomes. Mathematical models are created from the knowledge of physical mechanisms during the ion irradiation process. The results of these models are compared to the experimental behavior of surface morphology. An example of some experimental observations are presented in Figure 1.4 from the work of Madi and Aziz [18]. The resulting surface morphology can be determined from any combination of ion beam angle and ion beam energy within the given regime. Existing mathematical models largely reflect this relation. However, models may not accurately capture some surface pattern formations, leading to discrepancies between models and experiments that must be remedied by adjusting the model. For instance, an example of some experimental observations that are not well understood is shown in Figure 1.5 from Hofsäss, Bobes, and Zhang’s work [17]. At high ion beam energies, patterned ripples on the surface of the target vanish for all incidence angles, while mathematical models propose that the patterns persist for some ion beam angle. This disagreement indicates that current models may be missing an unconditionally stable mechanism. A possible missing mechanism is observed in Figure 1.6 (from work by Chini, Okuyama, Tanemura, and Nordlund [11]), where vacancies appear to grow in the amorphous layer and are larger near the surface. This observed swelling has not been considered in existing models. We are interested in the effects of this expansion mechanism and therefore include swelling in our model. Chapters 2 and 3 of this dissertation investigate swelling as a stabilizing mechanism during ion bombardment of thin films.

In Chapter 2, we propose a set of governing equations that allow for swelling and study their linear stability. After converting to a moving frame of reference, we non-dimensionalize the system and solve the steady state equations. Then we linearize the governing equations about the steady state solution as part of the linear stability analysis process. We find that these linearized equations are not analytically solvable in closed form, because they lead to an integro-differentiable equation with variable coefficients. In order to make this system solvable, we use a simplification in which the implantation-induced stress is still small by
Figure 1.4. Here is an example of some experimental observations. The resulting surface morphology can be determined from any combination of ion beam angle and ion beam energy within the given regime. For $\theta < 50$, surfaces will be stable and produce no patterns. For $\theta > 50$, surfaces will be unstable and produce rippled structures - either in the same direction as the ion beam (parallel mode) for $50 < \theta < 80$ or perpendicular to the beam (perpendicular mode) for $\theta > 80$. Mathematical models largely reflect this structure. This image is from work by Madi and Aziz [18] (reprinted from [18], with permission from Elsevier).
Figure 1.5. Here is an example of some experimental observations that are not well understood. For high ion beam energies, ripples vanish for all incidence angles. However, according to existing mathematical models, mechanisms should produce patterns at some $\theta$. This disagreement suggests that current models may be missing an unconditionally stable mechanism. This image is from Hofsäss, Bobes, and Zhang’s work [17] (reprinted from [17], with the permission of AIP Publishing).
Figure 1.6. A possible missing mechanism. At higher energies, cavities are created as the ions embed into the amorphous layer of the material. The cavities remain in the amorphous layer and grow over time until they are sputtered away. Existing mathematical models neglect this expansion. We are interested in the effect of this mechanism. These images are from work by Chini, Okuyama, Tanemura, and Nordlund [11] (reprinted figures with permission from [11] by the American Physical Society).
the time bulk parcels are sputtered away from the free interface. We solve this simplified system of equations and calculate the dimensionless perturbation’s growth rate as a function of its wavenumber. We find a region of stability in the system at all ion beam angles for the specific dimensionless wavenumbers in which experiments are typically conducted. This suggests that swelling is a contributing factor of vanishing ripples at higher ion beam energies.

In Chapter 3, we want to know what happens when we relax our previous simplification and allow for finite values of implantation-induced stress in our system. Because the original linearized equations are analytically insolvable, we perform a numerical study to determine the solutions of the variable-coefficient linear system. We use the Portable, Extensible Toolkit for Scientific Computations (PETSc) library to solve this system, because of its convenient built-in features, including parallelization, which performs large matrix operations more efficiently. This allows us to quickly solve the linear system over many wavenumbers, thus recreating the growth rate obtained earlier. We first choose a small parameter value for the expansion rate, in order to compare these solutions to our results from our simplified system in Chapter 2. After demonstrating that these agree, we solve the original linearized equations for increasing values of expansion rates and find that stability regimes persist. By allowing swelling to occur in our model, we have found wavenumbers, consistent with experimental parameters, that correspond to a stabilizing system and therefore predict vanishing ripples at high energies. This suggests that our model helps align mathematical models and experimental observations, especially as it pertains to vanishing ripples. Moreover, this confirms that the stability observed in Chapter 2 is not an artifact of the small swelling limit.

Next, in Chapter 4 we shift our focus to a different problem - the various forms of nanoscaled surface morphology over time, such as ripples, dots, cones, sawtooths, and terraces. Throughout the last several decades, researchers have sought to understand the underlying mechanics that are driving these surface deformations. Grazing-Incidence Small-Angle X-ray Scattering (GISAXS) is an experimental scattering technique at grazing incidence angles that aids in the study of these surface structures. An illustration of GISAXS experimentation is shown in Figure 1.7 from Yamamoto’s work [41]. Normally (without GISAXS),
experimentalists bombard the sample for a period of time, remove the sample from the chamber to microscopically investigate the surface morphology, then return it to the chamber for continued bombardment. This repeated “cook and look” technique requires multiple movements of the sample. Now with experiments performed with GISAXS, x-rays are run through the chamber and scattered to an area detector to determine surface behavior. This allows for “real time” results without having to remove the sample from the chamber. However, GISAXS scattering only reveals the structure factor instead of an image of the surface itself. Early work was performed in the linear regime, where the structure factor resulting from the GISAXS scattering can be directly compared to mathematical models by computing the squared amplitude of the Fourier Transform of the surface height. Figure 1.8 (from work by Norris, Perkinson, Mokhtarzadeh, Anzenberg, Aziz, and Ludwig [28]) shows an example of how mathematical models fit the GISAXS data well in the linear regime. However, recent GISAXS studies indicate the presence of higher-amplitude target structures, implying the failure of underlying assumptions in the linear regime. In this nonlinear regime, GISAXS scattering no longer equals the structure factor, so we are unable to directly compare this data to mathematical models. We choose to explore if there is anything we can learn from comparing nonlinear regime experiments to nonlinear simulations and simulated GISAXS. Specifically, we are interested in the emergence of sawtooth structures on the target. Entering the nonlinear regime requires models to now include nonlinear terms. We adopt and expand the higher-order anisotropic Kuramoto-Sivashinsky (aKS) equation as our nonlinear model and use PETSc, paired with Southern Methodist University’s high performance computer, ManeFrame, to solve this equation. After generating the resulting surface plots over time for appropriate parameter values, we see the presence of sawtooth structures. We then run simulated GISAXS across these surfaces and compare to experimental GISAXS results. We find that our simulated GISAXS plots capture several experimental features, such as sawtooths, chevrons, and asymmetry.

Finally, in Chapter 5 we explore strain at the atomistic level to inform models based on stress and swelling. Atoms often naturally arrange themselves in a crystal lattice, a
Figure 1.7. A schematic of GISAXS experimentation. The advantage of GISAXS is that one can obtain “real time” results of surface behavior during ion bombardment, but the cost is that GISAXS only provides the structure factor. This schematic is from Yamamoto’s work [41].

stable structure in which all of the atoms are neatly aligned at the same critical distance apart. These atoms are hit by incoming ions during irradiation, disrupting the lattice and balance between attracting atoms. Strain is introduced into the system by this departure from the relaxed crystallized state. Molecular Dynamics (MD) is an accurate, but costly way to study the displacement of these atoms. The Binary Collision Approximation (BCA) is cheaper and accurate enough for what we are interested in studying: if stress-free strain rates can be obtained from the BCA. We determine the single impact vector field, then calculate the velocity field from the displacement field. Next, we calculate the strain tensor field and average over many nearby impact before we add isotropic expansion to counteract ion-induced compression. The zeroth moment of the resulting deviatoric strain tensor is calculated and plotted alongside the incidence angle, yielding angle-dependence of the deviatoric strain tensor. We compare results to Norris’ work in [27], especially at low angles. We notice that our off-diagonal values are nonzero, which differs from Norris’ results. This could be due to our ordering of the calculations of the deviatoric strain tensor. But we do
Figure 1.8. A comparison of GISAXS data and mathematical models in the linear regime, where the data fits experimentation. These images are from work by Norris, Perkinson, Mokhtarzadeh, Anzenberg, Aziz, and Ludwig [28] (licensed under CC BY 4.0 [9]).
see agreement with the \( \{1, 1, -2\} \)-like pattern along the main diagonal and have calculated stress-free strain rates from the BCA.

This analytical and numerical study of ion bombardment aims to increase our knowledge and understanding of the underlying mechanics surrounding irradiation, especially as it pertains to surface pattern formation, as well as strengthen the relationships between models, simulations, and experiments.
2.1. Introduction

Various patterned structures can form on the surface of a bombarded material. At the irradiation energies typical of many existing experiments – i.e. at 1 keV and below – the pattern formation behavior depends strongly on irradiation angle $\theta$, but depends very little on irradiation energy $E$ [18]. In particular, such experiments generally exhibit a progression, as $\theta$ increases, from flat surfaces, to ripples with wave vector parallel to the direction of the ion beam, to ripples with wave vector perpendicular to the direction of the ion beam, regardless of the ion energy. However, limited experiments suggest that energy becomes more important for energies in the tens of keV range, and suggest that at some energies, ripples vanish for all incidence angles [17]. This is somewhat puzzling, as all the physical mechanisms described above are destabilizing for at least some values of $\theta$. This suggests the possible presence of a missing, unconditionally stable mechanism that becomes important above a critical value of the ion energy.

As noted above, a significant amount of recent work has considered the effect of ion-induced stress, and found it to be a significant contributor to pattern formation [7,8,22,29]. Indeed, recent GISAXS studies of 1 keV Ar+ bombardment of Si suggest that it may be the strongest contributing factor to determining the presence or absence of patterns [28]. However, all of these studies assume the irradiated thin film to be incompressible, and employ slight modifications of the incompressible Stokes equations to describe the film. At higher energies, these assumptions may no longer be justified. In Figure 1.6, from work by Chini et al. [11], cross-section TEM images clearly reveal the presence of bubbles or voids in Si.
irradiated by Ar$^+$ at 50 keV. These voids are small near the bottom of the film, but increase steadily in size toward the top of the film. Because position in an eroding film is correlated with time spent in the film, this observation implies that the voids grow in time, increasing in size the longer they exist within the irradiated film. Moreover, the significant size of these voids by the time the eroding surface reaches them suggests that significant swelling occurs within the film, violating the incompressibility assumption used in prior studies. The combined observations of puzzling experimental results on pattern formation, and a physical mechanism missing from prior models, naturally suggests a study of the latter as a possible explanation for the former.

In this chapter, we derive governing equations consistent with the physics of ion-induced swelling, and perform a linear stability analysis to understand the effect of swelling on surface stability. In the process, we find the full equations preclude analytical solution even in the small-amplitude limit, so we further simplify the problem by studying the limit of small expansion rate. In this limit, intriguingly, the linear growth rate due to swelling can change sign as a function of wavenumber. However, it is stabilizing at all angles for wavenumbers typically observed experimentally, and should become increasingly important as the ion energy increases. We therefore conclude that it may play an important role in the puzzling disappearance of patterns at higher energies.

2.2. Model and Non-Dimensionalization

2.2.1. Model

We consider a crystalline material that is irradiated from above by an ion beam. The coordinate $z$ points upward, toward the ion source, and the top surface of the film is described by the function $z = h_2(x, y)$. Ions are embedded into the material over a characteristic length, $h_0$, causing the top layer of the crystal to become amorphous. This amorphization then requires the introduction of a crystal/amorphous boundary at $z = h_1(x, y)$, where
Figure 2.1. A schematic illustration of our problem geometry, for the case of normal incidence irradiation. Here $h_0$ is the film depth (typically twice the characteristic distance that the ions are embedded into the material). As the bombardment process continues, ions are sputtered away from the surface, causing the top layer $h_2$ to recede at speed $V$. Note that the ion beam is perpendicular to the surface of the material and that as $h_2$ recedes and moves down into the amorphous layer, $h_1$ will also move down into the crystalline substrate at the same rate.
\[ h_1 = h_2 - h_0. \] Ions are sputtered away over time, causing the amorphous/vacuum boundary, \( h_2 \), to erode downward into the amorphous layer at a speed

\[ V = f \cos \theta Y(\theta) \Omega, \]

where \( f \) is the flux through a plane perpendicular to the beam, \( Y(\theta) \) is the incidence angle-dependent sputter yield, and \( \Omega \) is the atomic volume of the target atoms. Consequently, \( h_1 \) must also recede, to maintain a thickness of \( h_0 \). As a result, more of the crystalline material becomes part of the amorphous layer, and the two free boundaries recede together. This situation is illustrated schematically in Figure 2.1 for normal incidence irradiation, in which case the crystal/amorphous boundary, \( h_1 \), is an exact downward translation of the amorphous/vacuum boundary \( h_2 \). However, in the analysis that follows, we separately keep track of \( h_1 \) and \( h_2 \), allowing the consideration of non-normal incidence in Section 2.4.

**Bulk Equations.** In the bulk, we begin with the standard equations of conservation of mass and conservation of momentum in the viscous limit of Stokes’ flow:

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (2.1)
\]

\[
-\nabla p + \eta (\nabla \cdot (\nabla \mathbf{v}) + \nabla (\nabla \cdot \mathbf{v})) = 0, \quad (2.2)
\]

where \( \rho \) is the density of the amorphous layer, \( \mathbf{v} = \langle u, v, w \rangle \) is the velocity of the fluid, \( p \) is the pressure, and \( \eta \) is the kinematic viscosity. The left-hand side of Eq. (2.2) is the divergence of the stress tensor

\[
\mathbf{T} = -p \mathbf{I} + \eta (\nabla \mathbf{v} + (\nabla \mathbf{v})^T). \quad (2.3)
\]

Although the stress tensor traditionally includes a divergence term, we assume that our system behaves like a monatomic fluid with no contribution from this term, because the
bulk viscosity is negligible. Therefore, we have not included it in our definition of the stress
tensor. Note that, because we wish to describe swelling, which causes material expansion, we
have not assumed that $\rho$ is constant, nor have we employed the simplifications to Eqs. (2.1)-(2.2)
enabled by that assumption in the case of an incompressible fluid. Instead, we must
introduce a equation of state that accounts for swelling. As a simplest choice, we choose the
relation

$$\rho = \frac{\rho^*_0}{1 + \alpha a}. \quad (2.4)$$

Here $\rho^*_0$ is the original density of the crystalline solid, $\alpha$ is a constant expansion rate, and
$a$ is an age describing the accumulated time that a given parcel of material has been in the
amorphous layer and subject to irradiation damage. This statement reflects a very simple
model of swelling, in which volume is added to the amorphous film uniformly at a constant
rate $\alpha$. Finally, the age-tracking variable $a$ obeys a simple advection equation

$$\frac{\partial a}{\partial t} + \mathbf{v} \cdot \nabla a = 1, \quad (2.5)$$

where the right-hand side of Eq. (2.5) equals one to reflect the increasing age of amorphous
material parcels as they move about the film.

We pause briefly to discuss the equation of state, Eq. (2.4), which represents the most
significant mathematical assumption of this work. In a full treatment of compressibility,
Eq. (2.4) would normally include an additional term $\beta (p - p_0)$ on the right-hand side to
account for density changes due to pressure. This term is omitted for mathematical simplic-
ity, because it frustrates analytical solution of the resulting model. The resulting simplified
model could be said to exhibit “irresistible swelling” or “quasi-incompressibility” – the den-
sity of the material directly and irresistibly increases due to irradiation-induced swelling,
but otherwise behaves as in an incompressible fluid by ignoring changes in pressure. Beyond
the obvious benefit of enabling analytical solutions, this simplification allows us to focus
on the consequences of swelling rather than the precise mechanism driving it, and seems appropriate for a first study.

**Boundary Conditions.** Now we need to impose boundary conditions at the top and bottom of the film. These come from the conservation of mass and momentum criteria imposed at each boundary, plus a boundary condition on the age variable. At the crystalline/amorphous boundary, where \( z = h_1(x, y, t) \):

\[
\begin{align*}
a & = 0 \quad \text{(2.6)} \\
v & = 0 \quad \text{(2.7)}
\end{align*}
\]

The age variable, \( a \), is zero, since the material at the crystalline/amorphous boundary has not spent any previous time in the amorphous layer. The velocity, \( v \), is zero at this boundary due to no slip and no penetration conditions enforced in the system.

At the amorphous/vacuum boundary, where \( z = h_2(x, y, t) \):

\[
\begin{align*}
v \hat{n} & = v \cdot \hat{n} - \frac{V \rho v^*}{\rho} \\
[T] \cdot \hat{n} & = -\gamma \kappa \hat{n}, \quad \text{(2.9)}
\end{align*}
\]

where \( \gamma \) is the surface tension coefficient, \( \kappa \) is the curvature, and \([T]\) is the jump in the stress tensor. Here Eq. (2.8) is a modified kinematic condition obtained from the conservation of mass applied at the free surface – when material is lost due to sputtering at the top boundary, the interfacial velocity no longer equals the fluid velocity (see Appendix A for a brief derivation; we note that the dependence on density in the final term comes from the assumption that the sputter yield does not depend on the density at the surface of the amorphous film). Finally, Eq. (2.9) is the standard stress balance condition, obtained from
the conservation of momentum. In what follows, because the effect of surface tension is well-understood [33, 39], we will set $\gamma = 0$ to more clearly focus on the effect of swelling.

