

12-15-2014

Finite Element Analysis Simulations of Micro and Nano-Electromechanical Sensors for Design Optimization

Nicholas Frank DeRoller
University of South Carolina - Columbia

Follow this and additional works at: <http://scholarcommons.sc.edu/etd>

Recommended Citation

DeRoller, N. F. (2014). *Finite Element Analysis Simulations of Micro and Nano-Electromechanical Sensors for Design Optimization*. (Master's thesis). Retrieved from <http://scholarcommons.sc.edu/etd/2940>

This Open Access Thesis is brought to you for free and open access by Scholar Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact SCHOLARC@mailbox.sc.edu.

FINITE ELEMENT ANALYSIS SIMULATIONS OF MICRO AND NANO-
ELECTROMECHANICAL SENSORS FOR DESIGN OPTIMIZATION

by

Nicholas Frank DeRoller

Bachelor of Science
University of South Carolina, 2012

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Electrical Engineering

College of Engineering and Computing

University of South Carolina

2014

Accepted by:

Goutam Koley, Director of Thesis

MVS Chandrashekhar, Committee Member

Lacy Ford, Vice Provost and Dean of Graduate Studies

© Copyright by Nicholas Frank DeRoller, 2014
All Rights Reserved.

DEDICATION

This work is dedicated to my wife, Amanda DeRoller, who has been the muse and guiding light during this long journey. Also to my late grandmother, Patricia Condon, who showed me through example what it means to be a loving human being, and whose guidance led me into the relationship I have with my wife. And finally to God, without Him I may not have had so many great opportunities to take part in.

ACKNOWLEDGEMENTS

I would like to first thank Dr. Koley. In 2009 he was my professor in my sophomore level electronics course. During an idle conversation in the elevator I told him about how I enjoy simulating circuits, prototyping, and programming. Keeping this in mind he offered me a summer position running finite element analysis (FEA) simulations. I had no idea what I was getting into but he was confident in my ability to learn the subject. After learning FEA he offered me a longer term position, where my love for the research field grew. My hard work in his lab was returned by Dr. Koley in greater magnitude by his support within the lab, the classroom, and in my personal life. My time with Dr. Koley has opened many doors for me, and has allowed me to take part in some truly fantastic and groundbreaking research opportunities, a few of which are presented in this work. For this I will be forever thankful.

I would also like to thank all of my lab partners and friends who I have worked with through the years that made progress in these projects possible. First is Dr. Mohammed Qazi, who I considered my mentor when I started in Dr. Koley's lab and started me on the AlGaIn/GaN cantilever project. Also Abdul Talukdar (Hafiz) and Ifat Jahangir who continued this project and whose continued help allowed me to greatly refine the accuracy of my simulations. Alina Wilson, who I worked with on the InN nanowire project, was a fantastic resource for these simulations and a large help in understanding some of the deeper mathematics. Finally I would like to thank Dr. Amol

Singh, Ahsan Uddin, and Yihao Zhu, whose discussions helped me become a better and well-rounded researcher.

I would finally like to thank my parents. My father, Frank DeRoller, for showing me that even nerds can be cool guys. He introduced me to programming at an unusually early age, showing me how to make batch-file menus in DOS so that I could easily get to my games before there was such thing as a Windows operating system. This was the seed that began my love for programming and technology in general. My mother, Charlotte Choate, whose education and ascent to success may serve as an inspiration to anyone. She climbed to the top during a time when many things were considered a “man’s world” and showed that any amount of adversity could be overcome with hard work and dedication. She is also the person who taught me that, “All you need to do is ask, the worst anyone can say is ‘no’!” I would finally like to thank my stepfather, Dan Choate, who taught me through example that you can get far through a good social skills. By observing his charisma and how easily he talked to others I realized that I would do well by breaking out of my shell and opening up to my neighbors. Without this influence I may never have spoken with Dr. Koley that time in the elevator.

In general, I am very fortunate to have been able to surround myself with so many great people. I could go on writing about all of the other influences I have had, and I apologize to the ones left out. This journey was not a one-man feat, but required the help of many. For this I am truly thankful.

ABSTRACT

Micro and Nano-electromechanical sensors (MEMS and NEMS) provide a means of actively sensing minute changes in the surrounding environment. Small changes in temperature, momentum, and strain may be sensed in passive modes while greater sensing possibilities exist in active modes. Theoretical femto-gram resolution mass detection and heated element sensing methods may be used while volatile organic compound (VOC) sensing may be achieved when combined with a functionalization layer or device heating. These devices offer a great reduction in cost and offer increased mobility by allowing a "lab-on-chip" solution for the prospective user while also greatly reducing the amount of energy consumed by current sensor designs and equipment set-ups.

Work has been done in our lab on AlGaN/GaN MEMS cantilevers and InN NEMS nanowire sensors for use as NO_x and VOC sensing applications. Research and development of optimal mechanical designs for these proposed sensors can be a very iterative, and therefore an expensive process. The devices must be grown (in the case of NEMS) etched, and characterized; a process taking several weeks or months and often involving the use of advanced facilities. Because of this great time and material cost methods are needed to reduce the number of iterations to optimal design and streamline the research and development process to achieve a faster time to end-product.

Finite element analysis (FEA) simulations allow the researcher to test hundreds of proposed designs within weeks. Allowing a research and development team to reduce the

number of proposed design configurations as well as propose and test the feasibility of new sensor designs for future works. Using full-featured simulation packages such as COMSOL Multiphysics allows a researcher to not only model the mechanical properties of the proposed MEMS/NEMS sensors, but also provides the ability to couple together other attributes of the complete device model such as environmental losses, joule heating effects, and polarization. This can be done either within the simulation software itself, or through coupling with external packages such as MATLAB. Here COMSOL simulations, as well as the simulation methodology, will be presented for the cases of the AlGaIn/GaN resonant cantilever sensor as well as the InN nanowire based design. In the case of the AlGaIn/GaN MEMS sensor some simulation results will be compared to experimental measurements to show the feasibility of this research step for micro-scale and nanoscale mechanical sensor design.

TABLE OF CONTENTS

Dedication	iii
Acknowledgements.....	iv
Abstract.....	vi
List of Tables	x
List of Figures.....	xi
List of Symbols.....	xiii
List of Abbreviations	xiv
Chapter 1: Cantilever Based Sensing & Device Simulation.....	1
Chapter 2: AlGaIn/GaN Cantilever Model	4
2.1 Model Simplifications & Adjustments	5
2.2 Model Meshing	10
2.3 Model Simulation Sequence	12
2.4 Post-processing	12
2.5 Simulation Outcomes.....	13
2.6 System Optimization.....	17
Chapter 3: InN Resonator Model.....	19
3.1 Model Geometry & Mesh.....	19

3.2 InN Resonator Simulation Outcomes	20
3.2 InN Cantilever Simulation Outcomes	22
Chapter 4: Conclusion.....	26
References.....	27
Appendix A: AlGaIn/GaN Cantilever COMSOL Scripting MATLAB Code	28

LIST OF TABLES

Table 2.1 Simulated Material Constants	6
Table 2.2 Simulated Alloy Constants	6
Table 2.3 Spontaneous & Piezoelectric Polarization Parameters	13
Table 3.1 InN Material Constants	21
Table 3.2 InN Piezoelectric Constants	23

LIST OF FIGURES

- Figure 2.1: Schematic diagram of the AlGa_N/Ga_N microcantilever with mesa at the base. Inset shows the enlarged view of the mesa indicating source, drain, and gate contacts of the AlGa_N/Ga_N HFET4
- Figure 2.2: Cross section of mesa along y-direction showing different metal stacks (not drawn to scale) on the left. On the right we have consolidated these stacks using Vegard's law. This operation greatly reduces simulation size.5
- Figure 2.3: SEM of the base of the cantilever ~5 μm chamfer and filets can be seen on this geometry. Though, to save mesh density only chamfers will be used in the simulations.7
- Figure 2.4: Non-idealities are shown in this model. Chamfers are given to the base and top, as well as the underside of the cantilever.8
- Figure 2.5: SEM showing side-view of the AlGa_N/Ga_N Cantilevers. These show that there is some pre-strain present in the cantilevers that cause a static deflection after etching release. (SEM by Dr. Muhammad Qazi).....9
- Figure 2.6: Meshed model. (a) Model is colored black because of the high mesh density on the surface. This is caused by the thin, high-aspect-ratio domains. (b) A close-up is shown of the meshing area, showing a tetrahedral mesh over the body and a quad swept-mesh across the mesa. (c) An extreme zoom is shows the side of the mesa. (SEM by Dr. Muhammad Qazi)11
- Figure 2.7: Actual cantilever measured using the Laser Vibrometer MSA 500. Measurements are taken in real time by oscillating the cantilever using a piezo-chip while a laser scans the cantilever over the shown grid. This gives information about resonant frequency, velocity, and amplitude. The measured resonant frequency is 43.0 kHz. (Measured by Abdul Talukdar).....15
- Figure 2.8: 2DEG post-processing polarization output for all four bending simulations. The blue 'shadow' underneath the deformation plots is the calculated polarization for comparison. The red dotted line goes across the low polarization zones.16
- Figure 2.9: SEM of the optimized FET with simulated polarization given a bending of 20 μm. (SEM by Alina Wilson and device fabrication by Abdul Talukdar).....18
- Figure 3.1: SEM of the suspended nanowire this simulation models. (SEM by Dr. Ehtesham Bin Quddus and growth by Alina Wilson).....19

Figure 3.2: Resonator model simulated. Each of the Si ends is rigidly bound and acts as anchors for the model.....20

Figure 3.3: Resonant sweep output of the resonator (left). Attached mass resolution of the resonator (right)21

Figure 3.4: Cross-section of a very small nanowire with polarization charge signs noted.23

Figure 3.5: Top polarization of both 100 nm and 10nm diameter model.24

Figure 3.6: Bottom polarization of both 100 nm and 10nm diameter model.25

LIST OF SYMBOLS

f_0	Resonant frequency
B	General material constant variable
E	Young's modulus
ν	Poisson's ratio
ρ	Density
α	Coefficient of thermal expansion
κ	Thermal conductivity
C_p	Specific heat at constant pressure
ϵ_{xx}	Strain in the x-direction
a	Lattice constant
P	Polarization
P_{SP}	Spontaneous polarization
P_{PZ}	Piezoelectric polarization
t	Thickness
l	Length
ϵ_r	Relative Permittivity
σ	Conductivity

LIST OF ABBREVIATIONS

2DEG	Two-dimensional electron-gas
DOF.....	Degrees-of-freedom
FEA.....	Finite element analysis
HEMT	High-electron-mobility transistor
MEMS.....	Micro-electromechanical systems
NEMS	Nano-electromechanical systems
NESL.....	Nano-Electronics & Sensors Lab
SEM	Scanning electron microscope –or– Scanning electron micrograph
VOC	Volatile organic compounds

CHAPTER 1: CANTILEVER BASED SENSING & DEVICE SIMULATION

Cantilever-based Micro and Nano-electromechanical sensors (MEMS and NEMS) provide a means of actively sensing minute changes in the surrounding environment through many different avenues. Current common consumer-grade MEMS sensors may be found in portable devices and may detect humidity or acceleration. Recently a great amount of interest has been given towards using MEMS and NEMS sensors for selective gas detectors for use in many different fields including: medical, automotive, industrial, environmental, as well as the defense sector.

While there are currently means of detecting target chemicals in the market today with fair accuracy, many of these suffer from several disadvantages including high-cost, slow response time (in the order of minutes), bulkiness, high power consumption, and high cost. Several reports have shown that MEMS based detection is an attractive alternative to standard chemical sensors to improve detection sensitivity (Hu, Thundat and Warmack 2001). These devices may be made to detect specific chemicals or biological molecules through changes in resonant frequency or surface stress. This may be through an attached mass detection, electrostatic variation, to characteristic changes due to a nearby functionalization layer (Qazi and Koley 2008).

NEMS sensors provide an even greater potential given their higher mechanical sensitivity to changes. Because of their small relative size, individual molecules may be measured due to changes in surface stress or resonant frequency of the device.

Theoretical femtogram resolution mass detection and heated element sensing methods may be used to detect these molecules or volatile organic compounds (VOC).

These devices offer a great reduction in cost and offer increased mobility by miniaturizing devices and creating a "lab-on-chip" solution for the prospective user while also greatly reducing the amount of energy consumed by current sensor designs and equipment setups. While there is great promise in the avenue of MEMS and NEMS sensors for selective gas detection, there is still much work to be done to make the devices more sensitive, repeatable (concerning manufacturing), and selective.

Methods are needed to help better characterize these devices beyond standard physical measurement techniques commonly used today. There is a need to be able to visualize the physics taking place in these mechanical sensors using advanced applications of analytical theory in order to predict a model's behavior or troubleshoot a structure's shortcomings.

Using finite-element analysis (FEA) we can model these complex structures with great accuracy and observe the behavior of not just the device under test, but many other variations of the device with very little effort or extra cost to the researching laboratory. By doing this the laboratory may save thousands of dollars in fabrication costs and hundreds of man-hours in finding the optimal design by running through hundreds of potential designs to find a few optimal designs to fabricate and test, cutting out most of the iterative process in research and development.

In this manuscript we examine an AlGaIn/GaN based MEMS cantilever made at the Nano-electronics and Sensors Lab (NESL) that has a HEMT placed at the suspended area near the base. The operating concept of this sensor is that the HEMT acts like a

piezoresistive element. Because HEMT hetero-structures use a strain-induced 2D electron-gas (2DEG) we can expect the polarization at the 2DEG to be modulated by bending induced strains in this location, thus changing the conductivity across the channel.

