UNIVERSITY OF CALIFORNIA
SANTA CRUZ

Computational Study of Switching Mechanism and Data Retention in Dielectric Thin Film memristor Using Phase-Field Methodology

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

Doctor of Philosophy
in
Electrical Engineering

by
Foroozan S. Koushan
March 2021

The Dissertation of Foroozan Koushan is approved:

Dr. Nobuhiko P. Kobayashi, Chair

Dr. Sung-Mo(Steve) Kang

Dr. B Sriram Shastry

Quentin Williams
ActingVice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Computational Study of Switching Mechanism and Data Retention in Dielectric Thin Film memristor Using Phase-Field Methodology

Foroozan S. Koushan

The possibility of neuro-inspired computing with eNVMs has increased drastically within the last decade as these devices proved to have the required characteristics such as linearity and scalability to be used as synapses in order to bring together memory and computational process in the network. Memristors with metal oxide stack are demonstrated to have increased number of multi-level states, with long-term stability, making them strong candidates to be used as synaptic devices in STDP.

Since the conductive path formation in a metal oxide Memristor devices plays a major role in training process in Spiking Neural Network, this thesis focuses in using a self-consistent computational phase field method to study conducting channel morphology of resistive switching thin film structures. This approach successfully

predicts the formation and annihilation of conducting channels in typical dielectric thin film structures, comparable to a range of resistive switches, offering an alternative computational formulation based on metastable states treated at the atomic scale, as the system is biased by electric field potential, and as the external temperature of the system changes. In contrast to previous resistive switching thin film models, our formulation makes no a priori assumptions on conducting channel morphology and its fundamental transport mechanisms. This study, also, suggests that the generation and growth of nuclei sites in the system due to the influence of external electric field to be one possible root cause of retention failures of ON and OFF states, and eventual reliability degradation of the memristor device.

Dedication

To my beloved mother, and in memory of my dear father

Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Nobuhiko Kobayashi , for giving me a chance to work on my favorite topic of study, offering me insightful technical comments and recommendations, and teaching me the valuable lesson of how to enjoy the research when nothing seems to work! I will cherish our long discussions forever. Furthermore, I would like to thank Dr. Steve Kang who considered my application for the program worth evaluating, and supported me throughout this work. I am also thankful to Dr. Shastry for taking the time to be part of both my candidacy and defense committees, and grateful to UCSC School of Engineering and its staff for all their considerate guidance.

But at the end, I owe the successful ending of my journey in this wonderful program to my family (my husband Ramtin, and my sons Aryan and Ethan) for their continuous support, encourangement, and wisdom. Thank you for always being there for me, particularly whenever I needed it the most.

Chapter 1

Resistive Switching Metal Oxide Memristors: An Emerging Non-Volatile Memory Technology

1.1 Introduction

In recent years, artificial neural networks (Supervised Learning or Deep Learning) has significantly enhanced accuracy in performing large-scale visual/auditory recognition and classification tasks, some even going beyond humanlevel accuracy [1]. Convolutional neural network (CNN) [2] and recurrent neural network (RNN) [3] algorithms, in particular, have also improved their capability in the area of video, image, speech, and biomedical applications by increasing the size of data set, implementing better modeling schemes, and combining both. Deep learning algorithms have also helped improve accuracy by aggressively increasing the depth and size of the neural network. But the volume of computations imposes increasingly more significant challenges for hardware implementation of these systems[4].

Today, most of the deep learning training is done by GPUs and some application-specific accelerators [5, 6, 7]. These accelerators mostly use SRAM as a synaptic memory on-chip, which is extremely inadequate for storage of large amount of data required in deep leaning algorithms. As an alternative solution, emerging NVM devices are considered for on-chip weight storage in parallel computation with low leakage, low power consumption and high density[8].

One of the critical requirements in selecting an NVM device is the device's capability for multi-level resistance or conductance states which would mimic bio-synaptic device characteristics in neural networks. Among emerging NVM devices, Phase Change Memory (PCM), Resistive RAM or Memristors, Spin-Transfer Torque Magnetic RAM, and Floating Gate Memory devices have potential to fulfill this require-

ment, with minimum compromise in parallel computation of weighted sum.

Many comprehensive research outcomes have been published in the last few years mainly focusing on changes in conductance, or dielectric constant of various metal oxides between two electrodes to emulate basic characteristics of biological synapses at device levels [9, 10, 11] and even in arrays[12, 13, 14]. Intrinsic shortcomings of devices are mitigated by implementing elaborate device architecture and software algorithms to maximize efficiency of the whole system. In addition, there have been some papers attempting to highlight the key characteristics of NVM devices for neural network applications [15], focusing on energy-efficient computation. This paper aims at to discuss the findings on component-level search for best synapses in device and material level, which can be to emulate the realistic biological behavior of synapses in terms of plasticity and accuracy.

1.2 Neuromorphic Design

1.2.1 Review of the hardware design

Current hardware design for neuromorphic networks depends on whether training is done off-line or in-situ in the system. Off-line training is used for non-spiking hardware where information is loaded on edge devices as a one-time programming using software, and classification task is done afterwards by hardware. In-situ training can be done by two different methodologies: 1) Deep learning which uses supervised techniques to train hardware, based on error calculation of the results vs. real model; and 2) Spiking which uses non-supervised technique for hardware training,

using STDP (Spike-Timing Dependent Plasticity) mimicking the biological process of weighted summing in synapses.

Based on the STDP learning rule, the synapse's conductance or weight decreases if pre-synaptic neuron fires earlier than the post-synaptic neuron, and vice versa. The magnitude of the change in weight depends also on the timing between firing of the two neurons: shorter timimg results in larger change.

Emerging NVMs have proved to be a viable replacement for SRAM in GPUs due to their low leakage and potential for parallelism and low power consumption. For back propagation and non-spiking learning, the amount of transferred data increases significantly and power consumption to transfer the data back and forth between the memory and control unit increases exponentially [15]. But, to our best knowledge, for the STDP learning rule eNVMs have not been extensively studied or compared with each other and against bio-synapses.

In the next section, the main characterisctics needed for STDP in an ideal synaptic device are summarized, and then compared against some of most promising eNVMs in the market. The main eNVMs considered for this comparison are resistive memories, such as Phase Change Memory(PCM), Resistive RAM (RRAM), Conductive Bridge RAM (CBRAM), and Spin-Transfer Torque Magnetic RAM (STT-MRAM). Many of these devices have proved to be viable for replacing current flash memory as reported by Sony, Toshiba, Samsung, Micron [16, 17, 18].

1.2.2 Synaptic Devices

In this section, desirable characteristics of an ideal synaptic device for STDP application are discussed. The basic characteristics of a synapse can be replicated by using about ten transistors. However, given the fact that the human brain has approximately 10^{11} neurons and 10^{15} synapses, artificial neurons and synapses built on CMOS transistors are unlikely to accommodate the scalability required for neuromorphic computing. Thus, multi-level state capability satisfying the plasticity requirement, accuracy, and linearity to fulfill analog behavior of weight change in biological synapses are crucial characteristic requirements.

Different techniques such as binarization of the neural network parameters by trading off precision of weights to area [19] have been implemented in order to overcome the lack of ideal plasticity requirement in synaptic devices. Also a combination of write voltage and amplitude algorithms have shown to improve nonlinearity/asymmetric characteristic of synapses [20], at the cost of the area and complexity of the digital circuits. But the search to find an ideal synaptic device that performs learning and memory processes at the network level continues.

1.3 Emerging NV Memory Technologies

Resistance-based emerging NVMs have shown a great potential to be used as synaptic devices due to their scalability, low power consumption and multi-level state variability. Among this group are PCM, RRAM/CBRAM, and STT-MRAM. Scalability and low power consumptions in STT-RAM have made it a strong candidate to replace

current flash process compared to other eNVMs. Research at IMEC has shown that by switching to perpendicular Magnetic Tunnel Junction structure (pMTj), STT-RAM can scale down to 10nm and below [21]. Also research has shown that when STT-RAM is used in a non-convetional regimes, it acts as a stochastic memristor, which can implement synaptic functions [22]. But high write energy and long write latency are the most recent challenges in STT-RAM, which industry is trying to overcome by using architectural approaches.

PCM relies on interchangeable crystallization and amorphization of a chalcogenide layer, and achieves multi-level resistance states by modulating the volume of the amorphous state. Gradual device conductance state up to 100 levels have been shown [23]. But the realization of PCM as synaptic device has been difficult so far due to abrupt chatracteristics of its RESET state compared to SET. This asymmetric characteristics has proved hard to be mitigated by changes in programming algorithms.

Similar problem is observed in RRAM and CBRAM devices which inhibits a non-symmetric characteristics in SET and RESET operations, limiting multi-level state capability per cell and the stability of these states. CBRAM is a version of RRAM where the resistance change is due to the migration of cations from one or both electrodes in the oxide layer, rather than the ionized oxygen or their respective vacancy as in RRAM. Due to the abrupt nature of SET operation in these devices, many tradeoffs have been made in order to use CBRAM and RRAM as sysnapses for STDP applications. Tradeoffs such as reducing ON and OFF window by limiting the region of ON state and controlling the amount of current flowing through the

filament, which results in a gradual increase of conductance in analog form[24]; or adjusting pulse amplitude incremental steps during SET and RESET in RRAM with Pt/HfOx/TiN stack, that compromises speed for accuracy and adds to the latency of write operaqtion[25]. Also recent research[26] shows a tradeoff between the switching speed and the volatile behavior of RRAM by engineering a low mobility metal ions such as Ti within the active electrode, and a tradeoff between device footprint and its symmetric switching characteristics by connecting one RRAM to two CMOS transistors.

1.3.1 Metal-Oxide Memristor

Material engineering of stack in RRAM and Memristors have opened the path to overcome their basic challenges, like capability of storing multiple memory states per cell and long-term stability of these independent states, mainly for Spiking Neural Networks application. It has been shown that the interface between active layer and one of the electrodes influences the stability and switching characteristics in these devices [27], as it controls the location of oxygen vacancy layer formed during switching. In particular, most stable switching characteristics are observed when the active layer is deposited on top of bottom electrode in a bilayer oxide stack. Also the number of stable memory states is proved to be modulated by the types of oxides in RRAM devices[28]. Insertion of a thin interfacial layer between the active layer and the contact is proved to be critical in the stability of the filament.

Incorporation of a thin oxide barrier in Memristor and RRAM devices could also improve the plasticity of the device. This is an improtant requirement needed in

synaptic devices to function like a chemical synapse to enable learning. In Spiking Neural Network, the final connection between the neurons can be made by high conductive paths generated within the active layer, after training process has modulated the resistance of the barrier layer. Hence studying the dynamical evoluation of the conductive path formation within the thin insulating interfactial layer in RRAM and memristor devices becomes essential in understanding the device characteristics and reliability.

1.4 Dissertation Contribution and Organization

Since the conductive path formation in a metal oxide Memristor devices plays a major role in training process in Spiking Neural Network, in this dissertation, we study this phenamenon, using computational phase-field analysis. In contrast to molecular dynamics which tracks the motion of each charge carrier, the phase-field formulation tracks the dynamical evolution of the envelope of clusters of charge carriers, the aggregate boundary of which forms a conducting channel interface within the non-conducting layer. This investigation is intended to give an understanding on how resistive switching works in the oxide, as its behavior depends on dynamical chracateristics of atomic metastable states within the thin film dielectrics. Conductive path formation and annihilation, its stability under thermal force, and the possible role of nucleation in retention loss of data in memristors device are the topics of this dissertation.. Discussions in Chapter 2 of this study gives an overview and bench mark study of available models for switching behavior in RRAM and memris-

tor devices. In Chapter 3, Self-consistent continuum transient simulation of thermo electroformation and annihilation of conductive filaments in thin film dielectric will be modeled and demonstrated. As a follow up, Chaper 4, proposes nucleation force as possible root cause for retention loss of high resistance state, as well as reliability degradation due to ON-state instability and increased device variability, in filamentary memristors and thin film dielectric devices. Chapter 5, gives a summary and conclusion of the work, and a few proposal for future studies, followed by additional details on simulation test cases and the simulation environment and coding in the appendices.

References

[1] Y. LeCun, Y. Bengio, and G.Hinton. Deep learning. *Nature*, 521:436–444, 2015.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012.

[3] Alex Graves, AbdelRahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks, 2013.

[4] Quoc V. Le, Marc'Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng. Building high-level features using large scale unsupervised learning, 2012.

[5] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014.

[6] Yu-Hsin Chen, Tushar Krishna, Joel Emer, and Vivienne Sze. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. In *IEEE International Solid-State Circuits Conference, ISSCC 2016, Digest of Technical Papers*, pages 262–263, 2016.

[7] J. Sim, J. Park, M. Kim, D. Bae, Y. Choi, and L. Kim. 14.6 a 1.42tops/w deep convolutional neural network recognition processor for intelligent ioe systems. In *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 264–265, 2016.

[8] Shimeng Yu. *Neuro-inspired Computing Using Resistive Synaptic Devices*. Springer, Cham, 2017.

[9] Duygu Kuzum, Shimeng Yu, and H-S Philip Wong. Synaptic electronics: materials, devices and applications. *Nanotechnology*, 24(38):382001, sep 2013.

[10] Doo Seok Jeong, Inho Kim, Martin Ziegler, and Hermann Kohlstedt. Towards artificial neurons and synapses: a materials point of view. *RSC Adv.*, 3:3169–3183, 2013.

[11] Doo Seok Jeong, Kyung Min Kim, Sungho Kim, Byung Joon Choi, and Cheol Seong Hwang. Memristors for energy-efficient new computing paradigms. *Advanced Electronic Materials*, 2(9):1600090, 2016.

[12] Giacomo Indiveri, Bernabe LinaresBarranco, Tara Hamilton, André van Schaik, Ralph EtienneCummings, Tobi Delbruck, Shih-Chii Liu, Piotr Dudek, Philipp Häfliger, Sylvie Renaud, Johannes Schemmel, Gert Cauwenberghs, John Arthur, Kai Hynna, Fopefolu Folowosele, Sylvain SAÏGHI, Teresa SerranoGotarredona, Jayawan Wijekoon, Yingxue Wang, and Kwabena Boahen. Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5:73, 2011.

[13] Jennifer Hasler and Harry Marr. Finding a roadmap to achieve large neuromorphic hardware systems. *Frontiers in Neuroscience*, 7:118, 2013.

[14] Steve Furber. Large-scale neuromorphic computing systems. *Journal of Neural Engineering*, 13(5):051001, aug 2016.

[15] Shimeng Yu. Neuro-inspired computing with emerging nonvolatile memorys. *Proceedings of the IEEE*, 106(2):260–285, 2018.

[16] Y. Choi, I. Song, M. Park, H. Chung, S. Chang, B. Cho, J. Kim, Y. Oh, D. Kwon, J. Sunwoo, J. Shin, Y. Rho, C. Lee, M. G. Kang, J. Lee, Y. Kwon, S. Kim, J. Kim, Y. Lee, Q. Wang, S. Cha, S. Ahn, H. Horii, J. Lee, K. Kim, H. Joo, K. Lee, Y. Lee, J. Yoo, and G. Jeong. A 20nm 1.8v 8gb pram with 40mb/s program bandwidth. In *2012 IEEE International Solid-State Circuits Conference*, pages 46–48, 2012.

[17] T. Liu, T. Yan, Roy Scheuerlein, Yingchang Chen, J. K. Lee, G. Balakrishnan, G. Yee, Henry Zhang, A. Yap, Jingwen Ouyang, T. Sasaki, Sravanti Addepalli, A. Al-Shamma, C. Chen, M. Gupta, Greg Hilton, S. Joshi, Achal Kathuria, V. Lai, D. Masiwal, M. Matsumoto, A. Nigam, Anil Pai, Jayesh Pakhale, C. Siau, X. Wu, R. Yin, Liping Peng, Jang Yong Kang, S. Huynh, Huijuan Wang, Nicolas Nagel, Y. Tanaka, M. Higashitani, Tim Minvielle, C. Gorla, T. Tsukamoto, T. Yamaguchi, Mutsumi Okajima, T. Okamura, S. Takase, Takahiko Hara, Hirofumi Inoue, Luca Fasoli, M. Mofidi, R. Shrivastava, and

K. Quader. A 130.7mm2 2-layer 32gb reram memory device in 24nm technology. *2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pages 210–211, 2013.

[18] Akifumi Kawahara, Ryotaro Azuma, Yuuichirou Ikeda, Ken Kawai, Yoshikazu Katoh, Kouhei Tanabe, Toshihiro Nakamura, Yoshihiko Sumimoto, Naoki Yamada, Nobuyuki Nakai, Shoji Sakamoto, and Yukio Hayakawa. An 8mb multi-layered cross-point reram macro with 443mb/s write throughput. *IEEE International Solid-State Circuits Conference*, pages 432–434, 02 2012.

[19] J. Woo, K. Moon, J. Song, S. Lee, M. Kwak, J. Park, and H. Hwang. Improved synaptic behavior under identical pulses using alox/hfo2 bilayer rram array for neuromorphic systems. *IEEE Electron Device Letters*, 37(8):994–997, 2016.

[20] P. Chen, B. Lin, I. Wang, T. Hou, J. Ye, S. Vrudhula, J. Seo, Y. Cao, and S. Yu. Mitigating effects of non-ideal synaptic device characteristics for on-chip learning. In *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 194–199, 2015.

[21] E. Liu, J. Swerts, Y. C. Wu, A. Vaysset, S. Couet, S. Mertens, S. Rao, W. Kim, S. Van Elshocht, J. De Boeck, and G. S. Kar. Top-pinned stt-mram devices with high thermal stability hybrid free layers for high-density memory applications. *IEEE Transactions on Magnetics*, 54(11):1–5, 2018.

[22] A. F. Vincent, J. Larroque, N. Locatelli, N. B. Romdhane, O. Bichler, C. Gamrat, W. Zhao, Jacques-Olivier Klein, S. Galdin-Retailleau, and D. Querlioz. Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems. *IEEE Transactions on Biomedical Circuits and Systems*, 9:166–174, 2015.

[23] Duygu Kuzum, Rakesh G. D. Jeyasingh, Byoungil Lee, and H.-S. Philip Wong. Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Letters*, 12(5):2179–2186, 2012. PMID: 21668029.

[24] Yuhan Shi, Leon Nguyen, Sangheon Oh, Xin Liu, Foroozan Koushan, John R. Jameson, and Duygu Kuzum. Neuroinspired unsupervised learning and pruning with subquantum cbram arrays. *Nature Commun*, 9:5312, 2018.

[25] L. Gao and S. Yu. Programming protocol optimization for analog weight tuning in resistive memories. In *2015 73rd Annual Device Research Conference (DRC)*, pages 184–184, 2015.

[26] D. Ielmini. Brain-inspired computing with resistive switching memory (rram): Devices, synapses and neural networks. *Microelectronic Engineering*, 190:44–53, 2018.

[27] Liudmila Alekseeva, Toshihide Nabatame, Toyohiro Chikyow, and Anatolii Petrov. Resistive switching characteristics in memristors with al2o3/TiO2and TiO2/al2o3bilayers. *Japanese Journal of Applied Physics*, 55(8S2):08PB02, jun 2016.

[28] Spyros Stathopoulos, Ali Khiat, Maria Trapatseli, Simone Cortese, Alexander Serb, Ilia Valov, and Themis Prodromakis. Multibit memory operation of metal-oxide bi-layer memristors. *Scientific Reports*, 7, 12 2017.

Chapter 2

Conduction Path Study in Thin-Film Dielectric: Insulator- Metal Transition

2.1 Dielectric Thin Film Resistive Switching

Resistive switching devices are currently being explored as the best candidate for next-generation non-volatile memory applications, and as the result many research studies are conducted to model switching behavior and characteristics in these devices. In general, RRAM and memristor consist of two-terminal metal-insulator-metal (MIM) structure, and are typically classified into bipolar and unipolar devices, depending on the polarity of switching. Switching mechanism in these devices falls into two general chategories: 1) Formation and rupture of localized conductive filaments (CF) by the drift of metallic ions/ oxygen vacancies; 2) Combination/regeneration of the oxygen ions/vacancies[1], as shown in Fig. 2.1.