### 2.2.2. Conversion to a Moving Frame and Non-Dimensionalization

As illustrated in our model, Figure 2.1, the interfaces $h_2(x, y)$ and $h_1(x, y)$ translate downward due to sputtering at the free surface. We assume that in a suitable steady state, this occurs at a constant rate: $h_2(x, y) = h_2(x, y, t) - Vt$ (though see [5] as an interesting counter-example in a more complicated system). We therefore adopt a frame of reference following $h_2(x, y, t)$:

$$
\begin{align*}
    h_1 &\to h_1 - Vt \\
    h_2 &\to h_2 - Vt \\
v_n &\to v_n - V\left(\hat{k} \cdot \hat{n}\right) \\
z &\to z - Vt \\
v &\to v - V\hat{k}.
\end{align*}
$$

(2.10)

We also reduce the number of parameters in the problem by non-dimensionalization. We employ the following scalings:

$$
\begin{align*}
x &\to h_0\tilde{x} \\
\nabla &\to \frac{1}{h_0}\tilde{\nabla} \\
v &\to V\tilde{v} \\
\rho &\to \rho^*\tilde{\rho} \\
a &\to \frac{1}{\alpha} \tilde{a} \\
p &\to \frac{\eta V}{h_0}\tilde{p} \\
t &\to \frac{h_0}{V}\tilde{t} \\
v_n &\to V\tilde{v}_n.
\end{align*}
$$

(2.11)
These two operations produce the resulting non-dimensionalized bulk equations:

\[
\frac{\partial \bar{\rho}}{\partial \bar{t}} + \nabla \cdot (\bar{\rho} \bar{v}) = 0 \quad (2.12)
\]

\[-\nabla \bar{p} + (\nabla^2 \bar{v} + \nabla (\nabla \cdot \bar{v})) = 0 \quad (2.13)\]

\[\frac{\partial \bar{a}}{\partial \bar{t}} + \bar{v} \cdot \nabla \bar{a} = A \quad (2.14)\]

\[\bar{\rho} = \frac{1}{1+\bar{a}}. \quad (2.15)\]

Here we encounter the dimensionless parameter

\[\frac{\alpha h_0}{V} = A, \quad (2.16)\]

which represents the characteristic volumetric expansion of the material, since \(\alpha\) is the expansion rate, and \(\frac{h_0}{V}\) is the dwelling rate of the material in the amorphous layer.

At the boundaries, the dimensionless version of the boundary conditions become

\[\bar{a} = 0 \quad (2.17)\]

\[\bar{v} = \hat{k} \quad (2.18)\]

at \(\bar{z} = \bar{h}_1 (\bar{x}, \bar{y}, \bar{t})\) and

\[\bar{v}_n = \bar{v} \cdot \hat{n} - \frac{1}{\bar{\rho}} \quad (2.19)\]

\[\mathbf{T} \cdot \hat{n} = 0 \quad (2.20)\]

at \(\bar{z} = \bar{h}_2 (\bar{x}, \bar{y}, \bar{t})\).
2.3. Analysis

We now proceed to analyze the model obtained above, leading ultimately to a linear stability analysis giving the growth rate $\sigma (k)$ of small surface perturbations with wavenumber $k$.

2.3.1. Steady State Solution

We begin our analysis by looking for a steady solution, associated with flat boundaries $h_1$ and $h_2$. The steady hypothesis implies that any solution will be time invariant ($\frac{\partial}{\partial t} \to 0, v_\bar{n} \to 0$). The flatness hypothesis implies that this solution should be translation invariant ($\frac{\partial}{\partial x} \to 0, \frac{\partial}{\partial y} \to 0$). Finally, the isotropic nature of the swelling effect implies that the solution should be rotation invariant ($u = v = 0$). This produces even more simplified versions of equations (2.12)-(2.15), with the form

$$\frac{\partial}{\partial \bar{z}} (\bar{\rho}_0 \bar{w}_0) = 0 \quad (2.21)$$

$$- \frac{\partial \bar{p}_0}{\partial \bar{z}} + 2 \frac{\partial^2 \bar{w}_0}{\partial \bar{z}^2} = 0 \quad (2.22)$$

$$\bar{w}_0 \frac{\partial \bar{a}_0}{\partial \bar{z}} = A \quad (2.23)$$

$$\frac{1}{1 + \bar{a}_0} = \bar{\rho}_0, \quad (2.24)$$

together with further simplified boundary conditions

$$\bar{a}_0 = 0 \quad (2.25)$$

$$\bar{w}_0 = 1 \quad (2.26)$$

at $\bar{z} = 0$, and
\begin{align}
0 &= \bar{w}_0 - \frac{1}{\bar{\rho}_0} \tag{2.27} \\
-\bar{p}_0 + 2 \frac{\partial \bar{w}_0}{\partial \bar{z}} &= 0 \tag{2.28}
\end{align}

at \( \bar{z} = 1 \). Equations (2.21)-(2.28) are solved by (a) integrating Eq. (2.21) to obtain \( \bar{w}_0 \) in terms of \( \bar{\rho}_0 \), (b) using Eq. (2.24) to express \( \bar{w}_0 \) in terms of \( \bar{a}_0 \), (c) inserting this result into Eq. (2.23) and solving the resulting ODE for \( \bar{a}_0 \), thus also obtaining \( \bar{\rho}_0 \) and \( \bar{w}_0 \), and finally (d) inserting \( \bar{w}_0 \) into Eq. (2.22) and integrating. This produces the following steady state solutions:

\begin{align}
\bar{\rho}_0(\bar{z}) &= \frac{1}{\sqrt{1 + \frac{2}{c}(A\bar{z} + d)}} \tag{2.29} \\
\bar{w}_0(\bar{z}) &= c \sqrt{1 + \frac{2}{c}(A\bar{z} + d)} \tag{2.30} \\
\bar{p}_0(\bar{z}) &= \bar{p}_{00} + 2 \frac{\partial \bar{w}_0}{\partial \bar{z}} \tag{2.31} \\
\bar{a}_0(\bar{z}) &= -1 + \sqrt{1 + \frac{2}{c}(A\bar{z} + d)} \tag{2.32}
\end{align}

where \( c \) and \( d \) are integration constants, which can be determined by enforcing the boundary conditions. Applying these boundary conditions to the above equations result in the steady state solutions

\begin{align}
\bar{\rho}_0(\bar{z}) &= \frac{1}{\sqrt{1 + 2A\bar{z}}} \tag{2.33} \\
\bar{w}_0(\bar{z}) &= \sqrt{1 + 2A\bar{z}} \tag{2.34} \\
\bar{p}_0(\bar{z}) &= \frac{2A}{\sqrt{1 + 2A\bar{z}}} \tag{2.35} \\
\bar{a}_0(\bar{z}) &= -1 + \sqrt{1 + 2A\bar{z}}. \tag{2.36}
\end{align}
These solutions are plotted in Figure 2.2. In the moving frame, the material enters the amorphous layer with a fixed density, speed, and pressure. As the material moves toward the surface, the density decreases, the speed increases, and the pressure decreases. Additionally, we can observe that the age of the material is zero at the crystalline/amorphous boundary and increases as it approaches the top.

2.3.2. Linear Stability Analysis

To study ripple formation, we now proceed to linearize the full governing equations (2.12)-(2.20) about the steady solution (2.33)-(2.36). We let \( \Phi = [\bar{u}, \bar{w}, \bar{p}, \bar{a}, \bar{h}_1, \bar{h}_2] \) and perform the following expansion

\[
\Phi(\bar{x}, \bar{z}, \bar{t}) = \Phi_0(\bar{z}) + \varepsilon \Phi_1(\bar{z}) e^{ikx + \sigma t} = \Phi_0(\bar{z}) + \varepsilon \Phi_1(\bar{z}) e^{iQx + \Sigma t} \tag{2.37}
\]

where \( \varepsilon \) is a small positive number, \( k \) is the wavenumber of a Fourier mode, and \( \sigma (k) \) is its corresponding growth rate. Because the swelling mechanism is isotropic, it is sufficient to consider perturbations in only one direction, and we choose the \( x \)-direction. We also non-dimensionalize the new parameters \( k \) and \( \sigma \) via

\[
Q = h_0k \quad \Sigma = \frac{\sigma h_0}{\bar{v}}. \tag{2.38}
\]

Inserting this ansatz into Eqs. (2.12)-(2.20) and keeping terms to leading order in \( \varepsilon \), we obtain linearized equations:
Figure 2.2. Plots of the steady state solutions, Eqs. (2.33)-(2.36), where $A\bar{z} \in [0, 10]$ is an arbitrary domain chosen to illustrate qualitative behavior. The density (a) and pressure (b) of the material decrease as the material moves toward the surface. (c) In the moving frame of reference, the velocity is constant when material first enters the amorphous layer, then increases under continual swelling as material moves toward the surface. (d) The age of the material is zero and increases until parcels reach the surface and are sputtered away.
\[ \Sigma \tilde{\rho}_1 - A \frac{1}{\psi^3} \tilde{w}_1 + \frac{1}{\psi} (iQ \tilde{u}_1 + \tilde{w}'_1) + \psi \tilde{p}'_1 + A \frac{1}{\psi} \tilde{\rho}_1 = 0 \]  
\[ \Sigma \tilde{a}_1 + \psi \tilde{a}'_1 + A \frac{1}{\psi} \tilde{w}_1 = 0 \]  
\[ \psi \tilde{p}_1 + \frac{1}{\psi} \tilde{a}_1 = 0 \]  
\[ -2Q^2 \tilde{u}_1 + \tilde{u}''_1 + iQ \tilde{w}'_1 = iQ \tilde{p}_1 \]  
\[ -Q^2 \tilde{w}_1 + 2 \tilde{w}''_1 + iQ \tilde{u}'_1 = \tilde{p}'_1, \]  
\[ (2.39) \]
\[ (2.40) \]
\[ (2.41) \]
\[ (2.42) \]
\[ (2.43) \]

where \( \psi = \sqrt{1 + 2A \bar{z}} \). The unknowns are subject to similarly linearized boundary conditions

\[ \bar{a}_1 = -\bar{h}_{11} A \]  
\[ \bar{u}_1 = 0 \]  
\[ \bar{w}_1 = -\bar{h}_{11} A \]  
\[ (2.44) \]
\[ (2.45) \]
\[ (2.46) \]

at \( \bar{z} = 0 \) and

\[ \bar{w}_1 = \Sigma \bar{h}_{21} - (1 + 2A) \bar{\rho}_1 \]  
\[ \bar{u}'_1 + iQ \bar{w}_1 = -2iQ \frac{A}{\sqrt{1 + 2A}} \bar{h}_{21} \]  
\[ -\bar{p}_1 + 2 \bar{w}'_1 = 0 \]  
\[ (2.47) \]
\[ (2.48) \]
\[ (2.49) \]

at \( \bar{z} = 1 \).

2.3.3. Small-A limit.

We find that even the linearized Eqs. (2.39)-(2.49) are not analytically solvable in closed form (they lead to an integro-differential equation with variable coefficients). Therefore, we consider a further mathematical simplification, \( A << 1 \). This is the limit in which the implantation-induced stress is still small by the time bulk parcels are sputtered away from
the free interface. We note that if $A = 0$, then no stress is being generated, and therefore we expect no material evolution whatsoever; indeed, it is easy to verify that Eqs. (2.39)-(2.49) have zero as a solution when $A = 0$. Therefore we need to obtain the solution at $\mathcal{O}(A)$. We write

$$\Phi_1 = 0 + A\Phi_{11} + \mathcal{O}(A^2),$$

(2.50)

where each variable has higher-order contributions in $A$ that have been neglected (note that $\bar{h}_{11}$ and $\bar{h}_{21}$ are not expanded in $A$, because the height perturbations are not dependent on the swelling rate). Additionally, $\Sigma$, which is not included in $\Phi$, has the following expansion: $\Sigma = 0 + A\Sigma_1 + \mathcal{O}(A^2)$. This leads to the further simplified system of equations

$$iQ\bar{u}_{11} + \bar{w}_{11}' + \bar{\rho}_{11}' = 0$$

(2.51)

$$\bar{a}_{11}' = 0$$

(2.52)

$$\bar{\rho}_{11} + \bar{a}_{11} = 0$$

(2.53)

$$-2Q^2\bar{u}_{11} + \bar{u}_{11}'' + iQ\bar{w}_{11}' = iQ\bar{p}_{11}$$

(2.54)

$$-Q^2\bar{w}_{11} + 2\bar{w}_{11}'' + iQ\bar{a}_{11}' = \bar{p}_{11}'$$

(2.55)

and boundary conditions

$$\bar{a}_{11} = -\bar{h}_{11}$$

(2.56)

$$\bar{u}_{11} = 0$$

(2.57)

$$\bar{w}_{11} = -\bar{h}_{11}$$

(2.58)

at $\bar{z} = 0$, and
\[ w_{11} = \sum \bar{h}_{21} - \bar{\rho}_{11} \]  
(2.59)

\[ \bar{u}'_{11} + iQ\bar{w}_{11} = -2iQ\bar{h}_{21} \]  
(2.60)

\[ -\bar{p}_{11} + 2\bar{w}'_{11} = 0 \]  
(2.61)

at \( \bar{z} = 1 \).

2.3.4. Solution.

To solve Eqs. (2.51)-(2.61), we first use Eqs. (2.52), (2.53) and (2.56) to show that

\[ \bar{a}_{11} = -\bar{h}_{11} \]  
(2.62)

\[ \bar{\rho}_{11} = \bar{h}_{11} \]  
(2.63)

We now must use Eqs. (2.51)-(2.55) to obtain a solution for \( \bar{u}_{11}, \bar{w}_{11}, \) and \( \bar{p}_{11} \). Our first step will be to use Eqs. (2.54) and (2.55) to obtain \( \bar{u}_{11} \) and \( \bar{w}_{11} \) in terms of \( \bar{p}_{11} \) by treating the latter as if it were a known forcing function. Particular solutions associated with the forcing by \( \bar{p}_{11} (\bar{z}) \) are obtained using Laplace Transforms. To find homogeneous solutions, we re-cast the equations as a system of four first-order ODEs. Two solutions to this system can be found using eigenvalue analysis, and the other two can be found using techniques from the study of dynamical systems in the case of repeated eigenvalues. The resulting solution for
\{\bar{u}_{11}, \bar{w}_{11}\} \text{ in terms of } \bar{p}_{11} \text{ is}

\[
\begin{bmatrix}
\bar{u}_{11} \\
\bar{w}_{11}
\end{bmatrix} =
\begin{bmatrix}
\bar{a} \\
\bar{c}
\end{bmatrix}
\cosh( Q \bar{z}) +
\begin{bmatrix}
\bar{b} \\
\bar{d}
\end{bmatrix}
\sinh( Q \bar{z}) +
\bar{z}
\begin{bmatrix}
\frac{Q}{3} (\bar{b} - i \bar{c}) \cosh( Q \bar{z}) + \frac{Q}{3} (\bar{a} - i \bar{d}) \sinh( Q \bar{z}) \\
-\frac{Q}{3} i (\bar{a} - i \bar{d}) \cosh( Q \bar{z}) - \frac{Q}{3} i (\bar{b} - i \bar{c}) \sinh( Q \bar{z})
\end{bmatrix}
+ \frac{i}{2} \int_0^\bar{z} \bar{p}_{11}(\bar{\zeta}) \sinh( Q (\bar{z} - \bar{\zeta})) d\bar{\zeta}
+ \frac{1}{2} \int_0^\bar{z} \bar{p}_{11}(\bar{\zeta}) \cosh( Q (\bar{z} - \bar{\zeta})) d\bar{\zeta},
\]

(2.64)

where \(\bar{a}, \bar{b}, \bar{c}, \text{ and } \bar{d}\) are potentially complex constants. We now insert these expressions for \(\bar{u}_{11}\) and \(\bar{w}_{11}\) into Eq. (2.51), solve the resulting equation in \(\bar{p}_{11}(\bar{z})\), and insert the solution back into Eqs. (2.64), obtaining

\(\bar{u}_{11}(\bar{z}) = \bar{a} \cosh( Q \bar{z}) + \left(\frac{2}{3} \bar{b} + \frac{i}{3} \bar{c}\right) \sinh( Q \bar{z}) + \frac{2}{3} Q \bar{z} \left[ (\bar{b} - i \bar{c}) \cosh( Q \bar{z}) + (\bar{a} - i \bar{d}) \sinh( Q \bar{z}) \right] \)

(2.66)

\(\bar{w}_{11}(\bar{z}) = \bar{c} \cosh( Q \bar{z}) + \left(\frac{2}{3} \bar{d} - \frac{i}{3} \bar{a}\right) \sinh( Q \bar{z}) - \frac{2}{3} Q \bar{z} \left[ (\bar{d} + i \bar{a}) \cosh( Q \bar{z}) + (\bar{c} + i \bar{b}) \sinh( Q \bar{z}) \right] \)

(2.67)

We must next apply the boundary conditions. Equations (2.57)-(2.58) immediately yield

\(\bar{a} = 0\)

\(\bar{c} = -\bar{h}_{11}\).
Then, expressions for $\bar{b}$ and $\bar{d}$ are obtained from Eqs. (2.60)-(2.61); this results in the matrix system

$$
\begin{bmatrix}
\cosh(Q) + Q \sinh(Q) & -iQ \cosh(Q) \\
iQ \cosh(Q) & \cosh(Q) + Q \sinh(Q)
\end{bmatrix}
\begin{bmatrix}
\bar{b} \\
\bar{d}
\end{bmatrix}
= \begin{bmatrix}
(\frac{i}{2} \cosh(Q) - iQ \sinh(Q))\bar{h}_{11} - \frac{3}{2}i\bar{h}_{21} \\
(\frac{3}{2} \sinh(Q) - Q \cosh(Q))\bar{h}_{11}
\end{bmatrix}.
$$

Solving using Cramer’s Rule, we obtain values of

$$
\begin{align*}
\bar{b} &= \frac{i(-Q^2 + \frac{1}{2} \cosh^2(Q))\bar{h}_{11} - i(\frac{3}{2} \cosh(Q) - \frac{3}{2}Q \sinh(Q))\bar{h}_{21}}{Q^2 + \cosh^2(Q)} \\
\bar{d} &= \frac{(-\frac{3}{2}Q + \frac{3}{2} \sinh(Q) \cosh(Q))\bar{h}_{11} + (\frac{3}{2}Q \cosh(Q))\bar{h}_{21}}{Q^2 + \cosh^2(Q)}.
\end{align*}
$$

Finally, the value of $\Sigma_1$ is obtained from Eq. (2.59). Inserting values for $\bar{a}, \bar{b}, \bar{c},$ and $\bar{d}$ into Eq. (2.67), and then inserting Eqs. (2.63), (2.67) into Eq. (2.59), we obtain an expression for the dimensionless growth rate

$$
\Sigma_1 = \left(1 - \frac{\cosh(Q) + Q \sinh(Q)}{Q^2 + \cosh^2(Q)}\right)\frac{\bar{h}_{11}}{\bar{h}_{21}} - \frac{Q^2}{Q^2 + \cosh^2(Q)}.
$$

(2.68)

This result is plotted in Figure 2.3 for the case $\bar{h}_{11} = \bar{h}_{21}$, corresponding to a vertically-displaced bottom boundary expected during normal-incidence irradiation.