This MEMS model has shown some responsiveness, but has generally been plagued with low sensitivity to bending. The FEA simulations herein are constructed to attempt to examine what exactly is happening in the 2DEG region, and show some possible design optimizations which may improve this sensitivity.

Additionally, this document will cover the simulation of two InN NEMS sensors, a resonator and a nanowire cantilever. These devices are in the early stages of development and thusly have not yet been made into devices or fully characterized. Here we hope to show how FEA may be used to examine possible uses for these new and upcoming devices (in the case of the resonator) or characterize a property of a device where standard characterization methods may fall short or be prohibitively difficult (nanowire-based cantilever case).

CHAPTER 2: ALGaN/GaN CANTILEVER MODEL

Figure 2.1 shows the COMSOL model of the AlGaN/GaN microcantilever ($250\mu\text{m}\times 50\mu\text{m}\times 2\mu\text{m}$) used in this work. A $35\mu\text{m}$ by $35\mu\text{m}$ mesa is situated at the base (fixed end) of the microcantilever. The mesa is $0.2\mu\text{m}$ high with a very thin layer (17nm) of AlGaN above that and an additional 2nm GaN layer on top. The source, gate, drain, and tip metal is placed on the top of the entire structure. An extra GaN block is created at the base assigning fixed displacement to replicate the fabricated microcantilever that hangs from a rigid body.

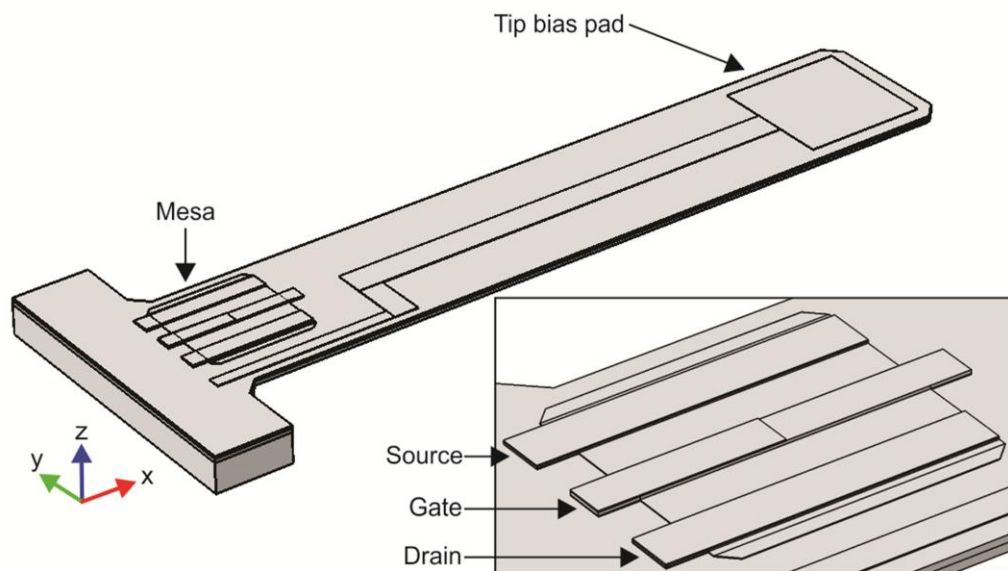


Figure 2.1: Schematic diagram of the AlGaN/GaN microcantilever with mesa at the base. Inset shows the enlarged view of the mesa indicating source, drain, and gate contacts of the AlGaN/GaN HFET.

2.1 MODEL SIMPLIFICATIONS & ADJUSTMENTS

2.1.1 ALLOY APPROXIMATIONS

The model used contains very thin subdomains. To reduce memory requirements and processing time, we first simplify the metal deposition stacks using the principles of Vegard's law. Shown in Figure 2.2, similar domains are found and consolidated through all material constants using Equation 2.1 (altered Vegard's law).

$$B_{composite} = xB_1 + yB_2 + zB_3 \quad [2.1]$$

Here P represents the general material constant property, which can be any constant presented in Table 2.1 and x , y , and z represent the relative percentage of the stack that the particular metal occupies. In the end, the consolidated stack will represent an alloy. Simulated alloy material constants appear in Table 2.2.

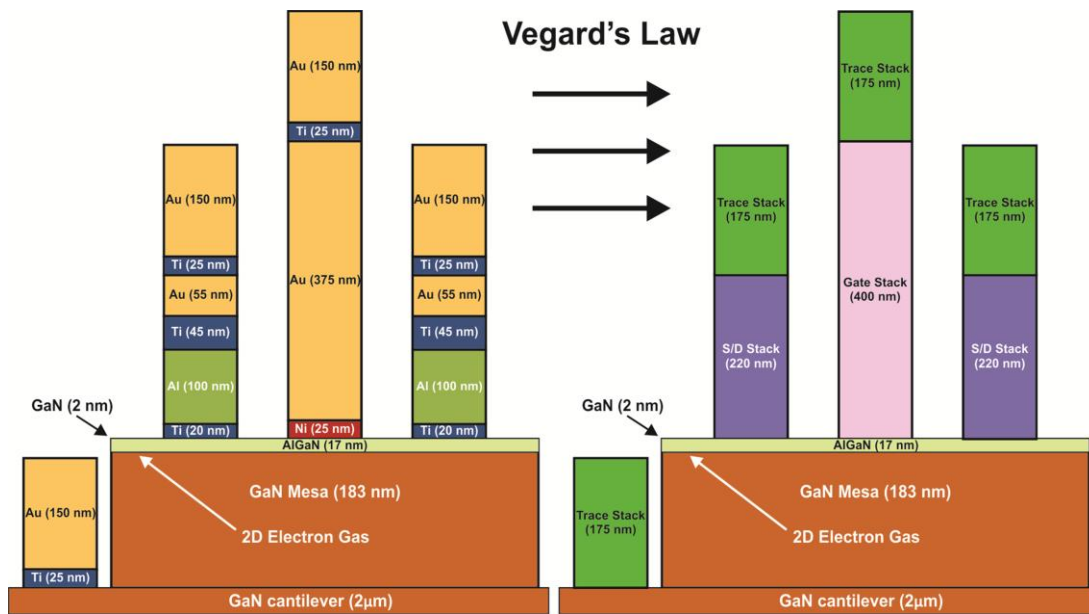


Figure 2.2: Cross section of mesa along y-direction showing different metal stacks (not drawn to scale) on the left. On the right we have consolidated these stacks using Vegard's law. This operation greatly reduces simulation size.

By doing this we greatly reduce the number of subdomains in the metal stacks and by extension the number of meshing points required to solve the problem. This brings the metal stack subdomains to a lower aspect ratio. Because we are not particularly interested in the strain interactions within the stack, but rather how the stack acts upon the mesa and entire cantilever, this is a great approximation to use.

TABLE 2.1 SIMULATED MATERIAL CONSTANTS

	AlGa _N	GaN	Au	Ti	Al	Ni	UNITS
E	185	390	70	115.7	70	219	(GPa)
ν	0.2	0.352	0.44	0.321	0.35	0.31	
ρ	5740	6150	19300	4506	2700	8900	(kg/m ³)
α	10	24	14.2	8.6	24	13.4	$\times 10^{-6}$ (1/K)
κ	52.7	41	318	15.8	205	90.9	W/(m·K)
C _p	0.41	0.24	0.126	0.523	0.9	0.444	J/(g·K)

Using Table 2.1 and Equation 2.1 we may use the metal stack approximation alloy constants shown in Table 2.2.

TABLE 2.2 SIMULATED ALLOY CONSTANTS

	Trace Stack	S/D Stack	Gate Stack	UNITS
E	76.5	83.5	79.2	(GPa)
ν	0.423	0.364	0.348	
ρ	17185	7383	18655	(kg/m ³)
α	13.4	17	14.2	$\times 10^{-6}$ (1/K)
κ	274.8	117.4	303.9	W/(m·K)
C _p	0.183	0.595	0.145	J/(g·K)

2.1.2 GEOMETRY MODIFICATIONS

Over time we have found that while a standard cantilever shape may be easily verified using Hooke's law, it does not make a simulation that will agree with a real-world micro-cantilever. These cantilevers will have several non-idealities that will alter its resonant frequency and strain profile at the base. As a quick example we can see in Figure 2.3 chamfers at the base. We can estimate through observation that these are about $5\ \mu\text{m}$ in width. This is used as the standard chamfer width throughout the model, which results in a $\sim 4\text{kHz}$ increase of the resonant frequency, reducing the model error (discussed later). These non-idealities are shown in Figure 2.4.

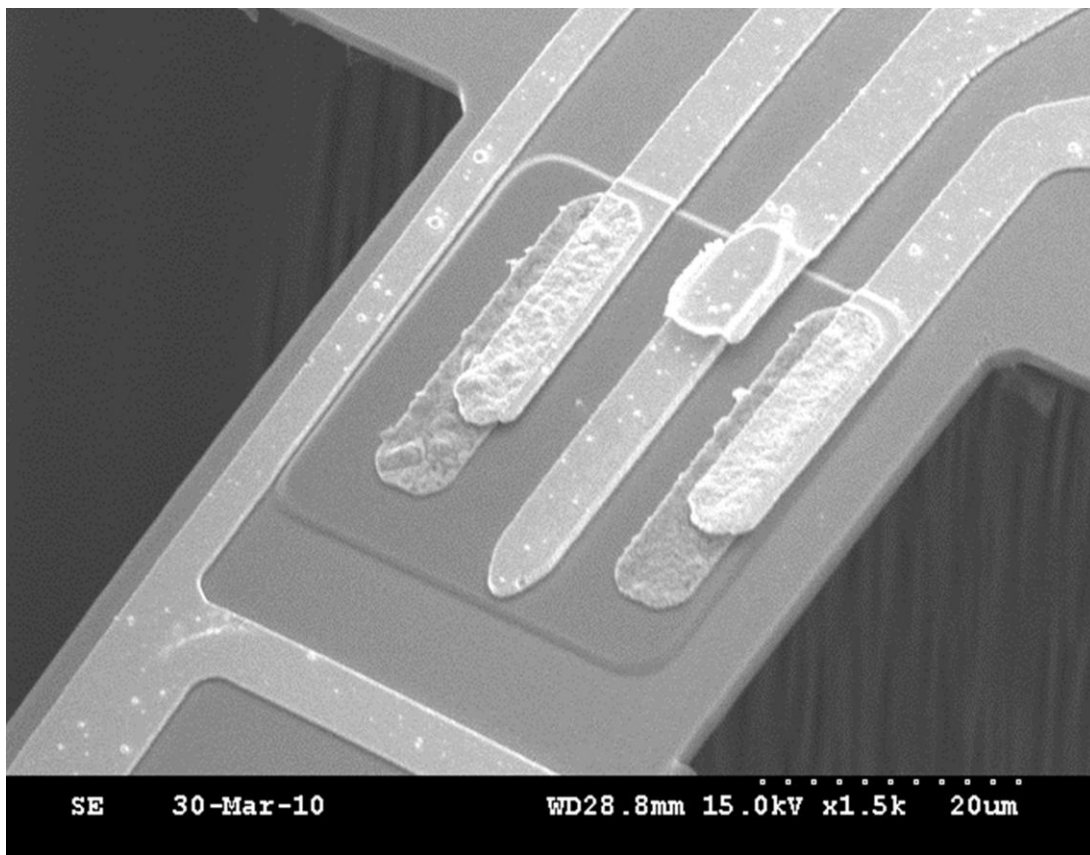


Figure 2.3: SEM of the base of the cantilever $\sim 5\ \mu\text{m}$ chamfer and fillets can be seen on this geometry. Though, to save mesh density only chamfers will be used in the simulations. (SEM by Dr. Muhammad Qazi)

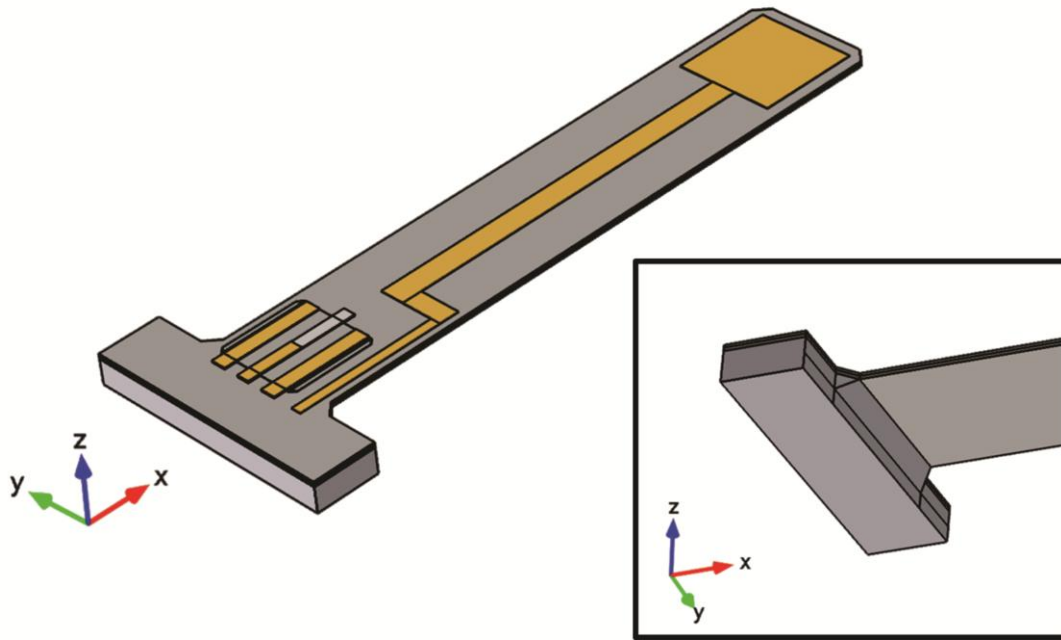


Figure 2.4: Non-idealities are shown in this model. Chamfers are given to the base and top, as well as the underside of the cantilever.