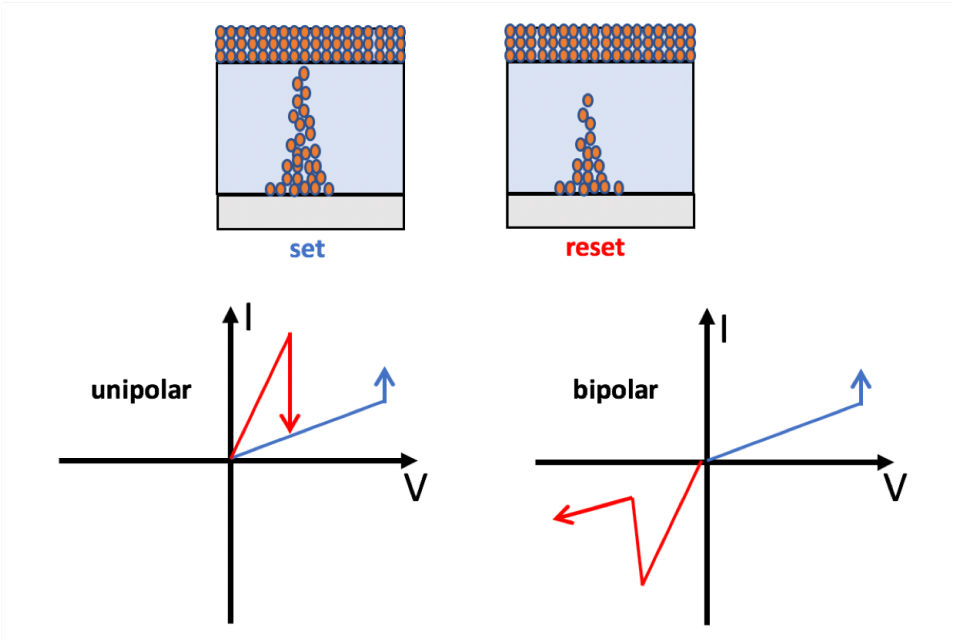

Figure 2.1: Current-Voltage characteristics of a typical unipolar and bipolar memristor

Experimental studies have shown that performance of resistive switching strongly is affected by the choice of the active oxide layer[2]. Various qualitative models have been adopted to study resistive switching behavior of dielectric thin films, largely based on the formation of conducting channels composed of clusters of charged species. Some of these models are summarized by D.Panda [2].

According to this summary, modeling of switching characteristics of resistive devices has evolved over time by using different variables such as width of the doped region in ion drift model, or filament gap in joule heating model, or concentration of ions as well as CF radius in COMSOL and electroforming model. These models invoke both electronic transport and ionic transport, often treated as electrothermal and electrochemical processes, respectively, reproducing both unipolar and bipolar resistive switching behavior.

Given the complexity of the ionic transport of thin film dielectric system, a computational formulation, such as phase field method, that does not impose assumptions on conducting channel morphology, transport phe- nomena, or interface uniformity and instead treats resistive switching from its origin at the atomic scale could be advantages to use. Phase field method can model the dynamical evolution of cluster-like charged aggregates that are subject to their atomic and interfacial electrothermal interaction, naturally producing conducting channels in a nonconducting host. A short comparison summary of available models is discussed in the next section.

2.2 Literature Benchmarking and Phase Field Formulation

In contrast with the lumped equivalent and continuum formulations, various qualitative models have been adopted to study resistive switching behavior of dielectric thin films, largely based on formation of so-called conducting channels composed of clusters of charged species[3, 4]. These models invoke both electronic transport and ionic transport, often treated as electrothermal and electrochemical processes, respectively, reproducing both unipolar and bipolar resistive switching behavior. The conducting channel formalism suggests an initial irreversible growth of cluster-like aggregates composed of charged species forming conducting channels, as illustrated by Fig. 2.1. These conducting channels subsequently form and rupture under the influence of an external electric potential, yielding resistive switching behavior.

Self-consistent solution of the continuum transport equations dynamically emulates advection and diffusion of thermally-activated charged species, and their interaction with local electric potential and temperature, to model bulk resistive switching phenomena. The continuum formulation, however, depends vitally on an a priori conducting channel transport model and correct identification of diffusion and mobility expressions for each specific transport mechanism, for example Poole-Frenkel transport for ionic vacancy conduction[5, 6, 7].

A computational formulation that does not a priori impose assumptions on conducting channel morphology, transport phenomena, or interface uniformity, and in- stead treats resistive switching from its origin at the atomic-scale, may offer significant ad-

vantages over existing methods. Such a method might model the dynamical evolution of cluster-like charged aggregates, as illustrated by Fig. 2.1, subject to their atomic and interfacial electrothermal interaction, naturally producing conducting channels in a non-conducting host. The phase field method is one such method[8, 9].

With the phase field formualtion, the assumptions of an a priori conducting channel model and the presence of specific transport phenomena to explain resistive switching are abandoned, and the model is instead formulated as a diffuse interface problem subject to a variational principle[10]. The phase field formulation thus avoids the mathematically onerous problem of expressing dynamic boundary conditions over an interface whose location is part of the unknown solution. As demonstrated in Fig. 2.2, a qualitative comparison of the various computational formulations with phase filed formulation. highlights that the user of the phase field formulation accrues many benefits favorable to the study of the dynamical behavior of dielectric thin film resistive switching phenomena, as the phase field formulation successfully predicts the formation of conducting channels in typical dielectric thin films comparable to a range of resistive switch structures, offering an alternative computational formulation based on metastable states treated at the atomic scale, requiring no assumptions on conducting channel morphology and its fundamental transport mechanisms. This approach applies to both electronic transport and ionic transport, e.g. ionic oxygen vacancies.

Implementastion of the phase field formulation is explained in the following two chapters.

Capability	Published Lumped Element	Published Continuum	This Dissertation	
			Continuum	Phase-Field
Self-Consistent Electrothermal Transport	Assumed Ideal	Yes	Yes	Yes
Self-Consistent Charge Transport	Assumed Ideal	Yes	Yes	Yes
Conductivity from First Principles	No	Yes	Yes	Yes
Continuum Phase Change Formulation	No	Yes	Yes	Yes
Atomic-Origin Phase Change Formulation	No	No	No	Yes
Scales to Nanoscale Transport	No	Assumed	Assumed	Yes
Transient Dynamics	Yes	Yes	Yes	Yes
Electroforming Dynamics	No	No	Yes	Yes
Conducting Channel Morphology and Growth	No	No	No	Yes

Figure 2.2: Literature benchmarking comparing various capabilities of lumped element, continuum, and phase field formulations for the computational study of dielectric thin film resistive switching phenomena. Capabilities of the work produced in this disserta- tion are in the two right columns

References

[1] R. Waser and M. Aono. Nanoionics-based resistive switching memories. *Nature Mater*, 6:833–840, 2007.

[2] D. Panda, P.P. Sahu, and T.Y.Tseng. A collective study on modeling and simulation of resistive random access memory. *Nanoscale Res Lett*, 13, 2018.

[3] F. Pan, C. Chen, Z. shun Wang, Y. chao Yang, J. Yang, and F. Zeng. Non-volatile resistive switching memories-characteristics, mechanisms and challenges. *Progress in Natural Science: Materials International*, 20:1–15, 2010.

[4] T. Ahmed, S. Walia, E. L. Mayes, R. Ramanathan, P. Guagliardo, V. Bansal, M. Bhaskaran, J. J. Yang, and S. Sriram. Inducing tunable switching behavior in a single memristor. *Applied Materials Today*, 11:280–290, 2018.

[5] G. Gibson, S. Musunuru, J. Zhang, K. Vandenberghe, J. Lee, C. Hsieh, W. Jackson, Y. Jeon, D. Henze, Z. Li, and R. Williams. An accurate locally active memristor model for s-type negative differential resistance in nbox. *Applied Physics Letters*, 108(2):023505, 2016.

[6] F. Nardi, S. Larentis, S. Balatti, D. Gilmer, and D. Ielmini. Resistive switching by voltage-driven ion migration in bipolar rram - part i: Experimental study. *IEEE Trans. on Electron Devices Letters*, 59(9):2461—-2467, 2012.

[7] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams. The missing memristor found. *Nature*, 453:80, 2008.

[8] W. J. Boettinger, J. A. Warren, C. Beckermann, and A. Karma. Phase-field simulation of solidification. *Annual Review of Materials Research*, 32(1):163–194, 2002.

[9] Q. C. Sherman and P. W. Voorhees. Phase-field model of oxidation: Equilibrium. *Phys. Rev. E*, 95:032801, 2017.

[10] N. Provatas and K. Elder. *Phase-Field Methods in Materials Science and Engineering*. Wiley-VCH Verlag, Weinheim, Germany, 2010.

Chapter 3

An electro-thermal computational study of conducting channels in dielectric thin films using self-consistent phase-field methodology: A view toward the physical origins of resistive switching

3.1 Introduction:

Morphological evolution of the interface separating two chemically distinct materials is inherently dynamic, producing distinctive microstructure due to various interactions between interface energy, bulk free-energy, electrothermal, and electrochemical phenomena[1]. Resistive switching behavior of dielectric thin film is obtained as dynamic evolution of complex microstructures emerging at the interface minimizes the total energy, forming a conductive path resulting in characteristic transport properties of electric charges[2]. The formation and annihilation of electrically conductive filaments (ECFs) in dielectric thin films have been studied extensively to understand distinctive functionaries of resistive switching devices that would benefit a variety of technical fields encompassing nonvolatile memories and neuromorphic technology[3].

Experimental assessments on ECFs that operate within dielectric thin films have been an important issue in the field of memristive device in recent years. Strachan el al. studied ECFs at the atomic-scale and identified the presence of phase changes that occurred in a functioning TiO_2-based memristive device using synchrotron-based x-ray absorption spectromicroscopy and transmission electron microscopy[4], providing experimental evidence of regions of dynamic conductivity modulations during resistive switching. In another study, Nallagatla el al. used an Au-coated probe tip as a mobile top electrode to show that localizing electric field and using an epitaxial homogeneous thin-film improved the performance and repeat-ability of resistive switching[5].

Along with these experimental assessments, the nature of ECFs has been in-

vestigated extensively using various computational approaches that employ phenomenological electro-thermal modeling often in conjunction with the finite-element method[6, 7]. In many existing models, the formation and annihilation of ECFs are resolved on the basis of solutions obtained self-consistently for the continuum transport equations that dynamically emulate advection and diffusion of electrical charges under the influence of electric field and temperature. In other words, the presence – rather than the formation – of a single ECF is set as the initial condition, and then, the dependence of physical properties (i.e., electrical conductance) of the pre-conditioned ECF on such parameters as electrical potential and temperature is evaluated to reproduce unique electrical characteristics of resistive switching. As a result, these models heavily rely on diffusion and mobility formulations based on the transport of, for instance, the Frenkel-type vacancies[8]. For instance, Strukove[9] first proposed a linear and then later an exponential ion drift mobility, and Yang[10] suggested a non-linear dependence of dopant drifts under applied voltage. A ECF combined with a gap was also used along with stochastic properties of ions having spatial variations within the ECF[11].

Furthermore, an electro-thermal modeling was carried out for a device represented by a cylindrical shape that contained a pre-existing single ECF assumed to have a plain cylindrical shape[6]. When various thermal processes are included in the formulation of ECFs, the Fourier equation is used to model annihilation of ECFs by invoking Joule heating, which was found to be consistent with the temperature of ECFs, experimentally obtained[12]. In multiple studies, an electro-thermal model in conjunction with the finite-element method was performed by using the migration

of such electrical charges as oxygen vacancies[1, 2, 3, 7]. A similar method was used for a structure that consists of heterogeneous bi-layered dielectric stack[13]. Moreover, multiple ECFs were considered to participate in electrical properties to explain multi-level switching characteristics[13, 14].

In this paper, the phase-field method[15, 16] is employed to explicitly illustrate the formation and annihilation of ECFs in a dielectric thin film to assess characteristic electrical properties of resistive switching devices. The previous studies described earlier all assumed the presence of a single ECF that has specific dimensions and geometrical morphologies in conjunction with general descriptions on the transport of electrical charges under the influence of electric potential and heat. In contrast, our approach based on the phase-field method tracks the dynamical formation and evolution of domains (i.e., ECFs) made of electrical charges present in a dielectric thin film as the energy associated with the system – a dielectric thin film containing electrical charges forming domains – is reduced over time. While the phase-field method successfully predicted the formation of multiple ECFs under the influence of electric field externally applied under isothermal conditions in our previous studies[17], this paper presents a substantial extension of our previous study by including the dependence of the formation and annihilation of ECFs on local electrical potential and local temperature to elucidate the on and off states of a resistive switching device.

3.2 Formulation for Phase-Field method

Molecular dynamics applied to the current context would track the motion of charged particles present in a dielectric thin film in a resistive switching device. In contrast, the phase-field method used in our study tracks the dynamical evolution of clusters made of charged particles in a dielectric thin film. These charge-clusters that represent regions of electrically conducting, collectively define geometrical boundaries (i.e., interfaces) that separate an electrical conducting region from a non-conducting region, allowing us to avoid the necessity of expressing dynamic boundary conditions over an evolving interface and significantly reducing computational burdens without forfeiting overall integrity of the modeling.

In our study, a 50 nm x 10 nm dielectric thin film, illustrated in Fig. 3.1, is considered as a system. This system is viewed as an as-fabricated resistive switch comprising a dielectric thin film in which two distinct regions separated by an interface, exemplifying a resistive switch made of a dielectric thin film in which mobile charges are initially distributed in a certain way by which the two regions, one conducting and the other non-conducting are formed. As illustrated in Fig. 3.1, the system consists of two distinctive regions: a conducting region with normalized concentration of charged particles $c_(\mathbf{r}, t)$, where $\mathbf{r}$ is a position vector within the system and t is time, set to a random local value varied in the range of 0.7-0.9 at $t = 0$, and a non-conducting region with $c_(\mathbf{r}, t)$ set to a random local value chosen in the range of 0.1-0.3 at $t = 0$, separated by an interface along which c also varies randomly. In our previous study[17], the presence of the non-uniformity in c along the

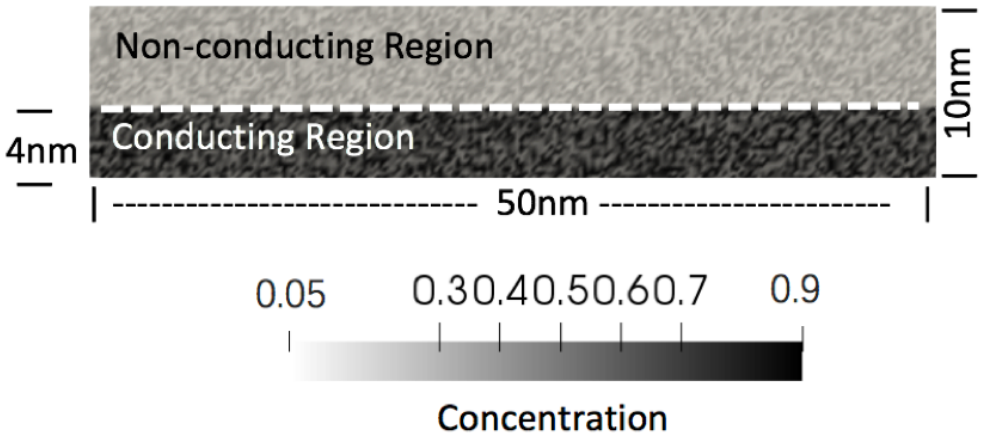

Figure 3.1: A 50 nm x 10 nm dielectric thin film composed of a conducting region with initial concentration in the range of 0.7-0.9 and a non-conduction region with concentration in the range of 0.1-0.3.

interface between the two regions was found to induce the formation of ECFs under the influence of external electric potential . This specific initial structure allows a double-well free-energy density function and the diffuse interface approximation to suitably describe dynamical structural evolution of the dielectric thin film. The bulk free-energy density function $f_{bulk}(c_{(\mathbf{r},t)})$ associated with the system in Fig. 3.1 is given by:

$$f_{bulk}(c_{(\mathbf{r},t)}) = A[c_{(\mathbf{r},t)} - c_1]^2[c_{(\mathbf{r},t)} - c_2]^2 \tag{3.1}$$

where A is the magnitude of the double-well potential, c_1 and c_2 are the normalized concentrations of conducting and non-conducting states. Variable $c_{(\mathbf{r},t)}$ in Eq. (3.1) is bounded to $0<c_{(\mathbf{r},t)}<1$, where zero corresponds to pure non-conducting state and unity corresponds to pure conducting state.

In a dielectric thin film structure, with specific dimensions set in our modeling, the bulk free energy density of the system is expected to be dominated by the total surface tension experienced by the entire conductive regions made of charged particles. In this view, the dependence of the surface tension on temperature T is translated into the dependence of the bulk free energy density on temperature in the modeling. The dependence of the surface tension on temperature shown by Guggenheim-Katayama[18] is expressed as follows:

$$\gamma = \gamma^o(1 - \frac{T}{T_c})^n \tag{3.2}$$

where γ^o is the concentration-dependent surface tension, n is an empirical factor, and T_c is the critical temperature of the system at which the phase of the material changes. So as temperature T raises above T_c, the thin-film undergoes an irreversible phase change impacting its electrical characteristics in such a way that the system no longer works as a resistive switching device.

Eq. (3.2) suggests that surface tension, and thus its corresponding surface energy, decreases as the temperature T increases, and becomes zero when T matches T_c. Empirical factor n that depends on a specific material system modulates the rate at which γ responds to changes in T. In other words, n determines the order of the polynomial equation relating γ to T, and its specific value is obtained experimentally. In this paper, n is set to 2 as it usually takes a value in the range between 1 and 2 for liquids[19].

Although Eq. (3.2) conventionally applies to an interface that separates two liquid phases, in this paper we are assuming that utilizing the dependence of γ on T as

in Eq. (3.2) is deemed valid for interfaces of two solid phases – conducting and non-conducting regions as shown in Fig. 3.1 – present in a dielectric thin film, because of virtual melting theorized for solid-solid interfaces at the nano-meter scale[20]. Therefore, in our modeling, free energy density of the system in Fig. 3.1 is considered to be equivalent to the total surface tension present within the system. This treatment allows us to incorporate the T-dependence provided in Eq. (3.2) into Eq. (3.1), resulting in an expression for temperature-dependent bulk free-energy density function:

$$f_{bulk}(c_{(\mathbf{r},t)}, T) = A[c_{(\mathbf{r},t)} - c_1]^2[c_{(\mathbf{r},t)} - c_2]^2(1 - \frac{T}{T_c})^n \tag{3.3}$$

In general, the phase-field method is a natural extension of diffuse-interface models by Cahn and Allen[21], Ginzburg and Landau[22], and Cahn and Hilliard[23]. The equations involved in the phase-field approach are developed in terms of the following free-energy function:

$$F = \int_R [f_{bulk}(c_{(\mathbf{r},t)}, T) + \frac{1}{2}\epsilon|\nabla c_{(\mathbf{r},t)}|^2]dr \tag{3.4}$$

where R is the entire area of the system, $f_{bulk}(c_{(\mathbf{r},t)}, T)$ is the free-energy density function given in Eq. (3.3), $c_{(\mathbf{r},t)}$ is the concentration of charged particles.

Also in Eq. (3.4), ϵ is the interfacial gradient energy term relating to the energy stored per unit application of the potential of the gradient of $c_{(\mathbf{r},t)}$. This term represents potential energy of the interface, and it is assumed to be uniformly constant along the interface.

The dynamical evolution of charge-clusters as the system establishes thermal

equilibrium is obtained by invoking a general form of phase-field conservation law, as the first-order variation of free-energy functional equation results in phase-field flux:

$$\frac{\partial c_{(\mathbf{r},t)}}{\partial t} = \nabla.(M\nabla\frac{\partial F}{\partial c_{(\mathbf{r},t)}}) \tag{3.5}$$

where M is the mobility of the conserved variable representing a physical property of the system, and is assumed to be constant.

Eq. (3.5) is a Cahn-Hilliard phase field equation in concentration variable $c_{(\mathbf{r},t)}$ for the system, which incorporates effect of temperature on the movement of charge-clusters within the system. Self-consistent solutions to this equation are obtained by using the Multi-physics Object-Oriented Simulation Environment (MOOSE) finite-element platform [24, 25].