### 2.4. Discussion

**Intuitive Interpretation.** The shape of $\Sigma_1(Q)$ is rather unusual in that it changes sign – for $Q < 1.5$, the swelling mechanism is stabilizing, whereas for $Q > 1.5$, it is destabilizing. We will therefore first present in Figure 2.4 an intuitive, graphical exploration of the physics behind each regime. Before we begin, we will note that, in the limit $Q \to 0$, the velocity
Figure 2.3. A plot of the dimensionless growth rate $\Sigma_1$ given in Eq. 2.68, for the case $\bar{h}_{11} = \bar{h}_{21}$ (normal incidence). For $Q \lesssim 1.5$, the negative sign of $\Sigma_1$ indicates that swelling is stabilizing. For $Q \gtrsim 1.5$, the positive sign of $\Sigma_1$ indicates that it becomes destabilizing. Though not pictured here, $\Sigma_1$ saturates at a value of 1 as $Q \to \infty$.

The field $v_1(\bar{x}, \bar{z})$ assumes a very simple form:

$$\lim_{Q \to 0} v_1(\bar{x}, \bar{z}; Q) = -\bar{h}_{11}(x) \hat{k}.$$  

(2.69)

This term acts simply to ensure that the no-penetration condition at $\bar{z} = \bar{h}_{11}$ remains satisfied under the perturbation to the bottom surface. When $\bar{h}_{21} = \bar{h}_{11}$, it also yields a constant vertical velocity at the free interface $\bar{z} = 1 + \bar{h}_{21}$, which is neutrally stable (indeed, $\Sigma_1(0) = 0$). Now, for $Q > 0$, the flow field still strongly resembles that in Eq. (2.69), which makes visualizations of $v_1$ rather hard to interpret. Therefore, in Figure 2.4 we present quiver plots of the quantity $v_1(\bar{x}, \bar{z}; Q) - v_1(\bar{x}, \bar{z}; 0)$; this relative flow field isolates the components of the flow field contributing to stability or instability.

In Figure 2.4a, we illustrate the relative flow field for $Q = 0.7$. This value is near to its most stabilizing state (where $\Sigma_1$ is most negative), and reveals clearly a relative flow away from the hilltops and toward the valleys. This behavior may be attributed simply to the availability of lateral expansion at the hilltops: in valleys, the only available direction for expansion is upwards, whereas on hilltops both upward and lateral expansion are available. Consequently the valleys catch up to the hills. In Figure 2.4b, we illustrate the relative flow...
Figure 2.4. Visualizations of the relative flow field $v_{11}(\bar{x}, \bar{z}; Q) - v_{11}(\bar{x}, \bar{z}; 0)$ for $Q = \{0.7, 1.5, 2.0\}$. (a) For $Q = 0.7$, the availability of lateral expansion at the hilltops induces flow from hilltops into valleys, which is stabilizing. (b) For $Q = 1.5$, this former mechanism is essentially offset by shear forces pulling material originating in valleys uphill. (c) For $Q = 2.0$, shear forces have become stronger than the lateral hilltop expansion mechanism, and the result is destabilizing.
field for $Q = 1.5$, near the transition point. Here we see the relative flow field beginning to develop regions of circulation. These are caused by shear: because the material underneath valleys is amorphized sooner that the material underneath hilltops, the vertical velocity below a valley at any value of $\bar{z}$ is larger than the corresponding velocity below a hilltop. As $Q$ increases, these columns with different velocities approach each other, and the resulting shear forces pull material away from the fast-moving columns below valleys toward the slow-moving columns below hilltops. For $Q = 1.5$, this new sub-surface effect nearly balances the effect of lateral swelling on hilltops, and the surface is neutrally stable. Finally, in Figure 2.4c, we illustrate the relative flow field for $Q = 2.0$. By now, the valleys (exhibiting fast flow) have approached near enough to the hilltops (exhibiting slow flow) that the regions of circulation are well-developed. Although lateral swelling remains available near the hilltops, the effect of sub-surface shear forces has strengthened to the point that the net effect on the surface is destabilizing.

The discovery of a fundamental competition between lateral swelling and shear-induced vorticity governing the behavior observed in Figures (2.3)-(2.4) casts light on the appropriateness of our central simplification of “quasi-incompressibility.” The higher material velocities beneath valleys is a consequence of the basic geometry of the system, and the tendency of shear to bend fast-moving streamlines toward slower neighbors is a basic property of viscous fluids. So neither mechanism should be fundamentally altered by the introduction of a pressure term in Eq. (2.4).

**Angle Dependence.** In the discussion to this point, we have for simplicity considered only the case of normal-incidence irradiation. However, for an isotropic mechanism such as the one we have presented, the only difference between the case of normal and off-normal incidence is the relative location of the top and bottom boundaries. Because our analysis did distinguish between these boundaries, we can now retroactively explore the effect of dependence on the ion incidence angle. We shall assume that for a non-zero incidence angle $\theta$, the bottom boundary is simply shifted a distance $h_0$ in the direction of the ion beam, rather
Figure 2.5. A 2D heatmap plot of $\Sigma_1 (Q; \theta)$. Blue indicates stability, red indicates instability, and grey indicates neutral stability. The dashed line is a guide to the eye, and separates a domain of unconditional stability on the left, from a domain of alternating stability and instability on the right. Notably, long-wavelength perturbations are stabilized at all angles, and the range of stable wavenumbers grows with increasing $\theta$. 
than straight down. This produces a film with depth $d = h_0 \cos(\theta)$ and bottom boundary shifted laterally by a distance $w = h_0 \sin(\theta)$. If the top boundary is perturbed as before, then the perturbation to the bottom boundary is also shifted in this way, yielding values of $\bar{h}_1$ and $\bar{h}_2$ given by

$$h_2 = h_0 \cos(\theta) + \varepsilon e^{ikx+\sigma t}$$

$$h_1 = \varepsilon e^{ik(x-h_0 \sin(\theta))+\sigma t}.$$

Finally, because the film depth is now angle-dependent, we must retroactively re-scale $x$ and $k$ using $h_0 \cos(\theta)$ instead of $h_0$. Noting that $\bar{h}_{11}$ and $\bar{h}_{21}$ are the $O(\varepsilon)$ contributions to $\bar{h}_1$ and $\bar{h}_2$, we can obtain new definitions of $Q$ and $\bar{h}_{11} \bar{h}_{21}$ as follows:

$$Q_\theta = h_0 \cos(\theta) k$$

$$\frac{\bar{h}_{11}}{\bar{h}_{21}}|_{\theta} = e^{-iQ_\theta \tan(\theta)}. \hspace{1cm} (2.70)$$

By inserting expressions (2.70) into the right-hand side of Eq. (2.68), we obtain an equation for the growth rate $\Sigma_1$ that is dependent on both the wavenumber and the ion-beam incidence angle. Figure 2.5 shows a heatmap of $\Sigma_1(Q_\theta(Q,\theta))$ for various wavenumbers and incidence angles, exhibiting interesting changes in sign with both $Q$ and $\theta$.

**Implications for Experiments.** The changing sign of $\Sigma_1$ somewhat complicates a characterization of the expected effect of swelling in physical problems. We make two observations. First, it is clearly stabilizing in the longwave limit $Q \rightarrow 0$; indeed the mechanism exhibits leading-order behavior at $O(Q^2)$:

$$\lim_{Q \rightarrow 0} \Sigma_1(Q;\theta) = -\frac{1}{2} \cos^2(\theta) Q^2 + O(Q^4). \hspace{1cm} (2.71)$$

This indicates that it competes directly with mechanisms such as prompt atomic displacements [4, 6, 31] and anisotropic plastic flow [13, 29] to determine the behavior of long-wavelength perturbations, and stabilizes these wavelengths at all irradiation angles! Second, as $\theta$ increases, the strength of the swelling mechanism decreases, but the range of $Q$ over
which it remains stabilizing increases. In most experiments in the low keV range, ripples are not seen until the irradiation angle exceeds a critical value of around 45°. From Figure 2.5, we see that for angles above 45°, swelling remains stabilizing until $Q \approx 6$. This value is significantly higher than the wavenumber of ripples observed experimentally – often around $Q = 1$. Therefore, at experimentally-observed ripple wavenumbers, swelling is stabilizing for all angles, and should therefore inhibit the formation of ripples. We note that higher wavenumbers are stabilized by other mechanisms not considered here, notably ion-enhanced thin-film viscous flow [33,39], which exhibits increasingly negative values of $\sigma$ as $Q \to \infty$ [28], in contrast to the constant approached in the same limit by the swelling mechanism.

**Energy Dependence.** We conclude this section with some speculation on the nature of the energy dependence of this mechanism. Though we have not invoked a specific mechanism, we postulate that the increased importance of swelling in the range of tens of keV ion bombardment may be related to the time a given parcel of material spends in the amorphous layer, between its initial amorphization and its final sputtering. In short, the slower the erosion rate relative to the film depth, the longer the amorphous target has to collect defects and implanted ions that lead to bubble/void formation. This suggests computing the “dwell time”

$$\tau = \frac{h_0}{V}$$

where $h_0$ is the average film depth and $V$ is the erosion rate. This can be estimated up to a constant factor using Binary Collision Approximation (BCA) simulations, which we have preformed in TRI3DST [26] via the PyCraters library [27]. For a variety of ions and energies, we divided the expected film depth $h_0$ (taken to be the mean penetration depth of the ions plus two times the standard deviation) by the sputter yield (which, when multiplied by the flux and atomic volume, gives the erosion rate). The results, shown in Figure 2.6, reveal a minimum for each ion precisely in the range of energies where ripples are readily observed, with increases at both larger and smaller energies. Moreover, there is a clear dependence on ion mass, suggesting that lighter ions may lead to more significant swelling at lower energies.
than larger ions. We are aware of no published studies dedicated exploring energies below \( \approx 100 \text{ eV} \), nor the ion species dependence of ripple disappearance. Figure 2.6 may therefore suggest interesting topics of future research.

2.5. Conclusions

Although most existing models of ion beam induced nanostructure formation predict ripples for some angle of incidence of the incoming ions, experiments reveal energy ranges in which no structures are formed at any irradiation angle. We have hypothesized that a failure to account for material swelling, which is observed in these energy ranges, may explain this discrepancy between theory and experiment, and have therefore studied a simple model of swelling during irradiation. After performing a linear stability analysis in the limit of small expansion rates, we find that although the linear amplification rate exhibits interesting changes in sign, due to a competition between lateral expansion and shear forces, the swelling mechanism should be stabilizing for all angles at the wavenumbers for which ripples are observed in experiments. This strongly suggests that swelling could contribute to the disappearance of ripples at higher energies, and should therefore be a critical component of models attempting to describe observations at those energies.
Figure 2.6. A plot of film depth divided by sputter yield (which is proportional to the dwell time in Eq. (2.72)) for different ions and energies. The minimum of values align well with the range 250 eV – 2 keV in which the majority of studies on ripple formation have been performed.
Chapter 3

The Effect of Swelling on Ion-Induced Pattern Formation:

Numerical Implementation

3.1. Introduction

In the previous chapter, we analytically solved our simplified system, Eqs. (2.51) - (2.61), in which expansions in the material are small, and found the corresponding stability regime for experimentally observed wavelengths. This suggests that swelling may help account for the vanishing ripples that have been observed in experiments, but are not present in mathematical models. Ultimately, we want to know what happens to stability in our system as we allow the expansion rate to increase. This requires that we relax the previous simplification of small expansion, which brings us back to the original linearized bulk equations, Eqs. (2.39) - (2.49). Since these equations are analytically insolvable, we must implement a numerical study.

We choose to use the Portable, Extensible Toolkit for Scientific Computations (PETSc) library (version 3.8 [2], [3], [1]) to help us solve our original linearized system. PETSc is a useful tool for solving nonlinear PDEs, because it is already parallelized, handles periodicity, and can overwrite variables in the executable. It also performs root-finding methods and iterative methods more efficiently, since these costly solves are performed in parallel.

As it pertains to this work, we use PETSc to solve our nonlinear system using its built-in Scalable Nonlinear Equations Solver (SNES) routine. This feature performs a Newton iteration, and therefore requires routines to compute the nonlinear residual vector, the Jacobian matrix, and an initial guess. As a result, we must supply PETSc with our discretized right-hand-side function and an initial guess. Instead of creating our own discretized right-hand-side Jacobian, we choose to use the run-time option argument \texttt{--snes_fd_color}, which
automatically creates a Jacobian matrix by using finite differences with coloring. Briefly, this is a technique that maps each nonzero row entry to its corresponding nonzero column entries, then partitions the matrix by structurally orthogonal columns. Color assignments are used to group elements into a compressed column. From there, the Jacobian can be calculated with fewer functional evaluations than if it was calculated traditionally [15].

To begin, we solve the simplified system numerically with PETSc and sweep over appropriate wavenumber values to obtain our stability function. This results in a stability regime that agrees with the results from our analytical solution of the simplified system. Next, we implement the linearized equations and solve the system over the same range of wavenumbers, for a small value of the expansion rate parameter. These solutions agree with the results from both the numerical and analytical study of the simplified equations. Now that all of these solutions agree with each other, we increase our expansion parameter to observe what happens to surface stability outside of the small expansion simplification. We find that stability regimes are still present when allowing for increasing expansion rates in the system. Lastly, we perform convergence tests on the numerical methods and find first-order convergence for our values of $A$ and $Q$. These results support our investigation of swelling as a stabilizing mechanism and suggest that surface stability corresponding to the inclusion of swelling in the previously studied simplified system in Chapter 2 is not an artifact of the small expansion simplification.

### 3.2. Small-A Equations

We start with the simplified equations from our analytical study. We recall from Chapter 2 that these equations are for small expansion in our system, which is represented by
investigating behavior at $O(A)$. For convenience, we restate these small-A equations here:

\[
\begin{align*}
   iQ\bar{u}_{11} + \bar{\omega}'_{11} + \bar{\rho}'_{11} &= 0 \quad (3.1) \\
   \bar{a}'_{11} &= 0 \quad (3.2) \\
   \bar{\rho}_{11} + \bar{a}_{11} &= 0 \quad (3.3) \\
   -2Q^2\bar{u}_{11} + \bar{u}''_{11} + iQ\bar{w}'_{11} &= iQ\bar{p}_{11} \quad (3.4) \\
   -Q^2\bar{w}_{11} + 2\bar{w}''_{11} + iQ\bar{u}'_{11} &= \bar{p}'_{11} \quad (3.5)
\end{align*}
\]

with boundary conditions

\[
\begin{align*}
   \bar{a}_{11} &= -\bar{h}_{11} \quad (3.6) \\
   \bar{u}_{11} &= 0 \quad (3.7) \\
   \bar{w}_{11} &= -\bar{h}_{11} \quad (3.8)
\end{align*}
\]

at $\bar{z} = 0$ and

\[
\begin{align*}
   \bar{w}_{11} &= \Sigma_1\bar{h}_{21} - \bar{\rho}_{11} \quad (3.9) \\
   \bar{u}'_{11} + iQ\bar{w}_{11} &= -2iQ\bar{h}_{21} \quad (3.10) \\
   -\bar{p}_{11} + 2\bar{w}'_{11} &= 0 \quad (3.11)
\end{align*}
\]

at $\bar{z} = 1$.

