Session #

Undergraduate Research Project: Computer Modeling of Nanotube Embedded Chemicapacitive Sensors

Tal Rusak

School of Electrical and Computer Engineering and Department of Computer Science
Cornell University, Ithaca, New York 14853, USA

Abstract

Carbon nanotubes (CNTs) are rolled sheets of carbon whose atoms are arranged in a hexagonal pattern. CNTs measure about a millionth of a millimeter in diameter and show great promise for nanotechnology applications. Use of CNTs in nanoelectronics can lead to nanoscale sensitive devices, such as sensors that can detect environmentally toxic microscopic agents. To investigate the electrical properties of such sensors, CNT-embedded metal-oxide-semiconductor structures are analyzed using numerical modeling. The Poisson equation which describes the electric field in the sensor is solved numerically to obtain the electrostatic potential and capacitance of sensor devices. The effects of CNTs on device performance are examined by graphing results in the one and two dimensional cases.

Motivation

Since the earliest days of recorded history, the deliberate contamination of the environment has been an unfortunate but common tactic in warfare[1]. For the past two and a half millennia and perhaps even earlier, mechanisms such as “the catapulting of plague victims to the deliberate use of infected clothes, insect vectors, and specialized weapon systems”[2] have been used to carry out such warfare and to cause an innumerable number of deaths. Environmental contagion is thus one of the longest-surviving, continuously-used techniques in offensive warfare. Today, a range of microscopic chemical agents have been developed and are being held by a number of countries and organizations[3]. These include the famed nerve agent sarin and the biological toxin ricin.[4]

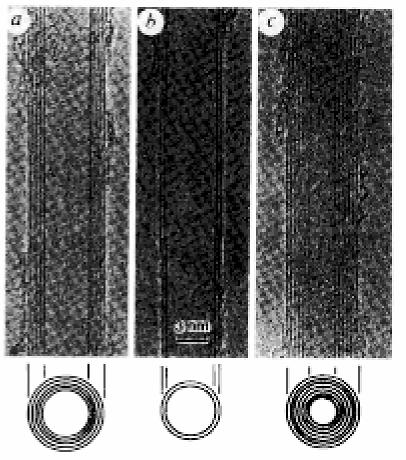


Figure 2-1 (above)
Electron micrograph of the first carbon nanotubes produced by Sumio Iijima. The scale gives an idea of the tubes' extremely small size, and there are even smaller ones available.[6]

The popularity and longevity of such methods are a direct result of the sheer difficulty of detecting invisible, toxic substances in the air, water, and food supply. The advent of modern packaging and distribution methods for toxins, as illustrated by the deadly anthrax attacks on the United States in 2002[5], has made the detection of contaminants even more elusive. The issues compound when soldiers are operating in a foreign, unfamiliar and possibly hostile territory and/or in a crowded urban environment.

Recent advances in nanoengineering[*] pose a potential solution to the problem of detecting toxins. The proposed sensor makes use of nanoscale components and will have a high sensitivity to chemical and biochemical substances that potentially cause disease. It is being designed to report the presence of such hazards even at extremely low quantities and with a very fast response rate. In addition, the device will be very compact and it will be easy to carry around on any mission, thanks to its tiny components. Thus, such a device may serve as the first line of defense against environmental toxins for soldiers and civilians in any situation imaginable.

The purpose of this undergraduate research project is to elucidate the properties of such a device proposed by the United States Army Research Laboratory, as will be described in the following pages. The project was conducted as a part of the University of Maryland Engineering Research Internship Teams (MERIT). I would like to express my sincerest thanks to Prof. Neil Goldsman and Dr. Akin Akturk for guiding and supporting me throughout this project. I am also indebted to the United States Army Research Laboratory for the funding and the idea which made this research possible. The interest and understanding that this project instilled in me have enabled and encouraged me to continue doing research in this area at my home institution, Cornell University.

1 Introduction

We propose to build a fast, sensitive sensor based on a **metal-oxide-semiconductor (MOS)** structure with an embedded **carbon nanotube (CNT)**. A description of these concepts follows this introduction. Subsequently, we develop numerical modeling to study the electrostatic properties of the device. Among the factors varied in the simulation are the diameter and potential of the sensing plate (such as a nanotube) and the dimensions of each component of the structure. The Poisson equation is solved numerically for the electrostatic potential of each MOS device and the capacitance is calculated using the electric field. Finally, we analyze and compare the results to find trends in capacitance. In addition, we validate our solutions by several methods and compare our results to other capacitive sensor designs.

[*] **Nanoengineering** is the study of the construction of devices measuring on the order of 10^{-9} meter or 1 nanometer.

2 Overview of Key Concepts

2.1 Carbon Nanotube (CNT)

The carbon nanotube was discovered by Sumio Iijima of NEC in 1991[6] (see figure 2-1). A nanotube, pictured in figure 2-2, is a rolled sheet of graphene, which is in turn comprised of hexagonal rings of carbon atoms bonded in sp^2 hybridization[6]. The discovery ushered a wave of advanced research in the field of nanotube-based electronics. We propose one such application in this project.