As seen in Figure 2.4, etching inconsistencies are given to the model. We can expect that during the etching process, when the GaN layer is being removed there may be ‘over-etch’ in the convex corners and ‘under-etch’ in the concave corners. The same is assumed on the bottom of the model, shown in the Fig. 2.4 inset. We may expect some Si to be left behind in the corners leaving a chamfer behind.

2.1.3 PRE-STRAIN PREPARATION

The structure being observed is under strain at all times. During growth of the wafer we can expect there to be strains present between the hetero-layers due to lattice mismatch. Some of this strain is attempted to be relieved by a superlattice strain buffer placed between the Si substrate and the GaN layer. When the substrate is removed the superlattice is allowed to relax, creating the deformation seen in Figure 2.5.

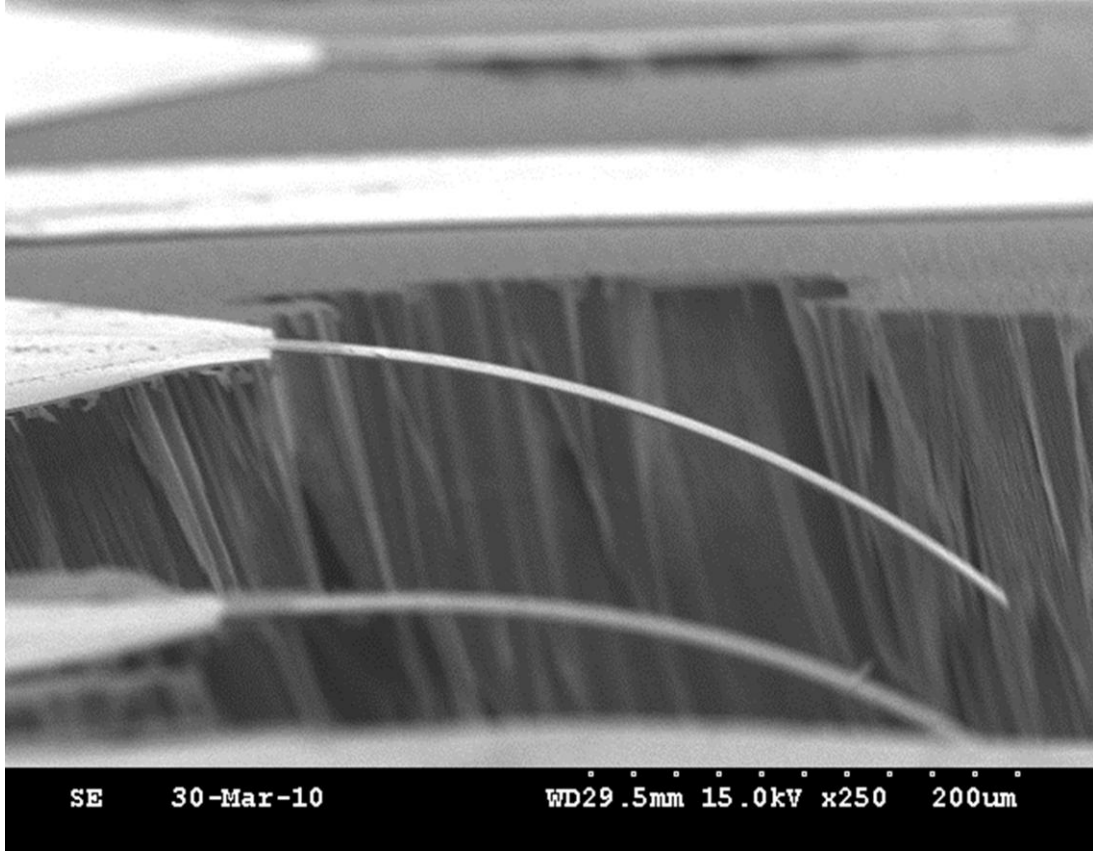


Figure 2.5: SEM showing side-view of the AlGaIn/GaN Cantilevers. These show that there is some pre-strain present in the cantilevers that cause a static deflection after etching release. (SEM by Dr. Muhammad Qazi)

We should also expect strains to exist between the GaN/AlGaIn layers. This strain contributes to the piezoelectric properties of the HEMT hetero-junction and should be modeled. The pre-strain induced deflection observed should also be modeled because we can also expect this deflection to alter the strain profile in our mesa's HEMT, which will in-turn alter its polarization.

To find the pre-strain present within the AlGaIn layer we use the simple lattice mismatch Equation 2.2. Here we can expect the strain to be the same in both x and y-directions.

$$\epsilon_{xy} = \epsilon_{11} = \epsilon_{22} = \frac{a_s - a_f}{a_f} \quad [2.2]$$

Finding the pre-strain present in the buffer layer is a bit more difficult. The wafer manufacturer, Nitronex, uses a proprietary superlattice to relieve stress between the substrate and the GaN layer. Because of this calculating the exact residual stress in this buffer layer can be complicated. Additionally, even if the layers were known the calculations and modeling of this layer would be tedious and very resource heavy. Here we may save some time by applying to the buffer layer an estimated pre-strain. To do this we use Equation 2.2 again to calculate a starting strain between GaN and Si, this will be our middle “best estimate” number. We again calculate the strain between Si and AlN, and again AlN and GaN. Using this range of numbers and some steps in-between we can run some static bending simulations on the model in sweep-mode to observe the tip deflection. This is compared against the measured tip deflection (10 μm) of an actual cantilever to heuristically find the appropriate buffer-layer strain.

2.2 MODEL MESHING

Meshing of this model is particularly challenging because of the extremely high aspect ratios found on the mesa. Here we can see a domain as thin as 2 nm, but as wide as 35 μm ; four-orders of magnitude difference. Unless we have a supercomputer at our disposal, using any ‘auto-mesher’ would be foolhardy. Even with a supercomputer we would still be wasting valuable resources computing this model. Instead we must creatively mesh our geometry starting with the most important part (the focus of the study, the HEMT) moving out to the less important parts. This will allow us to focus on applying the densest mesh around the HEMT, while allowing the ancillary geometry to accept the more course mesh.

We first start by applying a quad mesh with element size of 500 nm to the top of all of the metal alloy stacks and sweeping down to the top of the mesa. At the mesa we again apply the same element size quad mesh across the top. We again sweep down through the GaN layer making it one mesh-element thick. Next we face the AlGaIn/2DEG layer. Here we can again sweep down with one mesh element, but afterwards through the AlGaIn layer we must make it several elements thick and specify a growth rate for these elements, as seen in Figure 2.6.c.

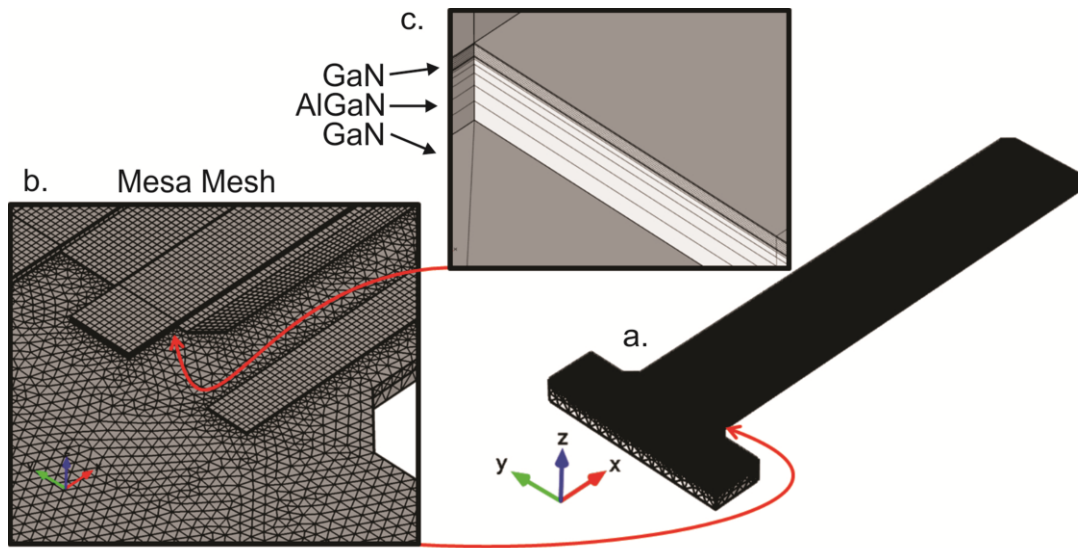


Figure 2.6: Meshed model. (a) Model is colored black because of the high mesh density on the surface. This is caused by the thin, high-aspect-ratio domains. (b) A close-up is shown of the meshing area, showing a tetrahedral mesh over the body and a quad swept-mesh across the mesa. (c) An extreme zoom is shown of the side of the mesa.

The same technique is used on the metal alloy trace running along the cantilever. A free-tetrahedral mesh is then applied to the remaining geometry. Here we should use the coarsest mesh allowable. Coarser meshes can be used if the tetrahedral growth rate is tweaked appropriately. This must be done heuristically, as too coarse of a mesh will not allow convergence.

2.3 MODEL SIMULATION SEQUENCE

Because the model is pre-strained we must account for this by solving for initial conditions. To do this we run the simulation in static mode, which allows the simulation to let the model ‘relax’ into its bent position. The model is then saved as an initial condition which will be used as a starting point for the desired simulation runs.

Because our simulation goal is to optimize the operation of the HEMT at the cantilever’s base, simple displacement simulations will show us what we need to know. So the sequence used will first solve for the system at rest, then displace the tip of the cantilever beyond the resting point to measure strains at the 2DEG for use in polarization calculations.

2.4 POST-PROCESSING

Once the simulation is complete we can use the resulting simulation values to run additional calculations. In our case we would like to find the polarization (given by Equation 2.2) within the 2DEG plane of the HEMT. To do this we will first need to calculate the spontaneous polarization, given by Equation 2.3 where x represents the mole fraction, which for our wafer is 0.26.

$$P = P_{SP} + P_{PZ} \quad [2.2]$$

$$P_{SP} = P_{SP,AlN} + P_{SP,GaN}(1 - x) \quad [2.3]$$

Additionally we need to calculate the piezoelectric polarization. This will be done using Equation 2.4 and using COMSOL to probe for the ε_x , ε_y , and ε_z values. These directional strain values match up with the internal COMSOL strain-tensor variables of `solid1.eXX`, `solid1.eYY`, and `solid1.eZZ` respectively.

$$P_{PZ} = e_{33}\varepsilon_z + e_{31}(\varepsilon_x + \varepsilon_y) \quad [2.4]$$

We may additionally calculate the theoretical piezoelectric polarization using Equation 2.5. This equation is commonly used to calculate the piezoelectric polarization of HEMTs which are not incorporated in MEMS devices. For our case it is helpful to calculate this not only as a basis to check if our outcome is correct (because we should expect the outcome to be within the same order of magnitude), but also as a basis of comparison against the theoretical value to show the advantages of this method. The constants used to calculate both Equations 2.3 and 2.5 may be found in Table 2.3 (Bernardini, Fiorentini and Vanderbilt 2001).

$$P_{PZ,est} = 2 \frac{a-a_0}{a_0} \left(e_{31} - e_{33} \frac{C_{13}}{C_{33}} \right) \quad [2.5]$$

TABLE 2.3 SPONTANEOUS & PIEZOELECTRIC POLARIZATION PARAMETERS

Material	P_{SP} C/m ²	a Å	e_{13} C/m ²	e_{13} C/m ²	C_{13} GPa	C_{33} GPa
GaN	-0.034	3.197	-0.37	0.67	68	354
AlN	-0.090	3.108	-0.62	-1.50	94	377

2.5 SIMULATION OUTCOMES

2.5.1 INITIAL CHECKS

Problems can arise during construction and configuration of a complex model such as this. Some domains could terminate close to other domains within a marginally close distance, which will cause a simulation to fail to start. Another failure is a common Jacobean error, which is generally caused by material constants or special boundary conditions being omitted. The more difficult conditions to catch will be meshing issues, incorrect material constants, and incorrect boundary conditions.

Starting a potentially long simulation which may possibly have several million degrees-of-freedom (DOF) without any type of verification of the model can be a

problem, and may provide misleading results. A good simple test that the model is set up correctly is to run a ‘quick’ Eigen-frequency simulation. This will provide the resonant frequency of the structure along with any added losses (the frequency will be shown as a complex number) without the need to run an entire frequency sweep, which in the case of large DOF high aspect-ratio systems may take days.

If the mechanical model being simulated is ‘ideal’ then it can be easily compared to analytical equations based on Hooke’s law, such as Equation 2.6 (Sarid 1994). When comparing the ideal model with the Hookean model we should expect no more than $\pm 10\%$ error. In the case of well-constructed models error less than $\pm 1\%$ is not uncommon.

$$f_0 \cong \frac{t}{2\pi l^2} \sqrt{\frac{E}{\rho}} \quad [2.6]$$

However, the model presented in Figure 2.1 at the beginning of this chapter is far from ideal. With chamfered corners, hetero-layers, and metal-deposition stacks we can expect our simulated Eigen-frequency to be different from the Hookean model. Here it is best to compare our simulation against an actual real-world cantilever. In our case we compare our resonant frequency against one of our better AlGaIn/GaN cantilevers, the same cantilever that the pre-strains were modeled after.