Next, we explain the discretization process we use to take this continuum system of PDEs and create a finite-dimensional approximation.

3.2.1. Discretizations of the Small-A Equations

Discretizations must be defined at each main portion of the vector: the bulk, the bottom boundary, and the top boundary to ensure that we are not stepping out of bounds. The vector is entered in such a way that the left-most column of the vector corresponds to
the bottom of the physical film, \( z = 0 \), while the right-most column of the vector corresponds to the surface of the film, \( z = Mz-1 \). For notational brevity in this chapter, we let \( \text{var} = (u,w,\rho,a,p,\sigma) \) represent the variables in our system. And we have \( dhz = 1.0/hz \), where \( hz \) is related to \( Mz \) and represents the step size in our grid.

First, we determine the discretizations in the bulk, where we implement second-order upwind differencing, allowing first derivatives to be discretized in such a way that they “look” toward their Dirichlet boundary conditions. It is useful that centered difference discretizations are not implemented here, because of stability issues within that scheme. Since \( u,w, \) and \( a \) have boundary conditions at \( z = 0 \), these first derivative discretizations look “left” in the vector:

\[
\text{varz} = (x[i-2].\text{var} - 4.0*x[i-1].\text{var} + 3.0*x[i].\text{var})*0.5*dhz.
\]

Notice that for \( i = 1 \), we will step out of bounds. Therefore, we define first derivatives at \( i = 1 \) to be the following:

\[
\text{varz} = (-3.0*x[1].\text{var} + 4.0*x[2].\text{var} - x[3].\text{var})*0.5*dhz.
\]

Similarly, \( p \) and \( \sigma \) have boundary conditions at \( z = Mz-1 \), so these first derivative discretizations look “right” in the vector:

\[
\text{varz} = (-3.0*x[i].\text{var} + 4.0*x[i+1].\text{var} - x[i+2].\text{var})*0.5*dhz.
\]

Here, we step out of bounds for \( i = Mz-2 \), so we implement

\[
\text{varz} = (x[Mz-4].\text{var} - 4.0*x[Mz-3].\text{var} + 3.0*x[Mz-2].\text{var})*0.5*dhz.
\]

Since \( \rho \) does not have a boundary condition at either location, we choose its first derivative discretization to be centered:

\[
\text{rhoz} = (x[i+1].\text{density} - x[i-1].\text{density})*0.5*dhz.
\]

Additionally, we have second derivatives in both \( u \) and \( w \), so we use a second-order centered difference method here:

\[
\text{varzz} = (x[i-1].\text{var} - 2.0*x[i].\text{var} + x[i+1].\text{var})*dhz*dhz.
\]

Next, we look at the left-most column of the vector, \( z = 0 \), the bottom boundary. We represent first derivatives by one-sided finite difference methods that look to the bulk, which means that here it looks “right” in the sense of the vector structure:

\[
\text{varz} = (-3.0*x[i].\text{var} + 4.0*x[i+1].\text{var} - x[i+2].\text{var})*0.5*dhz.
\]

Here, we also have a second derivative in \( w \), for which we extend our stencil at \( z = 0 \). By calculating the ap-
appropriate Lagrange polynomial, we obtain the discretization:

\[ w_{zz} = (2.0 \times x[0].z_{vel} - 5.0 \times x[1].z_{vel} + 4.0 \times x[2].z_{vel} - x[3].z_{vel}) \times dhz \times dhz. \]

Then, we move to the right-most column of the vector, \( z = Mz^{-1} \), the top boundary. Again, first derivatives are represented by one-sided finite difference methods that look to the bulk. Now, these discretizations look “left” in the sense of the vector:

\[ varz = (x[i-2].var - 4.0 \times x[i-1].var + 3.0 \times x[i].var) \times 0.5 \times dhz. \]

3.2.2. Creating the Vector for the System of Small-A Equations

Now that we have presented the discretizations that we will be implementing in our system, we now set up the vector of the small-A functions that we are discretizing.

Each entry in our vector corresponds to a specific variable in our system at a unique point in \( z \). Here, we declare a C struct with a field composite type, which is a PETSc specific term that allows us to define and solve our multi-variable system. For example, here, the variable \( f \) with a field composite type has a list of members which corresponds to our system variables. These variable members are accessed by dot notation.

These variable members corresponding to the field composite type have a different name than the variables seen in the bulk. This is due to a notational requirement that restricts the use of repeated names, even when different types. Under these conditions, an example of a function sent to PETSc’s SNES is \( f[i].x_{vel} = u \), where \( f[i].x_{vel} \) corresponds to the function entry for the “velocity in the x direction” member of the field composite type, while \( u \) represents the “velocity in the x direction” bulk variable. Again, this is notationally necessary to avoid repeating names and therefore creating errors within our program. For all intents and purposes, the particular name of the members of the field composite type can be ignored. They are simply placeholders for our system variables as a means of keeping track of how many functions we have supplied and are unrelated to the associated variable or function.

We have six variables in our system: \( u \) (velocity in the x-direction), \( w \) (velocity in the z-direction), \( \rho \) (density), \( a \) (age), \( p \) (pressure), and \( \Sigma \) (dimensionless growth rate). As a
reminder, these variables are expansions of the linearized variables at $O(A)$. They are represented as $x_{vel}$, $z_{vel}$, density, age, pressure, sigma when members of the field composite type and as $u$, $w$, $rho$, $a$, $p$, sig when bulk variables.

Now we need to determine our numerical equations for this system. Having six variables indicates that we need six equations in the bulk, at the bottom boundary, and at the top boundary. As part of the solution process using PETSc, we must supply a nonlinear residual function for PETSc to work to solve until it evaluates to zero. Note that both the order of appearance of the equations and the corresponding functions, $f[i].vars$, associated with the equations are arbitrary. Again, these are simply how we track the inclusion of our variables and are not a means to define each variable. We start with the equations that we have already defined.

First, we simply restate the known bulk equations Eqs. (3.1) - (3.5), setting each quantity equal to zero, as is required by PETSc, to obtain root-finding functions that encode our numerical bulk equations, Eqs. (3.12) - (3.16). This leads to the following system in the bulk:

\[
\begin{align*}
  f[i].x_{vel} = & \frac{-2.0*Q*Q*u + uzz + PETSC_i*Q*wz - PETSC_i*Q*p}{dhz*dhz}; \\
  f[i].z_{vel} = & \frac{-Q*Q*w + 2.0*wzz + PETSC_i*Q*uz - pz}{dhz*dhz}; \\
  f[i].density = & \rho + a; \\
  f[i].age = & \frac{az}{dhz}; \\
  f[i].pressure = & \frac{(PETSC_i*Q*u + wz + rhoz)}{dhz}; \\
  f[i].sigma = & \frac{sigz}{dhz};
\end{align*}
\]

where PETSc_i corresponds to the imaginary number, $Mz$ is the length of our mesh, and $dhz = 1.0/hz$, where $hz$ represents the step size used in our grid. Notice that some equations are divided by some power of dhz. This is to compensate for the derivatives present in the
equation, which are being multiplied by some power of $dhz$, to keep the equations of the same scale.

It is useful to note that $\sigma$ is only defined at the surface, but we choose to define it as a bulk variable for consistency with later treatments of more complex nonlinear equations. This suggests that the remaining bulk equation, Eq. (3.17), is $\sigma z/dhz$.

Similarly, our boundary conditions at the bottom ($\bar{z} = 0$), Eqs. (3.6) - (3.8), become our numerical boundary conditions at the bottom, $z = 0$: Eqs. (3.18), (3.21), and (3.22). So at the bottom boundary, $z = 0$, we have

\[
\begin{align*}
    f[i].x_vel &= u; \quad (3.18) \\
    f[i].z_vel &= (-Q*Q*w + 2.0*wzz + PETSC_i*Q*uz - pz)/(dhz*dhz); \quad (3.19) \\
    f[i].density &= \rho + a; \quad (3.20) \\
    f[i].age &= a + h11; \quad (3.21) \\
    f[i].pressure &= w + h11; \quad (3.22) \\
    f[i].sigma &= \sigma z/dhz; \quad (3.23)
\end{align*}
\]

We then choose the remaining boundary conditions, Eqs. (3.19), (3.20), and (3.23), at the bottom boundary, $z = 0$, to be the corresponding numerical bulk equations, Eqs. (3.13), (3.14), and (3.17), implemented at the bottom boundary.

Then, our boundary conditions at the top ($\bar{z} = 1$), Eqs. (3.9) - (3.11), become our numerical boundary conditions at the top, $z = Mz-1$: Eqs. (3.24), (3.25), and (3.29). So at
the top boundary, $z = Mz-1$, we have

$$f[i].x_{vel} = \left(uz + PETSC_i*Q*w + 2.0*PETSC_i*Q*h21\right)/dhz;$$ (3.24)

$$f[i].z_{vel} = (-p + 2.0*wz)/dhz;$$ (3.25)

$$f[i].density = \rho + a;$$ (3.26)

$$f[i].age = az/dhz;$$ (3.27)

$$f[i].pressure = \left(PETSC_i*Q*u + wz + \rho z\right)/dhz;$$ (3.28)

$$f[i].sigma = w - sig*h21 + \rho;$$ (3.29)

Like before, Eqs. (3.26), (3.27), and (3.28) at the top boundary, $z = Mz-1$, are set to be corresponding numerical bulk equations, Eqs. (3.14), (3.15), and (3.21), implemented at the top boundary.

Traditionally, one would now begin to form the Jacobian matrix needed for this solve. However, PETSc can construct the colored finite-difference approximation of the Jacobian matrix for this system of equations. Therefore, we choose this method by adding the run-time option `snes_fd_color` as described in the introductory section of this chapter. Now, we have supplied all of the information needed by PETSc’s SNES routine and proceed to solve the system.

3.2.3. Small-A Equations Results

For the solution process, we write a Python script to sweep over values of $Q$ from 0.0 to 5.0, with increments of 0.05, in order to obtain our dimensionless growth rate, $\sigma$, which is a function of our dimensionless wavenumber, $Q$. We choose our remaining parameters to be $h11 = 0.1$, $h21 = 0.1$, and $Mz = 10001$. Beginning with $Q=0.0$, our Python script writes the initial guesses to an input file, which we choose to be: $x = 0.0$, $z = -0.1$, $den = 0.1$, $a = -0.1$, $p = 0.0$, and $sig = 0.0$ from the solution at small-A. Our script then runs the executable PETSc created, that reads the initial guesses from the given input file, uses SNES to solve the system, and prints the results to an output file. Then it extracts the $\sigma$ value
corresponding to the current \( Q \) by reading the final value recorded for the \( \sigma \) variable from the output file. For this study, that corresponds to the last line of the output file, because equations that govern \( \sigma \) are sixth of the six functions. This \( \sigma \) is stored, before repeating the process for the next value of \( Q \). For the remaining \( Q \) values, the only difference from the process just described is that our script writes the input file of initial guesses as the results of the previous \( Q \)'s output file. We had to define initial conditions for \( Q = 0.0 \), because it did not have a previous \( Q \)'s output in which to use as its initial guess. Once the \( Q \) and \( \sigma \) pairs are stored, we can plot them on a single figure.

We now present the numerical solution of the discretized small-A system just introduced and compare these results to our analytical small-A solution from the previous chapter. In that analytical study, we solved for \( \Sigma_1 \), the dimensionless growth rate of our system at \( \mathcal{O}(A) \). In this numerical study, we solve for the dimensional growth rate, \( \sigma \), which is also at \( \mathcal{O}(A) \), due to implementing the small-A equations. This allows for a direct comparison between the two studies.

The plot of our numerical and analytical solutions can be viewed in Figure 3.1, where we see agreement for the solution plots. We acknowledge that these results are indiscernible from one another in the figure, so we continue this work by performing a convergence study between the two studies.

Since we have the exact solution from our analytical study, we can directly evaluate the error in our numerical small-A results by comparing the solutions to our analytical results. For our discretized method, we use the following equation to determine the convergence rate:

\[
\frac{\log(\text{err}[i]-\text{err}[i-1])}{\log(\text{dMz}[i]-\text{dMz}[i-1])}
\]  

where \( \text{err} \) is the difference between the numerical small-A solutions and the analytical small-A solutions, and \( dMz=1.0/Mz \) is the step size. Here, we choose our parameters to be \( A=0.01 \), \( Q=1.0 \), \( h1=0.1 \), and \( h2=0.1 \). For each step size, \( dMz=\{.2, .1, .05, .02, .01, .005, .002, .001, .0005, .0002\} \), we store the resulting value for \( \sigma \) and calculate the \( \text{err} \)
Figure 3.1. Visualizations of the analytical and numerical solutions of the small-A equations, with $h_{11}=0.1$, $h_{21}=0.1$, and $M_z=10001$. We sweep over $Q$ values from 0.0 to 5.0, at increments of 0.05. The results appear to agree and are indiscernible from one another in this figure, with stability regimes (where $\Sigma_1 < 0$) for $Q \leq 1.5$, which is within experimental parameters.

by subtracting this $\text{sigma}$ from our analytical result, $\Sigma_1$, at the same $Q$. As long as this is not our first $\text{err}$ that we have calculated, we can find the difference between this current $\text{err}$, $\text{err}[i]$, and the $\text{err}$ from the previous step size, $\text{err}[i-1]$. Similarly, we calculate the difference between the our current step size $dM_z[i]$ and our previous step size $dM_z[i-1]$.

Results of the convergence test are displayed in Figure 3.2, where we see that we have second-order convergence in our numerical small-A system. This is the convergence rate that we expect, due to the chosen discretizations being second-order.

Both the visual agreement in Figure 3.1 and the convergence obtained by comparing the numerical small-A solutions to the analytical small-A solutions increase our confidence in the results themselves, since the simplified system is solved two different ways. Additionally, this confirms that our implementation of the numerical approximation to the model is correct. Moreover, we can alter the computational model to investigate the original linearized equations. This will eventually lead us to discovering the effect of increasing $A$ in our system.
Figure 3.2. The convergence plot for our numerical small-A equations, with solutions compared to our analytical small-A solutions. Parameter values are chosen to be $A=0.01$, $Q=1.0$, $h_{11}=0.1$, and $h_{21}=0.1$, with various step sizes, $dMz=\{0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.002, 0.001, 0.0005, 0.0002\}$. The convergence rate is of order 2, which is consistent with our second-order upwinding scheme.
3.3. Original Linearized Equations

Now that we have reproduced our analytical results with our numerical investigation of the small-A equations, we relax the $A \to 0$ simplification in order to recover the original linearized equations. For convenience, we restate the original governing equations here:

\[
\begin{align*}
\Sigma \bar{\rho}_1 - A \frac{1}{\psi^3} \bar{w}_1 + \frac{1}{\psi}(iQ \bar{u}_1 + \bar{w}_1') + \psi \bar{p}_1' + A \frac{1}{\psi} \bar{\rho}_1 &= 0 \quad (3.31) \\
\Sigma \bar{a}_1 + \psi \bar{a}_1' + A \frac{1}{\psi} \bar{w}_1 &= 0 \quad (3.32) \\
\psi \bar{p}_1 + \frac{1}{\psi} \bar{a}_1 &= 0 \quad (3.33) \\
-2Q^2 \bar{u}_1 + \bar{u}_1'' + iQ \bar{w}_1' &= iQ \bar{p}_1 \quad (3.34) \\
-Q^2 \bar{w}_1 + 2 \bar{w}_1'' + iQ \bar{u}_1' &= \bar{p}_1' \quad (3.35)
\end{align*}
\]

where $\psi = \sqrt{1 + 2A\bar{z}}$, and with boundary conditions

\[
\begin{align*}
\bar{a}_1 &= -\bar{h}_{11}A \quad (3.36) \\
\bar{u}_1 &= 0 \quad (3.37) \\
\bar{w}_1 &= -\bar{h}_{11}A \quad (3.38)
\end{align*}
\]

at $\bar{z} = 0$ and

\[
\begin{align*}
\bar{w}_1 &= \Sigma \bar{h}_{21} - (1 + 2A)\bar{\rho}_1 \quad (3.39) \\
\bar{u}_1' + iQ \bar{w}_1 &= -2iQ \frac{A}{\sqrt{1 + 2A}} \bar{h}_{21} \quad (3.40) \\
-\bar{p}_1 + 2\bar{w}_1' &= 0 \quad (3.41)
\end{align*}
\]

at $\bar{z} = 1$.

Next, we explain the discretization process we use to take this continuum system of PDEs and create a finite-dimensional approximation.
3.3.1. Discretizations of the Linearized Equations

The discretizations of the system are the same as with the small-A equations, with \( \text{var} = (u, w, \rho, a, p, \sigma) \) representing the variables in our system. We let \( \text{dhz} = 1.0/\text{hz} \), where \( \text{hz} \) is related to \( Mz \) and represents the step size in our grid. We allow upwinding in our system, so first derivatives will be discretized so that they “look” toward their Dirichlet boundary conditions.