It has been observed that nanotubes are superior in a large number of physical properties when compared to today's leading materials.[7] First, the size of nanotubes gives a distinct advantage in constructing nanoscale electronic devices because the diameter of nanotubes has a lower limit of 0.6 to 1.8 nanometers in diameter, compared to devices produced by lithography that are usually about an order of magnitude larger.[7] Nanotubes are able to carry 1 billion amperes per centimeter squared, a thousand times more current than copper wire.[7] This research investigates advantages for using a nanotube in a MOS chemicapacitor and examines the capacitance of various such devices when compared to more traditional sensors.

2.2 Metal-Oxide-Semiconductor (MOS) Capacitor

The metal-oxide-semiconductor (MOS) capacitor is common in many semiconductor applications. A MOS capacitor can be made by shorting the drain and source terminals of a MOSFET (metal-oxide-semiconductor field affect transistor).[8] As the name suggests, a MOS capacitor has three layers: a metal, an oxide, and a semiconductor. However, variants of this structure are often used in modern electronics. In fact, the metal is usually replaced by a highly conductive polycrystalline silicon.[8] In this project we proposed to replace a traditional, "rectangular" semiconductor plate with a nanotube. The general structure of a MOS capacitor is illustrated in figure 2-3. For the MOS capacitor that we will study in this project, please see figure 3-1.

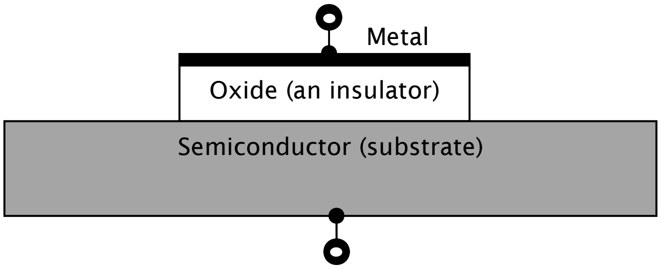


Figure 2-2 (above) Schematic of a typical MOS capacitor.[8]

2.3 Chemicapacitive Sensing and Scope of this Research

The purpose of a chemical sensor is to detect the presence of a certain substance in the environment. This can be done readily with a capacitor since the capacitance changes as the charge on the plates of the sensor is altered, which happens when a foreign particle attaches. In order to selectively sense particles, we use a polymer sensitizer to limit entry into the capacitor to certain substances. We will not focus on specific substances or on the polymer sensitizer in this project. Instead, we will have a generic "particle" that acts as the absorbate to the sensor.

Assume that we have the following device, a "traditional" microelectronic capacitor. The diagrams below represent a parallel-plate chemicapacitive sensor,[9] a model that has been discussed in the published literature since 1983.[10] This design has had applications as a sensor for the vapor phase of water (humidity),[11] for detecting organic compounds,[9] and for detecting ammonia in an industrial setting.[12] In the diagram to the right, a particle that we are trying to detect has entered the sensor.

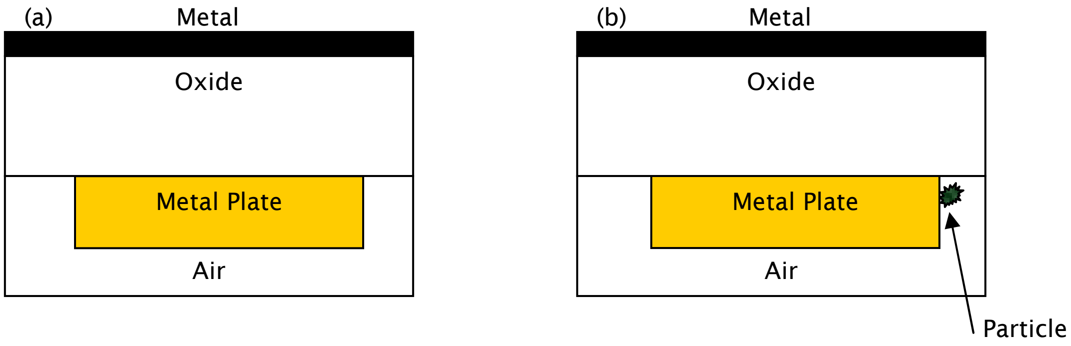


Figure 2-3 (above) Cross section of a traditional MOS capacitor that will act as a sensor. In (a) there is no particle and in (b) there is a particle attached to the metal plate.

The particle will cause a change in the capacitance of the device. A more sensitive sensor will have a greater *percent change* in capacitance when the particle enters so that it will be more easily detected.

In this project, we investigate whether embedding a carbon nanotube into the air of the MOS using the single nanotube planar capacitor geometry,[13] shown in figure 2-5, makes a better capacitive sensor and if it induces a higher sensitivity (as measured by percent change) in capacitance when the particle enters.