3.3 Evolution of Charge-Clusters at Fixed Temperature

To elucidate the sole effect of temperature T on the evolution of charge-clusters, a system of an 80 nm x 40 nm dielectric thin film, as shown in Fig. 3.2, was considered. The initial charge concentration $c_{(\mathbf{r},t)}$ at $t = 0$ was varied in the range between 0.1 and 0.8 randomly across the system. Dimensions for this system were set to values much larger than those of the system drawn in Fig. 3.1 so that variations in the grey-scale across the figure appears visible and distinct. A fixed temperature, T was

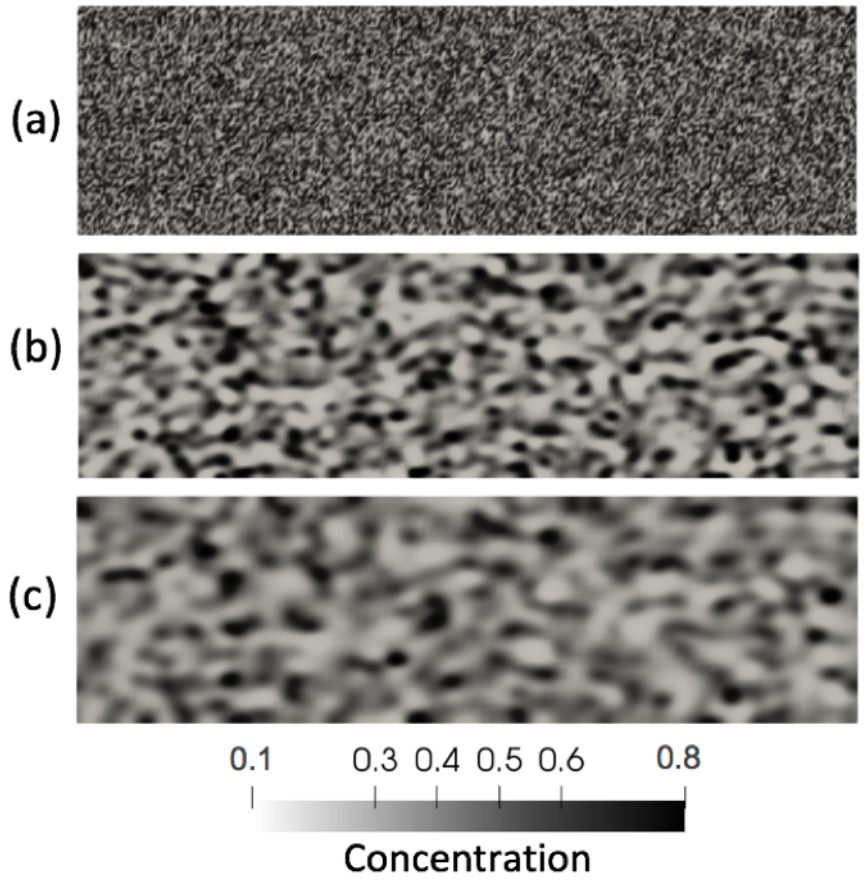

Figure 3.2: Effect of external temperature on Spinodal decomposition of a system with 80 nm x 40 nm dimension, consisting of conducting and non-conducting materials. (a) the initial state at t_0 defined by the initial charge concentration $c_{(\mathbf{r},t)}$ at t = 0 s. The evolusion of charge-clusters at t = 80 s, for (b) T = 350 K, and (c) T = 450 K.

uniformly applied to the system. Periodic boundary conditions were imposed on the left and the right sides of the system for $c_{(\mathbf{r},t)}$. To produce Fig. 3.2, simulation time was kept constant by purposely halting the computation before the system reached the absolute steady-state. The dynamic evolution of the charged clusters was captured at $T = 350$ K in Fig. 3.2(b) and $T = 450$ K Fig. 3.2(c), respectively, highlighting how the temperature, in the absence of electrical potential, contributes to defining the cluster boundaries as total free energy of the system reduces. High temperature forms clusters with less defined boundaries, whereas the boundary and shape of clusters formed at lower temperature are more well distinct. In the next section, the effect of electrical potential on the free energy formulation is studied at constant temperature $T = 400$ K.

3.4 Electro-thermal modeling: The formation and annihilation of ECFs

After evaluating the effect of temperature on dynamic distribution of $c_{(\mathbf{r},t)}$, the effect of applying external electric potential was assessed in an isothermal environment. The dependence of bulk free energy, F, on the external electric potential $V_{(\mathbf{r},t)}$ is incorporated by adding electrostatic energy g_{elec} to Eq.(3.4), resulting in the following formulation, where the external electric potential is coupled to the electric Laplace equation:

$$F = \int_R [f_{bulk}(c_{(\mathbf{r},t)}, T) + \frac{1}{2}\epsilon|\nabla c_{(\mathbf{r},t)}|^2 + g_{elec}(c_{(\mathbf{r},t)}, V)]dr$$

g_{elec} is defined as,

$$g_{elec}(c_{(\mathbf{r},t)}, V) = \frac{q}{\Omega} V_{(\mathbf{r},t)} c_{(\mathbf{r},t)} \tag{3.6}$$

where $V_{(\mathbf{r},t)}$ is the electrical potential applied across a dielectric thin film, q is the electric charge, and Ω is the differential volume unit of the mesh cell used in the system. Self-consistent solutions for $c_{(\mathbf{r},t)}$ can be calculated by incorporating Eq. (3.6) and Eq. (3.6) into Eq. (3.5),self-consistently coupled to electronic Laplace equation[26, 17]:

$$\nabla . \sigma(c_{(\mathbf{r},t)}) \nabla V_{(\mathbf{r},t)} = 0 \tag{3.7}$$

where electrical conductivity σ is a linear function of variable $c_{(\mathbf{r},\mathbf{t})}$. Solving the phase field conservation law allows us to demonstrate the formation and annihilation of ECFs in a dielectric thin film as the system seeks the minimum bulk free energy under the influence of the external electrical potential at a specific temperature and thermal force. The system used for the electro-thermal modeling is illustrated in Fig. 3.3(a) that represents a dielectric thin film at its pristine state. In Fig. 3.3(a), a conductive region - the lower region with uniform light contrast - with high $c_{(\mathbf{r},t)}$ is separated from a non-conductive region - the upper region with non-uniform contrast - with low $c_{(\mathbf{r},t)}$ by an interface along which $c_{(\mathbf{r},t)}$ varies.

Electrical resistance, along the thickness of the system in the pristine-state, is the highest throughout the lifetime of the system. Fig. 3.3(b) represents the programmed-state established when $V = 1$ V was applied to the top electrode while

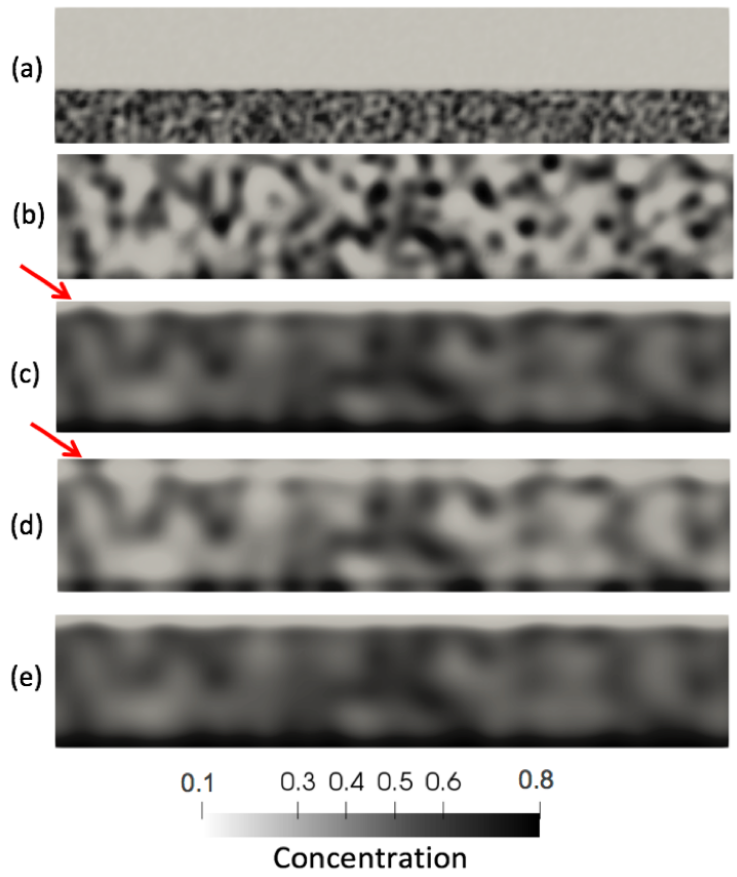

Figure 3.3: The system consisting of dielectric thin film with dimension of 50 nm x 10 nm at 400 K. (a) pristine state, (b) the first programmed state (ON-state) established by applying electrical potential V_{top} = 1 V and V_{bottom} = 0 V, and (c) the first erased state (OFF-state) obtained subsequently by reversing the electrical potential. (d) the second programmed state (ON-state) and (e) the second erased state (OFF-state) are obtained by repeating the biases as in (a) and (b), respectively .

the bottom electrode was set to 0V, and the temperature was set to 400 K. Under these conditions, multiple ECFs were formed, connecting the top and the bottom electrodes, resulting in low electrical resistance (i.e., ON-state). Fig. 3.3(c) represents the erased-state, and it was achieved when the polarity of the external electric potential was reversed. Fig. 3.3(c) shows that the ECFs were annihilated, resulting in high electrical resistance (i.e., OFF-state). The fact that the thickness of the non-conductive region near the top electrode in Fig. 3.3(c) is smaller than that of the non-conductive region in Fig. 3.3(a) indicates that electrical resistance of the OFF-state in Fig. 3.3(c) is lower than that in the pristine-state in Fig. 3.3(a),which is consistent with RRAM and memristor devices that experimentally exhibit initial electrical resistance at the pristine-state much higher than that of the OFF-state.

In order to study whether the location of the ECFs, formed in Fig. 3.3(b) and erased in Fig. 3.3(c), changes or remains unchanged, the polarity of external electric potential was reversed to establish the ON-state again in Fig. 3.3(d). A comparison between the two consecutive ON-states in Fig. 3.3(b) and (d) highlights that stochastic nature of the formation and evolution of charge-clusters can result in the formation of ECFs at different locations everytime the ON-state is established.

Also, the results indicate that the location at which the dominant ECF forms is likely to depend on a specific configuration of charge-clusters that continues to exist after an erase operation is completed. For instance, a large part of Fig. 3.3(b) with dark contrast representing the first ON-state, except the thin region underneath the uppoer boundary of the film, seems to resemble a bulk part of pf charge-clusters - the dark contrast - in Fig. 3.3(c), suggesting that the OFF-state was established

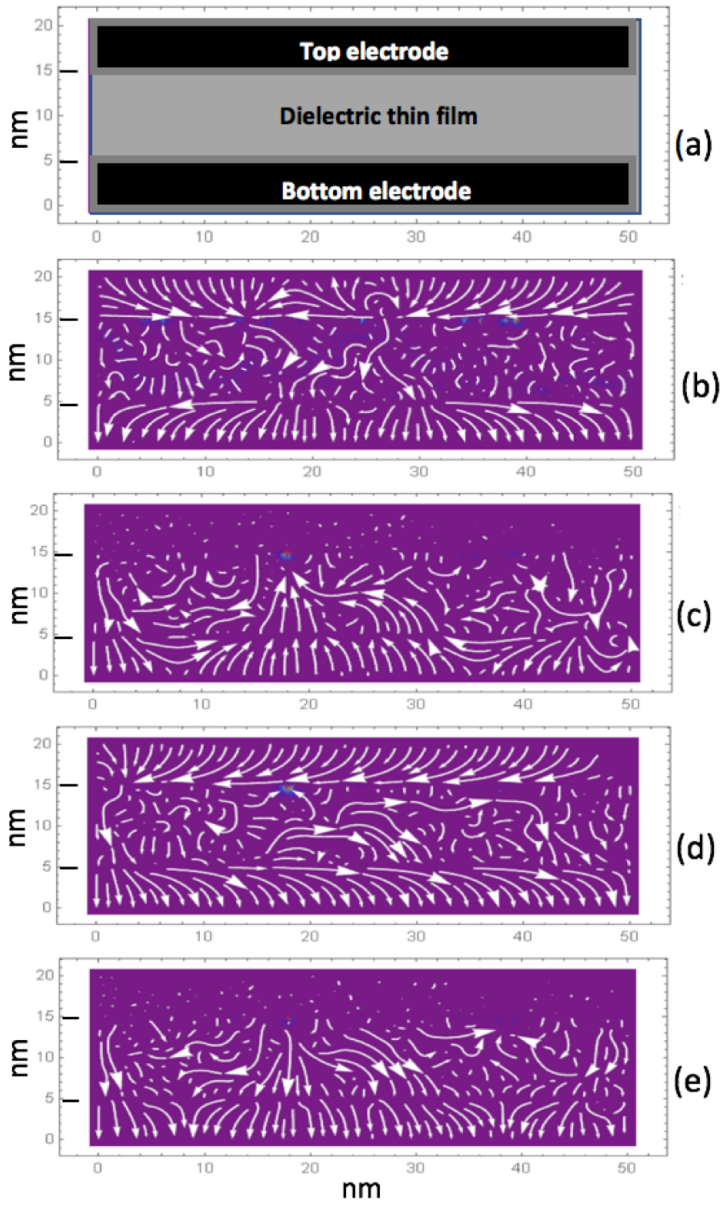

Figure 3.4: Electrical current density maps correponding to Fig. 3.3(b)-(e). A pair of 5 nm thick electrodes are added to the top and bottom of the film in Fig. 3.3, as shown in (a). Local electrical current density is represented by arrows with length proportional to the relative magnitude and direction of local electric current flow in (b) the first ON-state, (c) the first OFF-state, (d) the second ON-state, and (e) the second OFF-state.

when a fraction of the ECFs present in the ON-state in Fig. 3.3(b) is ruptured (Note: the total number of charges is conserved, no matter what state the film is set to). Moreover, in Fig. 3.3(c), the thickness of non-conducting region near the top left corner, as indicated by an arrow, is much smaller than that of the other locations along the lateral extend of the film. As a result, the dominant ECF in Fig. 3.3(d) representing the second ON-state was formed preferentially at the same location pointed by an arrow, presumably because electric field at that location was strongest during a program operation.

In Fig. 3.3(e), the electric potential across the film was again reversed to establish the second OFF-state, which is similar to the first OFF-state shown in Fig. 3.3(c). Specific configurations of charge-clusters presented in Fig. 3.3(b)-(e) were translated into electrical current density maps as presented in Fig. 3.4(b)-(e). In order to appropriately observe how electrical current is funneled form the top electrode into ECFs, a pair of 5 nm thick electrodes were added to the dielectric film in Fig. 3.3, as shown in Fig. 3.4(a). The electrical current density maps were obtained by applying electrical potential of 100 mV to the top electrode, while the bottom electrode was set to 0 V. When the system is in the ON-state as depicted in Fig. 3.4(b), a few long downward arrows - those appearing near the center (x ˜25 nm) of the film - connecting the top electrode and the bottom electrode, clearly indicate the presence of dominant ECFs that funnel electrical current flowing laterally across the top electrode.

As described earlier in reference to Fig. 3.3(c), in the second ON-state depicted in Fig. 3.4(d), the location of dominant ECFs - the downward arrows close to the left boundary (x ˜2 nm) of the film - is different from that in the first ON-state in

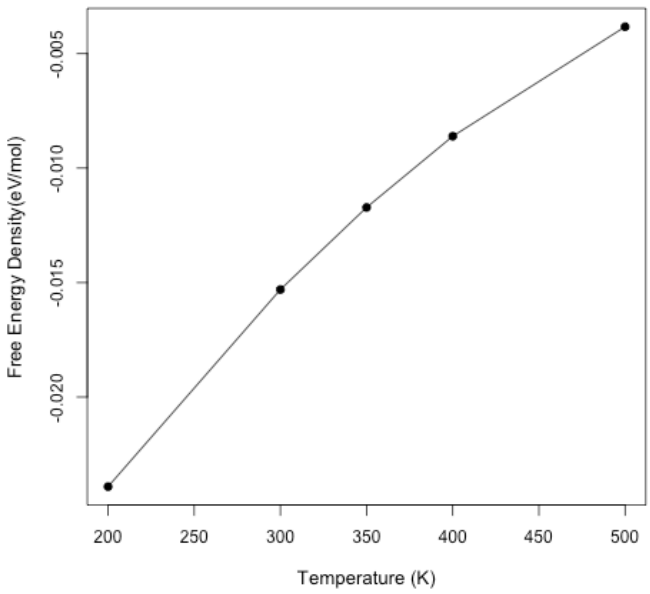

Figure 3.5: The free energy density of the system at various temperatures. In this simulation, critical Temperature T_c was set to 700K, and empirical factor n was set to 2 in Eq.(3.2). The number of iterations to reach the convergence criteria for a self-consistence solution to Eq.(3.6) and (3.6) was fixed so that the system did not reach the ground state at various temperatures.

Fig. 3.4(b). In contrast, as the electric potential was reserved, and thus, the system entered the OFF-state, all the arrows present in the top electrode in Fig. 3.4(b) and (d) disappeared as seen in Fig. 3.4(c) and (e), indicating that the flow of electrical current discontinued.

Fig. 3.3 and Fig. 3.4 were obtained for T=400K. In order to evaluate the overall endurance of resistive switching, which is expected to depend on the stability of the formation and annihilation of ECFs, the effect of increasing temperature on the free energy density of the system was assessed. In the assessment, the number

of computational iterations was fixed in reducing the free energy density for the pristine-state shown in Fig. 3.3(a) at various temperatures.

In other words, at various temperatures, the iteration process was stopped when a certain number of iterations was completed and the free energy density of the system was recorded. Fig. 3.5 shows the free energy density obtained at various temperatures, indicating that reduction in the free energy density gradually decreases as the temperature increases, that is to say that as the temperature increases, it takes longer time for the system to reach minimum free energy. Thus, the stability of the system – reversible and repeatable formation and annihilation of ECFs – reduces as the temperature rises.

This trend is valid as long as the temperature is well below T_c of the system. In other words, as the T_c is expected to be determined by a range of factors including material properties, it needs to be as high as possible for the device to achieve high stability at a larger temperature range.

3.5 Summary

Electro-thermal characteristics of the formation and annihilation of ECFs in a system consisting of dielectric thin film were studied by using the phase-field method. The dependence of interface energy on local temperature and variations in local concentration of electrical charges were formulated and incorporated in describing bulk free energy density of the system in our treatment based on the Cahn-Hilliard

model. Our results suggest that as the system temperature increases, the formation of charge-clusters with weakly defined boundaries is promoted, meaning the free energy density of the system reduces at slower rates as the system temperature drifts higher. In other words, the stability of the system - reversible and repeatable formation and annihilation of ECFs - reduces as the temperature rises, extending significant implications for endurance of resistive switching devices and memristors.

Our results clearly illustrate that the formation and annihilation of ECFs are distinctly accomplished by controlling electrical potential applied across the system as resistive switches and memristors normally operate. The resulting electrical current density maps provide a realistic representation of a complex conducting path through which electrical current flows in the ON-state and a cessation of the electrical current when ECFs rupture in the OFF-state as the polarity of applied electrical potential is reversed across the system. A significant difference in the thickness of non-conductive regions in the pristine-state and in the OFF-state highlights that the irreversible characteristics of a process often referred to as electroforming result in the resistance of the OFF-state to be always smaller than that of the pristine-state. Furthermore our results clearly capture the stochastic nature of the formation of ECFs in consecutive program operations, and distinctively show how partial rupture of ECFs in erase operations leads to a preferential formation of ECFs during the subsequent program operation.

References

[1] M. Seul and D. Andelman. Domain shapes and patterns: The phenomenology of modulated phases. *Science*, 267:476–483, 1995.

[2] R. Waser and M.Aono. Nanoionic–based resistive switching memories. *Nature Materials*, 6:833–840, 2007.

[3] Foroozan S. Koushan and Sung-Mo Steve Kang. Resistance-based embedded non-volatile memories as synaptic devices in spiking neural networks. *IEEE Workshop on Microelectronics and Electron Devices (WMED)*, pages 1–4, 2019.