In the bulk, where we implement upwind differencing, \( u, w, \) and \( a \) look “left” in the vector:

\[
\text{var}_z = \frac{1}{2} \left( x[i-2].\text{var} - 4.0 \times x[i-1].\text{var} + 3.0 \times x[i].\text{var} \right) \times \text{dhz}
\]

with the following at \( i = 1 \):

\[
\text{var}_z = \frac{1}{2} \left( -3.0 \times x[1].\text{var} + 4.0 \times x[2].\text{var} - x[3].\text{var} \right) \times \text{dhz}
\]

to avoid stepping out of bounds. Similarly, \( p \) and \( \sigma \) look “right” in the vector:

\[
\text{var}_z = \frac{1}{2} \left( -3.0 \times x[i].\text{var} + 4.0 \times x[i+1].\text{var} - x[i+2].\text{var} \right) \times \text{dhz}
\]

with the following at \( i = Mz-1 \):

\[
\text{var}_z = \frac{1}{2} \left( x[Mz-4].\text{var} - 4.0 \times x[Mz-3].\text{var} + 3.0 \times x[Mz-2].\text{var} \right) \times \text{dhz}
\]

to avoid stepping out of bounds. Since \( \rho \) does not have a boundary condition at either location, we choose its first derivative discretization to be centered:

\[
\text{rho}_z = \left( x[i+1].\text{density} - x[i-1].\text{density} \right) \times \text{dhz}
\]

Additionally, we have second derivatives in both \( u \) and \( w \), so we use a second-order centered difference method here:

\[
\text{var}_{zz} = \left( x[i-1].\text{var} - 2.0 \times x[i].\text{var} + x[i+1].\text{var} \right) \times \text{dhz} \times \text{dhz}
\]

At the left-most column of the vector, \( z = 0 \), the bottom boundary, first derivatives are represented by one-sided finite difference methods that look “right” in the sense of the vector structure:

\[
\text{var}_z = \frac{1}{2} \left( -3.0 \times x[i].\text{var} + 4.0 \times x[i+1].\text{var} - x[i+2].\text{var} \right) \times \text{dhz}
\]

Here, we also have a second derivative in \( w \), for which we extend our stencil at \( z = 0 \). By calculating the appropriate Lagrange polynomial, we obtain the discretization:

\[
\text{wzz} = \left( 2.0 \times x[0].z\_vel - 5.0 \times x[1].z\_vel + 4.0 \times x[2].z\_vel - x[3].z\_vel \right) \times \text{dhz} \times \text{dhz}
\]
At the right-most column of the vector, $z = Mz-1$, the top boundary, first derivatives are represented by one-sided finite difference methods that look “left” in the sense of the vector:

$$\text{varz} = (x[i-2].\text{var} - 4.0*x[i-1].\text{var} + 3.0*x[i].\text{var})*0.5*\text{dhz}.$$ 

### 3.3.2. Creating the Vector for the System of Linearized Equations

Now that we have presented the discretizations that we will be implementing in our nonlinear system, we now set up the vector of the original linearized functions that we are discretizing.

We repeat the same process used earlier for our numerical small-A equations to create the vector for our original linearized equations. We maintain the six variables in our system: $u$ (velocity in the x-direction), $w$ (velocity in the z-direction), $\rho$ (density), $a$ (age), $p$ (pressure), and $\Sigma$ (dimensionless growth rate). Again, we need six equations in the bulk, at the bottom boundary, and at the top boundary.

To determine our numerical equations for this system, we start with the equations that we have already defined. First, we simply restate the known bulk equations Eqs. (3.31) - (3.35), setting each quantity equal to zero, as is required by PETSc, to obtain our numerical bulk equations:

$$f[i].x_{\text{vel}} = (-2.0*Q*Q*u + uzz + PETSC_i*Q*wz - PETSC_i*Q*p)/(dhz*dhz); \quad (3.42)$$

$$f[i].z_{\text{vel}} = (-Q*Q*w + 2.0*wzz + PETSC_i*Q*uz - pz)/(dhz*dhz); \quad (3.43)$$

$$f[i].\text{density} = \psi*\rho + a/\psi; \quad (3.44)$$

$$f[i].\text{age} = (\sigma*a + \psi*a + A*w/\psi)/dhz; \quad (3.45)$$

$$f[i].\text{pressure} = (\sigma*\rho - A*w/(\psi*\psi*\psi) + (PETSC_i*Q*u + wz)/\psi + psi*rhoz + A*rho/\psi)/dhz; \quad (3.46)$$

$$f[i].\text{sigma} = sigz/dhz; \quad (3.47)$$
where PETSC_i corresponds to the imaginary number, Mz is the length of our mesh, and dhz = 1.0/hz, where hz represents the step size used in our grid. Like before, some equations are divided by some power of dhz to keep the equations of the same scale.

Here we see why it is convenient to treat sigma as if it were a bulk variable, and why we used SNES in Section 3.2 even though it was linear. When we recover the original linearized equations, nonlinearities appear in the bulk at Eqs. (3.31) and (3.32) due to the unknown nature of Σ. Allowing sigma to be a bulk variable means that we have to add equations into the vector in the bulk, bottom boundary, and top boundary. These additions increase the vector size as well as the complexity of the internal solves needed to obtain the resulting solution. Therefore, we employ PETSc’s SNES feature, instead of a slower numerical computing environment like Matlab, to be more computationally efficient.

Since sigma is only defined at the surface, the remaining bulk equation, Eq. (3.47) is sigz/dhz.

Similarly, our boundary conditions at \( \bar{z} = 0 \), Eqs. (3.36) - (3.38) become our numerical boundary conditions at \( z = 0 \):

\[
\begin{align*}
  f[i].x_{\text{vel}} &= u; \\
  f[i].z_{\text{vel}} &= \frac{(-Q*Q*w + 2.0*wzz + PETSC_i*Q*uz - pz)/(dhz*dhz)}{dhz*dhz}; \\
  f[i].\text{density} &= \frac{psi*rho + a/psi}{dhz*dhz}; \\
  f[i].\text{age} &= a + h11*A; \\
  f[i].\text{pressure} &= w + h11*A; \\
  f[i].\text{sigma} &= \frac{sigz}{dhz};
\end{align*}
\]

We then choose the remaining boundary conditions, Eqs. (3.49), (3.50), and (3.53), at the bottom to be the corresponding numerical bulk equations, Eqs. (3.43), (3.44), and (3.47), implemented at the bottom boundary.
Then, our boundary conditions at \( \bar{z} = 1 \), Eqs. (3.39) - (3.41) become our numerical boundary conditions at \( z = Mz-1 \):

\[
\begin{align*}
\mathbf{f}_i.\text{x}_\text{vel} &= (u_z + \text{PETSC}_i*Q*w \\
&+ 2.0*\text{PETSC}_i*Q*h21*(A/sqrt(1+2*A))/dhz; \\
\mathbf{f}_i.\text{z}_\text{vel} &= (-p + 2.0*\text{w}z)/dhz; \\
\mathbf{f}_i.\text{density} &= \psi*\rho + a/\psi; \\
\mathbf{f}_i.\text{age} &= (\sigma*a + \psi*az + A*w/\psi)/dhz; \\
\mathbf{f}_i.\text{pressure} &= (\sigma*\rho - A*w/(\psi*\psi*\psi) + (\text{PETSC}_i*Q*u + wz)/\psi \\
&+ \psi*\rho\text{oz} + A*\rho/\psi)/dhz; \\
\mathbf{f}_i.\text{sigma} &= w - \sigma*h21 + (1+2*A)*\rho;
\end{align*}
\]

Like before, Eqs. (3.56), (3.57), and (3.58) at the top boundary are set to be corresponding numerical bulk equations, Eqs. (3.44), (3.45), and (3.51), implemented at the top boundary.

Like the small-A case, one would now begin to form the Jacobian matrix needed for this solve. We again choose to use \texttt{snes_fd_color} to approximate our Jacobian matrix have therefore supplied all of the necessary information for PETSc’s SNES routine. We can now solve the system and obtain our numerical results.

3.3.3. Linearized Equations Results with Small-A Value

The solution process is similar to the numerical small-A study, where we write a Python script to sweep over values of \( Q \) from 0.0 to 5.0, with increments of 0.05, in order to obtain our dimensionless growth rate, \( \Sigma \), which is a function of our dimensionless wavenumber, \( Q \). We choose our remaining parameters to be \( h11 = 0.1 \), \( h21 = 0.1 \), \( Mz = 10001 \), and \( A = 0.00001 \), since we need a small value in order to compare it to the small-A results. Beginning with \( Q=0.0 \), our Python script writes the initial guesses to an input file, which we choose to be: \( x = 0.0*A \), \( z = -0.1*A \), \( \text{den} = 0.1*A \), \( a = -0.1*A \), \( p = 0.0*A \), and \( \text{sig} = 0.0*A \) from the solution at small-A, but multiplied by \( A \) to recover the full value.
Our script then runs the executable PETSc created, that reads the initial guesses from the given input file, uses SNES to solve the system, and prints the results to an output file. Then it extracts the \texttt{sigma} value corresponding to the current \texttt{Q} by reading the final value recorded for the \texttt{sigma} variable from the output file. Again, this corresponds to the last line of the output file, because equations that govern \texttt{sigma} are sixth of the six functions. This \texttt{sigma} is stored, before repeating the process for the next value of \texttt{Q}. Like previously, for the remaining \texttt{Q} values, the only difference from the process just described is that our script writes the input file of initial guesses as the results of the previous \texttt{Q}'s output file. Once the \texttt{Q} and \texttt{sigma} pairs are stored, we can plot them on a single figure.

Now we present our numerical results for the original linearized governing equations. As mentioned before, we want to compare them to our numerical and analytical results for the small-A equations, so we let \( A = 0.00001 \). Remember that the small-A solutions are \( \Sigma_1 \) which is an expansion at \( O(A) \) of \( \Sigma : \Sigma \approx \Sigma_0 + A \Sigma_1 \), where \( \Sigma_0 = 0 \) from our work in the previous chapter. Therefore, we must divide our linearized \( \Sigma \) solutions by \( A \) in order to compare these new results to solutions of \( \Sigma_1 \).

The corresponding comparison plot of our analytical and numerical results can be found in Figure 3.3. Our results agree across all three solution methods, with the stability region existing for \( Q < 1.5 \).

Next, we determine the convergence rate of our linearized equations. Unlike the small-A equations, we do not have an exact solution for our linearized equations. Therefore, we will choose a larger number of grid points, \( Mz = 50000 \), to act as our “true” solution in which to compare our results and calculate error. Again, for our discretized method, we use the following equation to determine the convergence rate:

\[
\frac{\log(\text{err}[i]-\text{err}[i-1])}{\log(Mz[i]-Mz[i-1])}
\]

(3.60)

where \text{err} is the difference between the numerical small-A solutions and the “true” numerical small-A solutions and \( dMz = 1.0/Mz \) is the step size. Like before, we choose our parameters to
Figure 3.3. Visualizations of the analytical and numerical solutions of the small-A equations, along with numerical solutions of the linearized equations, with a small value for $A$: $A=0.00001$. Additionally, we choose $h_{11}=0.1$, $h_{21}=0.1$, $M_z=10001$ and sweep over $Q$ values from 0.0 to 5.0, at increments of 0.05. The results between all three methods agree, with stability regimes ($\Sigma/A<0$) for $Q \leq 1.5$, which is within experimental parameters.

be $A=0.01$, $Q=1.0$, $h_{11}=0.1$, and $h_{21}=0.1$. For each step size, $dMz=\{.2, .1, .05, .02, .01, .005, .002, .001, .0005, .0002\}$, we store the resulting value for $\sigma$ and calculate the $err$ by subtracting this $\sigma$ from our “true” solution, where we enforce a smaller step size, $dMz=.00002$. As long as this is not our first $err$ that we have calculated, we can find the difference between this current $err$, $err[i]$, and the $err$ from the previous step size, $err[i-1]$. Similarly, we calculate the difference between the our current step size $dMz[i]$ and our previous step size $dMz[i-1]$. Results of the convergence test are displayed in Figure 3.4, where we see that we have second-order convergence in our numerical linearized system. This is the convergence rate that we expect, due to our chosen discretizations being second-order.

Both the visual agreements in Figure 3.3 and the convergence obtained by comparing the numerical linearized solutions to the finer numerical linearized solutions increase our confidence in the results themselves. Now that we trust our original linearized equations for small parameter values of $A$, we can finally increase $A$ in this system and continue our study.
Figure 3.4. The convergence plot for our numerical linearized equations, with chosen parameter values \( A=0.01 \), \( Q=1.0 \), \( h_{11}=0.1 \), and \( h_{21}=0.1 \), with various step sizes, \( dMz=\{0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.002, 0.001, 0.0005\} \). The error is calculated by comparing these solutions to results when the step size is smaller, \( dMz=0.00002 \). The convergence rate is of order 2, which is consistent with our second-order upwinding scheme.
3.3.4. Linearized Equations Results with Increasing A

Since our numerical results from both the small-A equations and the linearized equations (with a small value for A) match the analytical results, we increase the value of A to investigate what happens to the stability in the system. The only change in this solution process from what has previously been described is that we are looping over values of A as well, specifically:

\[ A = \{0.001, 0.005, 0.01, 0.05, 0.1, 0.2\} \]

and we let our Q values be 50 evenly spaced values between \( Q = 0.0 \) and \( Q = 5.0 \). We present the results of the A parameter sweep in Figure 3.5.

In general, the results at larger values of \( A \) are consistent with those in the small-A limit, suggesting that the basic competition between lateral swelling and viscosity-induced vortices is preserved as \( A \) increases. However, at \( Q = 0 \), we notice that the plot becomes increasingly distant from the origin as \( A \) increases. This suggests that \( \sigma/A \neq 0 \), which implies that there is a nonzero dimensionless growth rate in our system even when the dimensionless wavenumber, \( Q \), of the system is 0. This is puzzling, as we expect this problem to have vertical translation invariance, which usually results in \( \Sigma(Q) = 0 \). One idea to explain such an occurrence, is that there are multiple solution branches for \( \Sigma \), and we have found the less positive branch, whereas the more positive branch governs stability. To test this theory, we force both \( Q \) and \( \sigma \) to be 0 in the linearized equations and then have PETSc solve for the remaining variables. However, the results of this approach diverge, which suggests that there is not a solution where both \( Q \) and \( \sigma \) are exactly 0 for all values of \( A \). More study on this topic remains necessary.

3.4. Discussion and Conclusion

We wanted to know what happens to stability in our system as \( A \) increases, which requires a numerical solve of the original linearized bulk equations. We discretized our system in the bulk and at the boundaries, then employed the PETSc library’s SNES function to solve our system of nonlinear equations more efficiently.

We started with the small-A equations so that we could directly compare the numerical results to our analytical work. The results agreed, so we moved to the original linearized
Figure 3.5. We plot solutions for $\sigma$ when increasing $A$ in the linearized equations, with parameter values $h_{11}=0.1$, $h_{21}=0.1$, and $M_2=501$. We sweep over $Q$ values from 0.0 to 5.0 and find that stability regimes exist for each value of $A$. This suggests that swelling has an effect on stability for various expansion rates.
equations, with small value for our $A$ parameter. Again, our solutions matched. Finally, we reached the motivation of this study: to observe how increased $A$ values effect surface stability. For each value of $A$ in our linearized system, we swept over values of the dimensionless wavenumber, $Q$, and obtained stability regimes that are experimentally appropriate for $Q$.

Stability regimes persist for increasing values of $A$, which suggests that the resulting stability regime in the small-$A$ simplification study in Chapter 2 is not a convenient artifact. Moreover, swelling does have a stabilizing effect on our system and may help account for the vanishing ripples, a phenomenon that has been previously absent in related mathematical models. Furthermore, this supports the idea that swelling should be included in future ion bombardment models that seek to capture surface behavior at higher energies.
4.1. Introduction

Early work in surface dynamics generally concentrates on the linear regime of pattern formation, which results in small amplitude variations of the surface. This is proven to be convenient to study both mathematically and experimentally, because nonlinear terms in mathematical models can be ignored and corresponding experiments can be performed using low-flux ion sources.

Grazing-Incidence Small-Angle X-ray Scattering (GISAXS) is an experimental technique that scatters photons at grazing incidence angles, often less than one degree, in order to study resulting nanostructures. After hitting the surface and scattering, the location of the photons are recorded on an area detector. The surface patterns can be obtained by analyzing the corresponding intensities from the area detector. In the linear regime, GISAXS analysis is simply the Fourier transform of the intensity, the structure factor. This structure factor is the first term in the Taylor expansion of the amplitude.

Low-flux ion experiments are expected to behave within the linear regime, but recent studies using GISAXS indicate the presence of higher-amplitude target surface structures. This entrance into the nonlinear regime requires the reintroduction of the previously neglected nonlinear terms into mathematical equations. This complicates the problem, because the now nonlinear equations cannot be solved analytically. Implementation of numerical nonlinear models should be able to capture the behavior of the larger amplitudes forming on the surface. Specifically, we focus on the experimentally observed sawtooth structures which have been unaccounted for in mathematical mechanisms.
A common mathematical model for ion sputtering is the Kuramoto-Sivashinsky (KS) equation. This equation is a simple, nonlinear PDE with desirable properties such as being isotropic, unstable, and well-posed. The nonlinearity of this model allows for modeling large surface slopes, as seen in the nonlinear regime. Notice that this model does not include any swelling, which is different from our work in Chapters 2 and 3. Solutions of the KS equation are unstable and do not produce sawtooths, as seen experimentally. Introducing asymmetry into this system results in the anisotropic Kuramoto-Sivashinsky (aKS) equation. Solutions of the aKS equation suggest ripples form on the surface, which still does not capture the sawtooth structures. Recently, Pearson and Bradley [34] introduced a higher-order aKS equation in 1D, by including the cubic term of the Taylor expansion of the nonlinear terms. Solutions of the higher-order aKS equation were shown to result in a terraced sawtooth surface.