The resonant frequency of the actual cantilever is measured to be 43.9 kHz, shown in Figure 2.7. Compared to the simulated resonant frequency of 44.7 kHz this is 1.8% off of the actual value. As previously stated, this is entirely because of the chamfers and other non-idealities added to the model. Additionally, when the alloy stack is compared to the actual non-alloy metal stack in the Eigen-frequency simulation outcome the error changes only by a few tenths of a percent while the simulation time for the non-alloy stack increases by about 12 hours, more than justifying this simplification

procedure. Given the permissible error level (below $\pm 10\%$) we can conclude that the model is properly set-up and continue on with further simulations.

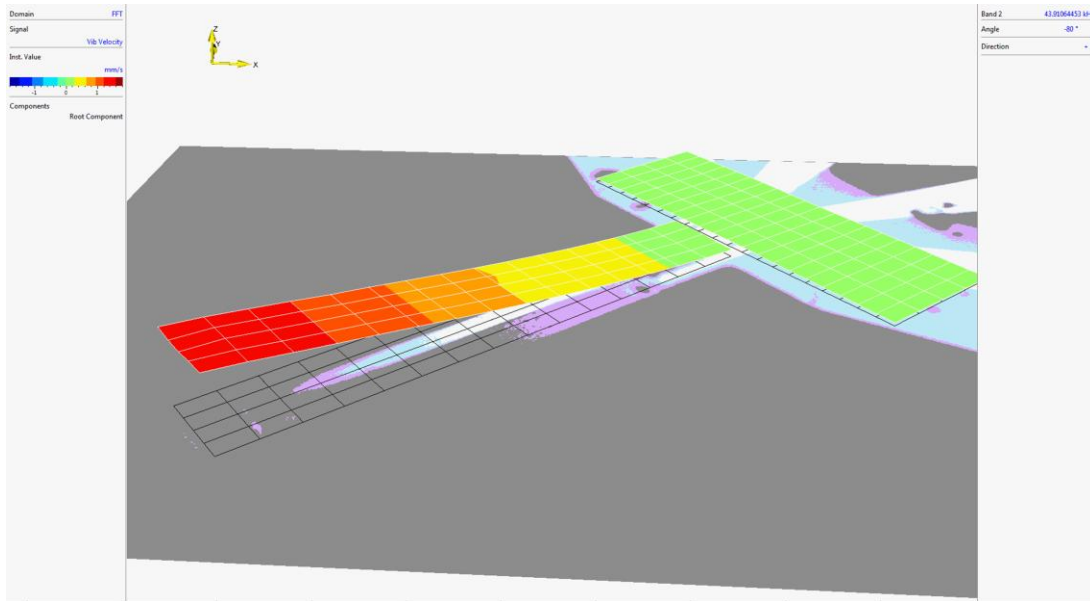


Figure 2.7: Actual cantilever measured using the Laser Vibrometer MSA 500. Measurements are taken in real time by oscillating the cantilever using a piezo-chip while a laser scans the cantilever over the shown grid. This gives information about resonant frequency, velocity, and amplitude. The measured resonant frequency is 43.0 kHz. (Measured by Abdul Talukdar)

2.5.2 STATIC BENDING & POLARIZATION

To measure polarization change from pre-strains and bending we will run static-bending simulations. Here, as mentioned before, we will apply a pre-strain to the model and allow it to deform until it is at rest. After this the model is saved and used as the initial condition for static-bending simulations. Here the tip of the cantilever will be displaced downward to a prescribed distance (0, 10 μm , 20 μm , & 30 μm) and saved for post processing where the polarization at the 2DEG will be calculated.

By using Equations 2.2, 2.3, and 2.4 together we can visualize the polarization at the 2DEG as seen in Figure 2.8. In this picture we see the output from the four simulations

run together for comparison. The color scale on the right is normalized for better comparison between the outputs. Below each of the deformation plots can be seen a dark blue 'shadow'. This is actually the theoretical polarization calculated using Equations 2.2, 2.3 and 2.5 (0.0414 C/m^2). Especially compared to the system at rest this shows the advantage that simulations like this give into gaining insight into what is actually happening at the 2DEG level.

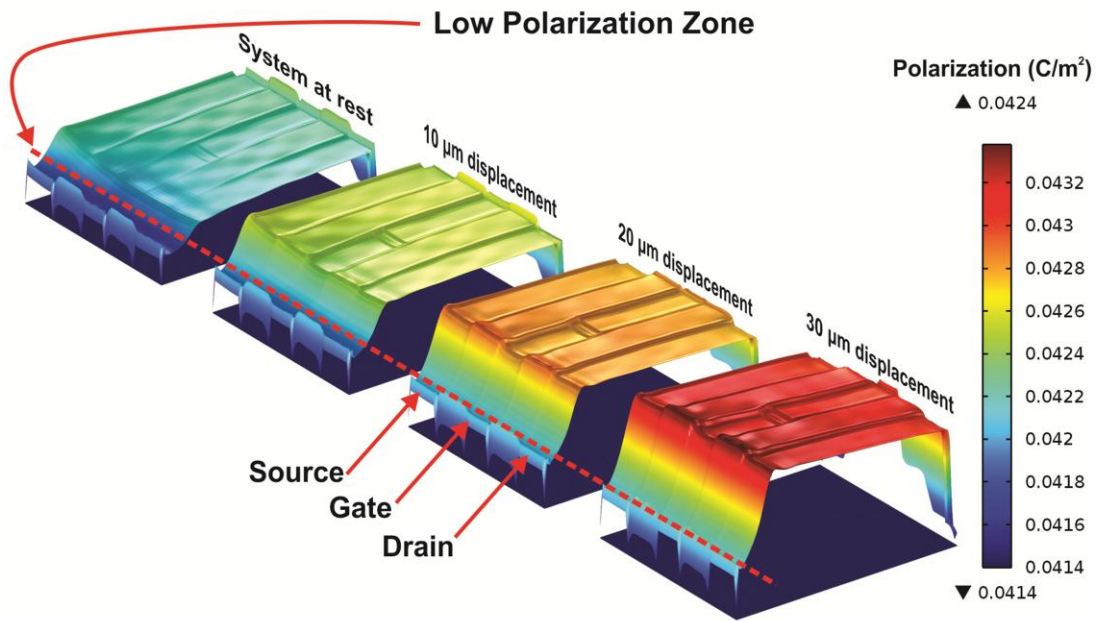


Figure 2.8: 2DEG post-processing polarization output for all four bending simulations. Shown from left to right: system at rest, $10 \mu\text{m}$, $20 \mu\text{m}$, & $30 \mu\text{m}$ deflection. The blue 'shadow' underneath the deformation plots is the calculated polarization for comparison. The red dotted line goes across the low polarization zones.

The most notable part of these simulations when compared to each other is that we see a consistent trend in low polarization in each of the displacement outputs near the HEMT base. This zone coincides with the edge of the cantilever base. Considering that in this device current will flow from the source to the drain, and resistivity is directly proportional to polarization, we see that in each case the current is given a lower

resistance path across the device. If we imagine the device as a several discrete parallel resistors we see that in this base region that there will always be a low-value unchanging resistance. This will have a large effect on the device sensitivity, and in-fact does affect the sensitivity of our cantilevers fabricated in lab. By moving the HEMT further onto the cantilever we may lessen this steep polarization drop-off, but we will still always have a lower polarization zone between the source and drain, again lowering our sensitivity.

2.6 SYSTEM OPTIMIZATION

Because in this presented model we will always have a lower sensitivity because of the lower polarization closest to the base, we may overcome this by turning our HEMT by 90° . Turning the HEMT will cause the current to flow along the length of the cantilever, giving the current path no route around the high polarization area. This will improve the sensitivity of the device compared to the current characterized design.

After constructing a model with the HEMT turned by 90° parametric simulations are run to find where to position the HEMT to best make use of the strains exhibited at this area of the cantilever. After 3 days running the parametric sweep, we find that the best strain distribution is given when the HEMT's gate is positioned $15\ \mu\text{m}$ away from the base of the cantilever. This new device configuration, shown in Figure 2.9, has been fabricated and as of writing this manuscript has yet to be tested.

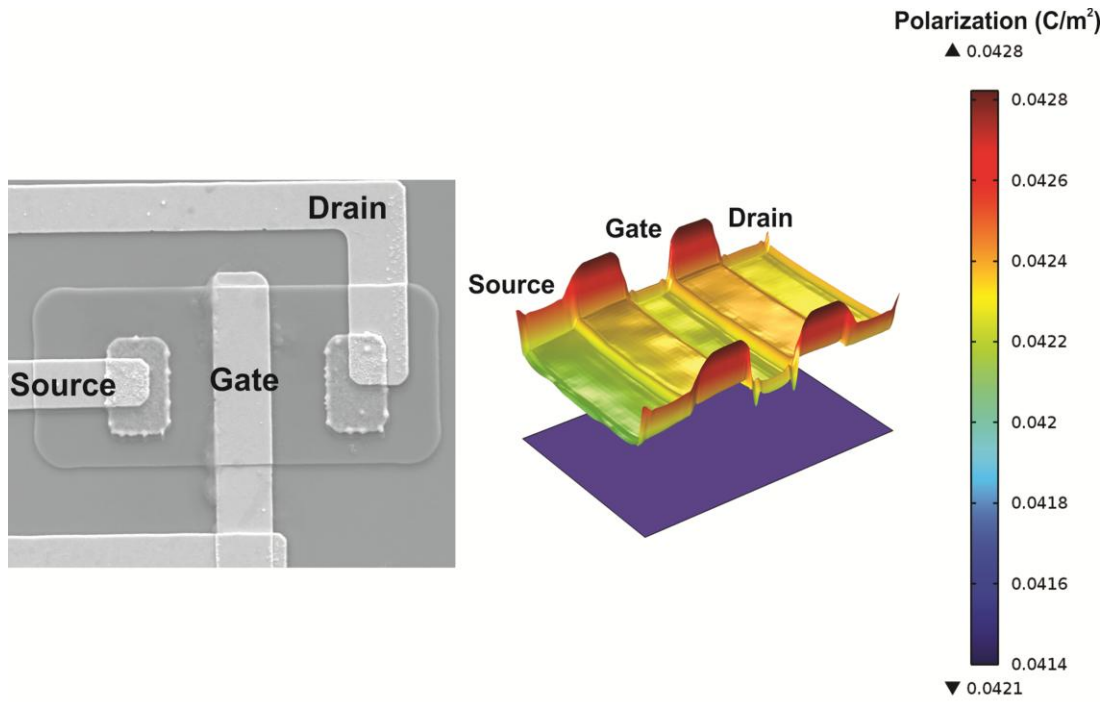


Figure 2.9: Microscope photograph of the optimized FET with simulated polarization given a bending of 20 μm . (SEM by Alina Wilson and device fabrication by Abdul Talukdar)

CHAPTER 3: INN RESONATOR MODEL

We have shown how a model may be simulated to troubleshoot and optimize a structure. Now we examine a structure that is difficult to characterize in order to find some properties that may be challenging to find otherwise. Here we will examine the resonator structure seen in Figure 3.1, as well as a cantilever variation of this model.

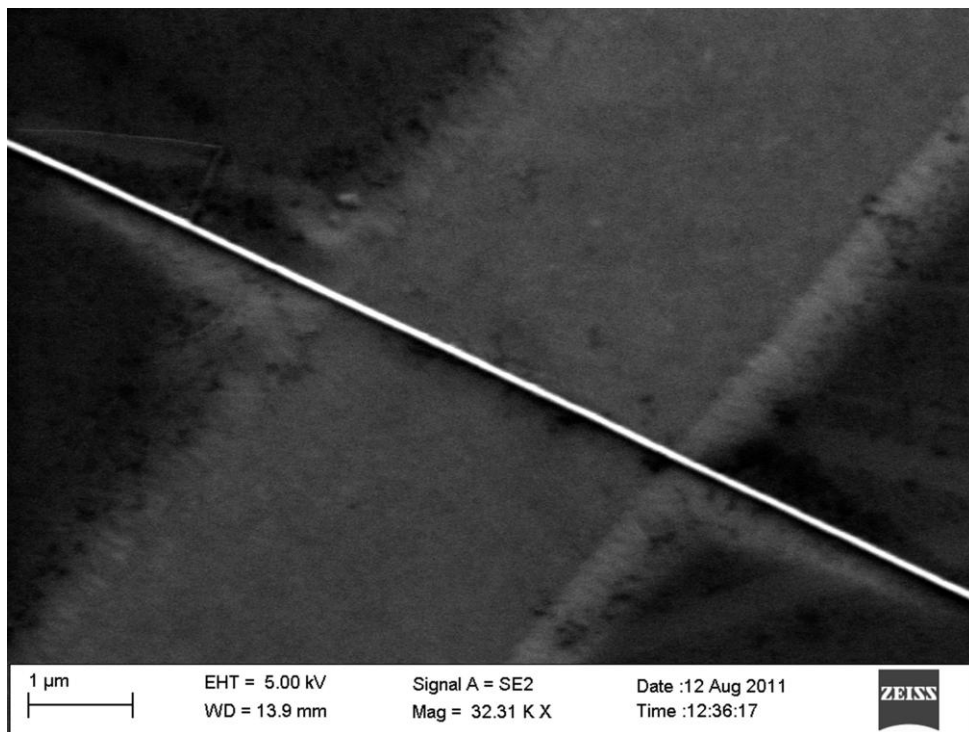


Figure 3.1: SEM of the suspended nanowire this simulation models. (SEM by Dr. Ehtesham Bin Quddus and growth by Alina Wilson)

3.1 MODEL GEOMETRY & MESH

The model is built as a simple InN cylinder with two larger cylindrical anchors made of Si which is rigidly bound, shown in Figure 3.2. A model like this provides no

difficulty in meshing and is a simple model to solve even for a standard laptop. While a course mesh will suffice it is better to use a fine mesh because we will be extracting strain values later to examine polarization. Because data is interpolated between meshing points this will produce a better graph and clear up errors near the binding points.