[4] J. P. Stracha, M. D. Pickett, J. J. Yang, and et al. Direct identification of the conducting channels in a functioning memristive device. *Adv Mater*, 22:3573–3577, 2010.

[5] V. R. Nallagatla, J. Jo, S. K. Acharya, and et al. Confining vertical conducting filament for reliable resistive switching by using a au–probe tip as the top electrode for epitaxial brownmillerite oxide memristive device. *Sci. Rep.*, 9:1188, 2019.

[6] F. Nardi, S. Larentis, and et al. Resistive switching by voltage–driven ion migration in bipolar rram–part 1: Experimental study. *IEEE Transactions on Electron Devices*, 59:2461–2464, 2012.

[7] S. Kim, S. J. Kim, K. M. Kim, and et al. Physical electro–thermal model of resistive switching in bi–layered resistance–change memory. *Scientific reports*, 3:1680, 2013. DOI:10.1038/srep01680.

[8] E.W. Lim and R. Ismail. Conduction mechanism of valence change resistive switching memory: A survey. *Electronics*, 4:586–613, 2015.

[9] D.B. Strukov, G.S. Snider, and et al. The missing memristor found. *Nature*, 443:80–83, 2008.

[10] J. J. Yang, M. D. Pickett, and et al. Memristive switching mechanism for metal/oxide/metal nanodevices. *Nat. Nanotechnol.*, 3:429–433, 2008.

[11] X. Guan, S. Yu, and H-SP. Wong. A spice compact model of metal oxide resistive switching memory with variations. *IEEE Electron Device Lett*, 33:1405–1407, 2012.

[12] J. Kwon, A. A. Sharma, C. Chen, and et al. Transient thermometry and high–resolution transmission electron microscopy analysis of filamentary resistive switches. *IEEE Transactions on Electron Devices*, 8(31):20176–10184, 2016. DOI:10.1021/acsami.6b05034.

[13] S. Asapu and T. Maiti. Electrothermal numerical modeling of multifilamentary conduction in ta_2o_5 and wo_3 bilayer oxides based rram. *Ferroelectronics*, 500:1:229–240, 2016. DOI:10.1080/00150193.2016.1217706.

[14] S. Asapu and T. Maiti. Multifilamentary conduction modeling in transition metal oxide–based rram. *IEEE Transactions on Electron Devices*, 64(8):3145–3150, 2017.

[15] Long-Qing Chen. Phase–field models for microstructure evolution. *Annual Review of Materials Research*, 32(1):113–140, 2002. DOI:10.1146/annurev.matsci.32.112001.132041.

[16] Y. Shindo, F. Narita, and T. Kobayashi. Phase field simulation on the electromechanical response of poled barium titanate polycrystals with oxygen vacancies. *Journal of Applied Physics*, 117, 2015.

[17] J. F. Sevic and N. P. Kobayashi. A computational phase field study of conducting channel formation in dielectric thin–films: A view toward the physical origins of resistive switching. *Journal of Applied Physics*, 126:6, 2019.

[18] Neil Kensington Adam. *The Physics and Chemistry of Surfaces.* Oxford University Press, 1941.

[19] E. A. Guggenheim. The principle of corresponding states. *Journal of Chemical Physics*, 13:253–261, 1945.

[20] Valery I. Levitas, Bryan F. Henson, Laura B. Smilowitz, and Blaine W. Asay. Solid–solid phase transformation via internal stress-induced virtual melting, significantly below the melting temperature. application to hmx energetic crystal. *The Journal of Physical Chemistry B*, 110(20):10105–10119, 2006. PMID: 16706472.

[21] J. Cahn and S. Allen. A Microscopic Theory of Domain Wall Motion and Its Experimental Verification in Fe–Al Alloy Domain Growth Kinetics. *Journal de Physique Colloques*, 38(C7):C7–51–C7–57, 1977. doi: 10.1051/jphyscol:1977709.

[22] V.L. Ginzburg and L.D. Landau. On the theory of superconductivity. *Zh. Eksp. Teor. Fiz.*., 20:1064–1082, 1950. doi: 10.1016/B978-0-08-010586-4.50078-X.

[23] J.W. Cahn and J.E. Hilliard. Interfacial free energy of non–uniform system. *Journal of Chemical Physics*, 28(2), 1958. doi:10.1063/1.1744102.

[24] D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandié. Moose: A parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239:1768–1778, 2009.

[25] M. R. Tonks, D. Gaston, P. C. Millett, D. Andrs, and P. Talbote. An object–oriented finite element framework for multiphysics phase field simulations. *Computational Materials Science*, 1:20–29, 2012.

[26] N. Provatas and K. Elder. *Phase–Field Methods in Materials Science and Engineering.* Wiley–VCH Verlag, Weinheim, 2010.

Chapter 4

Nucleation and its impact on data retention and the stability of ON and OFF states in thin-film dielectric memristor, using phase-field methodology

4.1 Introduction:

The emergence of novel resistive memories such as resistive RAM (RRAM) [1], phase change memory (PCM) [2], and spin transfer torque RAM (STTRAM) [3], aiming at replacing or complementing flash memories and other silicon-based memories including dynamic random-access memory and static random-access memory, offers many advantages in terms of size, speed, and scaling. Among these resistive memories, filamentary RRAM deems to be most promising from the perspective of its structural simplicity, ease of integration at the back-end-of-line (BEOL), scalability, and fast switching speed [4, 5]. However, for a successful implementation of the technology, various challenges such as variabilities of electrical properties at the device-to-device and cycle-to-cycle level [6], random telegraph noise [7], and loss of data [8].must be resolved. Of these challenges, the loss of data refers to the loss of the ability of retaining low-resistance-state (i.e., ON-state) and/or high-resistance-state (i.e., OFF-state) – the retention loss.

The retention loss is found to be the trade-off [9], that is, an improvement in the retention of data often results in a degradation of endurance and vice versa. Furthermore, the retention of data is found to be adversely interfered with changes in temperature, application of SET/RESET biases, and structural alternations occurring at the bottom and top electrode interfaces [10]. The loss of retention that occurs over a period of time shorter than 1 minute, is attributed to statistical fluctuations of electrical conductance [11], whereas the retention loss that takes place over a period of time longer than 1 minute is often modeled based on the diffusion

of oxygen vacancies [12].

Regardless of underlying physics, the retention loss results when resistance of OFF-state, R_{HRS}, decreases over time and/or resistance of ON-state, R_{LRS}, becomes unstable and increases over time, even when the device is no longer under external electrical bias. Additionally, reversible transitions between OFF-state and ON-state – cyclic resistive switching – are often interpreted as the formation and annihilation of electrically conducting filaments (ECFs) [13, 14, 15]; thus, one possible scheme for the retention loss can be illustrated by allowing a dielectric film – switching layer – responsible for the resistive switching to evolve structurally away from those that define OFF-state and ON-state. Such structural evolutions in the switching layer with or without the presence of ECFs are delineated by, for instance, introducing the nucleation of clusters – nuclei – made of electrical charges, which is described in the context of the formation of metallic nuclei that can occur in dielectric films under the influence of high electric field and lead to such a phase transition as the insulator-metal transition[16]. Furthermore, studies have shown that even when the metallic phase is energetically unfavorable, the insulator-metal transition can still take place with electric field being sufficiently high[17], leading to the development of models based on the formation of electrically conducting nuclei in describing switching mechanisms for PCM and ferroelectric memory[18].

Since the nucleation of charge-clusters under the influence of electric field appears to be responsible for the insulator-metal transition regardless of the underlying microscopic mechanisms (e.g., densification, crystallization, electron solvation)[19], it is sensible to extend this view to filamentary RRAM and memristors in which elec-

trically insulating regions in the switching layer experience localized electric field during cyclic switching operations [20]. In RRAM, relative magnitude of chemical potential of insulating, unstable conductive, and metastable conductive phases and their corresponding thermodynamic barriers determine the nucleation and growth of charge-clusters, and thus, their effects on OFF-state and LRS[21]. Emerging charge-clusters in the switching layer can potentially modify fractions of ECFs present in OFF-state and resulting in reduction of ROFF; they can also possibly connect multiple ECFs present in the switching layer at ON-state and push ROFF to even lower resistance and cause a failure during an erase operation. Furthermore, since the number of charges in a switching layer is conserved, the emergence of charge-clusters can also fracture ECFs, resulting in the retention loss of ON-state. Given all these possible scenarios exhibited by the formation of charge-clusters, in this paper, we incorporate the nucleation and growth of charge-clusters in a switching layer of RRAM initially set at either ON-state or OFF-state by leveraging our previously developed approach based on the phase-field method in illustrating the formation and annihilation of ECFs [15] in order to describe the retention loss of RRAM. Our results suggest that even if charge-clusters do not contribute to the resistive switching of RRAM during cyclic operations, they can potentially play a crucial role in setting a root cause of the retention loss for both OFF-state and ON-state.

4.2 Nucleation and Cahn-Hilliard Model

Classical nucleation theory (CNT) developed by Becker and Döring[22] is based on a thermodynamic approach by which the system's Gibbs free energy is minimized, using energies associated with macroscopic entities such as surface to develop expressions for the rate of nucleation. This thermodynamic approach is eventually extended to various types of phase transitions[23] being established as a traditional way of describing the crystallization of solid. In general, there are two types of nucleation: the nucleation that occurs at nucleation sites located on solid surfaces contacting liquid or vapor is referred to as heterogeneous nucleation. In contrast, homogeneous nucleation occurs spontaneously and randomly in a host phase brought to a supercritical state such as a supersaturation.

The presence of external electric field influences the homogeneous nucleation, and the rate of the nucleation depends on the ratio of the dielectric constant of solution and that of solid[24]. Several experimental studies clarified the influence of electric field on nucleation processes[25, 26], highlighting the existence of critical electric field above which nucleation was induced. For example, electric field in the range of 0.1 to 1 MV/cm was found to discernibly increase the rate of nucleation of ice[27]. Given these findings, our premise is that the homogeneous nucleation of charge-clusters occurs as a consequence of the presence of local electric-field higher than the average electric-field defined globally by the applied electric potential across a switching layer in a memristor. Consequently, the likelihood of the formation of charge-clusters increases as a switching layer undergoes increasing number of switch-

ing cycles throughout its lifetime (i.e., the total time over which a switching layer is stressed by the local high electric-field).

While the CNT describes the nucleation, the growth of nuclei is expressed as the evolution of size distribution of nuclei over time[28]. In general, the change in the total free energy ΔF associated with the formation of a nucleus with radius r in a homogeneous system is composed of a term representing the reduction in the free energy due to the generation of a spherical nucleus $V\Delta f_{(c)}$ and a term due to an increase in the interfacial energy $A\gamma$:

$$\Delta F = f_n = -V\Delta f_{(c)} + A\gamma = -\frac{4}{3}\pi r^3 \Delta f_{(c)} + 4\pi r^2 \gamma \tag{4.1}$$

where V is volume and A is the area of nucli center, γ is the surface energy, $\Delta f_{(c)}$ is the free energy associated with the volume of the nucli site, and $c_{(r,t)}$ is the composition of the nucleus.

In an inhomogeneous system, the composition $c_{(r,t)}$ is a field and not a scalar variable, thus, the Cahn-Hilliard model adds a correction to its homogeneous free energy function to account for spatial inhomogeneity:

$$F = \int_V [f_n(c_{(r,t)}) + \frac{1}{2}\epsilon|\nabla c_{(r,t)}|^2]dV \tag{4.2}$$

where $f_(c_{(r,t)})$ is the homogeneous free energy, and gradient energy, $\epsilon(\nabla c_{(r,t)})^2$, provides the first-order correction for the inhomogeneity, allowing interfacial energy to be modeled in the phase-field approach. Since $c_{(r,t)}$ is a conserved quantity, the dynamical evolution of $c_{(r,t)}$ is obtained by invoking a general form of phase-field

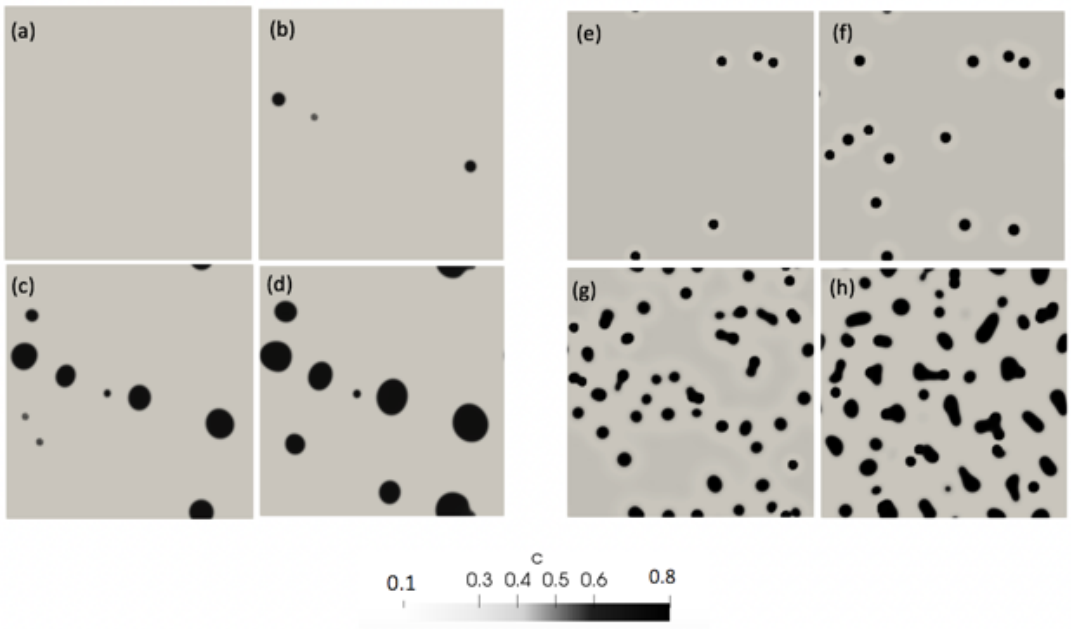

Figure 4.1: The nucleation and growth of clusters in a 500 nm x 500 nm dielectric film over time. periodic boundary conditions are assumed for all around the system in both x and y directions. Panels (a) through (d) represent nucleation growth when small probability function; whereas panels (e) through (h) represent nucleation growth as the probability function increases by 1.5 order of magnitude. Nucleation growth are compared at fixed time interval for each simulation: (a) and (e)$t = 25\ s$, (b) and (f)t =75 s , (c) and (g)t =100 s , (d) and (h)t =150 s.

conservation law, as the first-order variation of the free-energy functional equation which eventually results in:

$$\frac{\partial c_{(\mathbf{r},t)}}{\partial t} = \nabla . M \nabla \left[\frac{\partial f_n(c_{(r,t)})}{\partial c_{(\mathbf{r},t)}} - \nabla . \epsilon \nabla c_{(r,t)} \right] \tag{4.3}$$

In the use of the phase-field method, the Eq. (4.3) was applied to a large system of 500 nm x 500 nm to illustrate the nucleation and capture the growth of charge-

clusters over time. Within the system, charges were initially distributed at $c_{(\mathbf{r},\mathbf{t})}$ randomly chosen in the range of 0 to 0.3 representing low conductivity, leading to a spontaneous nucleation driven by the presence of variations in $c_{(\mathbf{r},\mathbf{t})}$, through the concentration dependent probability function, as the periodic boundary conditions were imposed for the top and bottom, and the left and right sides of the system in Fig. 4.1. Starting with classical nucleation theory, nucleation rate was defined to be:

$$R = P_n \exp\left(\frac{-\Delta f_n^*}{K_B T}\right) \tag{4.4}$$

where Δf_n^* is the critical free energy of nucleation at which an stable nucleus is achieved, P_n is the nucleation probability function which depends on the mass and density of charge-clusters in interaction with dielectric layer in the system [29], k_B is the Boltzmann constant, and T is the temperature.

Since the studies [14, 29, 30] have shown that among all the parameters, supersaturation and charge density have the highest impact on nucleation probability, then in these simulations we used a concentration-dependent function to represent nucleation probability. Fig. 4.1(a) (d) and Fig. 4.1(e) and Fig. 4.1(h) display two examples obtained by using two different probability functions of $c_{(\mathbf{r},\mathbf{t})}$*1e-7 and $c_{(\mathbf{r},\mathbf{t})}$*5e-6, respectively. In each of the two cases, a series of panels shows chronological evolution of the formation and growth of charge-clusters for time intervals of 25 s, 55 s, 100 s, and 150 s as indicated on the figures. The formation of a charge-cluster in this figure is defined by the emergence of a region with $c_{(\mathbf{r},\mathbf{t})} > 0.7$. These results highlight the dependence of the population and size distribution of charge-clusters overtime on the nucleation rate; a larger nucleation probability results in higher number density but

smaller size nuclei centers, whereas lower probability function results in less number of nuclei centers, but larger in size over comparable time range.

In the next section, we describe how the formation and growth of charge-clusters leads to reliability degradation of memristor devices.

4.3 The Emergence of Nuclei Centers and Retention Loss

Resistive switching through the formation and annihilation of ECFs in a dielectric film – switching layer – is depicted by coupling electrical and thermal transport using the Cahn-Hilliard phase-field model[15]. The results highlight how the formation and evolution of charge-clusters within the switching layer are driven under the influence of electric potential applied across the layer.

The findings also confirm that existing experimental results that ROFF is always lower than that of the pristine state (i.e., the state established in as-fabricated resistive switches before a necessary conditioning often referred to as electroforming is performed). ROFF established during a reset operation is dominated by the gap between one of the two electrodes and the tip of an ECF located closest to the electrode; thus, the highest electric field E_{gap} is expected to appear over the gap during the rest of the reset operation and the subsequent set operation. It is this E_{gap} that would initiate the formation of charged-clusters as discussed in section 4.1. The formation of charged-clusters is illustrated in the view of the classical nucleation theory that encompasses an initial slow nucleation stage and a subsequent fast nucleation stage

before the coalescence of nuclei (i.e., the formation of charged-clusters) as described in section 4.2.

Fig. 4.2(a) shows a system made of a $50nm$ x $10nm$ switching layer in its pristine state. The system consists of two distinctive regions: the lower region being electrically conducting and the upper region being electrical insulating. The conducting region is represented by varying contrast that signifies local variations in relative charge density $c_{(\mathbf{r},\mathbf{t})}$. The conducting region is distinctly separated from the upper region – the insulating region with uniform contrast of $c_{(\mathbf{r},\mathbf{t})}$ – by an interface along which $c_{(\mathbf{r},\mathbf{t})}$ varies.

At $t = 0\ s$, the variations in $c_{(\mathbf{r},\mathbf{t})}$ in the conducting region was randomly assigned to relative values in the range of $0.7 < c_{(\mathbf{r},\mathbf{0})} < 0.9$ while $c_{(\mathbf{r},\mathbf{0})}$ was set to $0.1 < c_{(\mathbf{r},\mathbf{0})} < 0.3$ in the insulating region. Fig. 4.2(b) represents ON-state established by applying electrical potential of $1\ V$ to the top electrode (i.e., the upper bound of the insulating region in Fig. 4.1(a)) at the ambient temperature of $400\ K$. The presence of multiple ECFs connecting the top electrode and the bottom electrode are readily identified. Subsequently, the polarity of the electrical potential was reversed to obtain OFF-state as show in Fig. 4.2(c) which shows that the ECFs that existed in ON-state were ruptured. A gap is clearly visible below the top boundary.