In this chapter, we study surface morphology in the nonlinear regime by analyzing simulations and experiments under appropriate conditions. Therefore, we expand the higher-order aKS equation to 2D and use the Portable, Extensible Toolkit for Scientific Computations (PETSc) library (version 3.8 [2], [3], [1]) to solve our system numerically, yielding the surface morphology results over time. We find that the surface deformation continues to generate sawtooth and terrace structures.

Now that we have our numerically generated surface, we use simulated GISAXS to perform post-processing analysis on our higher-order aKS solution. Simulated GISAXS is performed on generated surface plots from computational models. Since there is no physical area detector, like in experiments, the structure factor equation used for simulated GISAXS is proposed by Sinha [37]:

\[
\frac{d\sigma}{d\Omega} = N^2 b^2 \frac{1}{(q \cdot A)^2} \int_S \int_S (dS \cdot A)(dS' \cdot A)e^{-iq(r-r')}.
\]  

This allows us to determine the simulated surface patterns. We find that our simulated GISAXS results align with those seen experimentally.
4.2. Nonlinear GISAXS Results

Figure 4.1 from Perkinson and Aziz (private communication) shows AFM topographs of sample morphology saturation with ion beam angles of 55°, 70°, 75°, and 85°, where sawtooth structures are observed. These sawtooths are sharp, which result in chevron shapes. There are asymmetries in the x-direction, along the vertical axis, and a variation of length scales in the y-direction, along the horizontal axis. Ripple orientation is determined by FFT analysis of the saturated AFM topographs. Pattern formations at 55°, 70°, and 75° are characterized by both parallel-mode and perpendicular-mode roughening, which diverges from the typically parallel-mode dominance seen in the linear regime. However, surface structures at 85° exhibit roughening perpendicular to the direction of the ion beam, which is consistent with observations in the linear regime.

GISAXS analysis of the 70° and 75° ion beam angle saturation cases are found in Figure 4.2 from Perkinson and Aziz (private communication). Specifically, we see asymmetrical scattering resulting from the GISAXS analysis in the x-direction of the 70° case, because of the emergence of sawtooth structures perpendicular to the ion beam. We also notice a “smeared out” final scattering profile resulting from the GISAXS analysis in the y-direction of the 75° case, due to the simultaneous roughening of various wavelengths. Note that no asymmetry is seen in the GISAXS spectra of the y-direction, because sawtooth structures are not forming parallel to the ion beam.

We want to computationally duplicate this experiment, by running simulations on an adopted model and performing simulated GISAXS on the generated surface plots.

4.3. Model: aKS Equation

Early work by Bradley and Harper [4] suggest a theory involving curvature and the ion beam angle to account for parallel or perpendicular ripples on the surface of irradiated materials. However, this model is restricted to the linear regime. A recent need for nonlinear terms has led to additional models for the nonlinear regime.
Figure 4.1. Here are AFM images of saturated patterns with ion beam angles of 55° (top left), 70° (top right), 75° (bottom left), and 85° (bottom right). Brightness in this image represents structure height. As you scan the image vertically, from bottom to top, the brightness increases, then suddenly becomes darker, then increases again. This corresponds to sawtooth structures in the X direction. Additionally, we observe sharp, chevron shapes near the peaks. We note that in these images, ions are aimed in the vertical direction. In models, the ion direction is traditionally called the ‘x’ direction. The original images are from Perkinson and Aziz (private communication), with emphasis added to show chevrons and the direction of the ions.
Figure 4.2. (top) GISAXS analysis in the $x$-direction of the resulting patterns with an ion beam angle of $70^\circ$ shows asymmetrical scattering. (bottom) GISAXS analysis in the $y$-direction of the resulting patterns with an ion beam angle of $75^\circ$ shows a “smeared out” final scattering profile. These lines represent times during the ion bombardment process, starting with purple and ending with yellow. These images are from Perkinson and Aziz (private communication).
An ion sputtering model with nonlinear terms is the Kuramoto-Sivashinsky (KS) equation obtained by Cuerno and Barabasi [12]:

\[ u_t = -\beta \nabla^2 u - \gamma \nabla^4 u + \delta |\nabla u|^2, \quad (4.2) \]

where \( u_t \) is the film height and \( \beta, \gamma, \) and \( \delta \) are constant parameters. This simple, nonlinear PDE is isotropic, value-invariant, unstable, and well-posed. The Laplacian term represents the curvature-dependence of the sputter yield, the bi-Laplacian term represents the surface energy dissipation, and the third term represents the slope-dependence of the sputter yield.

An extension of the previous work with the KS equation is to observe what happens when asymmetry is introduced in the system. Therefore, the anisotropic Kuramoto-Sivashinsky (aKS) equation

\[ u_t = \beta_1 u_{xx} + \beta_2 u_{yy} + \gamma_1 u_{xxxx} + 2\gamma_2 u_{xyy} + \gamma_3 u_{yyyy} + \delta_1 u_x^2 + \delta_2 u_y^2, \quad (4.3) \]

is studied by Makeev et al. [20]. In addition to exploring the effects of the asymmetry, Pearson and Bradley more recently proposed the inclusion of an additional term that cubes the first derivative in \( x \), therefore resulting in a 1D higher-order aKS equation [34]:

\[ u_t = \beta_1 u_{xx} + \gamma_1 u_{xxxx} + \delta_1 u_x^2 + \epsilon_1 u_x^3. \quad (4.4) \]

This new higher-order term results from including an additional term of the Taylor series of the slope-dependence of the sputter yield, \( |\nabla u|^2 \). We continue their work on this higher-order aKS equation by extending the equation to 2D:

\[ u_t = \beta_1 u_{xx} + \beta_2 u_{yy} + \gamma_1 u_{xxxx} + 2\gamma_2 u_{xyy} + \gamma_3 u_{yyyy} + \delta_1 u_x^2 + \delta_2 u_y^2 + \epsilon_1 u_x^3 + \epsilon_2 u_y^3 \quad (4.5) \]

and choosing parameters likely to be representative of our system. The new terms arise from including additional terms in the Taylor expansion of \( Y(u_x) \).
4.4. Simulation

Now we present an algorithm for the computational simulation.

4.4.1. Time-Stepping

In a general sense, when solving nonlinear autonomous PDEs with implicit time-steps, we start with the time-dependent equation

\[
\frac{d\vec{u}}{dt} = \vec{f}(\vec{u}),
\]  

(4.6)

where \(\vec{u}\) is the discretized solution and \(\vec{f}(\vec{u})\) is a discretization of the spatial operator. (Note that a linear function can be solved using a simplified version of this nonlinear approach.) We must employ a time-stepping method to find the future value. In the study, we choose the implicit Crank-Nicolson method, because it is both implicit and unconditionally stable. This method is represented as

\[
\frac{\vec{u}_{n+1} - \vec{u}_n}{\Delta t} = \frac{1}{2} \left( \vec{f}(\vec{u}_{n+1}) + \vec{f}(\vec{u}_n) \right).
\]  

(4.7)

Crank-Nicolson is a combination of the forward Euler and backward Euler methods, but yields second-order convergence in time, since it is based on the trapezoid rule. We want to solve for the next point in time, so we move the terms involving the next step, \(\vec{u}_{n+1}\), to the left-hand side, while the terms for the current time step, \(\vec{u}_n\), are gathered on the right-hand side:

\[
\vec{u}_{n+1} - \frac{\Delta t}{2} \vec{f}(\vec{u}_{n+1}) = \vec{u}_n + \frac{\Delta t}{2} \vec{f}(\vec{u}_n).
\]  

(4.8)

When \(\vec{f}\) is nonlinear, this must be solved iteratively. We construct the residual, \(\vec{r}\), given an estimate of \(\vec{u}_{n+1}\), called \(\vec{u}^*\), and subtract the terms at the current step, \(\vec{u}_n\):

\[
\vec{r}(\vec{u}^*) = \left[ \vec{u}^* - \frac{\Delta t}{2} \vec{f}(\vec{u}^*) \right] - \left[ \vec{u}_n + \frac{\Delta t}{2} \vec{f}(\vec{u}_n) \right].
\]  

(4.9)
The residual informs us of how close our guess, $\vec{u}^*$, is to the true next step, $\vec{u}_{n+1}$. The true next step satisfies $\vec{r}(\vec{u}_{n+1}) = 0$, so we need a root-finding method to iterate toward the solution. In this study, we use Newton’s method

$$\vec{u}^{*k+1} = \vec{u}^{*k} - \left[ \frac{\partial \vec{r}}{\partial \vec{u}^{*k}} \right]^{-1} \cdot \vec{r}(\vec{u}^{*k})$$ (4.10)

and iterate until $|\vec{r}(\vec{u}^*)| < 10^{-12}$. This is to say that we enforce a tight nonlinear tolerance so that the nonlinear solver tolerance does not interfere with the overall solution error. Calculating $\left[ \frac{\partial \vec{r}}{\partial \vec{u}} \right]^{-1} \cdot \vec{r}(\vec{u}^*)$ leads to a large matrix solve. Hence, PETSc is useful here, because its parallelized nature is more computationally efficient for these solves, and it computes the above algorithm internally for us. These solves can become large, depending on the chosen spatial mesh size and the problem dimension.

4.4.2. Spatial Discretization

Next, we must discretize the right-hand-side function, calculate the Jacobian of the function, and form the initial condition. We discretize the problem in parallel, so that each MPI task “owns” a portion of the 2D grid. This work is explained below, and for convenience, we will rewrite the higher-order aKS equation here in expanded form:

$$u_t = \beta_1 u_{xx} + \beta_2 u_{yy} + \gamma_1 u_{xxxx} + 2\gamma_2 u_{xxyy} + \gamma_3 u_{yyyy} + \delta_1 u_x^2 + \delta_2 u_y^2 + \epsilon_1 u_x^3 + \epsilon_2 u_y^3. \quad (4.11)$$
First, we must discretize each component of the right-hand-side on the locally owned part of the grid:

\[
\begin{align*}
    u_{xx} &= \frac{u[j][i-1]-2u[j][i]+u[j][i+1]}{hx^2} \\
    u_{yy} &= \frac{u[j-1][i]-2u[j][i]+u[j+1][i]}{hy^2} \\
    u_{xxxx} &= \frac{u[j][i-2]-4u[j][i-1]+6u[j][i]-4u[j][i+1]+u[j][i+2]}{hx^4} \\
    u_{xxyy} &= \frac{u[j+1][i-1]-2u[j+1][i]+u[j+1][i+1]-2u[j][i-1]+4u[j][i]+u[j][i+1]-2u[j-1][i]+u[j-1][i+1]}{hx^2hy^2} \\
    u_{yyyy} &= \frac{u[j-2][i]-4u[j-1][i]+6u[j][i]-4u[j+1][i]+u[j+2][i]}{hy^4} \\
    u^2_x &= \frac{(u[j][i+1]-u[j][i-1])(u[j][i+1]-u[j][i-1])}{4hx^2} \\
    u^2_y &= \frac{(u[j+1][i]-u[j-1][i])(u[j+1][i]-u[j-1][i])}{4hy^2} \\
    u^3_x &= \frac{(u[j][i+1]-u[j][i-1])(u[j][i+1]-u[j][i-1])(u[j][i+1]-u[j][i-1])}{8hx^2} \\
    u^3_y &= \frac{(u[j+1][i]-u[j-1][i])(u[j+1][i]-u[j-1][i])(u[j+1][i]-u[j-1][i])}{8hy^2}
\end{align*}
\]

(4.12)

Notice that we have chosen discretizations that are all second-order accurate. Now, we can piece the discretizations together

\[
    f[j][i] = \beta_1 u_{xx} + \beta_2 u_{yy} + \gamma_1 u_{xxxx} + 2\gamma_2 u_{xxyy} + \gamma_3 u_{yyyy} + \delta_1 u^2_x + \delta_2 u^2_y + \epsilon_1 u^3_x + \epsilon_2 u^3_y \tag{4.13}
\]

in order to evaluate this right-hand-side function.

Next, we will compute the Jacobian of the right-hand-side function by taking the derivative of \(f[j][i]\) at each point in space where discretizations occur. For example, the Laplacian operator uses a stencil over the following grid points: \(u[j][i-1]\), \(2u[j][i]\), and \(u[j][i+1]\) for \(u_{xx}\) and \(u[j-1][i]\), \(2u[j][i]\), and \(u[j+1][i]\) for \(u_{yy}\). Therefore, we compute \(\frac{\partial f[j][i]}{\partial u_{xx}}\) and \(\frac{\partial f[j][i]}{\partial u_{yy}}\) at each grid point:

\[
\begin{align*}
    \frac{\partial f[j][i]}{\partial u_{xx}} : \frac{\partial f[j][i]}{\partial u[j][i-1]}, \frac{\partial f[j][i]}{\partial u[j][i]}, \text{ and } \frac{\partial f[j][i]}{\partial u[j][i+1]} \\
    \frac{\partial f[j][i]}{\partial u_{yy}} : \frac{\partial f[j][i]}{\partial u[j-1][i]}, \frac{\partial f[j][i]}{\partial u[j][i]}, \text{ and } \frac{\partial f[j][i]}{\partial u[j+1][i]}
\end{align*}
\]

(4.14)
where

\[
\frac{\partial f[j][i]}{\partial u[j][i-1]} = \frac{\beta_1}{h_x^2}
\]

\[
\frac{\partial f[j][i]}{\partial u[j][i+1]} = \frac{\beta_1}{h_x^2}
\]

\[
\frac{\partial f[j][i]}{\partial u[j][i]} = \frac{-2\beta_1}{h_x^2} \text{ or } \frac{-2\beta_2}{h_y^2}
\]

\[
\frac{\partial f[j][i]}{\partial u[j-1][i]} = \frac{\beta_2}{h_y^2}
\]

\[
\frac{\partial f[j][i]}{\partial u[j+1][i]} = \frac{\beta_2}{h_y^2}
\]

(4.15)

can be directly calculated from the previously defined function and its associated discretizations. Note that \(\frac{\partial f[j][i]}{\partial u[j][i]}\) has two different options of values, depending on if the derivative is taken with respect to \(u_{xx}\) or \(u_{yy}\). Now, we assemble the Jacobian terms by grid point location. The Jacobian values for the Laplacian operator are shown below.

\[
J[j][i-1] = \frac{\partial f[j][i]}{\partial u[j][i-1]} = \frac{\beta_1}{h_x^2}
\]

\[
J[j-1][i] = \frac{\partial f[j][i]}{\partial u[j-1][i]} = \frac{\beta_2}{h_y^2}
\]

\[
J[j][i] = \frac{\partial f[j][i]}{\partial u[j][i]} (\text{in x}) + \frac{\partial f[j][i]}{\partial u[j][i]} (\text{in y}) = \frac{-2\beta_1}{h_x^2} + \frac{-2\beta_2}{h_y^2}
\]

(4.16)

\[
J[j][i+1] = \frac{\partial f[j][i]}{\partial u[j][i+1]} = \frac{\beta_1}{h_x^2}
\]

\[
J[j+1][i] = \frac{\partial f[j][i]}{\partial u[j+1][i]} = \frac{\beta_2}{h_y^2}
\]

Note that \(J[j][i]\) has two \(\frac{\partial f[j][i]}{\partial u[j][i]}\) because this part of the grid appears in both the \(x\) and \(y\) directions. To keep track of these directions, we define this component twice, for each direction. Similarly, we calculate the derivatives and Jacobian values for the bi-Laplacian and square gradient operators. Since these operators also use some of the same grid points as the Laplacian operator, the above Jacobian values will be modified to include the addition of these terms. The resulting values of \(J\) are inserted into a stencil to create the Jacobian matrix.
4.4.3. PETSc Implementation

It is not necessary to write a program from scratch that will perform the steps described above, because mature toolkits for such purposes exist. Here, we choose the PETSc library to assist us in solving the higher-order aKS equation. As mentioned in the previous chapter, the PETSc library has built-in mechanisms that can handle many difficult aspects that can arise in numerical problems. These PETSc properties include inherent parallelization, periodicity, and the ability to overwrite variables in the executable. Additionally, the more costly numerical aspects, such as time-stepping, root-finding methods, and iterative methods are performed more efficiently due to PETSc’s built-in parallelization, which is particularly convenient for 2D and 3D systems.

We implement the simulation algorithm detailed in the previous section by computing the given discretizations of the right-hand-side function and the right-hand-side Jacobian to PETSc. Lastly, we form the initial solution consisting of random values uniformly distributed between 0 and 0.01.

Now that we have calculated the right-hand-side function, the right-hand-side Jacobian, and the initial solution in a way that PETSc can use, we can employ helpful PETSc functions to solve our aKS equation.