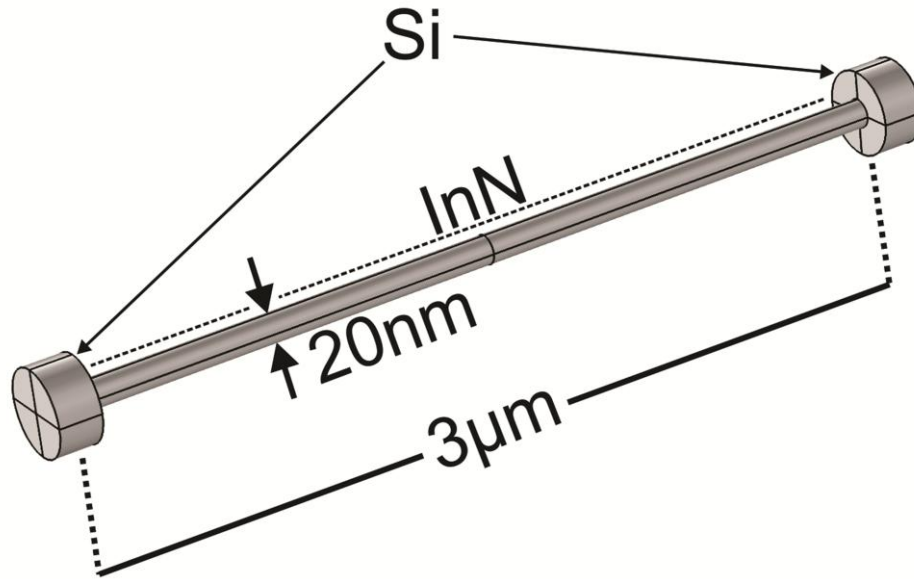


Figure 3.2: Resonator model simulated. Each of the Si ends is rigidly bound and acts as anchors for the model.

While the figure shown above (Fig3.2) is used as the resonator model, a slightly altered version is used for the polarization model. In the polarization model one of the Si anchors is removed and the cantilever shortened to 2 μm . The diameter in this model is also varied between 10 nm and 100 nm to view the effects on polarization for these different diameter structures.

3.2 INN RESONATOR SIMULATION OUTCOMES

Given the bulk material constants shown in Table 3.1, the resonant frequency is simulated to be 10.690 MHz. The frequency sweep shown in Figure 3.3 (left) is run with

a low quality factor of 100 to account for a high-loss free-air case. The outcome of this is used as the basis of comparison for frequency shift given an attached mass.

TABLE 3.1 INN MATERIAL CONSTANTS

Material	E GPa	ν	ρ kg/m ³	α $\times 10^{-6}$ (1/K)	κ W/(m·K)	C_p J/(g·K)	ϵ_r	σ S/m
InN	140	0.21	6810	3.8	45	0.32	15.3	2/3

At the middle of the beam a parametrically controlled added mass is placed to observe the attached mass resolution of the model. Shown in the Figure 3.3 (right), the expected frequency shift compared against the vacuum f_0 and is computed to be $\Delta f = 10\text{Hz}$ given an attached mass of 5 yg (He atomic mass is 6.65 yg). The relative change in frequency in proportion to the attached mass is less as the attached mass approaches the resonator's mass (6.40 fg), as expected.

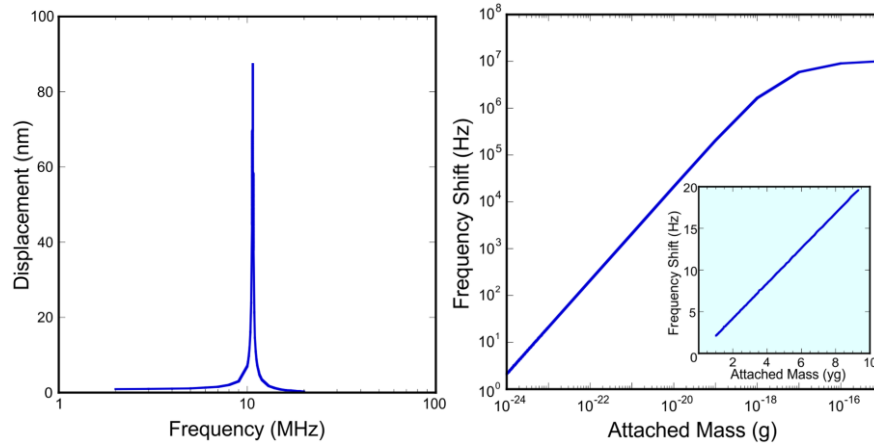


Figure 3.3: Resonant sweep output of the resonator (left). Attached mass resolution of the resonator (right)

Finally, because this resonator could be used as a VOC sensor, Joule heating simulations are run to find the needed operating conditions to reach 300 °C. Here, the anchors are modeled as perfect heat sinks, with a constant temperature of 30 °C so that

we may assume that they represent a large thermal mass. Also, the surface of the resonator is given a heat transfer coefficient of $5 \text{ W}/(\text{m}^2 \text{ }^\circ\text{C})$ which would allow the resonator to act as if it was in a free non-moving air domain, where the external temperature is $30 \text{ }^\circ\text{C}$. Given these conditions the center of the resonator can be expected to reach $300 \text{ }^\circ\text{C}$ with an applied voltage of 2 V (143.4 pA).

Given their small size, these resonators lend themselves to a much higher resonant frequency than MEMS devices offer, making them less susceptible to noise. These structures also show a very high resolution for attached mass measurements as well, with the ability to sense the presence of a mere helium atom. However, as shown in the Joule heating simulations, the strongest application for these resonators lies in the VOC sensing field. Given that current metal oxide based VOC sensing solutions consume such large amounts of power because of the heating required, a projected power consumption in the hundreds of pico-watts (nine orders of magnitude improvement) provides a welcome alternative to current solutions.

3.2 INN CANTILEVER SIMULATION OUTCOMES

For our experimental case observing InN nanowires we can expect that these structures are in an unstrained state when they are finished growing. These structures are grown in the a-direction, as the cross-section pictured in Figure 3.4 shows. Given the stacked wurtzite structure we may expect all charges at the internal shared c-planes to maintain a neutral charge. However we may also expect a net positive or negative charge on the top and bottom surfaces of this structure. When observing the top and bottom surfaces we can expect the same surface charge per unit area for both sub-micron scale as well as nano-scale structure from spontaneous polarization effects.

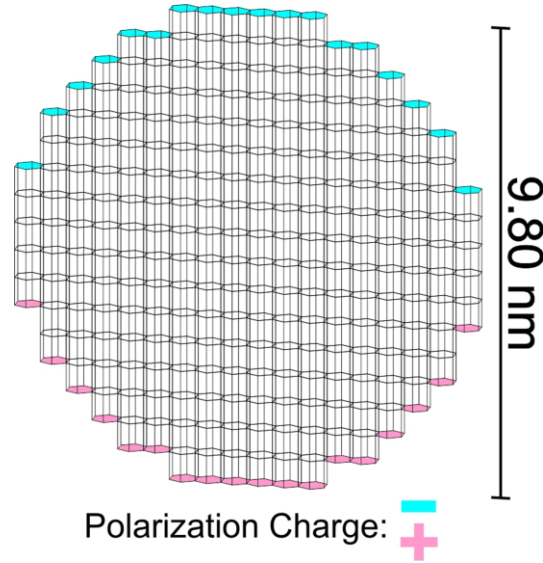


Figure 3.4: Cross-section of a very small nanowire with polarization charge signs noted.

However, because this device is intended to be a NEMS device, which will be subjected to strains, we can expect more interesting effects from piezoelectric polarization when in use. In post-processing strains will be extracted and used to compute the total polarization on the surface given some small displacement. We will compare the polarization changes for both sub-micron and nano-scale devices given the same displacement. In each case the cantilever tip will be given a prescribed displacement of 100 nm. Polarization output will be solved in post-processing by extracting the x, y, and z strain tensors, which are used in the polarization Equations 2.2, 2.3, and 2.4 along with the piezoelectric constants found in Table 3.2 below.

TABLE 3.2 INN PIEZOELECTRIC CONSTANTS

Material	P_{SP} C/m ²	a Å	e_{13} C/m ²	e_{13} C/m ²	C_{13} GPa	C_{33} GPa
InN	-0.042	4.580	-0.45	0.81	70	205

The simulation results are probed along the top and bottom domains along a line because this is where we can most likely expect our polarization charges. In post processing the strain values are extracted here and used to compute the polarization along that line using the constants provided by Bernardini et.al. As expected we can see (Figures 3.5 & 3.6) piezoelectric polarization increasing our polarization value within the higher strain regions, with the strains approaching the spontaneous polarization value at the cantilever tip.

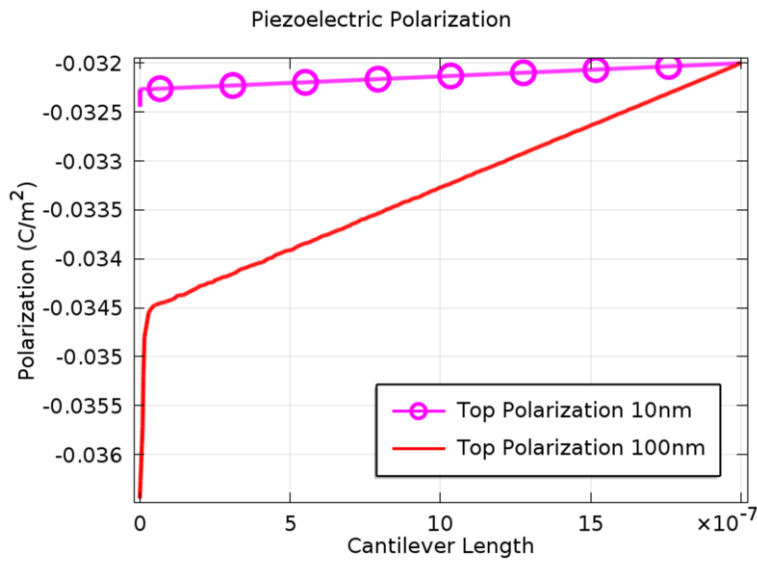


Figure 3.5: Top polarization of both 100 nm and 10nm diameter model.

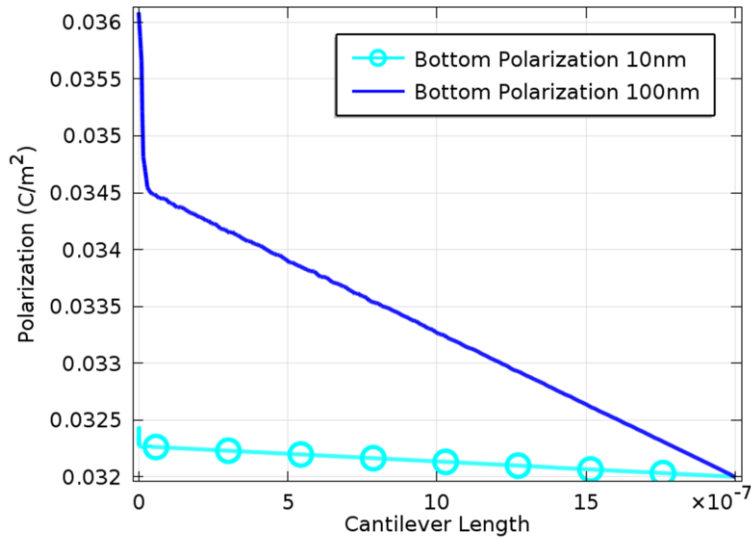


Figure 3.6: Bottom polarization of both 100 nm and 10nm diameter model.

In the resulting plots we can see that there is a difference between the 100 nm and 10 nm diameter model surface polarizations, but this is small. The largest change in piezoelectric polarization occurs at the anchor-point of the larger-diameter cantilever, leading to the conclusion that in some cases the larger-diameter cantilever may be more desirable if it is being used in a resonant or piezo-resistive application, at the loss of mass attachment sensitivity. While the smaller diameter cantilevers will be affected less by the piezoelectric effects and may be more useful for static purposes.

CHAPTER 4: CONCLUSION

We have shown that by using finite element analysis simulations we can parametrically characterize proposed structures to greatly reduce the amount of time and money spent on MEMS research and development.

In the AlGaN/GaN MEMS cantilever we have shown FEA's ability to find issues with a design that may not easily be able to be found through normal characterization methods. Simulations in this case gave us the ability to look 'inside' of the device and observe its operation. This gained insight allowed us to propose design modifications which would improve the sensitivity of the device and possibly solve the devices current sensitivity shortcomings.

In the case of the InN NEMS structures we were able to characterize these suspended devices theoretically. This is important because at the current state of technology it is either impossible or prohibitively expensive to run some of these characterizations on the currently grown devices. By running these simulations we are able to examine the feasibility of the proposed devices, or possibly find other uses for them, to determine if further research into the device is worth the investment.

Further work may be done to add greater complexity to these systems. Such complexities include optimized electrostatic reactions, functionalization layer chemical reactions, and surface acoustic responses.

REFERENCES

- Bernardini, Fabio, Vincenzo Fiorentini, and David Vanderbilt. "Accurate calculation of polarization-related quantities in semiconductors." *Physical Review B* 63, no. 19 (2001): 193201.
- Hu, Zhiyu, T. Thundat, and R. J. Warmack. "Investigation of adsorption and absorption-induced stresses using microcantilever sensors." *Journal of Applied Physics* 90 (2001): 427-431.
- Mandelis, Andreas, and Constantinos Christofides. *Chemistry and Technology of Solid State Gas Sensor Devices*. New York: Wiley & Sons, 1993.
- Qazi, Muhammad, and Goutam Koley. "NO₂ Detection Using Microcantilever Based Potentiometry." *Sensors* 8, no. 11 (2008): 7144-7156.
- Sarid, Dror. *Scanning Force Microscopy: With Applications to Electric, Magnetic, and Atomic Forces*. New York: Oxford University Press, 1994.