As described in the previous section, areas, within a switching layer, that experience electric-field locally, much higher than the nominal electric filed (i.e., applied electric potential divided by the physical thickness of a switching layer) – hot spots – during cyclic operations are most likely to be the epicenter of the nucleation of charge-clusters. In Fig. 4.2(b) and Fig. 4.2(c), such hot spots are shown as gaps in

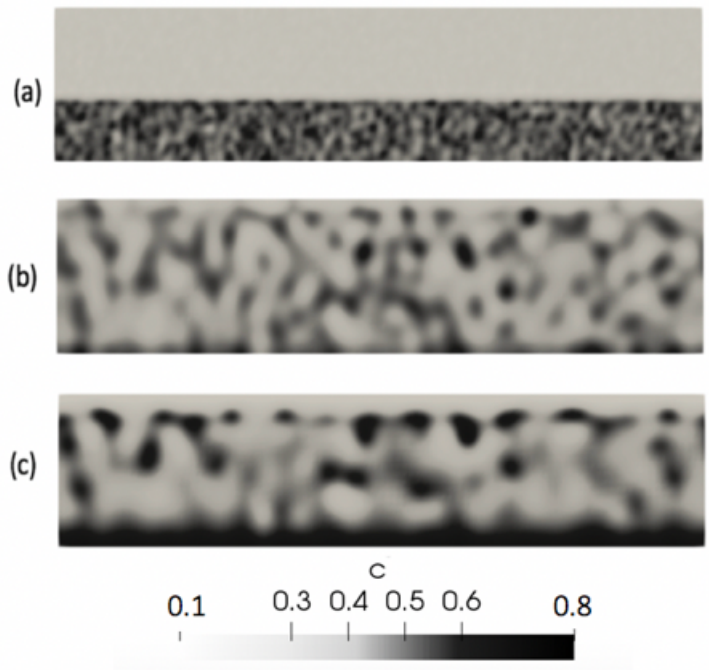

Figure 4.2: A switching layer with dimension of $50nm$ x $10nm$ set at temperature of $400\ K$. (a) pristine-state, (b) ON-state established by applying electrical potential of $1V$ to the upper bound of the layer and (c) OFF-state obtained subsequently by reversing the polarity of the electrical potential

lighter contrast between two adjacent conductive regions each of which is connected to an opposing electrode. With the system being in OFF-state, a SET operation causes the birth of hot spots where nuclei centers form and remain, eventually leading to ON-state as a result of the nucleation growth and formation of ECF . On the other hand, with the system being in ON-state, a RESET operation causes the annihilation of ECFs, which subsequently generates gaps between remnants of ECFs; these gaps act as hot spots during a subsequent SET operation. In a microscopic view, the spatial distribution of gaps/hot spots is expected to vary because of the random nature of the formation and movements of charge-clusters.

In our assertion, the degree at which ON-state and OFF-states remain stable depends on how nuclei centers that are charge-clusters in nature and are formed during the cyclic switching operation grow, evolve and rearrange themselves after the operation is ceased by removing electric potential. Self-diffusion of charge-clusters and/or fractions of charge-clusters is expected to govern the rearrangement since no electric potential operates, and thus, locations and the number of nuclei – the precursor of a charge-cluster – that form during a SET and a RESET operations are expected to influence the retention. In phase-field simulation, a free energy penalty based approach is used to incorporate nucleation by avoiding direct modification of concentration, but modifying local energy density to make nucleation state a lower energy state, forcing the neighboring charge-clusters to diffuse toward the nucleation site. This approach allows us to write the total homogeneous free energy density of the system, f_T, to be comprised of bulk free energy density of as two-phase system[15] – a system consisting of electrically conducting and non-conducting regions and nucleation free

energy density f_n:

$$f_T(c,T) = f_{bulk}(c,T) + f_n(c,T) \tag{4.5}$$

This expression of f_T is used in the phase-field method to calculate the total free energy of the system that needs to be minimized:

$$F_{(c,T)} = \int_R [f_T(c_{(\mathbf{r},t)},T) + \frac{1}{2}\epsilon|\nabla c_{(\mathbf{r},t)}|^2]dr \tag{4.6}$$

where R is the entire area of the system, ϵ is the interfacial gradient energy term relating to the energy stored per unit application of the potential of the gradient of $c_{(\mathbf{r},t)}$. This term represents potential energy of the interface, and it is assumed to be uniformly constant along the interface.

The dynamical evolution of ECFs originated at the interface formed between a conducting and nonconducting regions is described by modified Cahn-Hilliard equation:

$$\frac{\partial c_{(\mathbf{r},t)}}{\partial t} = \nabla.(M\nabla\frac{\partial F_{(c,T)}}{\partial c_{(\mathbf{r},t)}}) \tag{4.7}$$

$$\frac{\partial c_{(\mathbf{r},t)}}{\partial t} = \nabla.M\nabla\left[\frac{\partial f_T(c,T)}{\partial c_{(\mathbf{r},t)}} - \nabla.\epsilon\nabla c_{(\mathbf{r},t)}\right] \tag{4.8}$$

where M is the mobility of the conserved variable $c_{(\mathbf{r},t)}$, and is assumed to be constant. In order to incorporate Eq. 4.8 to evaluate effect of nucleation in thin film

dielectric switching behavior, the following methodology was used to find the most likely locations within the system to introduce nuclei centers.

Since the probability of nuclei site generation is highest in regions with higher electric field, as explained in the previous section, an electric field map of the device in the OFF-state generated after addition of a thin top electrode and a bottom electrode across the thin film dielectric, as illustrated in Fig. 4.4, was used to find, regions experiencing high electric field, which according to this map were mainly located in the thin gap separating the top electrode from the conductive charge clusters connected to the bottom electrode. These regions would and potentially have the highest probability to be the epi-center of nuclei sites.

Adopting this methodology, we started with Fig. 4.3(a) illustrating the OFF-state of the thin film dielectric after five SET/RESET switching cycles. The impact of cyclic switching operations is pronounced in the overall blurriness of charge-clusters seen in Fig. 4.3(a) in comparison to those seen in Fig. 4.2(a).

The yellow arrow in Fig. 4.3(a) points to a selected gap region separating top electrode from conductive region connected to the bottom electrode, as a representative of regions experiencing high electric field during cycling, thus having highest probability of becoming the host of nuclei sites. In Fig. 4.3(b), two nucleus were introduced, as indicated by the red arrow, in the selected location. The numbers of nuclei added to the system are expected to be driven by the cumulative electrical stress the system has experienced. For this simulation we have made an assumption that minimum two nuclei centers were generated within five SET/RESET cycles. Considering Fig. 4.3(b) as when the cycling stopped and electrical bias across the

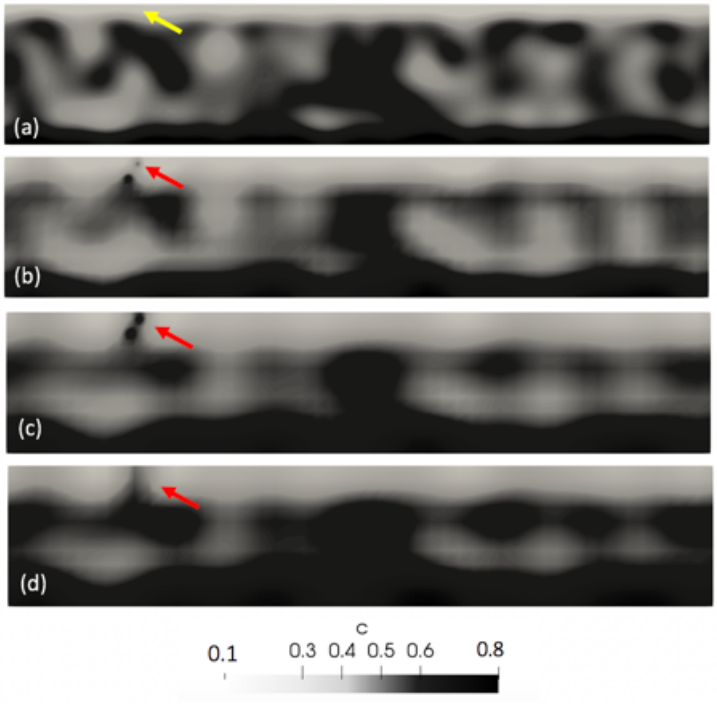

Figure 4.3: Formation and growth of charge-clusters in the system in the OFF-state, when T=400K. (a) the system in the OFF-state, after five cycles, shows the presence of a continuous non-conducting region near the top, with yellow arrow pointing to the region with smallest gap where nuclei centers will grow in the next panels after multiple cycles, (b) formation of two nuclei as indicated by the red arrow after one additional cycle. At this point electric field is removed and this state is considered at $t = 0$ for retention study, (c) the nuclei centers growing into a charge-cluster and eventually merging into a fractured ECF at t=40s, (d) and then forming a complete ECF, which results in the retention loss of the OFF-state, at t=100s

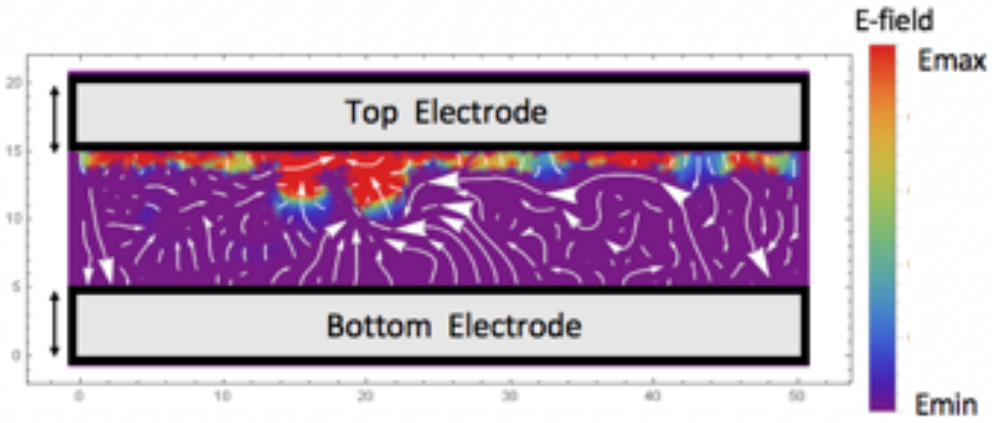

Figure 4.4: Electric field map of a system after RESET operation. A top electrode and a bottom electrode with 5 A thickness are added to the system and e-field is mapped with the system under electric potential difference of 1 V.

device was removed at $t = 0$, then Fig. 4.3(c) and Fig. 4.3(d) illustrate the growth of these nuclei over time, at $t = 40$ s and $t = 100$ s respectively; the two nuclei eventually merged and grew connecting the top electrode to the upper portion of the fractured ECFs as the free energy of the system is reduced. Using Fig. 4.3(b) and Fig. 4.3(d), electrical current density $j_{(r)}$ maps were obtained, as illustrated in Fig. 4.5(a) and Fig. 4.5(b), after addition of a 5 nm conductive layer as the top electrode and a 5 nm conductive layer as the bottom electrode to the switching layer as indicated by double-headed black arrows in Fig. 4.5. The $j_{(r)}$ maps were produced by applying an electric potential of $V_{top} = 100$ mV to the top electrode while the bottom electrode was grounded.

The magnitude and direction of current density at a specific location is expressed by the length and direction of an arrow. Lengths of white arrows are relative only within a map; relative electrical current calculated by integrating current density

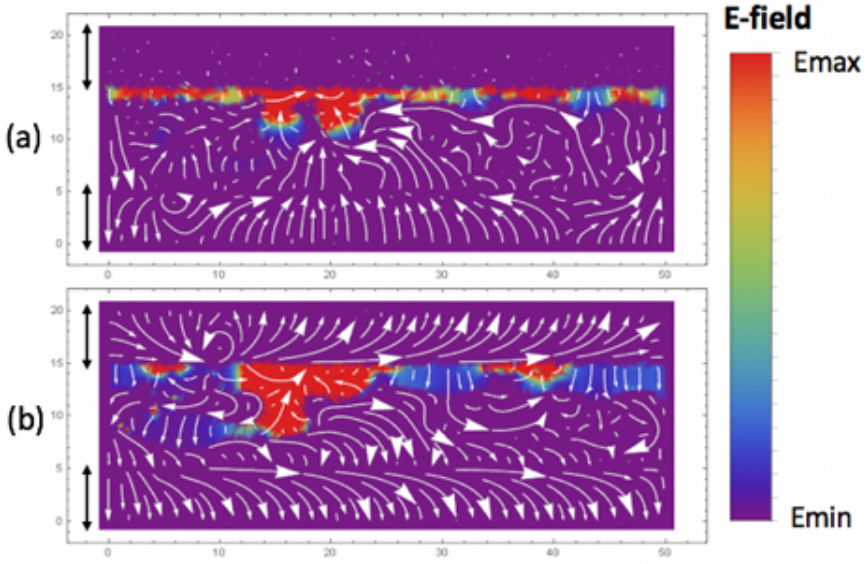

Figure 4.5: Current density maps of (a) the OFF-state as established by running five SET/RESET cycles and (b) after the formation of an infelicitous ECF. (a) $j_{(r)}$ vanishes in the top electrode as the system is set to the OFF-state as shown in Fig. 4.3(b). (b) $j_{(r)}$ is continuous from the top to the bottom electrode, indicating the presence of a ECF as seen in Fig. 4.3(d), indicating that the OFF-state established in panel (a) is destroyed and the system is in an erroneous OFF-state.

along the width (i.e., 0 - 50 nm) of the switching layer is 2x10-6 and 7 for Fig. 4.5(a) and Fig. 4.5(b), respectively. The $j_{(r)}$ maps clearly highlight that, even with a system being in the OFF-state and without electrical potential, an ECF forms as a result of the generation of nuclei centers during the previous SET/RESET operations, and their growth while the free energy of the system reduces, could potentially result in the loss of retention of the OFF-state that was established during the previous RESET operation.

Similarly, the introduction of nuclei to the system that is left in the ON-state was studied in comparison to the observations made for the OFF-state. Fig. 4.6(a) shows an example of the initial ON-state prepared after five SET/RESET switching cycles.

In a filamentary memristor, even though the electric field difference across the fully formed ECFs during SET operation reaches zero, the rest of the dielectric with no ECF experiences high electric field stress, similar to the stress applied during the RESET operation. So in our simulations, the same approach as explained for the OFF-state, was used to identify hot spots for introduction of two nuclei centers, as shown in Fig. 4.6(b), pointed by the red arrow. Fig. 4.6(c) and Fig. 4.6(d) illustrate the growth of these nuclei over time, at $t = 40\ s$ and $t = 100\ s$ respectively, after the removal of electric bias from the system.

According to the simulation results, the nuclei eventually merged and grew into a fractured ECFs as the total free energy of the system was reduced, and while the dominant ECF in the center of the figure was resolved within the system. For a better illustration, current density maps for Fig. 4.6(b) and (Fig. 4.6d), were produced

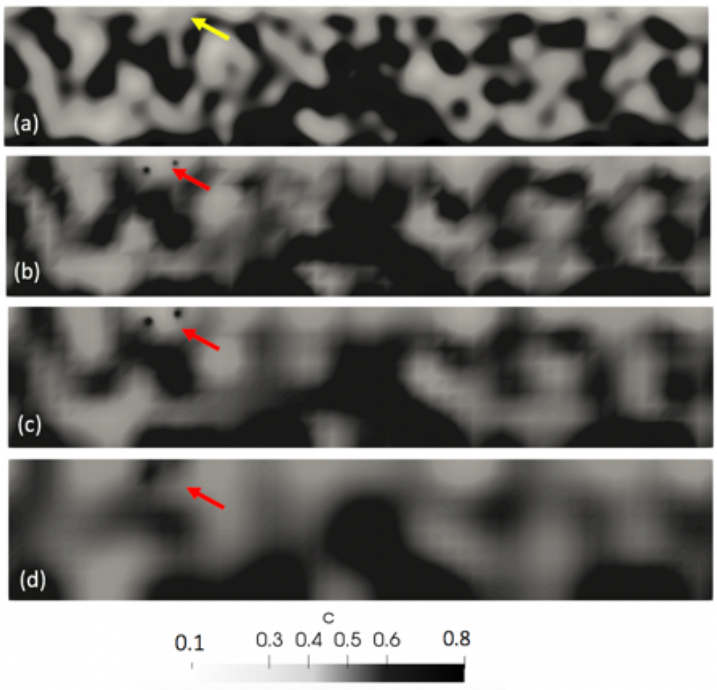

Figure 4.6: Generation and Growth of nuclei centers in thin-film dielectric after removal of electric field and placement of the system in the ON-state. (a) system in ON-state, with multiple ECFs connecting top and bottom electrode – the most dominant ECF at the center of the system, (b) generation of two nuclei centers within the gap in the top-left side of the system, (c) (d) growth of the nuclei centers into a ECF, eventually forming a different dominant ECF compared to the original one.

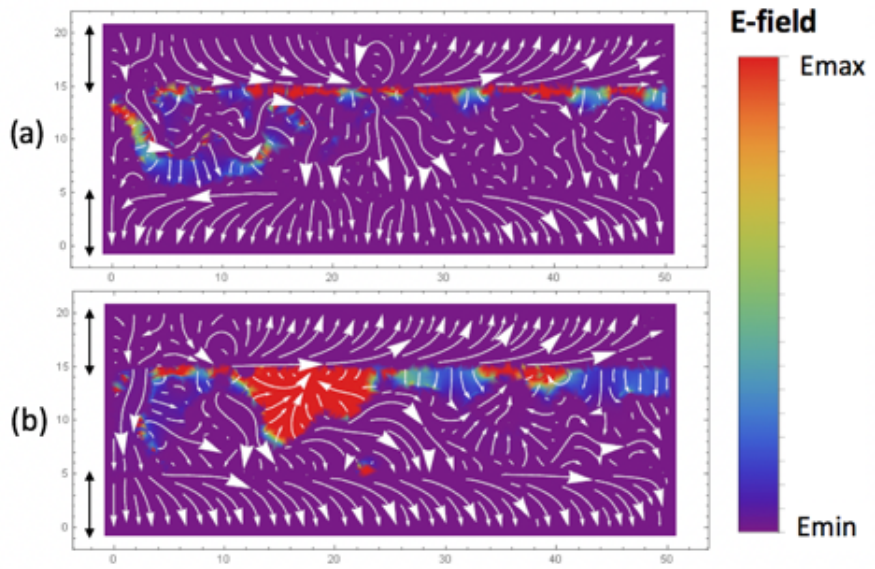

Figure 4.7: Current density map of the ON-state (a) before and (b)after nucleation simulation. (a) $J_{(r)}$ density continuity between top and bottom electrode as the system is placed in ON-state after set operation, generated based on Fig. 4.6(a); (b) current density map for Fig. 4.6(d), in which the arrangement and concentration of the white arrows connecting top and bottom electrodes have changed compared to panel (a) representing a change in the state of the device after the growth of nuclei centers over time, without the influence of external electric field, which is more gradual and not as abrupt as the change observed in the OFF state.

in the same way as that described for Fig. 4.5(a) and Fig. 4.5(b), and are shown in Fig. 4.7(a) and Fig. 4.7(b) respectively.

A comparison between Fig. 4.7(a) and Fig. 4.7(b) clearly shows that the dominant ECF from the center (i.e., $x \sim 25\ nm$) in Fig. 4.7(a) disappeared in Fig. 4.7(b), but two new ECFs were formed to the left of the system, one at the location of nuclei centers (i.e., $x \sim 10\ nm$), and the other in a random location (i.e., $x \sim 3\ nm$) due to growth of a fractured ECF remnant of the previous incomplete RESET operation. Relative electrical current calculated by integrating current density along the width (i.e., 0 - 50 nm) of the switching layer is 12 for Fig. 4.7(a) and is 24 for Fig. 4.7(b), that is, R_{ON} is smaller for Fig. 4.7(b) – an indication of deepening of the ON-state after retention bake. Even though the On-state retention was not lost per this simulation result, but it is speculated that increase in the ON-state conductance can eventually lead to the failure during the subsequent RESET operation, as more power is needed in order to disconnect the ECFs. Also, due to the randomness nature of an ECF, growth of nuclei sites could potentially increase instability of the On-state, and as the result contribute more to the cycle to cycle variability of the memristors.

4.4 Summary

We studied to include nucleation phenomenon into phase-field model capturing qualitative characteristics of electrothermal formation and annihilation of ECFs in dielectric thin film memristors. Considering the well-known effect of high electric field on

generation and growth of nuclei centers in a dielectric, we started our simulation with the assumption that the gap region separating top electrode from the conductive regions connected electrically to the bottom electrode experiencing high electric field during cycling in thin film dielectric system would be the epicenter of nuclei site generation.

Our study highlights how the nuclei centers can grow dynamically after the removal of electric bias and potentially change the state of the device and introduce more variation into a system that is already evolving. If the device is left in the OFF-state, growth of the nuclei centers can eventually generate conductive path connecting the top electrode to charge-clusters that are connected to the bottom electrode, resulting in the retention loss of the OFF-state. Similarly, in the device that is left in ON-state, growth of nuclei centers can potentially generate fractured ECF, as the total free energy of the system reduces and dominant ECF evolves dynamically, which in return introduce more variation within the system and potentially contribute to device to device and cycle to cycle variability, eventually reduce the device reliability.