PETSc has built-in parallelization, as long as we send it multiple CPU nodes to solve the equation. We first use PETSc to create a distributed array to manage the parallel nature of the problem. Here, we specify the periodicity of our problem, the stencil size, and the ghost zone by choosing the appropriate corresponding arguments. In this study, we impose periodic boundary conditions, a standard box stencil, and a ghost zone of 2 cells. Next, we set the parameters for our equation. This can be hard-coded, but we choose to read the parameters from an external file, which preserves the executable for use for multiple runs over varying parameter values.

Then, we have PETSc employ a time-stepper. There are many time-stepper options that PETSc can use, but we choose the implicit Crank-Nicolson time-stepping method, as described above in the general procedure. This is done by PETSc internally as part of the
time-stepping call. PETSc uses its Scalable Nonlinear Equations Solver (SNES) component as an interface to Newton-type methods.

Next, PETSc sets the Jacobian matrix and forms the initial solution from the information we supplied. Finally, PETSc solves the problem for the given number of time steps. Compiling the program with an MPI-enabled compiler creates the corresponding executable.

After the executable is made, there are several command line arguments that can be added to help in the study. We found that it was most helpful to have PETSc save each solution frame as a VTK file. We wrote a Python wrapper, utilizing Matplotlib, to convert these VTK files to a viewable format so we could observe each individual frame as well as the corresponding movie. Further post-processing analysis can be done from the data acquired from these VTK files which is detailed later in this study.

4.5. Simulation Results

Now that we have discretized our system and supplied the right-hand-side function, its Jacobian, and an initial guess, we can run our simulation. We choose our parameters of the higher-order aKS equation to be the following: $\beta_1 = -1$, $\beta_2 = 0$, $\gamma_1 = -1$, $\gamma_2 = -1$, $\gamma_3 = -1$, $\delta_1 = -0.5$, $\delta_2 = -0.5$, $\epsilon_1 = 0.05$, and $\epsilon_2 = 0$. We choose $\epsilon_2$ to be 0, due to reflection symmetry in the system. Note that if both $\epsilon$ terms are nonzero, sawtooths are no longer visible. Next, we choose $\epsilon_1$ so that we preserve the formation of structures. Sawtooths and terraces were previously produced by the 1D higher-order aKS equation in Pearson and Bradley’s Eq. (4.4) with a choice of 0.05 for the cubic term. This parameter value remains appropriate for our study, so we adopt it here. Additionally, we use up to 1024 nodes on Southern Methodist University’s high performance computer, ManeFrame, in order to run our simulation on a dimensionless 800x800 grid for a longer time of 300 (with $dt = 0.1$, to produce 3000 frames) and with greater efficiency. The larger domain is useful, because it minimizes the noise in the simulated GISAXS.

A visualization of the solution to the higher-order aKS equation, Eq. (4.5), with the previously mentioned parameters is presented in Figure 4.3. We begin with a random plot of
Figure 4.3. We present surface patterns of the 2D higher-order aKS with parameters $\beta_1 = -1$, $\beta_2 = 0$, $\gamma_1 = -1$, $\gamma_2 = -1$, $\gamma_3 = -1$, $\delta_1 = -0.5$, $\delta_2 = -0.5$, $\epsilon_1 = 0.05$, and $\epsilon_2 = 0$. We show one quarter of the simulated domain to better highlight the surface features. Starting with a random plot between 0 and 0.01, pattern formations are already visible by $t = 50$ (top left) and form ripples by $t = 500$ (top right). Sawtooth structures are present by $t = 1000$ (bottom left) and become increasingly chevron-shaped by $t = 3000$ (bottom right).
values between 0 and 0.01. By $t = 50$, pattern formation is already visible and these patterns become rippled structures by $t = 500$. The morphology continues by forming sawtooth structures by $t = 1000$ that become increasingly chevron-shaped sawtooths by $t = 3000$. It is important to notice that the sawtooth structures and various length scales seen at $t = 3000$ are also observed experimentally for incidence angles between $55^\circ$ and $75^\circ$ in saturated structures. In particular, chevrons are produced here, which is also seen experimentally in the AFM topographs of Figure 4.1. Limitations of the results from this higher-order aKS equation include the failure to duplicate the perpendicular-mode ripples seen in experiments. Including additional higher-order nonlinear terms may minimize behavioral discrepancies between the simulations and experiments.

Next, we determine the convergence rate of this higher-order aKS equation in both time and space. We first perform this convergence study in time. Since we do not have an exact solution for this equation, we choose our “true” solution to be the results of a smaller step size, $dt = 0.01$. We calculate convergence by implementing the following equation:

\[
\frac{\log(err[i]-err[i-1])}{\log(dt[i]-dt[i-1])} \tag{4.17}
\]

where $err$ is the difference between our current higher-order aKS solution and our finer high-order aKS solution, and $dt$ is the time step. We choose our parameters to be $hx = 0.1$ and $hy = 0.1$, with a final time of 100, and the same values from our higher-order aKS equation: $\beta_1 = -1$, $\beta_2 = 0$, $\gamma_1 = -1$, $\gamma_2 = -1$, $\gamma_3 = -1$, $\delta_1 = -0.5$, $\delta_2 = -0.5$, $\epsilon_1 = 0.05$, and $\epsilon_2 = 0$. We choose times $t = \{2, 5, 10, 12, 15, 20\}$, where $dt = 1/t$ and therefore range from 0.5 to 0.05. Like in Chapter 3, we print the solution to a file. We want to investigate the same location for each time so that we can compare the results with the differing time steps, so we save the last line of the file to retain as our solution. We calculate the $err$ by subtracting this solution from our “true” solution, where we enforce a smaller step size, $dt = 0.01$. As long as this is not our first $err$ that we have calculated, we can find the difference between this current $err$, $err[i]$, and the $err$ from the previous step size,
Similarly, we calculate the difference between the our current step size $dt[i]$ and our previous step size $dt[i-1]$.

In Figure 4.4, we see that the we have second-order convergence in our system. This is to be expected, because of the second-order nature of implicit Crank-Nicolson used in this method.

Now, we investigate convergence in space. The difference here is that instead of decreasing our time step, we decrease our step size in both $x$ and $y$ directions, represented as $hx$ and $hy$. Again, since we do not have an exact solution for this equation, we choose our “true” solution to be the results of a smaller step size, $dhx = 0.0001$ and $dhy = 0.0001$. We keep these step sizes identical in both $x$ and $y$ directions, so we can calculate convergence by implementing the following equation in just one direction:

$$\frac{\log(\text{err}[i] - \text{err}[i-1])}{\log(dhx[i] - dhx[i-1])} (4.18)$$

where $\text{err}$ is the difference between our current higher-order aKS solution and our finer high-order aKS solution, and $dhx$ is the step size in $x$. We choose our time step to be $dt = 0.1$, with a final time of $100$, and the same values from our higher-order aKS equation:

$\beta_1 = -1$, $\beta_2 = 0$, $\gamma_1 = -1$, $\gamma_2 = -1$, $\gamma_3 = -1$, $\delta_1 = -0.5$, $\delta_2 = -0.5$, $\epsilon_1 = 0.05$, and $\epsilon_2 = 0$. We choose spatial sizes $hx = \{2, 5, 10, 12, 15, 20, 50, 100, 200, 500\}$, where $dhx = 1/hx$ and therefore range from $0.5$ to $0.002$. Like before, we print the solution to a file and investigate the same location for each step size so that we can compare the results with the differing step sizes, so we save the last line of the file to retain as our solution. We calculate the $\text{err}$ by subtracting this solution from our “true” solution, where we enforce a smaller step size, $dhx = 0.0001$. As long as this is not our first $\text{err}$ that we have calculated, we can find the difference between this current $\text{err}$, $\text{err}[i]$, and the $\text{err}$ from the previous step size, $\text{err}[i-1]$. Similarly, we calculate the difference between the our current step size $dhx[i]$ and our previous step size $dhx[i-1]$. 

75
Figure 4.4. Here is the convergence plot for our higher-order aKS equations, with chosen parameters values $h_x = 0.1$, $h_y = 0.1$, with a final time of 100, and the same values from our higher-order aKS equation: $\beta_1 = -1$, $\beta_2 = 0$, $\gamma_1 = -1$, $\gamma_2 = -1$, $\gamma_3 = -1$, $\delta_1 = -0.5$, $\delta_2 = -0.5$, $\epsilon_1 = 0.05$, and $\epsilon_2 = 0$. The error is calculated by comparing solutions with $t = \{2, 5, 10, 12, 15, 20\}$, where $dt = 1/t$ and therefore range from 0.5 to 0.05 to results when the time step is smaller, $dt = 0.01$. The convergence rate is of order 2, which is consistent with the Crank-Nicolson time stepping scheme.
Figure 4.5. Here is the convergence plot for our higher-order aKS equations, with a chosen time step of $dt = 0.1$, with a final time of 100, and the same values from our higher-order aKS equation: $\beta_1 = -1$, $\beta_2 = 0$, $\gamma_1 = -1$, $\gamma_2 = -1$, $\gamma_3 = -1$, $\delta_1 = -0.5$, $\delta_2 = -0.5$, $\epsilon_1 = 0.05$, and $\epsilon_2 = 0$. The error is calculated by comparing solutions with $hx = \{2, 5, 10, 12, 15, 20, 50, 100, 200, 500\}$, where $dhx = 1/hx$ and therefore range from 0.5 to 0.002 to results when the time step is smaller, $dhx = 0.0001$. The convergence rate is of order 2, which is consistent with our chosen second-order discretizations.
In Figure 4.5, we see that the we have second-order convergence in our model. This is to be expected, because of the second-order discretizations we use in our system.

Moreover, this higher-order aKS model in 2D has second-order convergence in both space and time. Now, we can continue our comparison of numerical and experimental results, by performing some post-process analysis.

4.6. Simulated GISAXS

For the remainder of the study, we focus on analyzing the higher-order aKS equation, Eq. (4.5), and its solution. Our post-processing analysis consists of using simulated GISAXS in order to compare the results to the experimental GISAXS data.

We obtain the simulated GISAXS by using Python and Matplotlib to feed the height field into the GISAXS structure factor equation proposed by Sinha, Eq. (4.1). Minimizing the noise in the simulated GISAXS requires the simulation of a large domain over which to average. We would like to simulate this in parallel with minimal low-level design. For this reason, we use PETSc to solve this higher-order equation in 2D, as presented in Section 4.4.

Once the simulated GISAXS data is numerically run across the AFM images previously generated from solving the higher-order aKS equation with PETSc, we capture the corresponding simulated GISAXS scattering for each simulated surface. These scatterings are placed on a single plot in order to study the evolution of the data over time. The progression of the GISAXS signals in X and Y for the aKS equation are shown in Figure 4.6.

In both $x$ and $y$ directions, the profile begins with a low scattering intensity at high wavenumbers, which suggests an absence of structures. Asymmetry arises at $t = 500$, which corresponds to the presence of sawtooth structures. At later times, both directions take the likeness of a “smeared out” scattering pattern. This effect was only seen experimentally in the $y$-direction, which may be due to the mathematical model containing many length scales in both directions. Additionally, we see harmonic peaks in the simulated GISAXS plots, which are characteristic of weakly-nonlinear equations, where nonlinear terms with powers of the most-unstable mode $e^{ikx}$ excite modes at integer multiples of the wavenumber $k$. 
Figure 4.6. Visualizations of the corresponding GISAXS progression plots for the previously shown simulation of the higher-order aKS equation in 2D. The profiles begin with a low scattering intensity at high wavenumbers, which suggests an absence of structures. Slight asymmetry is visible in the X direction at \( t = 500 \) (near \( qx = 0.8 \)), at \( t = 1000 \) (near \( qx = -0.7 \)), and most noticeably at \( t = 3000 \) (near \( qx = -2.2 \)). In our simulated GISAXS spectra, we observe asymmetry (which corresponds to the emergence of sawtooths), harmonic peaks, and “smeared out” scattering patterns at late times.
4.7. Conclusion

We numerically investigate the nonlinear regime of pattern formation during ion bombardment by adopting and expanding the higher-order anisotropic Kuramoto-Sivashinsky (aKS) equation, Eq. (4.5), to 2D. This computational work utilizes the PETSc library, which is especially useful for our implemented 2D system, since the resulting matrices in this algorithm can become quite large due to the domain size needed for simulated GISAXS. PETSc methods perform the inner-most steps of the algorithm in parallel, which cut down on the computational cost needed to solve the system.

We allow the higher-order aKS solution to propagate in time, choosing parameter values appropriate for our study, where we find that both the sawtooth structures and the length scales produced are also present experimentally, under certain conditions. Additionally, the chevron patterns produced at the end of the simulation time are similar to those seen in some experimental AFM topographs. While the higher-order aKS equation indeed produces the desired sawtooth structures, it does not duplicate the experiment completely. Adding higher-order nonlinear terms into the aKS equation may help recover the missing structures between the simulations and experiments.

Finally, we perform post-process analysis of the results by simulating GISAXS data across the simulated surface structures and plotting the resulting simulated GISAXS scattering on a progression plot. The simulated GISAXS plot indicates an initial absence of structures. Asymmetry emerges by $t = 500$, suggesting the existence of sawtooths, before “smearing out” at late times. This is consistent with experimental GISAXS results in the $y$-direction only, which may be due to the mathematical model containing many length scales in both directions. Additionally, we see harmonic peaks in the simulated GISAXS plots, which are well understood.

By comparing nonlinear regime experiments to nonlinear simulations and simulated GISAXS, we find that the higher-order aKS equation in 2D is a reasonable model for capturing surface behavior. Specifically, the sawtooths that were not visible in the previous lower-order aKS models, are present in this study. Limitations in the model do not allow
for a complete capture of patterned structures, but this should not overshadow the similar features present in both experiments and simulations. Not only are sawtooths recovered in the higher-order aKS model, but simulated chevrons also emerge, as seen experimentally. Asymmetry in the GISAXS scattering is also a notable feature of both experiments and simulations. These agreements suggest that the higher-order aKS equation in 2D allows mathematical models to better align with experiments. By adding more higher-order terms, we are able to capture more experimental features.
5.1. Introduction

In previous chapters, we have considered the effect of swelling on the stability of ion bombarded surfaces, and described how that work extended earlier research that had assumed the strain induced by ion bombardment was volume-preserving. Here we explore the extent to which these strain rates – both volume-preserving and otherwise – might be obtained from atomistic simulations.

During ion bombardment, a series of ions hit the surface of a crystal lattice. Prior to impact, the atoms inside the structure are at a critical distance apart, satisfying the inherent push and pull of the atoms with their neighbors and therefore not moving. When an ion hits the surface, atoms are shifted out of the ideal lattice and may bump other atoms further down in the lattice. Therefore, this previous balance is no longer preserved. As a result, there must be a series of shifting in order to once again satisfy the attraction of the atoms to each other. Since there is new material present, the ion, the atoms cannot simply shift back into the original lattice formation. This can result in an interstitial (an unaligned atom) or a vacancy (an area of space without an atom). Due to their inherent attraction, as the atoms realign, some atoms may shift even if they were not directly hit by the ion or the resulting atom-to-atom hits. This departure from the relaxed crystal latticed introduces strain into the system.
In previous work, this stress-free strain was introduced into the constitutive equation governing the behavior of the viscous film as follows [29]:

\[
T = -pI + 2\eta \left( \dot{E} - \dot{E}_B \right),
\]

(5.1)

Here \( T \) is the stress, \( p \) is the pressure, \( \eta \) is the viscosity, \( \dot{E} \) is the strain rate, and \( \dot{E}_B \) describes an additional stress-free strain rate induced by the ion beam. (Our work in Chapters 2-3 may be considered as a study on the effect of the trace of the tensor \( \dot{E}_B \).) In other words, when ions hit the surface, they immediately induce a shape change in material elements. If the material is allowed to undergo this change, no stress is created, but if the presence of boundaries or other nearby parcels of material inhibit this change, then stress arises.

In existing work, the form of \( \dot{E}_B \) has simply been assumed based on symmetry arguments. In principle, however, Molecular Dynamics (MD) is a way to more carefully study the displacement of these atoms during ion bombardment and the resulting stresses and strains. MD simulations are quite detailed, which result in accurate findings. However, these simulations can be quite time consuming. The Binary Collision Approximation (BCA) is a much faster approach of simulating the atomistic displacement of the system, but sacrifices some accuracy in the process. With this approach, atoms are treated as hard spheres that can interact with only one other atom at a time, avoiding the expensive many-body interactions considered in MD. In this study, we use BCA in order to get a general idea of the atomistic displacements at different ion beam incidence angles. Further investigations may be performed later under a more rigorous molecular dynamics approach.

Ultimately, the goal of this study is to try to determine the angle-dependence of the stress-free strain tensor \( \dot{E}_B \) associated with a material undergoing ion irradiation. We can calculate this strain by using derivatives of the velocity of the atomistic movements, which is detailed later in this chapter. The velocity is determined by investigating the displacement field, which reiterates our need for BCA simulations. We use PyCraters [27], a Python framework designed to aid in the statistical analysis of crater functions. PyCraters builds
a wrapper for TRI3DST [26], which uses BCA for atomistic simulations. Next, we must smooth the data in order to remove statistical noise. Python’s SciPy package has a built-in Savitzky-Golay filter. This filter fits neighboring points to a polynomial by using linear least-squares. An analytical solution to the least-squares equations can be found in order to estimate the smoothed signal. This smoothed signal is plotted for all angles. Lastly, we calculate the zeroth moment and add isotropic expansion to counteract ion-induced compression by calculating the deviatoric strain tensor. The deviatoric strain tensor represents the deformations that change the physical structure, but preserves the volume. We compare the zeroth moment of this deviatoric strain tensor to values found in other literature.