APPENDIX A: ALGAN/GAN CANTILEVER COMSOL SCRIPTING MATLAB CODE

The following is an example of code that algorithmically constructs and runs a frequency sweep on a 3-dimensional AlGaIn/GaN cantilever model. This was created for use with COMSOL 3.5 but will also run with newer COMSOL packages as long as the MATLAB scripting interface is included.

```
flclear fem

cantilever.length = 250e-6;
cantilever.width = 50e-6;
cantilever.thick = 2e-6;

cantilever.hemt.thick = 17e-9;
cantilever.mesa.offset = 3e-6;
cantilever.mesa.side = 35e-6;
cantilever.mesa.thick = 200e-9 - cantilever.hemt.thick;

cantilever.hemt.side_offset = 3e-6;
cantilever.hemt.sd_width = 6e-6;
cantilever.hemt.g_width = 5e-6;

cantilever.trace.side_offset = 3e-6;
cantilever.trace.mesa_offset = 15e-6;
cantilever.trace.width1 = 4e-6;
cantilever.trace.width2 = 6e-6;
cantilever.trace.width3 = 6e-6;

cantilever.bias.side = 40e-6;
cantilever.bias.front_offset = 5e-6;

% ==== Metal Deposition Layers ====
% == Source and Drain
% Layer 1 -- [Ti]
cantilever.hemt.sd_11_thick = 20e-9;
% Layer 2 -- [Al]
cantilever.hemt.sd_12_thick = 100e-9;
% Layer 3 -- [Ti]
cantilever.hemt.sd_13_thick = 45e-9;
% Layer 4 -- [Au]
cantilever.hemt.sd_14_thick = 55e-9;

% == Gate
% Layer 1 -- [Ni]
cantilever.hemt.g_11_thick = 25e-9;
% Layer 2 -- [Au]
cantilever.hemt.g_12_thick = 375e-9;
```

```

% == Trace
% Layer 1 -- [Ti]
cantilever.trace.l1_thick = 25e-9;
% Layer 1 -- [Au]
cantilever.trace.l2_thick = 150e-9;

% ***   CONSTANTS   *****
% ***   CONSTANTS   *****
% ***   CONSTANTS   *****
% ***   CONSTANTS   *****
% ***   CONSTANTS   *****
% ***   CONSTANTS   *****

fem.const = {'F_0','0.1e-9', ...
            'F_dist','F_0/50e-6', ...
            'Def','1.877e-10', ...
            'k','F_0/Def'};
fem.descr.const= {'F_0','Force on tip',...
                'F_dist','Calculated distributed force',...
                'Def','deflection after force',...
                'k','Simulated spring constant'};

% ***   MATERIALS   *****
% ***   MATERIALS   *****
% ***   MATERIALS   *****
% ***   MATERIALS   *****
% ***   MATERIALS   *****
% ***   MATERIALS   *****

fem.lib.mat{1}.name='GaN';
fem.lib.mat{1}.varname='mat1';
fem.lib.mat{1}.variables.nu='0.2';
fem.lib.mat{1}.variables.E='210e9[Pa]';
fem.lib.mat{1}.variables.epsilonr='9.5';
fem.lib.mat{1}.variables.alpha='13.4e-6[1/K]';
fem.lib.mat{1}.variables.C='490[J/(kg*K)]';
fem.lib.mat{1}.variables.rho='6150[kg/m^3]';
fem.lib.mat{1}.variables.k='130[W/(m*K)]';
fem.lib.mat{2}.name='Ti';
fem.lib.mat{2}.varname='mat2';
fem.lib.mat{2}.variables.nu='0.36';
fem.lib.mat{2}.variables.E='40e9[Pa]';
fem.lib.mat{2}.variables.sigma='2.6e6[S/m]';
fem.lib.mat{2}.variables.alpha='8.60e-6[1/K]';
fem.lib.mat{2}.variables.C='522[J/(kg*K)]';
fem.lib.mat{2}.variables.rho='4506[kg/m^3]';
fem.lib.mat{2}.variables.k='21.9[W/(m*K)]';
fem.lib.mat{3}.name='Au';
fem.lib.mat{3}.varname='mat3';
fem.lib.mat{3}.variables.nu='0.44';
fem.lib.mat{3}.variables.E='70e9[Pa]';
fem.lib.mat{3}.variables.sigma='45.6e6[S/m]';
fem.lib.mat{3}.variables.alpha='14.2e-6[1/K]';
fem.lib.mat{3}.variables.C='129[J/(kg*K)]';
fem.lib.mat{3}.variables.rho='19300[kg/m^3]';
fem.lib.mat{3}.variables.k='317[W/(m*K)]';
fem.lib.mat{4}.name='Al';
fem.lib.mat{4}.varname='mat4';
fem.lib.mat{4}.variables.nu='0.35';
fem.lib.mat{4}.variables.E='70.0e9[Pa]';
fem.lib.mat{4}.variables.sigma='35.5e6[S/m]';
fem.lib.mat{4}.variables.alpha='23.1e-6[1/K]';
fem.lib.mat{4}.variables.C='904[J/(kg*K)]';

```

```

fem.lib.mat{4}.variables.rho='2700[kg/m^3]';
fem.lib.mat{4}.variables.k='237[W/(m*K)]';
fem.lib.mat{5}.name='Ni';
fem.lib.mat{5}.varname='mat5';
fem.lib.mat{5}.variables.nu='0.31';
fem.lib.mat{5}.variables.E='219e9[Pa]';
fem.lib.mat{5}.variables.sigma='13.8e6[S/m]';
fem.lib.mat{5}.variables.alpha='13.4e-6[1/K]';
fem.lib.mat{5}.variables.C='445[J/(kg*K)]';
fem.lib.mat{5}.variables.rho='8900[kg/m^3]';
fem.lib.mat{5}.variables.k='90.7[W/(m*K)]';
fem.lib.mat{6}.name='AlGaIn';
fem.lib.mat{6}.varname='mat6';
fem.lib.mat{6}.variables.nu='0.2';
fem.lib.mat{6}.variables.E='185e9[Pa]';
fem.lib.mat{6}.variables.alpha='10e-6[1/K]';
fem.lib.mat{6}.variables.rho='5740[kg/m^3]';

% *** GEOMETRY *****
% *** GEOMETRY *****
% *** GEOMETRY *****
% *** GEOMETRY *****
% *** GEOMETRY *****
% *** GEOMETRY *****

% =====
% === Cantilever ===
% =====
canti = block3(cantilever.length,cantilever.width,cantilever.thick,...
    'pos',{0,-cantilever.width/2,-cantilever.thick});

% =====
% === Base ===
% =====
cantilever.base.std = 1.5 * cantilever.thick;
temp.Y = cantilever.width + cantilever.thick;
base = block3(cantilever.base.std,temp.Y,cantilever.base.std,...
    'pos',{-cantilever.base.std,-temp.Y/2,-cantilever.base.std});

% =====
% === Mesa ===
% =====
temp.XY = cantilever.mesa.side;
temp.y = cantilever.width/2 - cantilever.mesa.side - cantilever.base.std;
mesa = block3(temp.XY,temp.XY,cantilever.mesa.thick,...
    'pos',{0,temp.y,0});
hemt = block3(temp.XY,temp.XY,cantilever.hemt.thick,...
    'pos',{0,temp.y,cantilever.mesa.thick});

% =====
% === Metal Deposition =====
% =====
% General computations
% Total layer xy dimensions
% Source % Drain
temp.sd.x = cantilever.base.std + cantilever.mesa.side;
temp.sd.y = cantilever.hemt.sd_width;

% Gate
temp.g.x = temp.sd.x;

```

```

temp.g.y = cantilever.hemt.g_width;

% =====
% === Source Contact (base) ===
% =====
% General computations
temp.y = cantilever.width/2 - cantilever.mesa.offset ...
        - cantilever.hemt.side_offset - cantilever.hemt.sd_width;

% ===== Base Layers =====
sbase0 =
block3(cantilever.base.std,cantilever.hemt.sd_width,cantilever.mesa.thick,...
        'pos',{-cantilever.base.std,temp.y,0});
sbase1 =
block3(cantilever.base.std,cantilever.hemt.sd_width,cantilever.hemt.thick,...
        'pos',{-cantilever.base.std,temp.y,cantilever.mesa.thick});

% ===== Layer 1 =====
% Altitude
cantilever.hemt.sd_l1_altitude = cantilever.mesa.thick + cantilever.hemt.thick;
source10 = block3(temp.sd.x,temp.sd.y,cantilever.hemt.sd_l1_thick,...
        'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l1_altitude));

% ===== Layer 2 =====
% Altitude
cantilever.hemt.sd_l2_altitude = cantilever.hemt.sd_l1_altitude +
cantilever.hemt.sd_l1_thick;
source20 = block3(temp.sd.x,temp.sd.y,cantilever.hemt.sd_l2_thick,...
        'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l2_altitude));

% ===== Layer 3 =====
% Altitude
cantilever.hemt.sd_l3_altitude = cantilever.hemt.sd_l2_altitude +
cantilever.hemt.sd_l2_thick;
source30 = block3(temp.sd.x,temp.sd.y,cantilever.hemt.sd_l3_thick,...
        'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l3_altitude));

% ===== Layer 4 =====
% Altitude
cantilever.hemt.sd_l4_altitude = cantilever.hemt.sd_l3_altitude +
cantilever.hemt.sd_l3_thick;
source40 = block3(temp.sd.x,temp.sd.y,cantilever.hemt.sd_l4_thick,...
        'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l4_altitude));

% ===== Layer 5 ===== TRACE
% Altitude
cantilever.hemt.sd_l5_altitude = cantilever.hemt.sd_l4_altitude +
cantilever.hemt.sd_l4_thick;
source50 = block3(temp.sd.x,temp.sd.y,cantilever.trace.l1_thick,...
        'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l5_altitude));

% ===== Layer 6 ===== TRACE
% Altitude
cantilever.hemt.sd_l6_altitude = cantilever.hemt.sd_l5_altitude +
cantilever.trace.l1_thick;
source60 = block3(temp.sd.x,temp.sd.y,cantilever.trace.l2_thick,...
        'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l6_altitude));

```

```

% =====
% === Gate Contact ===
% =====
% General computations
temp.y = cantilever.width/2 - cantilever.mesa.offset ...
        - cantilever.mesa.side/2 - cantilever.hemt.g_width/2;

% ===== Base Layers =====
gbase0 =
block3(cantilever.base.std,cantilever.hemt.g_width,cantilever.mesa.thick,...
        'pos',{-cantilever.base.std,temp.y,0});

gbase1 =
block3(cantilever.base.std,cantilever.hemt.g_width,cantilever.hemt.thick,...
        'pos',{-cantilever.base.std,temp.y,cantilever.mesa.thick});

% ===== Layer 1 =====
% Altitude
cantilever.hemt.g_l1_altitude = cantilever.hemt.sd_l1_altitude;
gate10 = block3(temp.g.x,temp.g.y,cantilever.hemt.g_l1_thick,...
        'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.g_l1_altitude});

% ===== Layer 2 =====
% Altitude
cantilever.hemt.g_l2_altitude = cantilever.hemt.g_l1_altitude +
cantilever.hemt.g_l1_thick;
gate20 = block3(temp.g.x,temp.g.y,cantilever.hemt.g_l2_thick,...
        'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.g_l2_altitude});

% ===== Layer 3 ===== TRACE
% Altitude
cantilever.hemt.g_l3_altitude = cantilever.hemt.g_l2_altitude +
cantilever.hemt.g_l2_thick;
gate30 = block3(temp.g.x,temp.g.y,cantilever.trace.l1_thick,...
        'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.g_l3_altitude});

% ===== Layer 4 ===== TRACE
% Altitude
cantilever.hemt.g_l4_altitude = cantilever.hemt.g_l3_altitude +
cantilever.trace.l1_thick;
gate40 = block3(temp.g.x,temp.g.y,cantilever.trace.l2_thick,...
        'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.g_l4_altitude});

% =====
% === Drain Contact ===
% =====
% General computations
temp.y = cantilever.width/2 - cantilever.mesa.offset ...
        - cantilever.mesa.side + cantilever.hemt.side_offset;

% ===== Base Layers =====
dbase0 =
block3(cantilever.base.std,cantilever.hemt.sd_width,cantilever.mesa.thick,...
        'pos',{-cantilever.base.std,temp.y,0});

dbase1 =
block3(cantilever.base.std,cantilever.hemt.sd_width,cantilever.hemt.thick,...
        'pos',{-cantilever.base.std,temp.y,cantilever.mesa.thick});

```

```

% ===== Layer 1 =====
drain10 = block3(temp.sd.x,temp.sd.y,cantilever.hemt.sd_l1_thick,...
                'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l1_altitude});

% ===== Layer 2 =====
drain20 = block3(temp.sd.x,temp.sd.y,cantilever.hemt.sd_l2_thick,...
                'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l2_altitude});

% ===== Layer 3 =====
drain30 = block3(temp.sd.x,temp.sd.y,cantilever.hemt.sd_l3_thick,...
                'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l3_altitude});

% ===== Layer 2 =====
drain40 = block3(temp.sd.x,temp.sd.y,cantilever.hemt.sd_l4_thick,...
                'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l4_altitude});

% ===== Layer 5 ===== TRACE
drain50 = block3(temp.sd.x,temp.sd.y,cantilever.trace.l1_thick,...
                'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l5_altitude});

% ===== Layer 6 ===== TRACE
drain60 = block3(temp.sd.x,temp.sd.y,cantilever.trace.l2_thick,...
                'pos',{-
cantilever.base.std,temp.y,cantilever.hemt.sd_l6_altitude});

% =====
% === Trace parts ===
% =====
% Trace, part 1
temp.X = cantilever.mesa.side + cantilever.trace.mesa_offset +
cantilever.base.std;
temp.y = -cantilever.width/2 + cantilever.trace.side_offset;
btrace10 = block3(temp.X,cantilever.trace.width1,cantilever.trace.l1_thick,...
                'pos',{-cantilever.base.std,temp.y,0});
btrace11 = block3(temp.X,cantilever.trace.width1,cantilever.trace.l2_thick,...
                'pos',{-
cantilever.base.std,temp.y,cantilever.trace.l1_thick});

% Trace, part 2
temp.Y = cantilever.width/2 + cantilever.trace.width3/2 ...
- cantilever.trace.side_offset;
temp.x = cantilever.mesa.side + cantilever.trace.mesa_offset;
btrace20 = block3(cantilever.trace.width2,temp.Y,cantilever.trace.l1_thick,...
                'pos',{temp.x,temp.y,0});
btrace21 = block3(cantilever.trace.width2,temp.Y,cantilever.trace.l2_thick,...
                'pos',{temp.x,temp.y,cantilever.trace.l1_thick});

% Trace, part 3
temp.X = cantilever.length - cantilever.bias.front_offset ...
- cantilever.bias.side - cantilever.trace.width2 ...
- cantilever.trace.mesa_offset - cantilever.mesa.side;
temp.x = cantilever.mesa.side + cantilever.trace.mesa_offset ...
+ cantilever.trace.width2;
btrace30 = block3(temp.X,cantilever.trace.width3,cantilever.trace.l1_thick,...

