

Transient Simulation Tool for RRAM Reset Processes

Why this work?

An essential aspect of the reset process in resistive random-access memory (RRAM) is the thermal inertia of the filament during cooling. When the applied voltage pulse decreases, the filament does not cool synchronously with the voltage waveform. Instead, its temperature exhibits a delayed tail: the decay of temperature is slower than the decay of the driving voltage that generates the heat. This thermal lag means that, for a given voltage drop, the filament temperature remains higher than it would if the cooling were instantaneous. As a result, the filament spends more time at elevated temperatures. Since filament dissolution follows an Arrhenius-type dependence on activation energy, the removal of filament material is exponentially accelerated with temperature. Consequently, these thermal inertia tails become fundamental to the reset process. They ensure that the filament is reduced more rapidly and irreversibly once the hysteresis loop is entered, providing a realistic explanation of why filament rupture accelerates during reset. Without considering this effect, simulations would fail to capture the physics of the process and would underestimate the rate of filament destruction.

It is important to note that **these thermal inertia tails are real, experimentally observed, and measured in the laboratory**. They can only be modeled through two possible approaches:

1. By artificially increasing the density and the constant-pressure heat capacity of the materials forming the filament and its dielectric matrix.
2. By introducing nonlinear Fourier-type heat transport terms in the thermal equations.

The first approach is unphysical, as it artificially adds matter that does not exist. The second approach, based on nonlinear transport, is more realistic and provides a consistent explanation of the observed hysteresis and accelerated filament rupture.

This work specifically addresses this matter by implementing nonlinear heat transport physics, ensuring that the experimentally observed thermal inertia and accelerated filament rupture are captured realistically in the reset process of RRAM.

Introduction

The core motivation behind developing this new simulation framework was the need to incorporate **nonlinear heat transport physics** into the model. In conventional approaches, the hysteresis observed in experimental data — particularly the acceleration of filament destruction during reset — could not be explained satisfactorily. Previously, to reproduce experimental results, it was common practice to **artificially increase** both the material density and its constant-pressure specific heat capacity in the simulation. While this could yield curves that matched measured data, the method was **physically unrealistic** and masked the true underlying mechanisms.

Our updated model resolves this by explicitly considering **nonlinear thermal transport processes**. This allows the simulation to capture the hysteresis effect naturally, without resorting to unphysical parameter adjustments, and to reproduce the observed acceleration of filament rupture under realistic conditions.

The following video explains this issue: <https://drive.upm.es/s/gG7RfgdbLpwyNDK>

The Repository

This repository (<https://github.com/maxwellfree/RRAM-models-with-COMSOL6.0-new-physics->) contains tools and scripts for the transient simulation of RRAM devices undergoing a **reset process** — the physical destruction of a conductive filament within the dielectric layer.

This repository contains the essential components required for transient simulations of resistive random-access memory devices during the reset process. It includes two MATLAB scripts: **connect.m**, which establishes the connection between MATLAB and the COMSOL Multiphysics server, and **FindResetTrans.m**, which runs the transient simulation of filament rupture while tracking temperature, voltage, and current. The COMSOL model file **Ni-HfO2-Si.mph** provides the device geometry and standard physics for the RRAM cell, while **RRAMNonLinear6_2.mphphb** introduces the new nonlinear heat transport physics developed to realistically capture hysteresis and accelerated filament destruction. Together with the licensing information and documentation files, these components form a complete environment for studying and reproducing reset dynamics in RRAM devices.

MATLAB+COMSOL Video

A walkthrough video is available that shows how to perform the transient reset simulation of an RRAM device using MATLAB in conjunction with COMSOL Multiphysics, leveraging LiveLink. In this video, you'll see how MATLAB repeatedly calls the COMSOL model in short time steps, updating parameters, running the simulation, and recording the evolution of key variables such as maximum temperature, current, and voltage. It offers a clear, step-by-step guide to automating transient simulations via LiveLink in a practical research context.

The video demonstrates **how a train of voltage pulses is divided into small time intervals** in order to evolve the transient simulation algorithm step by step. Within each interval, the geometry of the conductive filament is progressively updated according to the filament reduction equation, which reflects the physical shrinkage of the filament during reset. This incremental modification of the filament geometry, and therefore the computational domain, takes place while the applied voltage is increased or decreased. The process naturally incorporates a constructive feedback loop: at every step, MATLAB coordinates COMSOL to solve the Poisson equation, the heat equation, and the filament transformation equation. By doing so, the simulation dynamically accounts for both the electrical and thermal effects that accelerate filament rupture. In practice, MATLAB acts as the orchestrator, ensuring that the geometry inside the COMSOL model evolves consistently with the applied voltage pulses and the underlying physical processes.

Video (MATLAB): <https://drive.upm.es/s/LbDB6RLMEyDN2pM>

Video (Reset verification): <https://drive.upm.es/s/9W3CjiXP8B4fojg>

Formulation

The video at the end of this text provides an explanation of the weak formulation that has been employed to build the physics in the COMSOL Physics Builder. Following this, a second video will explain how this mathematical formulation is used to construct the physics within the builder itself. In particular, **the discussion will highlight the equivalences between the syntax that is naturally used when formulating a problem in weak form in the finite element method, and how that same structure is implemented in the COMSOL Physics Builder.** This includes the treatment of boundary conditions of both Dirichlet and Neumann types, showing how the standard weak formulation translates into the builder's framework

Video: <https://drive.upm.es/s/9kEmDfLS99fx5r4>

COMSOL Physics Builder

The video at the end of this text is a step-by-step visual explanation of how the physics has been created in COMSOL based on the weak formulation previously discussed. It shows in detail how

this formulation is translated into the COMSOL Physics Builder and then incorporated into the simulator. Finally, the video demonstrates how the constructed physics can be used within COMSOL to build and run a complete model.

Video: <https://drive.upm.es/s/em6rHxcLWy7n8sk>

COMSOL model using new physics

At the end of this text, a video is provided that details, step by step, the process of creating the COMSOL model that will be used in the simulations. **This model is built using the new physics** that we have developed through the Physics Builder, which was explained previously.

Video: <https://drive.upm.es/s/Eqatgf8FSsN5MgF>

For any questions or clarifications regarding the procedures described, you are welcome to contact me directly by email. I will be glad to provide support or further explanations concerning the use, creation, and application of these tools.

 [\[em.moreno@upm.es\]](mailto:em.moreno@upm.es)