References

[1] D. Strukov J. Yang and and D. Stewart. Memristive devices for computing. *Nature Nanotech*, 8:13, 2013.

[2] M. Wuttig and N. Yamada. Phase-change materials for rewriteable data storage. *Nature Mater*, 6:824–832, 2009.

[3] C. Chappert, A. Fert, and F.N. Van Dau. The emergence of spin electronics in data storage. *Nature Materials*, pages 813–823, 2003.

[4] F. Yuan, Z. Zhang, J. Wang, L. Pan, J. Xu, and C. Lai. Total ionizing dose (tid) effects of y ray radiation on switching behaviors of $Ag/AlOx/Pt$ *RRAM* device. *Nanoscale Research Letters*, 9:452–452, 2014.

[5] Antonio C Torrezan, John Paul Strachan, Gilberto Medeiros-Ribeiro, and R Stanley Williams. Sub–nanosecond switching of a tantalum oxide memristor. *Nanotechnology*, 22(48):485203, nov 2011.

[6] Stefano, Stefano, Simone Balatti, Antonio Cubeta, Alessandro Calderoni, Nirmal Ramaswamy, and Daniele Ielmini. Statistical fluctuations in hfox resistive-switching memory: Part i - set/reset variability. *Electron Devices, IEEE Transactions on*, 61:2912–2919, 08 2014.

[7] S. Ambrogio, Simone Balatti, V. McCaffrey, D. Wang, and Daniele Ielmini. Impact of low-frequency noise on read distributions of resistive switching memory (rram). *Technical Digest - International Electron Devices Meeting, IEDM*, 2015:14.4.1–14.4.4, 02 2015.

[8] G.S. Kim, T.H. Park, H.J. Kim, T.J. Ha, W.Y. Park, S.G. Kim, and C.S. Hwang. Investigation of the retention performance of an ultra–thin hfo2 resistance switching layer in an integrated memory device. *Journal of Applied Physics*, 124:024102, 2018.

[9] M. Azzaz, E. Vianello, B. Sklénard, P. Blaise, A. Roule, C. Sabbione, S. Bernasconi, C. Charpin, C. Cagli, E. Jalaguier, S. Jeannot, S. Denorme, P. Candelier, M. Yu, L. Nistor, C. Fenouillet-Béranger, and L. Perniola. Endurance/retention trade off in hfox and taox based rram. *2016 IEEE 8th International Memory Workshop (IMW)*, pages 1–4, 2016.

[10] Kentaro Kinoshita, Ayumi Okano, Koji Tsunoda, Masaki Aoki, and Yoshihiro Sugiyama. Dominant failure mechanism in data retention characteristics of resistance change memory consisting of nio at high temperature. *Applied Physics Express – APPL PHYS EXPRESS*, 1, 12 2008.

[11] Daniele Ielmini. Resistive switching memories based on metal oxides: mechanisms, reliability and scaling. *Semiconductor Science and Technology*, 31(6):063002, may 2016.

[12] Shinhyun Choi, Jihang Lee, Sungho Kim, and Wei D. Lu. Retention failure analysis of metal-oxide based resistive memory. *Applied Physics Letters*, 105(11):113510, 2014.

[13] G. Bersuker, D. C. Gilmer, D. Veksler, J. Yum, H. Park, S. Lian, L. Vandelli, A. Padovani, L. Larcher, K. McKenna, A. Shluger, V. Iglesias, M. Porti, M. Nafría, W. Taylor, P. D. Kirsch, and R. Jammy. Metal oxide rram switching mechanism based on conductive filament microscopic properties. In *2010 International Electron Devices Meeting*, pages 19.6.1–19.6.4, 2010.

[14] Randima Galhenage, Hui Yan, Samuel Tenney, Nayoung Park, Peter Albrecht, David Mullins, and Donna Chen. Understanding the nucleation and growth of metals on tio2: Co compared to au, ni, and pt. *The Journal of Physical Chemistry C*, 117:7191–7201, 04 2013.

[15] Foroozan S. Koushan and Nobuhiko P. Kobayashi. An electro-thermal computational study of conducting channels in dielectric thin films using self–consistent phase-field methodology: A view toward the physical origins of resistive switching. *Journal of Applied Physics*, 128(16):165302, Oct 2020.

[16] M. Nardone and V. G. Karpov. Nucleation of metals by strong electric fields. *Applied Physics Letter*, 100:151912, 2012.

[17] Stanford R. Ovshinsky. Reversible electrical switching phenomena in disordered structures. *Phys. Rev. Lett.*, 21:1450–1453, Nov 1968.

[18] V. G. Karpov, Y. A. Kryukov, I. V. Karpov, and M. Mitra. Field–induced nucleation in phase change memory. *Phys. Rev. B*, 78:052201, Aug 2008.

[19] Nevill Mott. *Metal–Insulator Transitions*. 1st edition edition, 1990.

[20] R. Meyer, L. Schloss, J. Brewer, R. Lambertson, W. Kinney, J. Sanchez, and D. Rinerson. Oxide dual-layer memory element for scalable non-volatile cross-point memory technology. pages 1–5, 2008.

[21] V.G. Karpov, D. Niraula, I.V. Karpov, and R. Kotlyar. Thermodynamics of phase transitions and bipolar filamentary switching in resistive random–access memory. *Physical Review Applied*, 8(2), Aug 2017.

[22] R. Becker and W. Döring. Kinetische Behandlung der Keimbildung in übersättigten Dämpfen. *Annalen der Physik*, 416(8):719–752, 1935.

[23] D. B. Dickens and J. J. Sloan. The nucleation and freezing of dilute nitric acid aerosols. *The Journal of Physical Chemistry A*, 106(44):10543–10549, 2002.

[24] R. Dhanasekaran and P. Ramasamy. Two–dimensional nucleation in the presence of an electric field. *Journal of Crystal Growth*, 79(1, Part 2):993–996, 1986. Proceedings of the Eighth International Conference on Crystal Growth.

[25] C C Chin, J B Dence, and J C Warren. Crystallization of human placental estradiol 17beta–dehydrogenase. a new method for crystallizing labile enzymes. *Journal of Biological Chemistry*, 251(12):3700–3705, 1976.

[26] Rosa Crespo, Pedro M. Martins, Luís Gales, Fernando Rocha, and Ana M. Damas. Potential use of ultrasound to promote protein crystallization. *Journal of Applied Crystallography*, 43(6):1419–1425, Dec 2010.

[27] C.A.Stan, S.K.Tang, K.J. Bishop, and G.M. Whitesides. Externally applied electric fields up to 1.6 $\times$ 10(5) v/m do not affect the homogeneous nucleation of ice in supercooled water. . *The journal of physical chemistry*, 115(5):1089–1097, 2010.

[28] J.Y. Rempel, M.G. Bawendi, and K.F.Jensen. Insights into the kinetics of semiconductor nanocrystal nucleation and growth. *Journal of the American Chemical Society*, 131:4479–4489, 2009.

[29] Nguyen T. K. Thanh, N. Maclean, and S. Mahiddine. Mechanisms of nucleation and growth of nanoparticles in solution. *Chemical Reviews*, 114(15):7610–7630, 2014.

[30] Soon Gu Kwon and Taeghwan Hyeon. Formation mechanisms of uniform nanocrystals via hot–injection and heat–up methods. *Small*, 7(19):2685–2702.

Chapter 5

Conclusion and Future Steps

This dissertation has successfully demonstrated dynamical evolution of conducting channel formation and annihilation of resistive switching dielectric thin films, using a computational phase field study. The electrothermal phase field formulation successfully predicts formation of conducting channels in SET operation, and rupture of conductive filaments in RESET operation, in typical dielectric thin film structures comparable to a range of resistive switches, offering an alternative computational approach based on metastable states treated at the atomic scale, requiring no assumptions on conducting channel morphology and its fundamental transport mechanisms. Abondaning the assumptions of an a priori conducting channel model and the presence of specific transport phenomena to explain resistive switching, our approach is based on formulating a diffuse interface between two different phases within the same system.

This study, also, proposes nucleation phenomenon as a possible root cause of reliability degradation and retention loss in thin film dielectric devices, as it enhances gradual formation of a fractured conductive path, in the absence of electrical bias, resulting in retention loss of the OFF-state. This approach can also explain reliability failure of the device after the system is left in the ON-state, for some period of time, as the gradual growth of nuclei can either connect multiple filaments together, driving the system into a lower ON resistance, or can rupture the already connected CF, resulting the retention loss of the ON state, as the total free energy of the system is reducing. All these dynamical changes whithin the system can potentially increase variability among devices at a given cycle, or among cycles in a given device.

Some assumptions are made in these simulations that each can potentially be a

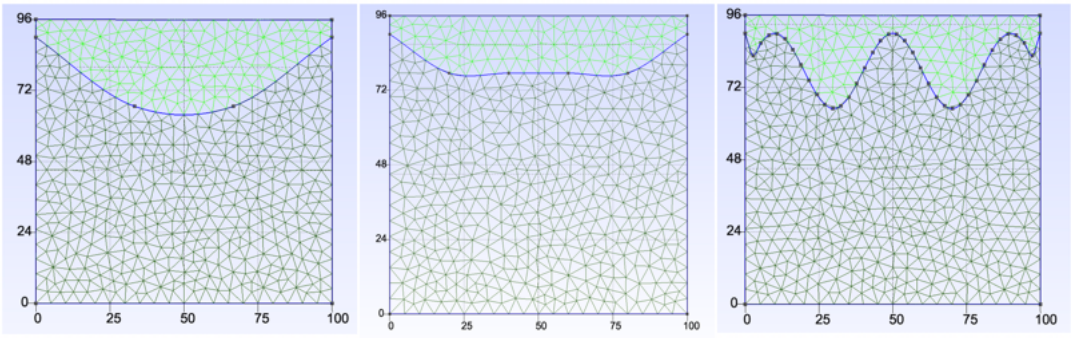

Figure 5.1: Example of possible different Bottom Electrode topographies in memristor device as the result of different etch profiles during RRAM stack process.

topic of separate focused study. For example, in this thesis, impact of temperature on nucleation process is ignored. But as the nucleation is driven by the electrical stress during cycling, and increase in temperature modulates the magnitude of elecric field stress the system experiences, then nucleation process can be affected by the temperature of the system. Also according to CNT, nucleation rate is directly dependent to temperature through the Eq. 4.4.

Another assumption made in this study is the uniformity of bottom electrode topography in small thin film memristor devices, which can directly impact magnitude of electric field experienced across the insulating layer modulating formation and annihilation process of ECFs within the system, as it is the case in the real device. Future studies can incorporate different topography in the bottom electrode, similar but not limitted to what is shown in Fig. 5.1, to capture its impact on ECF formation and annihilation during switching operation.

At the end, it is important to emaphasize our belief that eventhough all these assumptions would directly impact the parameters discussed in this thesis, they are

not expected to change the conclusion of the overall model presented in our study.

Appendix A

Multi Physics Object Oriented Simulation Environment - MOOSE

Introduction

This appendix introduces the Multiphysics Object-Oriented Simulation Environment (MOOSE) finite-element PDE solver framework[**?**]. Through an API, MOOSE provides an extensbible high-level framework to the PetSC nonlinear solver and libmesh mesher to solve a broad variety of partial differential equations typically encountered in engineering and physics. MOOSE was developed by the Idaho National Laboratory, while the PetSC[**?**] and libmemsh tools were developed at the Argrone and Sandia National Laboratories, respectively[**?**]. In contrast to commercial FEM tools, which are generally centered around a propreitary integrated design environment (IDE), the MOOSE workflow requires several additional applications, including a C compiler and visualization tools. A central feature of the MOOSE framework is the weak form specification of physics kernels in C, providing a flexibe mechanism for the formulation of complex anisotropic and nonlinear contiuum and hydrodynamic flow problems. A MOOSE workflow is presented, including installation instructions. A simual- tion example is provided to illustrate the application of MOOSE for self-consistent elec- trothermal simulation of a memristor structure.

About MOOSE

MOOSE solves systems of coupled partical differential equations. Each PDE is cast into an associated weak formualtion of the fintie element method to represent and solve the coupled system on a discretized 2-D or 3-D mesh. For example, consider

using modified Cahn-Hilliard equations,

$$\frac{\partial c_i}{\partial t} = \nabla.(M_i \nabla \frac{\partial F}{\partial c_i}) \tag{A.1}$$

where $c_i$is a conserved variable andM_i is the associated mobility. The free energy functional, for a phase field model using N conserved variables c_i is described by

$$F = \int_V [f_{loc}(c_1, ..., c_N) + f_{gr}(c_1, ..., c_N) + E_d] dV \tag{A.2}$$

where f_{loc} defines the local free energy density as a function of all concentrations and order parameters, and varies from model to model. The gradient energy density

$$f_{gr} = \sum_i^N \frac{k_i}{2} |\nabla c_i|^2 \tag{A.3}$$

where κ_i is the gradient energy coefficient, and E_d describes any additional sources of energy in the system.

To prepare for the FEM discretization, after combining the above equations with each other, a residual equation (equal to zero) in weak form is constructed, and,the weighted integral residual projection is generated using test function ψ_m. The residual equation

$$R_{c_i} = (\nabla c_i, \nabla(\kappa_i \psi_m)) - \langle \nabla c_i.\vec{n}, \kappa_i \psi_m \rangle + \left(\left(\frac{\partial f_{loc}}{\partial c_i} + \frac{\partial E_i}{\partial c_i} - \mu_i \right), \psi_m \right) \tag{A.4}$$

is achieved after the divergence theorem application to reduce the order of derivative, where the terms in the paranthesis represent different phhysics, and the term in the

bracket defines the boundary condition of the system.

MOOSE Workflow

This section discusses the minimum suite of applications to create an effective workflow, with a major feature being the broad discretion afforded in configuration. The MOOSE workflow depends largely on user preference, with a minimum workflow suite itemized below; each of these are disucssed in turn. Following description of custom physics, using the weak form cast in C code, which requires compiling, the majority of the workflow centers around simulation spec- ification by the MOOSE deck (text editor), executing the simulation (command line), and visualization (application).

- MOOSE application
- Text editor
- Mesh application
- Visualization application
- Compiler

MOOSE application

The MOOSE application is similar to Python or MATLAB in many ways, such as a command-line interface to an API (Application Program Interface). MOOSE must be complied when it is installed, unlike MATLAB, so a compiler is required. The

resulting executable provides an API that executes a simulation described by a text-file in which custom physics are linked to MOOSE run-time by user-supplied C code. The MOOSE API is itself based on the PetSC PDE solver from Argonne National Lab and the libmesh framework from Sandia National Lab.

Text editor

The text editor is the focal point of the MOOSE workflow, with the Atom being the endorsed text editor; a custom configuration file is avaiable for Atom for syntax high-lighting for the MOOSE API. It is suggested to read http://mooseframework.org/wiki/MooseTraining/InputFile/ for an introduction to basic elements of a MOOSE deck.

Mesh application

MOOSE requires a mesh to be supplied, inlcuding boundary and volume names. The libmesh framework from Sandia has a commercial version with academic pricing, which can be used. Simultaneously, the Gmsh open-source mesher is evaluated and found to be suitable for introductory research; it is this mesher that is the focus of the present note. The Gmsh application has stable builds for Mac, Linux, and Windows. MOOSE also provides a build in mesh system that can be defined within the MOOSE environment, for symlicity of the flow. This work around has also been evaluated for our study.

Visualization application

The MOOSE install provides a basic GUI interface the API and includes basic visualiztion. It also exports an industry standard Exodus file that can be read by many third-party visualization applications; for the present note, open-source ParaView provided execeptional visualization capability. For these stuidies, ParaView 5.5.2 version is used extensively.

Compiler

To invoke the MOOSE API, a machine-specific executable must be provided by a compiler. Moreover, custom physics in MOOSE must be compiled and linked to the MOOSE executable. Generally, once the physics are coded, no further compiling is necessary.

Installations

OXS Installer

This sections describes the El Captain OSX installation procedure for MOOSE, and the associated applications from the previous section, specifically the Atom text editor, the Gmsh mesher, the ParaView visualizer, and the gcc C compiler. Optimum installation results with a clean OSX install is followed by the MOOSE install. The remaining applications can be installed in any order.

MOOSE

The El Captain OSX installation instructions are found at $http://mooseframework.org/getting-started/osx/$. Note that Xcode, with command line tools, must be in- stalled prior to the MOOSE installation. Following the MOOSE installation, go to http://www.mooseframework.org/ getting-started/ and follow the instructions for cloning MOOSE and and compiling libmesh. Compiling takes about an hour. Run the test suite to confirm MOOSE has been properly installed.

ATOM

The Atom download is found at $https://atom.io$. It is recommended to install the MOOSE-specific syntax-highlighting package, found at
$https://atom.io/packages/language-moose$.

Gmsh

The Gmsh download is found at $http://gmsh.info$. Note that Gmsh offers both 2-D and 3-D mesh support.

ParaView

The ParaView download is found at $http://www.paraview.org$.

Compiler

The X-Code installation will install gcc, an OSX-compatible C/C++ compiler. To verify correct installtion, type cc –version at an X-Term command line.

Kernel Implementation

Each kernel in MOOSE environment requires a header (.h) file with declarations and a code file (.c) that implements the weak formulation, similar to what is given in Eq. A.4. Each kernel must also be registered with the MOOSE (.c). A complete listing of MOOSE classes is available at:
$http://mooseframework.org/docs/doxygen/moose/classes.html$.

The MOOSE Simulation Deck

The MOOSE simulation is specified by a text file composed of the following elements:

- Mesh
- Variables
- Mesh application
- Kernels
- Boundary Conditions
- Materials
- Executioner
- Output

A complete summarty of MOOSE simulation elements and systems is located at

http : *//mooseframework.org/wiki/MooseSystems/* with parameter definitions located at *http* : *//mooseframework.com/docs/syntax/moose/*.

Example codes related to the simulations presented in this work are listed in Appendix C.

Appendix B

Electrothermal Computational Study in Phase Field module

This Appendix lists the source code for three custom kernels as well as the MOOSE deck file for the phase field formulation of dielectric thin-film resistive switching phenomena. Although the heat transport kernel was complied and included, for the work presented in this dissertation, thermal resistance was set to zero, thereby inhibiting Joule heating. The Laplace kernel is coupled to phase field concentration, $c_{(r,t)}$, to couple this variable as a proxy for conductivity. In this simulation, electric potential is coupled to Cahn-Hilliard kernel, and effect of temperature on the whole system is also included in the phase field model, through the following equations:

$$\frac{\partial c_{(\mathbf{r},t)}}{\partial t} = \nabla.M\nabla\left[\frac{\partial f_{bulk}(c_{(\mathbf{r},\mathbf{t})},T)}{\partial c_{(\mathbf{r},\mathbf{t})}} - \nabla.\kappa\nabla c_{(\mathbf{r},t)} - \frac{q}{\Omega}V_{(\mathbf{r},t)}\right] \tag{B.1}$$

$$\nabla.\sigma(c_{(\mathbf{r},t)})\nabla V_{(\mathbf{r},t)} = 0 \tag{B.2}$$

The present simulation framework was constructed primarily to explore and validate the phase field formulation to study dielectric thin-film resistive switching phenomena. Many approximations were used, and in many cases, less emphasis was placed on absolute scaling of the state variables versus their relative ratios. While this leads to some unusual independent variable absolute values, the expected behavior was nevertheless observed.