```



```

        'pos', {temp.x, -cantilever.trace.width3/2, 0});
btrace31 = block3(temp.X, cantilever.trace.width3, cantilever.trace.l2_thick, ...
        'pos', {temp.x, -
cantilever.trace.width3/2, cantilever.trace.l1_thick});

% =====
% ===   Bias Pad   ===
% =====
temp.XY = cantilever.bias.side;
temp.x = cantilever.length - cantilever.bias.front_offset - temp.XY;
bpadl0 = block3(temp.XY, temp.XY, cantilever.trace.l1_thick, ...
        'pos', {temp.x, -temp.XY/2, 0});
bpadl1 = block3(temp.XY, temp.XY, cantilever.trace.l2_thick, ...
        'pos', {temp.x, -temp.XY/2, cantilever.trace.l1_thick});

% =====
% ===   Mesh Dividers   =====
% =====
temp.std = cantilever.base.std;
% Boundary 1
% Cuts across all contacts at HEMT.
temp.wrkpln = [0 0 0; 0 1 0; 0 0 1];
temp.x = cantilever.width+10e-6;
temp.y = temp.std * 2;
temp.pos = [-temp.x/2, -temp.y/2];
bound1 = rect2(temp.x, temp.y, 'pos', temp.pos);
bound1 = embed(bound1, 'Wrkpln', temp.wrkpln);

% Boundary 2
% Cuts 'wings' off of base extrusions (+y side)
temp.y = cantilever.width/2;
temp.wrkpln = [0 -1 0; temp.y temp.y temp.y; 0 0 1];
bound2 = rect2(temp.std, temp.std, 'pos', [0, -temp.std]);
bound2 = embed(bound2, 'Wrkpln', temp.wrkpln);

% Boundary 3
% Cuts 'wings' off of base extrusions (-y side)
garr=geomarrayr(bound2, 0, -cantilever.width, 0, 1, 2, 1);
[bound2, bound3]=deal(garr{:});

% Boundarys 4 & 5
% Cuts the bias trace
cantilever.trace.thick = cantilever.trace.l1_thick + cantilever.trace.l2_thick;

bound4 = geomcoerce('curve', {curve2([50e-6, 56e-6], [-18e-6, -18e-6], [1, 1])});
bound5 = geomcoerce('curve', {curve2([56e-6, 50e-6], [-3e-6, -3e-6], [1, 1])});

temp.wrkpln = [0 1 0; 0 0 1; 0 0 0];
bound4 =
extrude(bound4, 'distance', [cantilever.trace.thick], 'wrkpln', temp.wrkpln);
bound5 =
extrude(bound5, 'distance', [cantilever.trace.thick], 'wrkpln', temp.wrkpln);

% Boundary 6
% Cuts off hanging part of base
bound6 = geomcoerce('curve', {curve2([-3.0E-6, 0], [-2.0E-6, -2.0E-6], [1, 1])});
bound6 = extrude(bound6, 'distance', [52e-6], 'wrkpln', [0 2.3E-5 0; 2.6E-5 2.6E-5
2.6E-5; 0 0 2.3E-5]);

% =====
% ===   Assemble   =====

```

```

% =====
% Faces
fem.draw.f.objs={bound1,bound2,bound3,bound4,bound5,bound6};
fem.draw.f.name={'Bound1','Bound2','Bound3','Bound4','Bound5','Bound6'};
fem.draw.f.tags={'bound1','bound2','bound3','bound4','bound5','bound6'};

% Subdomains
fem.draw.s.objs={canti,base,mesa,hemt,...
    sbase0,sbase1,...
    source10,source20,source30,source40,source50,source60,...
    gbase0,gbase1,...
    gate10,gate20,gate30,gate40,...
    dbase0,dbase1,...
    drain10,drain20,drain30,drain40,drain50,drain60,...
    btrace10,btrace11',btrace20,btrace21,btrace30,btrace31,...
    bpad10,bpad11};

fem.draw.s.name={'Cantilever','Base','Mesa','HEMT',...
    'sbase0','sbase1',...

    'source10','source20','source30','source40','source50','source60',...
    'gbase0','gbase1',...
    'gate10','gate20','gate30','gate40',...
    'dbase0','dbase1',...

    'drain10','drain20','drain30','drain40','drain50','drain60',...

    'BTrace10','BTrace11','BTrace20','BTrace21','BTrace30','BTrace31',...
    'Bias Pad10','Bias Pad11'};

fem.draw.s.tags={'canti','base','mesa','hemt',...
    'sbase0','sbase1',...

    'source10','source20','source30','source40','source50','source60',...
    'gbase0','gbase1',...
    'gate10','gate20','gate30','gate40',...
    'dbase0','dbase1',...

    'drain10','drain20','drain30','drain40','drain50','drain60',...

    'btrace10','btrace11','btrace20','btrace21','btrace30','btrace31',...
    'bpad10','bpad11'};

fem.geom=geomcsg(fem);

clear base canti hemt mesa temp garr cantilever
clear bound1 bound2 bound3 bound4 bound5 bound6
clear bpad10 bpad11
clear btrace10 btrace11 btrace20 btrace21 btrace30 btrace31
clear dbase0 dbase1
clear sbase0 sbase1
clear gbase0 gbase1
clear drain10 drain20 drain30 drain40 drain50 drain60
clear gbase1 gate10 gate20 gate30 gate40
clear source10 source20 source30 source40 source50 source60

% *** MESH *****
% *** MESH *****
% *** MESH *****
% *** MESH *****
% *** MESH *****

```



```

0.7428571428571429 0.7714285714285715 0.8 0.8285714285714286 0.8571428571428571
0.8857142857142857 0.9142857142857143 0.9428571428571428 0.9714285714285714
1],306,[0 0.3333333333333333 0.6666666666666666 1]], ...
    'hauto',1, ...
    'face',[151,176,200,228], ...
    'meshstart',fem.mesh);

% Create swept mesh
fem.mesh=meshsweep(fem, ...
    'elsweeplayers',{5,[0 1],6,[0 1],7,[0 0.15384615384615388
0.48717948717948717 1],8,[0 1],9,[0 1],10,[0 1],11,[0 1],12,[0 1],13,[0
1],14,[0 1],15,[0 0.15384615384615388 0.48717948717948717 1],16,[0 1],17,[0
1],18,[0 1],19,[0 1],20,[0 1],21,[0 0.15384615384615388 0.48717948717948717
1],22,[0 1],23,[0 1],24,[0 1],25,[0 1],26,[0 1],27,[0 1],28,[0 1],32,[0
1],33,[0 1],34,[0 0.15384615384615388 0.48717948717948717 1],35,[0 1],36,[0
1],37,[0 1],38,[0 1],39,[0 1],40,[0 1],41,[0 1],42,[0 1],43,[0 1],44,[0
1],45,[0 1],46,[0 1],47,[0 1],48,[0 1],49,[0 1],50,[0 1],51,[0 1],52,[0
1],53,[0 1],54,[0 1],55,[0 1],56,[0 1],57,[0 1],58,[0 1],59,[0 1],60,[0
1],61,[0 1]], ...
    'hauto',1, ...
    'subdomain',[7,8,15,16,21,22,34,35], ...
    'meshstart',fem.mesh);

% Initialize mesh
fem.mesh=meshinit(fem, ...
    'hauto',1, ...
    'methodfac',{[7,14,23,51,79,111,125,135],'triaf'}, ...
    'point',[], ...
    'edge',[], ...
    'face',[7,14,23,51,79,111,125,135], ...
    'subdomain',[], ...
    'meshstart',fem.mesh);

% Convert mesh elements
fem.mesh=meshconvert(fem, ...
    'face','all', ...
    'subdomain','all');

% Initialize mesh
fem.mesh=meshinit(fem, ...
    'hauto',1, ...
    'methodfac',{[7,14,23,51,79,111,125,135],'triaf'}, ...
    'point',[], ...
    'edge',[], ...
    'face',[], ...
    'subdomain',[1,2,3,4,29,30,31], ...
    'meshstart',fem.mesh);

% (Default values are not included)
fem.border = 1;

% Multiphysics
fem=multiphysics(fem);

% *** APPLICATION MODE *****
% *** APPLICATION MODE *****
% *** APPLICATION MODE *****
% *** APPLICATION MODE *****
% *** APPLICATION MODE *****
% *** APPLICATION MODE *****

```



```

fem=multiphysics(fem);

% Extend mesh
fem.xmesh=mesextend(fem);

% Split fem variables for seperate analysis
femStatic = fem

% Solve problem
fem.sol=femstatic(fem, ...
    'solcomp',{'w','v','u'}, ...
    'outcomp',{'w','v','u'}, ...
    'blocksize','auto');

% Save current fem structure for restart purposes
fem0=fem;

% *** PLOT *****
% *** PLOT *****
% *** PLOT *****
% *** PLOT *****
% *** PLOT *****
% *** PLOT *****

postplot(fem, ...
    'tridata',{'disp_smsld','cont','internal'}, ...
    'trimap','Rainbow', ...
    'deformbnd',{'u','v','w'}, ...
    'lindata',{'disp_smsld','cont','internal'}, ...
    'linstyle',[0.0,0.0,0.0], ...
    'deformedg',{'u','v','w'}, ...
    'title','Boundary: Total displacement   Edge: Total displacement
Deformation: Displacement', ...
    'geom','off', ...
    'grid','on', ...
    'campos',[-7.941669892451432E-4,-
0.0012567197537588285,9.112058037219916E-4], ...
    'camtarget',[1.7014799414027948E-4,0,-3.351999566802988E-6], ...
    'camup',[0,0,1], ...
    'camva',[7.619656243309206]);

break;
% *** EXTRACT DATA *****
% *** EXTRACT DATA *****
% *** EXTRACT DATA *****
% *** EXTRACT DATA *****
% *** EXTRACT DATA *****
% *** EXTRACT DATA *****

%Get output from a 3D FEM structure
% Give as input:
% fem (the solved FEM structure)
% x_cords = [x_start: x_increment: x_end]
% y_cords = [y_start: y_increment: y_end]
% z_cords = [z_start: z_increment: z_end]
% targetVar = value sought to be extracted from the FEM setucture
% This variable must be a string. (Example: 'ex_smpn')
% solNum = The solution used for the values (1 is default)

% Large 3D Matrix
x_cords = [0: 0.1e-6 : 35e-6];
y_cords = [-13e-6: 0.1e-6 : 22e-6];
z_cords = [166e-9: 1e-9: 200e-9];

```

```

% From Boston Comsol Conference 2010 paper:
% "Characterization of an AlGaIn/GaN Electrostatically Actuated Cantilever
% using Finite Element Method"
% Figure 3 (b)
% All lines use 0.2um increments

% Line A (On surface of gate)
x_cords = [0: 0.1e-6 : 35e-6]; y_cords = 4.5e-6; z_cords = 775e-9;

% Line B (On surface of source)
x_cords = [0: 0.1e-6 : 35e-6]; y_cords = 16e-6; z_cords = 595e-9;

% Line C (below A at interface)
x_cords = [0: 0.1e-6 : 35e-6]; y_cords = 4.5e-6; z_cords = 200e-9;

% Line D (tip of A to 50um on canti surface)
x_cords = [35e-6: 0.1e-6 : 50e-6]; y_cords = 4.5e-6; z_cords = 0;

break;
% Use to retrieve values from 'fem'
% -----
targetVar = 'ex_smsld';
solNum = 1;
[X Y Z] = meshgrid(x_cords,y_cords,z_cords);
p = [X(:)';Y(:)';Z(:)'];
[out] = postinterp(fem,targetVar,p,'Solnum',solNum);
outputMatrix = reshape(out,size(X));
clear out p X Y Z x_cords y_cords z_cords targetVar solNum;
break;
% -----
targetVar = 'ex_smsld';solNum = 1;[X Y Z] = meshgrid(x_cords,y_cords,z_cords);p
= [X(:)';Y(:)';Z(:)'];[out] =
postinterp(fem,targetVar,p,'Solnum',solNum);outputMatrix =
reshape(out,size(X));clear out p X Y Z x_cords y_cords z_cords targetVar
solNum;

% *****
% *****
% *****
% ***   END   *****
% *****
% *****
% *****
% COMSOL Multiphysics Model M-file
% Generated by COMSOL 3.5a (COMSOL 3.5.0.603, $Date: 2008/12/03 17:02:19 $)

% COMSOL version
clear vrsn
vrsn.name = 'COMSOL 3.5';
vrsn.ext = 'a';
vrsn.major = 0;
vrsn.build = 603;
vrsn.rcs = '$Name: $';
vrsn.date = '$Date: 2008/12/03 17:02:19 $';
fem.version = vrsn;

% (Default values are not included)

% Application mode 1
clear appl
appl.mode.class = 'SmeSolid3';