5.2. Displacement Fields

5.2.1. PyCraters and TRI3DST

For this study, we use PyCraters [27] which is a Python framework designed to aid in the statistical analysis of crater functions. PyCraters builds a wrapper for TRI3DST, which uses the Binary Collision Approximation (BCA) for atomistic simulations.

We write a Python script to perform our simulation and analysis. The five parameters needed in this system the noble gas being ionized as the beam, the ion beam energy, the incidence angle of the ion beam, the number of impacts on our target, and the target material. We choose an Argon beam at 1000eV, with incidence angle values of \( \theta = \{0, 10, 20, 30, 40, 50, 60, 70, 80\} \). Additionally, we let there be 100,000 impacts on a Silicon target. The combination of PyCraters and TRI3DST uses this information to run the corresponding simulation and record the statistics from the bombardment.

5.2.2. Calculate the Displacement Fields

The first assumption we make in this study is that the average displacement field can be used as a substitute for the velocity field in a traditional description of the strain. Therefore, we first extract the average displacement field due to an ion impact. For each incidence
angle, $\theta$, we extract the average slices from the BCA simulations and define them as the following: $XZx = f[\text{'XZx_avg'}]$, $XZz = f[\text{'XZz_avg'}]$, $rddXZx = f[\text{'rddXZx_avg'}][0]$, and $rddXZz = f[\text{'rddXZz_avg'}][0]$, where $rddXZx$ represents $v_x$ and $rddXZz$ represents $v_z$. We construct a quiver plot of the 2D displacement field in Figure 5.1 for every angle. Here, we see a correlation between the ion beam angle and the atomistic displacement in the target.

5.3. Strain Tensor

Now that we have determined the displacement of the atoms as a result of our ion bombardment simulation, we want to measure the resulting strain from these movements. Therefore, we need to calculate each component in the stress tensor, $E$:

$$E = \frac{1}{2} \left( \nabla v + \nabla v^T \right) = \begin{bmatrix} v_{xx} & \frac{(v_{xx} + v_{xz})}{2} \\ \frac{(v_{xx} + v_{xz})}{2} & v_{zz} \end{bmatrix}$$  \hspace{1cm} (5.2)

From our previous calculation of the displacement field, we have $v_{x}$ and $v_{z}$. So now we must compute additional derivatives for both $v_{x}$ and $v_{z}$ with respect to each direction, $x$ and $z$.

5.3.1. Smooth Statistical Noise

In order to find these derivatives and smooth statistical noise, we use the built in Savitzky-Golay filter included in Python’s SciPy package. This smoothing function, `savgol_filter()` , is built for 1D arrays, but can appropriately filter our 2D system by allowing the user to specify on which axis the derivative is performed. We calculate the second derivatives in our velocity field by the following:
Figure 5.1. The displacement field for $\theta = \{0, 10, 20, 30, 40, 50, 60, 70, 80\}$, where we see the correlation between atomistic movement and ion beam incidence angle.

\begin{align*}
v_{xx} &= \text{sav} \text{gol\_filter}(rddXZx[:,0:40], \text{window}, \text{order}, \text{deriv}=1, \text{axis}=0) \quad (5.3) \\
v_{xz} &= \text{sav} \text{gol\_filter}(rddXZx[:,0:40], \text{window}, \text{order}, \text{deriv}=1, \text{axis}=1) \quad (5.4) \\
v_{zx} &= \text{sav} \text{gol\_filter}(rddXZz[:,0:40], \text{window}, \text{order}, \text{deriv}=1, \text{axis}=0) \quad (5.5) \\
v_{zz} &= \text{sav} \text{gol\_filter}(rddXZz[:,0:40], \text{window}, \text{order}, \text{deriv}=1, \text{axis}=1) \quad (5.6)
\end{align*}

where we choose the window length to be 11 and the polynomial order to be 5. We calculate $v_{xx}$ in Eq. (5.3) by using this Savitzky-Golay filter to take a derivative (deriv=1) of $v_x$ along the $x$ axis (axis=0). Then, we find $v_{xz}$ in Eq. (5.4) by calculating a derivative (deriv=1) of $v_x$ along the $z$ axis (axis=0). Similarly, we can obtain $v_{zx}$ and $v_{zz}$ by performing the same routine for $v_z$, as shown by Eqs. (5.5) and (5.6).

We present these components of the strain tensor in Figures 5.2 and 5.3.
Figure 5.2. The components of the strain tensor for $\theta = \{0, 10, 20, 30\}$. 
Figure 5.3. The components of the strain tensor for $\theta = \{40, 50, 60, 70, 80\}$. 
5.3.2. Calculate the Strain Field

Now we have calculated the additional derivatives of \( v \) needed to determine the strain field \( E \) in 5.2:

\[
e_{xx} = v_{xx} \quad (5.7)
\]
\[
e_{xz} = 0.5(v_{xz} + v_{zx}) \quad (5.8)
\]
\[
e_{zx} = e_{xz} \quad (5.9)
\]
\[
e_{zz} = v_{zz} \quad (5.10)
\]

5.3.3. Zeroth Moment

We also assume that the spatially-varying strain field affects the surface primarily through its zeroeth moment. This can be calculated by summing each component of the strain tensor.

\[
E_{11} = \text{np.matrix}(e_{xx}).\text{sum()} \quad (5.11)
\]
\[
E_{13} = \text{np.matrix}(e_{xz}).\text{sum()} \quad (5.12)
\]
\[
E_{31} = \text{np.matrix}(e_{zx}).\text{sum()} \quad (5.13)
\]
\[
E_{33} = \text{np.matrix}(e_{zz}).\text{sum()} \quad (5.14)
\]

where \text{np.matrix}().\text{sum()} provides the sum of the matrix elements. Note that the remaining matrix entries are 0, because this study does not consider the \( y \) direction. We then construct the zeroth moment of the strain tensor:

\[
E = \text{np.matrix}([[E_{11}, 0, E_{13}], [0, 0, 0], [E_{31}, 0, E_{33}]]) \quad (5.15)
\]
This strain field data as a function of ion beam incidence angle, $\theta$, can be acquired by repeating the above calculations for various angles $\theta = \{0, 10, 20, 30, 40, 50, 60, 70, 80\}$. For each $\theta$, we store the components of the zeroth moment of the strain tensor:

\[
\text{storeE11}[i] = E\cdot\text{item}((0,0)) \quad (5.16)
\]
\[
\text{storeE13}[i] = E\cdot\text{item}((0,2)) \quad (5.17)
\]
\[
\text{storeE33}[i] = E\cdot\text{item}((2,2)) \quad (5.18)
\]

Notice that we do not explicitly define $\text{storeE31}[i]$, which corresponds to the zeroth moment of $E_{31}$, because $E_{13} = E_{31}$ due to our definition of these values.

5.3.4. Deviatoric component

Finally, for comparison with existing theories, we assume that any measured isotropic component of $\dot{E}_B$ is relaxed more rapidly than the deviatoric component. The latter is calculated according to the expression

\[
E_{\text{dev}} = E - \frac{1}{3} tr(E)I \quad (5.19)
\]

where $E_{\text{dev}}$ is the deviatoric strain tensor, $E$ is the strain field, $tr(E)$ denotes the trace of the zeroth moment of the strain field, and $I$ is the identity matrix. Like before, we store the zeroth moment of the deviatoric strain tensor:

\[
\text{storeEdev11}[i] = E_{\text{dev}}\cdot\text{item}((0,0)) \quad (5.20)
\]
\[
\text{storeEdev13}[i] = E_{\text{dev}}\cdot\text{item}((0,2)) \quad (5.21)
\]
\[
\text{storeEdev33}[i] = E_{\text{dev}}\cdot\text{item}((2,2)) \quad (5.22)
\]
After gathering this data for each value of $\theta$, we plot the zeroth moments for both the strain tensor and deviatoric strain tensor in Figure 5.4.

### 5.4. Result Comparison

We compare the zeroth moment of each $E_{dev}$ component to a specific strain tensor $D(\theta)$ defined by Norris in [29], where

$$
D(\theta) = \begin{bmatrix}
\frac{3}{2} \cos(2\theta) - \frac{1}{2} & 0 & \frac{3}{2} \cos(2\theta) \\
0 & 1 & 0 \\
\frac{3}{2} \cos(2\theta) & 0 & \frac{3}{2} \cos(2\theta) - \frac{1}{2}
\end{bmatrix}
$$

(5.23)

represents thinning in the direction of the beam and is derived from $R(-\theta)D(0)R(\theta)$, where

$$
D(0) = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & -2
\end{bmatrix}
$$

(5.24)

is the strain tensor evaluated at $\theta = 0$, and $R(\theta)$

$$
R(\theta) = \begin{bmatrix}
\cos(\theta) & 0 & -\sin(\theta) \\
0 & 1 & 0 \\
\sin(\theta) & 0 & \cos(\theta)
\end{bmatrix}
$$

(5.25)
Figure 5.4. Zeroth moments for both the strain tensor and deviatoric strain tensor for incidence angles of \( \theta = \{0, 10, 20, 30, 40, 50, 60, 70, 80\} \).

describes rotation around the \( y \) axis. We define these components

\[
D_{11} = 1.5 \times \text{np.cos}(2.0 \times \text{theta} \times \text{np.pi}/180.0) - 0.5 \tag{5.26}
\]

\[
D_{13} = 1.5 \times \text{np.sin}(2.0 \times \text{theta} \times \text{np.pi}/180.0) \tag{5.27}
\]

\[
D_{31} = D_{13} \tag{5.28}
\]

\[
D_{33} = -1.5 \times \text{np.cos}(2.0 \times \text{theta} \times \text{np.pi}/180.0) - 0.5 \tag{5.29}
\]

and construct the strain tensor by piecing these components together by the following construction:

\[
D = \text{np.matrix}([[D_{11}, 0, D_{13}], [0, 1, 0], [D_{31}, 0, D_{33}]]) \tag{5.30}
\]
We choose to evaluate this matrix by choosing $\theta$ values that range from 0° to 80° at 1° increments. We store these components by

\[
\text{store}D_{11}[Di] = D_{11} \\
\text{store}D_{13}[Di] = D_{13} \\
\text{store}D_{33}[Di] = D_{33}
\]

and plot the results in Figure 5.5. We visually compare these results to our zeroth moment of the strain tensor in Figure 5.4 and notice that they do not agree.

5.4.1. Some Observations

We now present some observations. We expect that the results at low angles will be similar, but will differ at high angles. At $\theta = 0$,

\[
E(0) = \begin{bmatrix}
1.34211546 & 0 & -0.01605488 \\
0 & 1.34211546 & 0 \\
-0.01605488 & 0 & -2.68423091
\end{bmatrix}
\]

from this study, and

\[
D(0) = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & -2
\end{bmatrix}
\]
from Norris’ work. Notice that the \{1,1,-2\} structure can be recovered along the diagonal if we divide by the value in $E_{11}$. However, unlike $D(0)$, the off-diagonal values of $E(0)$ are nonzero. This value in $E_{13}$ is determined from Eqns. (5.12) and (5.8). As we see in Figure 5.2, the sum of $v_{z x}$ is 0, while the sum of $v_{x z}$ is $-0.032110$. Therefore, the strain related to $v_{x z}$ dominates the value of $E_{13}$ and $E_{13}$. Perhaps the order of this study (vector field → strain tensor field → average over nearby impacts → deviatoric strain tensor) contributes to this nonzero $v_{x z}$. A future study could switch the order to the following: vector field → average over nearby impacts → strain tensor field → deviatoric strain tensor. An additional study could involve redoing Norris’ work with atomistic data.

5.5. Conclusion

We want to determine a calculation strategy that will serve as a way to directly connect the strain used in continuum models to results available through simulation. In particular, we want to know if stress-free strain rates can be obtained from the BCA. We use PyCraters, with the help of TRI3DST and BCA, to extract the statistics from computational simula-
tions for our chosen incidence angle values of $\theta = \{0, 10, 20, 30, 40, 50, 60, 70, 80\}$. For each angle, we obtain values for the velocity field and plot the corresponding 2D displacement field. Next, we smooth the noise using the built-in Savitzky-Golay filter and calculate the derivatives needed to compute the strain tensor. Then, we plot the corresponding strain field components. Lastly, we find the deviatoric tensor in order to counteract ion-induced compression in the system. The zeroth moment of both the strain and deviatoric strain tensors is determined and plotted as a function of $\theta$.

We take these solutions and compare them to a strain tensor proposed by Norris, especially at low angles. There were some similarities at $\theta = 0$, particularly the $\{1, 1, -2\}$-type structure along the diagonal. However, at all other angles, the simulations produce behavior that is very different from that expected by symmetry arguments. This could indicate that the theoretical expectations are badly wrong, but given good agreement with experiments that those theories have shown, we assume that one or more of our assumptions in this study were not correct. These were that

1. The BCA could inform values of the stress-free strain due to ion impact.

2. The displacement field could serve as a substitute for a traditional velocity.

3. The effect of the resulting strain field is dominated by its zeroth moment.

4. The isotropic contributions can be subtracted to obtain a deviatoric component.

Although we were unable to obtain useful results with this approach, we hope it will prove useful to other efforts to obtain the stress-free strain by atomistic simulation.
Appendix A

Derivation of Kinematic Condition

The standard statement of conservation of mass at a moving interface is

$$\frac{d}{dt} \int_{\Omega} \rho \, dV = - \int_{\partial \Omega} [\rho (v - v_{\hat{n}})] \cdot \hat{n}_{B} \, dA + \int_{\Omega} S \, dV,$$

(A.1)

where $\Omega$ is a thin, moving volume element that straddles the free surface, $\partial \Omega$ is the boundary of that volume element, and $\hat{n}_{B}$ is the outward-pointing normal to the surface of the volume element (see Figure A.1. Note that this differs from the classical result for a static interface only through the inclusion of the term $-v_{\hat{n}} \hat{n}_{S}$). As in the main text, $\rho$ is the density, $v$ is the material velocity, $v_{\hat{n}}$ is the (normal) interfacial velocity, and $\hat{n}_{S}$ is the normal to the interface (the subscript is added here to distinguish it from $\hat{n}_{B}$). Additionally, $S$ is a source or sink of mass, that will prove key to modeling sputtering from the interface.

Rather than using the divergence theorem on the surface integral, as is done in the bulk, we here first consider the limit of a thin volume element, $\varepsilon \to 0$. In that limit one obtains

$$0 = - [- \rho (v \cdot \hat{n}_{S} - v_{\hat{n}}) + \nabla_{S} \cdot \mathbf{j}_{S}]_{\bar{x}_{2}} \Delta A - \left[ f \cos (\phi) Y (\phi) m_{a} \right]_{\bar{x}_{1}} \Delta A,$$

where $\Delta A$ is the area of the top/bottom surface elements, and $\bar{x}_{1}$ and $\bar{x}_{2}$ are points somewhere on the surface of the element. Here the left-hand side of Eq. (A.1) has vanished because as $\varepsilon \to 0$ the total mass in the box vanishes. The surface integral vanishes along the face inside the vacuum, but retains a contribution along the face inside the solid, as well as contributions around the edges due to surface fluxes $\mathbf{j}_{S}$. In addition, we have modeled the sputtering process as a sink of mass precisely at the interface via $S = -\delta (d_{I}) f \cos (\phi) Y (\phi) m_{a}$, which is nonzero only at points on the interface (i.e. where the distance to the interface $d_{I} = 0$),
and which depends on the local angle of incidence $\phi$ by way of the local flux $f(\phi)$ and the sputter yield $Y(\phi)$ ($m_a$ is the atomic mass).

Next, we divide both sides by $\Delta A$ and let the box shrink to a point $x$ on the surface, forcing both $\{\bar{x}_1, \bar{x}_2\} \rightarrow x$. Using $m_a = \rho_0^* \Omega$, dividing both sides by $\rho \Delta A$, and re-arranging, we obtain

$$v_n = \mathbf{v} \cdot \hat{n} - \nabla_S \cdot \mathbf{j}_S - f \cos(\phi) Y(\phi) \Omega \frac{\rho_0^*}{\rho}.$$

Finally, ignoring the effect of surface diffusion (which is well-understood [24]), and noting that we defined $V = f \cos(\theta) Y(\theta) \Omega$, we obtain Eq. (2.8) in the main text. We note that we are assuming that the sputter yield, in units of sputtered atoms per incoming ion, remains constant as a function of target density. This is the simplest possible assumption, and ensures that the recession rate of a flat surface is constant in the steady state. Also, within the context of the small-$A$ limit introduced in Section 2.3.3, the change in density throughout the film is small anyway. However, it should be noted that in principle this assumption may require revision if the density varies significantly over the surface of the film.
BIBLIOGRAPHY


[9] CC. Creative commons. CC BY 4.0 http://creativecommons.org/licenses/by/4.0/. xi, 12


99


[38] Skeeze. via Pixabay