Kernels

file#1

```
// Foroozan Koushan
// Department of Electrical Engineering, UC Santa Cruz
// April 2019

// This kernel returns the diffusion operator of LaPlace equation,
// and includes coupling to concentration variable c.


#ifndef CoupledLaplace_H
#define CoupledLaplace_H

#include "Kernel.h"


class CoupledLaplace;
template<>

InputParameters validParams<CoupledLaplace>();

class CoupledLaplace : public Kernel
{
public:
CoupledLaplace(const InputParameters & parameters);

protected:
virtual Real computeQpResidual() override;
virtual Real computeQpJacobian() override;

private:
const VariableValue & _coupled_c;
//  const MaterialProperty<Real> & _diffusivity;
//  const MaterialProperty<Real> & _diffEa;
Real _D;

};

#endif
```

file#2

```
// Foroozan Koushan
```

```
// Department of Electrical Engineering, UC Santa Cruz
// April, 2019

// This kernel returns the diffusion operator of electric potential V,
// it is derived from the CoupledLaplace.c code used in the
// continuum transport code. The present version couples
// concentration variable c, as a proxy for conductivity.

#include "CoupledLaplace.h"

registerMooseObject("ph02App", CoupledLaplace);

template<>
InputParameters validParams<CoupledLaplace>()
{
InputParameters params = validParams<Kernel>();
params.addRequiredCoupledVar("coupled_c", "This is c.");
return params;
}

CoupledLaplace::CoupledLaplace(const InputParameters & parameters) :
Kernel(parameters),
_coupled_c(coupledValue("coupled_c"))
{}

Real CoupledLaplace::computeQpResidual()
{
// For P-F investigation, make _D the effective conductivity based
// on c.
_D = 1.0 * _coupled_c[_qp];
return
_D * _grad_u[_qp] * _grad_test[_i][_qp];
}

Real CoupledLaplace::computeQpJacobian()
{
return _D * _grad_phi[_j][_qp] * _grad_test[_i][_qp];
}
```

file#3

```
//  Foroozan Koushan
//  Department of Electrical Engineering, UC Santa Cruz
//  April, 2019

// Phase field  simulation of formation of conductive filaments in
// thin film dielectric system, assuming an alloy composed of alpha
```

```
// and beta whose free energy density is a double well potential
// that interacts with an external electrtic potential, V.
//
// The idea is to select the correct free energy density and mobility
// parameters to model an non uniform interface. In this simulation
// The example, a double well free energy density function and an
// arbitrary surface mobility function are used.
// Also an interaction with the applied external potential, V, and
// system temperature, T, are added to the calculations.
// All BCs are periodic, unless otherwise noted.
// Electric potential boundary counditions are Neumann on sides
// and Dirichlet on top and bottom. IC is specified as a random
// concentration, c.
//
// this deck is simulated when charged clusters with high
// concentration are scattered randomely within the thin
// film dielectric, as the starting point.

[Mesh]
type = GeneratedMesh
dim = 2
elem_type = QUAD4
nx = 50
ny = 10
nz = 0
xmin = 0
xmax = 50
ymin = 0
ymax = 10
zmin = 0
zmax = 0
uniform_refine = 2
[]

[Variables]
[./c]    // Mole fraction of Cr (unitless)
order = FIRST
family = LAGRANGE
[../]
[./w]    // Chemical potential (eV/mol)
order = FIRST
family = LAGRANGE
[../]
[./V]    // Electric potential (V)
order = FIRST
family = LAGRANGE
```

```
[../]
[./T]    // Temperature (T)
order = FIRST
family = LAGRANGE
[../]
[]

[AuxVariables]
[./f_density]    // Local energy density (eV/mol)
order = CONSTANT
family = MONOMIAL
[../]
[./c_dot]
# considering c as electrical charge (Q) --> c_dot will be dQ/dt
order = FIRST
family = LAGRANGE
[../]
[./c_order1]
[../]
[./c_order2]
[../]
[./c_order3]
[../]
[]


[ICs]
[./concentrationIC]
type = RandomIC
min = 0.1
max = 0.8
seed = 505
variable = c
[../]
[]




[BCs]
[./Periodic]
[./c_bcs]
auto_direction = 'x'
variable = c
[../]
[./w_bcs]
```

```
variable = w
auto_direction = 'x'
[../]
[./V_bcs]
variable = V
auto_direction = 'x'
[../]
[./T_bcs]
variable = T
auto_direction = 'x'
[../]
[../]
[./V_top]
type = DirichletBC
variable = V
boundary = 'top'
value = 10e-5  ////moves the charges to the bottom
[../]
[./V_bottom]
type = DirichletBC
variable = V
boundary = 'bottom'
value = 0
[../]
[./T_top]
type = DirichletBC
variable = T
boundary = 'top'
value = 400
[../]
[./T_bottom]
type = DirichletBC
variable = T
boundary = 'bottom'
value = 400
[../]
[]

[Kernels]
[./w_dot]
variable = w
v = c
type = CoupledTimeDerivative
[../]
[./coupled_res]
variable = w
```

```
type = SplitCHWRes
mob_name = M
[../]
[./coupled_parsed]
variable = c
type = SplitCHParsed
f_name = f_loc
kappa_name = kappa_c
w = w
args = 'c_T'
[../]
[./coupled_parsed_elec]
variable = c
type = SplitCHParsed
f_name = f_ele
kappa_name = kappa_e
w = w
args = 'c_V'
[../]
[./laplace]
variable = V
type = CoupledLaplace
coupled_c = c
[../]
[./HeatConduction]
type = HeatTransportMetal
variable = T
coupled_T = T
[../]
[./ThermalCoupling]
type = ThermalCouplingConductor
variable = T
coupled_V = V
[../]
[./T_dot]
variable = T
type = TimeDerivative
[../]
[./heat]
type = HeatConduction
variable = T
[../]
[]

[AuxKernels]
# Calculates the energy density by combining the local and
```

```
# gradient energies
[./f_density]    // (eV/mol/nm^2)
type = TotalFreeEnergy
variable = f_density
f_name = 'f_loc'
kappa_names = 'kappa_c'
interfacial_vars = 'c'
[../]
[./coupled_dot]    // (eV/mol/nm^2)
type = DotCouplingAux
variable = c_dot
v = c
[../]
[./MidpointTimeIntegrator]
type = VariableTimeIntegrationAux
variable_to_integrate = c
variable = c_order1
order = 1
[../]
[./TrapazoidalTimeIntegrator]
type = VariableTimeIntegrationAux
variable_to_integrate = c
variable = c_order2
order = 2
[../]
[./SimpsonsTimeIntegrator]
type = VariableTimeIntegrationAux
variable_to_integrate = c
variable = c_order3
order = 3
[../]
[]


[Materials]
// d is a scaling factor that makes it easier for the solution to
// converge without changing the results. It is defined in each
// of the first three materials and must have the same value
// in each one.
[./kappa]              // Gradient energy coefficient (eV nm^2/mol)
type = GenericFunctionMaterial
prop_names = 'kappa_c'
prop_values = '8.125e-16*6.24150934e+18*1e+09^2*1e-27'
// kappa_c *eV_J*nm_m^2* d
//prop_values = '8.12*6.24*1e-6'
[../]
```

```
[./kappa_e]
// kappa_e *eV_J*nm_m^2* d
type = GenericFunctionMaterial
prop_names = 'kappa_e'
prop_values = '5e-10'
//prop_values = '5e-8'
[../]
[./mobility]                    // Mobility (nm^2 mol/eV/s)
// mobility is defined as a function of c in this simulation
// it can also be constant for simiplicity
type = DerivativeParsedMaterial
f_name = M
args = c
constant_names =        'Acr    Bcr    Ccr    Dcr
Ecr    Fcr    Gcr
Afe    Bfe    Cfe    Dfe
Efe    Ffe    Gfe
nm_m    eV_J    d'
constant_expressions = '-32.770969  -25.8186669  -3.29612744  17.669757
37.6197853  20.6941796    10.8095813
-31.687117  -26.0291774  0.2286581    24.3633544
44.3334237  8.72990497    20.956768
1e+09       6.24150934e+18          1e-27'
function = 'nm_m^2/eV_J/d*((1-c)^2*c*10^
(Acr*c+Bcr*(1-c)+Ccr*c*log(c)+Dcr*(1-c)*log(1-c)+
Ecr*c*(1-c)+Fcr*c*(1-c)*(2*c-1)+Gcr*c*(1-c)*(2*c-1)^2)
+c^2*(1-c)*10^
(Afe*c+Bfe*(1-c)+Cfe*c*log(c)+Dfe*(1-c)*log(1-c)+
Efe*c*(1-c)+Ffe*c*(1-c)*(2*c-1)+Gfe*c*(1-c)*(2*c-1)^2))'
derivative_order = 1
outputs = exodus
[../]

[./local_energy]            // Local free energy function (eV/mol)
type = DerivativeParsedMaterial
f_name = f_loc
args = 'c T'
derivative_order = 2
enable_jit = true
constant_names =        'a0 a1 a2 a3 a4 a5 a6 a7
b0 b1 b2 b3
c0 c1 c2 c3
d0 d1 d2 d3
f0 f1 f2 f3 f4 f5 f6 f7
eV_J d T0 Kb
A   B   C   D   E   F   G'
```

```
constant_expressions = '0.3856 -0.0973 -0.0467 -0.1945
 -0.1856 -0.0044 -3.8366 -4.1231
 0.0003 4.69e-5 -4.9588e-5 1.13e-5
  3.14e-9 -2.203e-8 3.9597e-8 -3.09e-8
 -1.76e-13 5.5788e-12 -5.93e-12 1.23e-11
 0.3817 -0.1007 -0.0485 -0.154 -0.1684
 -0.0416 -3.86 -4.167
 6.24150934e+18 1e-27 700 8.6173e-5
 -2.446831e+04 -2.827533e+04 4.167994e+03
 7.052907e+03 1.208993e+04 2.568625e+03
 -2.354293e+03'
function = 'Ha:=(c*(1-c)*((a0*(1-2*c))+(a1*(1-2*c))+
(a2*(1-2*c))+(a3*(1-2*c))+(a4*(1-2*c))+(a5*(1-2*c)))+
(a6*c)+(a7*(1-c)));
Hb:=b0*(1-c)+b1*(1-c)+b2*(1-c)+b3*(1-c);
Hc:=c0*(1-c)+c1*(1-c)+c2*(1-c)+c3*(1-c);
Hd:=d0*(1-c)+d1*(1-c)+d2*(1-c)+d3*(1-c);
He:=-Ha/T0 + Hc*T0 + Hb*log(T0) + Hd*0.5 * T0^2;
G0:=(c*(1-c)*((f0*(1-2*c))+(f1*(1-2*c))+
(f2*(1-2*c))+(f3*(1-2*c))+(f4*(1-2*c))+
(f5*(1-2*c)))+(f6*c)+(f7*(1-c)));
1e+4*eV_J*d*(G0*T/T0+Ha-Hb*T-
Hc*T^2+0.5*Hd*T^3+He*T+
T*Kb*(c*log(c)+(1-c)*log(1-c)))' ;
//10*eV_J*d*(c^2)*((1-c)^2)*(1-T/T0);
[../]

[./electric_field_energy_density]
// Electric potential contribution to local free-energy density (eV/mol).
type = DerivativeParsedMaterial
f_name = f_ele
args = 'c V'
constant_names = 'J_eV A B'
constant_expressions = '1.6e-19 1.0e+19 1e4'
function = 'V*c'
derivative_order = 2
enable_jit = true
[../]
[./heat]
type = HeatConductionMaterial
specific_heat = 1.0
thermal_conductivity = 1.0
[../]
[]

[Postprocessors]
```

```
[./step_size]         // Size of the time step
type = TimestepSize
[../]
[./iterations]       // Number of iterations needed to converge timestep
type = NumNonlinearIterations
[../]
[./nodes]            // Number of nodes in mesh
type = NumNodes
[../]
[./evaluations]     // Cumulative residual calculations for simulation
type = NumResidualEvaluations
[../]
[./active_time]       // Time computer spent on simulation
type = PerfGraphData
section_name = "Root"
data_type = total
[../]
[]

////line sample generation in csv format
[VectorPostprocessors]
[./line_sample1]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '1_0_0'
end_point = '1_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample2]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '2_0_0'
end_point = '2_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample3]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '3_0_0'
end_point = '3_10_0'
num_points = 201
sort_by = y // sort by: x y z id
```

```
execute_on = linear //timestep_end
[../]
[./line_sample4]
type = LineValueSampler
variable = 'c f_density c_dot'
start_point = '4 0 0'
end_point = '4 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample5]
type = LineValueSampler
variable = 'c f_density c_dot'
start_point = '5 0 0'
end_point = '5 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample6]
type = LineValueSampler
variable = 'c f_density c_dot'
start_point = '6 0 0'
end_point = '6 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample7]
type = LineValueSampler
variable = 'c f_density c_dot'
start_point = '7 0 0'
end_point = '7 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample8]
type = LineValueSampler
variable = 'c f_density c_dot'
start_point = '8 0 0'
end_point = '8 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
```

```
[../]
[./line_sample9]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '9_0_0'
end_point = '9_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample10]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '10_0_0'
end_point = '10_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample11]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '11_0_0'
end_point = '11_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample12]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '12_0_0'
end_point = '12_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample13]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '13_0_0'
end_point = '13_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
```

```
[./line_sample14]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '14_0_0'
end_point = '14_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample15]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '15_0_0'
end_point = '15_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample16]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '16_0_0'
end_point = '16_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample17]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '17_0_0'
end_point = '17_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample18]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '18_0_0'
end_point = '18_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample19]
```

```
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '19_0_0'
end_point = '19_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample20]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '20_0_0'
end_point = '20_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample21]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '21_0_0'
end_point = '21_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample22]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '22_0_0'
end_point = '22_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample23]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '23_0_0'
end_point = '23_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample24]
type = LineValueSampler
```

```
variable = 'c_f_density_c_dot'
start_point = '24_0_0'
end_point = '24_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample25]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '25_0_0'
end_point = '25_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample26]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '26_0_0'
end_point = '26_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample27]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '27_0_0'
end_point = '27_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample28]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '28_0_0'
end_point = '28_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample29]
type = LineValueSampler
variable = 'c_f_density_c_dot'
```

```
start_point = '29 0 0'
end_point = '29 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample30]
type = LineValueSampler
variable = 'c_f_density c_dot'
start_point = '30 0 0'
end_point = '30 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample31]
type = LineValueSampler
variable = 'c_f_density c_dot'
start_point = '31 0 0'
end_point = '31 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample32]
type = LineValueSampler
variable = 'c_f_density c_dot'
start_point = '32 0 0'
end_point = '32 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample33]
type = LineValueSampler
variable = 'c_f_density c_dot'
start_point = '33 0 0'
end_point = '33 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample34]
type = LineValueSampler
variable = 'c_f_density c_dot'
start_point = '34 0 0'
```

```
end_point = '34 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample35]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '35 0 0'
end_point = '35 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample36]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '36 0 0'
end_point = '36 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample37]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '37 0 0'
end_point = '37 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample38]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '38 0 0'
end_point = '38 10 0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample39]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '39 0 0'
end_point = '39 10 0'
```

```
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample40]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '40_0_0'
end_point = '40_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample41]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '41_0_0'
end_point = '41_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample42]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '42_0_0'
end_point = '42_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample43]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '43_0_0'
end_point = '43_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample44]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '44_0_0'
end_point = '44_10_0'
num_points = 201
```

```
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample45]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '45_0_0'
end_point = '45_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample46]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '46_0_0'
end_point = '46_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample47]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '47_0_0'
end_point = '47_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample48]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '48_0_0'
end_point = '48_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[./line_sample49]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '49_0_0'
end_point = '49_10_0'
num_points = 201
sort_by = y // sort by: x y z id
```

```
execute_on = linear //timestep_end
[../]
[./line_sample50]
type = LineValueSampler
variable = 'c_f_density_c_dot'
start_point = '50_0_0'
end_point = '50_10_0'
num_points = 201
sort_by = y // sort by: x y z id
execute_on = linear //timestep_end
[../]
[]

[Preconditioning]
[./coupled]
type = SMP
full = true
[../]
[]

[Executioner]
type = Transient
solve_type = NEWTON
l_max_its = 30
l_tol = 1e-6
nl_max_its = 50
nl_abs_tol = 1e-9
end_time = 6000 //86400 //1day  604800 // 7 days
petsc_options_iname = '-pc_type_-ksp_gmres_restart_-sub_ksp_type
-sub_pc_type_-pc_asm_overlap'
petsc_options_value = 'asm______31__________________preonly
ilu__________1'
[./TimeStepper]
type = IterationAdaptiveDT
dt = 10
cutback_factor = 0.8
growth_factor = 1.5
optimal_iterations = 7
[../]
[./Adaptivity]
coarsen_fraction = 0.1
refine_fraction = 0.7
max_h_level = 2
[../]
[]
```

```
[Debug]
show_var_residual_norms = true
[]

[Outputs]
exodus = true
//console = true
//csv = true
////comment out "checkpoint" if simulation does not start from scratch
Checkpoint = true
//[./console]
//  type = Console
//  max_rows = 10
//[../]
[./mycsv]
type = CSV
file_base = csvdir/linevalues
[../]

//[./csv]
//  type = CSV
//  //execute_on = final
//[../]
////to limit number of files generated from checkpoint
[./my_checkpoint]
type = Checkpoint
num_files = 25  ////needs to be higher than 2
interval = 5
[../]
[]
```

Appendix C

Mesh profile for different interface and states

This Appendix explains how to generate mesh profile for different concentration profiles of thin-film dielectric device, to represent ON and OFF states, in order to study impact of multiple cycles of set and reset operations in device reliability. Also, in this section, generation of different interface profile between condutive and nonconductive regions within thin film dielectric system is explained, eventhough it is not used for these dissertation. But it opens the filed to run a wide range of studies to understand the impact of interface topography on conductive filament formation and annihilation, under the influence of electric field potential.

As it was explained in Appendix B, starting mesh profile for the simulation consists of a random distribution of charged clusteres represented with different c value within a dielectric system, represented with low c value. The first set of simulation will rearrange c profiel within the system according to reduction of free energy of the whole bulk, as well as the direction of electric potential applied across the thin film dielectric. Concentration profile of the system can be saved in csv and exodus formats, after the following commands are added to the input file:

```
[Outputs]
exodus = true
## addition of checkpoint generates a directory named "checkpoint"
## in order to save the concentration profile of the system, peiodically
Checkpoint = true

## to limit number of files generated from checkpoint
[./my_checkpoint]
type = Checkpoint
num_files = 25
interval = 5
[../]
[]
```

After this step, a concentration profile at any given time that represents ON,

OFF or intermediate states of the device can be extracted and used as a starting mesh profile for the next simulation. An example of the implementation of the new mesh profile, is given below, in the input file of the simulation. In this example, simulation results of ON state after program operation at time step 15, is used as the starting point, in order to simulate erase operation in the device:

```
## use this mesh to restart the sim from a previously finished
## sim results
[Mesh]
  file = PR_state_cp/0015_mesh.cpr
[]
[Problem]
  restart_file_base = PR_state_cp/LATEST
[]
```

After the above simulation, again, the mesh profile of erase operation at the given time step n can be used as the starting point to run another program operation on the device, and so on, and so forth.

Interface Layer

Defining the interface layer between conductive and non-conductive regions within the mesh can also be done through the function command, in order to study impact of bottom electrode profile on device formation and annihilation of CFs. These functions can define concentration profile ofvthe interface layer based on x and y coordinates in a 2-D simulation. Forllowing lines are given as an example for such a run:

```
[Mesh]
type = GeneratedMesh
```

```
dim = 2
distribution = DEFAULT
elem_type = QUAD4
nx = 50
ny = 10#50
nz = 0
xmin = 0
xmax = 50
ymin = 0
ymax = 10#50
zmin = 0
zmax = 0
uniform_refine = 2
[]
[ICs]
[./concentrationIC]
type = FunctionIC
function = c_func
variable = c
[../]
[]
[Functions]
active = 'c_func'
[./c_func]
type = ParsedFunction
#vars = 'alpha'
#vals = '4'
#value = '(tan((pi*alpha*y)/180))/2.742'
#value = '(y+1)/(y+8)'
value = '(1.2*y)/(y+2)'
[../]
[]
```

Appendix D

Nucleation in Phase Field module

In this section, we explain how nucleation simulation is done through phase field module using Cahn-Hilliard formulation, in MOOSE. Implementation of discrete nucleation approach within phase field module can be found at $https://mooseframework.inl.gov/modules/phase\_field/Nucleation/Discrete-Nucleation.html$.