```



```

lib.mat{1}.variables.E='210e9[Pa]';
lib.mat{1}.variables.epsilonr='9.5';
lib.mat{1}.variables.alpha='13.4e-6[1/K]';
lib.mat{1}.variables.C='490[J/(kg*K)]';
lib.mat{1}.variables.rho='6150[kg/m^3]';
lib.mat{1}.variables.k='130[W/(m*K)]';
lib.mat{2}.name='Ti';
lib.mat{2}.varname='mat2';
lib.mat{2}.variables.nu='0.36';
lib.mat{2}.variables.E='40e9[Pa]';
lib.mat{2}.variables.sigma='2.6e6[S/m]';
lib.mat{2}.variables.alpha='8.60e-6[1/K]';
lib.mat{2}.variables.C='522[J/(kg*K)]';
lib.mat{2}.variables.rho='4506[kg/m^3]';
lib.mat{2}.variables.k='21.9[W/(m*K)]';
lib.mat{3}.name='Au';
lib.mat{3}.varname='mat3';
lib.mat{3}.variables.nu='0.44';
lib.mat{3}.variables.E='70e9[Pa]';
lib.mat{3}.variables.sigma='45.6e6[S/m]';
lib.mat{3}.variables.alpha='14.2e-6[1/K]';
lib.mat{3}.variables.C='129[J/(kg*K)]';
lib.mat{3}.variables.rho='19300[kg/m^3]';
lib.mat{3}.variables.k='317[W/(m*K)]';
lib.mat{4}.name='Al';
lib.mat{4}.varname='mat4';
lib.mat{4}.variables.nu='0.35';
lib.mat{4}.variables.E='70.0e9[Pa]';
lib.mat{4}.variables.sigma='35.5e6[S/m]';
lib.mat{4}.variables.alpha='23.1e-6[1/K]';
lib.mat{4}.variables.C='904[J/(kg*K)]';
lib.mat{4}.variables.rho='2700[kg/m^3]';
lib.mat{4}.variables.k='237[W/(m*K)]';
lib.mat{5}.name='Ni';
lib.mat{5}.varname='mat5';
lib.mat{5}.variables.nu='0.31';
lib.mat{5}.variables.E='219e9[Pa]';
lib.mat{5}.variables.sigma='13.8e6[S/m]';
lib.mat{5}.variables.alpha='13.4e-6[1/K]';
lib.mat{5}.variables.C='445[J/(kg*K)]';
lib.mat{5}.variables.rho='8900[kg/m^3]';
lib.mat{5}.variables.k='90.7[W/(m*K)]';
lib.mat{6}.name='AlGaIn';
lib.mat{6}.varname='mat6';
lib.mat{6}.variables.nu='0.2';
lib.mat{6}.variables.E='185e9[Pa]';
lib.mat{6}.variables.alpha='10e-6[1/K]';
lib.mat{6}.variables.rho='5740[kg/m^3]';

fem.lib = lib;

% Multiphysics
fem=multiphysics(fem);

% Extend mesh
fem.xmesh=meshextend(fem);

% Solve problem
fem.sol=femstatic(fem, ...
    'solcomp',{'w','v','u'}, ...
    'outcomp',{'w','v','u'}, ...
    'blocksize','auto', ...

```



```

lib.mat{3}.name='Au';
lib.mat{3}.varname='mat3';
lib.mat{3}.variables.nu='0.44';
lib.mat{3}.variables.E='70e9[Pa]';
lib.mat{3}.variables.sigma='45.6e6[S/m]';
lib.mat{3}.variables.alpha='14.2e-6[1/K]';
lib.mat{3}.variables.C='129[J/(kg*K)]';
lib.mat{3}.variables.rho='19300[kg/m^3]';
lib.mat{3}.variables.k='317[W/(m*K)]';
lib.mat{4}.name='Al';
lib.mat{4}.varname='mat4';
lib.mat{4}.variables.nu='0.35';
lib.mat{4}.variables.E='70.0e9[Pa]';
lib.mat{4}.variables.sigma='35.5e6[S/m]';
lib.mat{4}.variables.alpha='23.1e-6[1/K]';
lib.mat{4}.variables.C='904[J/(kg*K)]';
lib.mat{4}.variables.rho='2700[kg/m^3]';
lib.mat{4}.variables.k='237[W/(m*K)]';
lib.mat{5}.name='Ni';
lib.mat{5}.varname='mat5';
lib.mat{5}.variables.nu='0.31';
lib.mat{5}.variables.E='219e9[Pa]';
lib.mat{5}.variables.sigma='13.8e6[S/m]';
lib.mat{5}.variables.alpha='13.4e-6[1/K]';
lib.mat{5}.variables.C='445[J/(kg*K)]';
lib.mat{5}.variables.rho='8900[kg/m^3]';
lib.mat{5}.variables.k='90.7[W/(m*K)]';
lib.mat{6}.name='AlGaIn';
lib.mat{6}.varname='mat6';
lib.mat{6}.variables.nu='0.2';
lib.mat{6}.variables.E='185e9[Pa]';
lib.mat{6}.variables.alpha='10e-6[1/K]';
lib.mat{6}.variables.rho='5740[kg/m^3]';

fem.lib = lib;

% Multiphysics
fem=multiphysics(fem);

% Extend mesh
fem.xmesh=mesnextend(fem);

% Solve problem
fem.sol=femstatic(fem, ...
    'solcomp',{'w','v','u'}, ...
    'outcomp',{'w','v','u'}, ...
    'blocksize','auto', ...
    'pname','freq_smsld', ...
    'plist',[ colon(15400,10,15650)], ...
    'oldcomp',{}, ...
    'ntol',1e-006, ...
    'linsolver','spooles');

% Save current fem structure for restart purposes
fem0=fem;

% Plot solution
postplot(fem, ...
    'tridata',{'disp_smsld','cont','on'}, ...
    'trimap','Rainbow', ...
    'deformbnd',{'u','v','w'}, ...
    'lindata',{'disp_smsld','cont','on'}, ...

```

```

        'linestyle',[0.0,0.0,0.0], ...
        'deformedg',{'u','v','w'}, ...
        'solnum','end', ...
        'title','Boundary: Total displacement   Edge: Total displacement
Deformation: Displacement', ...
        'geom','off', ...
        'grid','on', ...
        'campos',[-7.681720030885453E-4,-
0.0012272492383750423,8.856857753091889E-4], ...
        'camtarget',[1.7352945849324897E-4,0,-7.425329357602095E-6], ...
        'camup',[0,0,1], ...
        'camva',7.3280972481526545);

% Plot in cross-section or along domain
postcrossplot(fem,0,[234], ...
    'pointdata','disp_smsld', ...
    'title','Total displacement', ...
    'axislabel',{'freq_smsld','Total displacement'}, ...
    'refine','auto');

% Plot in cross-section or along domain
postcrossplot(fem,0,[234], ...
    'pointdata','w', ...
    'title','z-displacement', ...
    'axislabel',{'freq_smsld','z-displacement'}, ...
    'refine','auto');

% Plot in cross-section or along domain
postcrossplot(fem,0,[234], ...
    'pointdata','disp_smsld', ...
    'title','Total displacement', ...
    'axislabel',{'freq_smsld','Total displacement'}, ...
    'refine','auto');

% Plot in cross-section or along domain
postcrossplot(fem,0,[234], ...
    'pointdata','v', ...
    'title','y-displacement', ...
    'axislabel',{'freq_smsld','y-displacement'}, ...
    'refine','auto');

% Plot in cross-section or along domain
postcrossplot(fem,0,[234], ...
    'pointdata','u', ...
    'title','x-displacement', ...
    'axislabel',{'freq_smsld','x-displacement'}, ...
    'refine','auto');

% Plot in cross-section or along domain
postcrossplot(fem,0,[234], ...
    'pointdata','w', ...
    'title','z-displacement', ...
    'axislabel',{'freq_smsld','z-displacement'}, ...
    'refine','auto');

% Plot in cross-section or along domain
postcrossplot(fem,0,[234], ...
    'pointdata','disp_smsld', ...
    'title','Total displacement', ...
    'axislabel',{'freq_smsld','Total displacement'}, ...
    'refine','auto');

% (Default values are not included)

```



```

clear lib
lib.mat{1}.name='GaN';
lib.mat{1}.varname='mat1';
lib.mat{1}.variables.nu='0.2';
lib.mat{1}.variables.E='210e9[Pa]';
lib.mat{1}.variables.epsilonr='9.5';
lib.mat{1}.variables.alpha='13.4e-6[1/K]';
lib.mat{1}.variables.C='490[J/(kg*K)]';
lib.mat{1}.variables.rho='6150[kg/m^3]';
lib.mat{1}.variables.k='130[W/(m*K)]';
lib.mat{2}.name='Ti';
lib.mat{2}.varname='mat2';
lib.mat{2}.variables.nu='0.36';
lib.mat{2}.variables.E='40e9[Pa]';
lib.mat{2}.variables.sigma='2.6e6[S/m]';
lib.mat{2}.variables.alpha='8.60e-6[1/K]';
lib.mat{2}.variables.C='522[J/(kg*K)]';
lib.mat{2}.variables.rho='4506[kg/m^3]';
lib.mat{2}.variables.k='21.9[W/(m*K)]';
lib.mat{3}.name='Au';
lib.mat{3}.varname='mat3';
lib.mat{3}.variables.nu='0.44';
lib.mat{3}.variables.E='70e9[Pa]';
lib.mat{3}.variables.sigma='45.6e6[S/m]';
lib.mat{3}.variables.alpha='14.2e-6[1/K]';
lib.mat{3}.variables.C='129[J/(kg*K)]';
lib.mat{3}.variables.rho='19300[kg/m^3]';
lib.mat{3}.variables.k='317[W/(m*K)]';
lib.mat{4}.name='Al';
lib.mat{4}.varname='mat4';
lib.mat{4}.variables.nu='0.35';
lib.mat{4}.variables.E='70.0e9[Pa]';
lib.mat{4}.variables.sigma='35.5e6[S/m]';
lib.mat{4}.variables.alpha='23.1e-6[1/K]';
lib.mat{4}.variables.C='904[J/(kg*K)]';
lib.mat{4}.variables.rho='2700[kg/m^3]';
lib.mat{4}.variables.k='237[W/(m*K)]';
lib.mat{5}.name='Ni';
lib.mat{5}.varname='mat5';
lib.mat{5}.variables.nu='0.31';
lib.mat{5}.variables.E='219e9[Pa]';
lib.mat{5}.variables.sigma='13.8e6[S/m]';
lib.mat{5}.variables.alpha='13.4e-6[1/K]';
lib.mat{5}.variables.C='445[J/(kg*K)]';
lib.mat{5}.variables.rho='8900[kg/m^3]';
lib.mat{5}.variables.k='90.7[W/(m*K)]';
lib.mat{6}.name='AlGaN';
lib.mat{6}.varname='mat6';
lib.mat{6}.variables.nu='0.2';
lib.mat{6}.variables.E='185e9[Pa]';
lib.mat{6}.variables.alpha='10e-6[1/K]';
lib.mat{6}.variables.rho='5740[kg/m^3]';

```

```
fem.lib = lib;
```

```
% Multiphysics
fem=multiphysics(fem);
```

```
% Extend mesh
fem.xmesh=meshextend(fem);
```

```
% Solve problem
```



```

fem.sol=femstatic(fem, ...
    'solcomp',{'w','v','u'}, ...
    'outcomp',{'w','v','u'}, ...
    'blocksize','auto', ...
    'pname','freq_smsld', ...
    'plist',[ colon(15400,10,15650)], ...
    'oldcomp',{'}, ...
    'ntol',1e-006);

% Save current fem structure for restart purposes
fem0=fem;

% Plot solution
postplot(fem, ...
    'tridata',{'disp_smsld','cont','on'}, ...
    'trimap','Rainbow', ...
    'deformbnd',{'u','v','w'}, ...
    'lindata',{'disp_smsld','cont','on'}, ...
    'linstyle',[0.0,0.0,0.0], ...
    'deformedg',{'u','v','w'}, ...
    'solnum','end', ...
    'title','Boundary: Total displacement   Edge: Total displacement
Deformation: Displacement', ...
    'geom','off', ...
    'grid','on', ...
    'campos',[-7.681720030885453E-4,-
0.0012272492383750423,8.856857753091889E-4], ...
    'camtarget',[1.7352945849324897E-4,0,-7.425329357602095E-6], ...
    'camup',[0,0,1], ...
    'camva',[7.3280972481526545]);

% Plot in cross-section or along domain
postcrossplot(fem,0,[234], ...
    'pointdata','disp_smsld', ...
    'title','Total displacement', ...
    'axislabel',{'freq_smsld','Total displacement'}, ...
    'refine','auto');

% Plot in cross-section or along domain
postcrossplot(fem,0,[234], ...
    'pointdata','disp_smsld', ...
    'title','Total displacement', ...
    'axislabel',{'freq_smsld','Total displacement'}, ...
    'refine','auto');

```