In order to generate figures similar to the following input file needs to be used:

```
# Foroozan Koushan
# Department of Electrical Engineering, UC Santa Cruz
# December 2020

# This input file tests the discrete nucleation in a dielectric film,
# by locally modifying the free energy to force growth of nuclei

[Mesh]
type = GeneratedMesh
dim = 2
nx = 120
ny = 120
xmax = 500
ymax = 500
elem_type = QUAD
[]

[Modules]
[./PhaseField]
[./Conserved]
[./c]
free_energy = F
mobility = M
kappa = kappa_c
solve_type = REVERSE_SPLIT
[../]
[../]
[../]
[]

[ICs]
[./c_IC]
type = RandomIC
```

```
variable = c
min = 0.2
max = 0.21
[../]
[]

[Materials]
[./pfmobility]
type = GenericConstantMaterial
prop_names  = 'M_kappa_c'
prop_values = '1_25'
[../]
[./chemical_free_energy]
# simple double well free energy
type = DerivativeParsedMaterial
f_name = Fc
args = 'c'
constant_names       = 'barr_height__cv_eq'
constant_expressions = '0.1__________0'
function = 16*barr_height*c^2*(1-c)^2
derivative_order = 2
outputs = exodus
[../]
[./probability]
# Nucleation rate for this simulation is assumed to be independent
# to the material. This, of course, can be changed, based on
# the different systems. For this simulation, rate of the nucleation
# would not have a difference in the conclusion.
type = ParsedMaterial
f_name = P
args = c
function = c*1e-7
outputs = exodus
[../]
[./nucleation]
# The nucleation material is configured to insert nuclei into
# the free energy to force the concentration to go to 0.9,
# and holds this enforcement for 100 time units, according
# to "inserter" section.
type = DiscreteNucleation
f_name = Fn
op_names  = c
op_values = 0.90
penalty = 5
penalty_mode = MIN
map = map
```

```
outputs = exodus
[../]
[./free_energy]
# adding the chemical and nucleation free energy contributions
# together.
type = DerivativeSumMaterial
derivative_order = 2
args = c
sum_materials = 'Fc Fn'
[../]
[]

[UserObjects]
[./inserter]
# The inserter runs at the end of each time step to add nucleation
# events that happend during the timestep (if it converged) to
# the list of nuclei
type = DiscreteNucleationInserter
hold_time = 100
probability = P
[../]
[./map]
# The map converts the nucleation points into finite area objects with
# a given radius.
type = DiscreteNucleationMap
radius = 10
periodic = c
inserter = inserter
[../]
[]

[Preconditioning]
[./SMP]
type = SMP
full = true
[../]
[]

[BCs]
[./Periodic]
[./all]
auto_direction = 'x y'
[../]
[../]
[]
```

```
[Postprocessors]
[./dt]
type = TimestepSize
[../]
[]

[Executioner]
type = Transient
scheme = bdf2
solve_type = 'PJFNK'
petsc_options_iname = '-pc_type -sub_pc_type'
petsc_options_value = 'asm      lu           '

nl_max_its = 20
l_tol = 1.0e-4
nl_rel_tol = 1.0e-10
nl_abs_tol = 1.0e-10
start_time = 0.0
num_steps = 1200

[./TimeStepper]
type = IterationAdaptiveDT
dt = 10
growth_factor = 1.5
cutback_factor = 0.5
optimal_iterations = 5
[../]
[]

[Outputs]
exodus = true
[]
```

In appendix E, input file, for inserttion of nulceation force within Cahn-Hilliard formulation of thin film dielectric during set and reset steps will be presented.

Appendix E

Electrothermal and Nucleation Simulation of ON-state and OFF-state

This appendix explains how nucleation force is combined with phase field simulation of conductive filament formation and annihilation formulation, in order to generate plots resented in chapter 4 of this document. In order to combine these effects, two changes are made to the simulation input file:

I - Incoming mesh profile

In order to study impact of nucleation on retention of CFs within the device, the incoming mesh profile is taken after the thin film has gone through set operation, for ON state, or reset operation, for OFF state, as it was explained in appendix B.

```
## use this mesh to restart the sim from a previously finished
## sim results, from here ....
[Mesh]
file = ph02_repeat_ers_my_checkpoint_cp/0015_mesh.cpr
[]
[Problem]
restart_file_base = ph02_repeat_ers_my_checkpoint_cp/LATEST
[]
##... to here
```

II - Inserting nuclei at desired mesh location

Also, nuclei generation can be forced within any x,y coordinate of the system, by defining those locations in a csv file and inputing the file in the following section. These selections of the coordinates, of course, needs to follow the already proven point that probablity of nucleation is larger at the points where electric field is highest. Based on this knowledge, and study of concentration map of ON and OFF

states, location of nuclei can be selected where non-conductive region of the thin film dielectric is thinnest. Using this logic, the following lines are added to the input file:

```
[UserObjects]
[./inserter]
# The inserter runs at the end of each time step to add
# nucleation events that happend during the timestep
# (if it converged) to the list of nuclei
#type = DiscreteNucleationInserter
type = DiscreteNucleationFromFile
# nuclei is held valid for the length of simulation time
hold_time = 100000
# csv file is used to define location of nucleir centers
file = nuclei.csv
[../]
```

The following is the format of the csv file that is used to defince nuclei centers within the system:

time x y

15 26 9.5

25 27 8.5

Input file

Finally, after all the updates, below is the example of input file used to capture impact of nucleation on retention of the ON and OFF state within the thin film dielectric system.

```
# Foroozan Koushan
# Department of Electrical Engineering, UC Santa Cruz
# January 2021
```

```
## use this mesh to restart the sim from a previously finished sim
## results from here ....
[Mesh]
file = ph02_repeat_ers_my_checkpoint_cp/0015_mesh.cpr
[]
[Problem]
restart_file_base = ph02_repeat_ers_my_checkpoint_cp/LATEST
[]
##... to here

[Modules]
[./PhaseField]
[./Conserved]
[./c]
free_energy = F
mobility = M
kappa = kappa_c
solve_type = REVERSE_SPLIT
[../]
[./w]
free_energy = F
mobility = M
kappa = kappa_c
solve_type = REVERSE_SPLIT
[../]
[./V]
free_energy = f_ele
mobility = M
kappa = kappa_e
args = 'c'
solve_type = REVERSE_SPLIT
[../]
[../]
[../]
[]

[Variables]
[./T]    #
order = FIRST
family = LAGRANGE
[../]
[]

#[ICs]
#[./c_IC]
##type = ConstantIC
```

```
#type = RandomIC
#variable = c
##value = 0.2
#min = 0.1
#max = 0.8
#[../]
#[]

[BCs]
#  [./Periodic]
#    [./all]
#      auto_direction = 'x y'
#    [../]
#  [../]
[./Periodic]
[./c_bcs]
auto_direction = 'x'
variable = c
[../]
[./w_bcs]
variable = w
auto_direction = 'x'
[../]
[./V_bcs]
variable = V
auto_direction = 'x'
[../]
[../]
[./V_top]
type = DirichletBC
variable = V
boundary = 'top'
# for this simulation, elevctric bias is removed
value = 0  #
[../]
[./V_bottom]
type = DirichletBC
variable = V
boundary = 'bottom'
# for this simulation, elevctric bias is removed
value = 0
[../]
[./T_top]
type = DirichletBC
variable = T
boundary = 'top'
```

```
value = 293  #RT
[../]
[./T_bottom]
type = DirichletBC
variable = T
boundary = 'bottom'
value = 293 #RT
[../]
[]

[AuxVariables]
[./f_density]   # Local energy density (eV/mol)
order = CONSTANT
family = MONOMIAL
[../]
[]

[Kernels]
[./HeatConduction]
type = HeatTransportMetal
variable = T
coupled_T = T
[../]
[]

[Materials]
[./pfmobility]
type = GenericConstantMaterial
prop_names  = 'M kappa_c kappa_e'
prop_values = '1e-5 3 5e-10'
[../]

[./chemical_free_energy]
# simple double well free energy
type = DerivativeParsedMaterial
f_name = Fc
args = 'c T'
constant_names =        'a0 a1 a2 a3 a4 a5 a6 a7
b0 b1 b2 b3
c0 c1 c2 c3
d0 d1 d2 d3
f0 f1 f2 f3 f4 f5 f6 f7
eV_J_d_T0_Kb
A   B   C   D   E   F   G'
constant_expressions = '0.3856 -0.0973 -0.0467 -0.1945
-0.1856 -0.0044 -3.8366 -4.1231
```

```
0.0003 4.69e-5 -4.9588e-5 1.13e-5
3.14e-9 -2.203e-8 3.9597e-8 -3.09e-8
-1.76e-13 5.5788e-12 -5.93e-12 1.23e-11
0.3817 -0.1007 -0.0485 -0.154 -0.1684 -0.0416
-3.86 -4.167
6.24150934e+18 1e-27 700 8.6173e-5
-2.446831e+04 -2.827533e+04 4.167994e+03
7.052907e+03 1.208993e+04 2.568625e+03 -2.354293e+03'
function = 'Ha:=(c*(1-c)*((a0*(1-2*c))+(a1*(1-2*c))+(a2*(1-2*c))+
(a3*(1-2*c))+(a4*(1-2*c))+(a5*(1-2*c)))+(a6*c)+(a7*(1-c)));
Hb:=b0*(1-c)+b1*(1-c)+b2*(1-c)+b3*(1-c);
Hc:=c0*(1-c)+c1*(1-c)+c2*(1-c)+c3*(1-c);
Hd:=d0*(1-c)+d1*(1-c)+d2*(1-c)+d3*(1-c);
He:=-Ha/T0 + Hc*T0 + Hb*log(T0) + Hd*0.5 * T0^2;
G0:=(c*(1-c)*((f0*(1-2*c))+(f1*(1-2*c))+(f2*(1-2*c))+
(f3*(1-2*c))+(f4*(1-2*c))+(f5*(1-2*c)))+(f6*c)+(f7*(1-c)));
1e+4*eV_J*d*(G0*T/T0+Ha-Hb*T-Hc*T^2+0.5*Hd*T^3+
He*T+T*Kb*(c*log(c)+(1-c)*log(1-c)))'    #

#G0:=(A*c+B*(1-c)+C*c*log(c)+D*(1-c)*log(1-c)+E*c*(1-c)+
F*c*(1-c)*(2*c-1)+G*c*(1-c)*(2*c-1)^2);
#eV_J*d*(G0*T/T0+Ha-Hb*T*log(T)-Hc*T^2+0.5*Hd*T^3+
He*T+T*Kb*(c*log(c)+(1-c)*log(1-c)))

derivative_order = 2
outputs = exodus
[../]
[./electric_field_energy_density]
## Electric potential contribution to local free-energy density (eV/mol).
type = DerivativeParsedMaterial
f_name = f_ele
args = 'c V'
constant_names = 'J_eV A B'
constant_expressions = '1.6e-19 1.0e+19 1e4'
function = 'V*c'
derivative_order = 2
enable_jit = true
[../]
#[./probability]
## This is a made up toy nucleation rate it should be replaced by
## classical nucleation theory in a real simulation.
#  type = ParsedMaterial
#  f_name = P
#  args = c
#  function = 'if(c<0.21,c*1e-8,0)'
#  outputs = exodus
```

```
#[../]
 [./nucleation]
# The nucleation material is configured to insert nuclei into the
# free energy to force the concentration to go to 0.9, and
# holds this enforcement for the whole simulation
# time units (100,000 units)
 type = DiscreteNucleation
 f_name = Fn
 op_names  = c
 op_values = 0.9 ##2
#penalty = 5
#penalty_mode = MIN
 map = map
 outputs = exodus
 [../]
 [./free_energy]
# add the chemical and nucleation free energy contributions together
 type = DerivativeSumMaterial
 derivative_order = 2
 args = c
 sum_materials = 'Fc_Fn_f_ele'
 [../]
 []

 [UserObjects]
 [./inserter]
# The inserter runs at the end of each time step to add nucleation
# events that happend during the timestep (if it converged) to
# the list of nuclei
#type = DiscreteNucleationInserter
 type = DiscreteNucleationFromFile
 hold_time = 100000
#probability = P
#probability = 0.01
 file = nuclei.csv
 [../]
 [./map]
# The map converts the nucleation points into finite area
# objects with a given radius.
 type = DiscreteNucleationMap
 radius = 0.2   #0.5   #10
 periodic = c
 inserter = inserter
 [../]
 []
```

```
[Preconditioning]
[./SMP]
type = SMP
full = true
[../]
[]

##line sample generation in csv format
## disabled for now - as file is being developed
[VectorPostprocessors]
[./line_sample1]
type = LineValueSampler
variable = 'c_f_density'
start_point = '1 0 0'
end_point = '1 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample2]
type = LineValueSampler
variable = 'c_f_density'
start_point = '2 0 0'
end_point = '2 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample3]
type = LineValueSampler
variable = 'c_f_density'
start_point = '3 0 0'
end_point = '3 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample4]
type = LineValueSampler
variable = 'c_f_density'
start_point = '4 0 0'
end_point = '4 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
```

```
[./line_sample5]
type = LineValueSampler
variable = 'c_f_density'
start_point = '5_0_0'
end_point = '5_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample6]
type = LineValueSampler
variable = 'c_f_density'
start_point = '6_0_0'
end_point = '6_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample7]
type = LineValueSampler
variable = 'c_f_density'
start_point = '7_0_0'
end_point = '7_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample8]
type = LineValueSampler
variable = 'c_f_density'
start_point = '8_0_0'
end_point = '8_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample9]
type = LineValueSampler
variable = 'c_f_density'
start_point = '9_0_0'
end_point = '9_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample10]
```

```
type = LineValueSampler
variable = 'c_f_density'
start_point = '10 0 0'
end_point = '10 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample11]
type = LineValueSampler
variable = 'c_f_density'
start_point = '11 0 0'
end_point = '11 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample12]
type = LineValueSampler
variable = 'c_f_density'
start_point = '12 0 0'
end_point = '12 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample13]
type = LineValueSampler
variable = 'c_f_density'
start_point = '13 0 0'
end_point = '13 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample14]
type = LineValueSampler
variable = 'c_f_density'
start_point = '14 0 0'
end_point = '14 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample15]
type = LineValueSampler
```

```
variable = 'c_f_density'
start_point = '15_0_0'
end_point = '15_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample16]
type = LineValueSampler
variable = 'c_f_density'
start_point = '16_0_0'
end_point = '16_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample17]
type = LineValueSampler
variable = 'c_f_density'
start_point = '17_0_0'
end_point = '17_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample18]
type = LineValueSampler
variable = 'c_f_density'
start_point = '18_0_0'
end_point = '18_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample19]
type = LineValueSampler
variable = 'c_f_density'
start_point = '19_0_0'
end_point = '19_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample20]
type = LineValueSampler
variable = 'c_f_density'
```

```
start_point = '20 0 0'
end_point = '20 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample21]
type = LineValueSampler
variable = 'c_f_density'
start_point = '21 0 0'
end_point = '21 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample22]
type = LineValueSampler
variable = 'c_f_density'
start_point = '22 0 0'
end_point = '22 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample23]
type = LineValueSampler
variable = 'c_f_density'
start_point = '23 0 0'
end_point = '23 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample24]
type = LineValueSampler
variable = 'c_f_density'
start_point = '24 0 0'
end_point = '24 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample25]
type = LineValueSampler
variable = 'c_f_density'
start_point = '25 0 0'
```

```
end_point = '25 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample26]
type = LineValueSampler
variable = 'c_f_density'
start_point = '26 0 0'
end_point = '26 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample27]
type = LineValueSampler
variable = 'c_f_density'
start_point = '27 0 0'
end_point = '27 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample28]
type = LineValueSampler
variable = 'c_f_density'
start_point = '28 0 0'
end_point = '28 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample29]
type = LineValueSampler
variable = 'c_f_density'
start_point = '29 0 0'
end_point = '29 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample30]
type = LineValueSampler
variable = 'c_f_density'
start_point = '30 0 0'
end_point = '30 10 0'
```

```
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample31]
type = LineValueSampler
variable = 'c f_density'
start_point = '31 0 0'
end_point = '31 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample32]
type = LineValueSampler
variable = 'c f_density'
start_point = '32 0 0'
end_point = '32 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample33]
type = LineValueSampler
variable = 'c f_density'
start_point = '33 0 0'
end_point = '33 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample34]
type = LineValueSampler
variable = 'c f_density'
start_point = '34 0 0'
end_point = '34 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample35]
type = LineValueSampler
variable = 'c f_density'
start_point = '35 0 0'
end_point = '35 10 0'
num_points = 201
```

```
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample36]
type = LineValueSampler
variable = 'c_f_density'
start_point = '36_0_0'
end_point = '36_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample37]
type = LineValueSampler
variable = 'c_f_density'
start_point = '37_0_0'
end_point = '37_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample38]
type = LineValueSampler
variable = 'c_f_density'
start_point = '38_0_0'
end_point = '38_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample39]
type = LineValueSampler
variable = 'c_f_density'
start_point = '39_0_0'
end_point = '39_10_0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample40]
type = LineValueSampler
variable = 'c_f_density'
start_point = '40_0_0'
end_point = '40_10_0'
num_points = 201
sort_by = y # sort by: x y z id
```

```
execute_on = linear #timestep_end
[../]
[./line_sample41]
type = LineValueSampler
variable = 'c f_density'
start_point = '41 0 0'
end_point = '41 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample42]
type = LineValueSampler
variable = 'c f_density'
start_point = '42 0 0'
end_point = '42 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample43]
type = LineValueSampler
variable = 'c f_density'
start_point = '43 0 0'
end_point = '43 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample44]
type = LineValueSampler
variable = 'c f_density'
start_point = '44 0 0'
end_point = '44 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample45]
type = LineValueSampler
variable = 'c f_density'
start_point = '45 0 0'
end_point = '45 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
```

```
[../]
[./line_sample46]
type = LineValueSampler
variable = 'c_f_density'
start_point = '46 0 0'
end_point = '46 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample47]
type = LineValueSampler
variable = 'c_f_density'
start_point = '47 0 0'
end_point = '47 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample48]
type = LineValueSampler
variable = 'c_f_density'
start_point = '48 0 0'
end_point = '48 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample49]
type = LineValueSampler
variable = 'c_f_density'
start_point = '49 0 0'
end_point = '49 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
[./line_sample50]
type = LineValueSampler
variable = 'c_f_density'
start_point = '50 0 0'
end_point = '50 10 0'
num_points = 201
sort_by = y # sort by: x y z id
execute_on = linear #timestep_end
[../]
```

```
[]

[Postprocessors]
[./dt]
type = TimestepSize
[../]
[./ndof]
type = NumDOFs
[../]
[./rate]
type = DiscreteNucleationData
value = RATE
inserter = inserter
[../]
[./dtnuc]
type = DiscreteNucleationTimeStep
inserter = inserter
p2nucleus = 0.0005
dt_max = 10
[../]
[./update]
type = DiscreteNucleationData
value = UPDATE
inserter = inserter
[../]
[./count]
type = DiscreteNucleationData
value = COUNT
inserter = inserter
[../]
[]

[Adaptivity]
[./Indicators]
[./jump]
type = GradientJumpIndicator
variable = c
[../]
[../]
[./Markers]
[./nuc]
type = DiscreteNucleationMarker
map = map
[../]
[./grad]
type = ValueThresholdMarker
```

```
variable = jump
coarsen = 0.1
refine = 0.2
[../]
[./combo]
type = ComboMarker
markers = 'nuc_grad'
[../]
[../]
marker = combo
cycles_per_step = 3
recompute_markers_during_cycles = true
max_h_level = 3
[]

[Executioner]
type = Transient
scheme = bdf2
solve_type = 'PJFNK'
petsc_options_iname = '-pc_type -sub_pc_type'
petsc_options_value = 'asm      lu          '

nl_max_its = 20
l_tol = 1.0e-4
nl_rel_tol = 1.0e-10
nl_abs_tol = 1.0e-10
start_time = 0.0
num_steps = 120

[./TimeStepper]
type = IterationAdaptiveDT
dt = 10
growth_factor = 1.5
cutback_factor = 0.5
optimal_iterations = 8
iteration_window = 2
timestep_limiting_postprocessor = dtnuc
[../]

[./Adaptivity]
coarsen_fraction = 0.1
refine_fraction = 0.7
max_h_level = 2
[../]
[]
```

```
[Outputs]
exodus = true
print_linear_residuals = false
[./mycsv]
type = CSV
file_base = csvdir/linevalues
[../]
[]
```