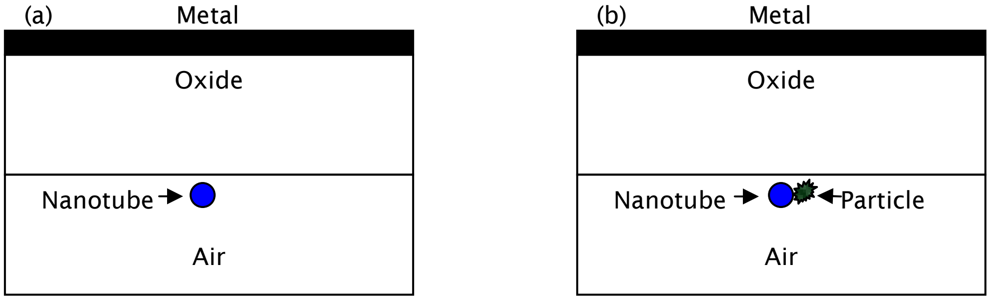


Figure 2-4 (above) Cross sections of (a) a MOS Sensor with embedded nanotube and (b) the same device with a particle attached to the nanotube.

3 Background to the Numerical Model

3.1 Notations and Conventions

We have already seen several figures of different MOS structures. For the purpose of analyzing the sensor and creating the associated numerical models, we use the geometry presented below in figure 3-1. Notice that the metal plate at the top of the oxide represents the entire top boundary, followed by a rectangular region representing the oxide and then a rectangular region representing air. Just below the *interface* between the oxide and air, a plate or nanotube of varying size and shape may be added to serve as a second capacitor plate. The *x*, *y*, and *z* axes illustrated here will be used to develop all of the simulations. The arrows represent the direction of increasing value for each variable. The *z*-axis is illustrated below by an **X**, which means that this axis is perpendicular to the page, with values increasing in the direction down *into* the page. Please also note the example values of ε given on this diagram. These will be used for the models developed in this project.

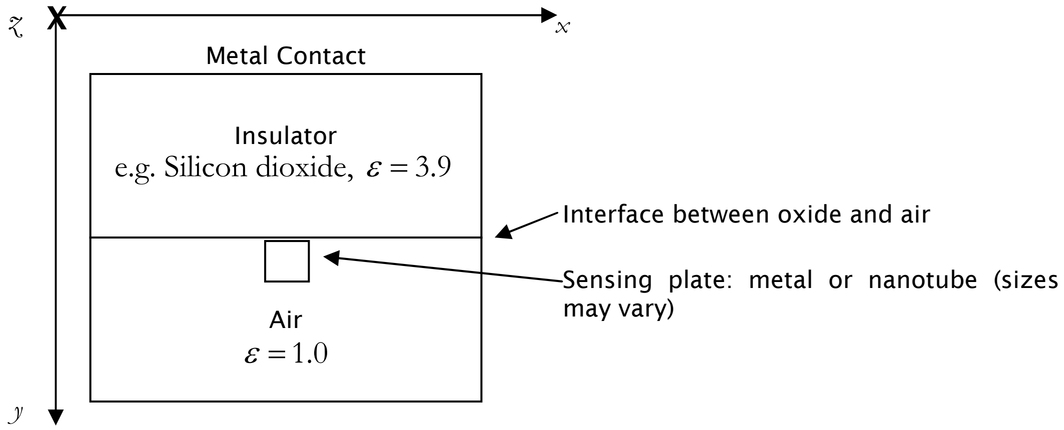

z
x
Metal Contact
Insulator
e.g. Silicon dioxide, $\varepsilon = 3.9$
Interface between oxide and air
Sensing plate: metal or nanotube (sizes may vary)
Air
$\varepsilon = 1.0$
y

Figure 3-1 (above) This structure will be investigated in this research.

3.2 Electrostatic Modeling

We solve the Poisson equation to obtain the electrostatic potential of our chemicapacitors. We then calculate measurable quantities such as capacitance and electric field that characterize the performance of our sensors. The **Poisson equation** is a partial differential equation that can be used to relate potential to net charge density,[14] as written below:

$$\nabla \cdot \left(\varepsilon \nabla \phi \right) = -q\rho \tag{3.1}$$

Above, ε is the space dependent dielectric constant, ϕ is the electrostatic potential, ρ is the net charge density, and *q* is the elementary charge, which is equal to 1.6×10^{-19} C.

We take ρ as zero within the sensor device. We do not have trapped, surface, or free charges in our sensor device except at the gate terminal and the sensing plate, shown in figure 3-1.

In other words, using the reciprocity theorem, charge is not dealt with directly, but indirectly by setting the potential at certain locations. Therefore, equation (3.1) simplifies to

$$\nabla \cdot \left(\varepsilon \nabla \phi \right) = 0 \qquad \textbf{(3.2)}$$

In this research, we will introduce both Dirichlet and Neumann boundary conditions for the numerical solution of the Poisson equation. In a **Dirichlet (fixed)** boundary, potential is set to a constant, i.e., $\phi = c$. In a **Neumann (floating)** boundary, the *derivative* of the potential is constant in a direction perpendicular to the boundary, e.g., if the bottom *x* boundary of the MOS is floating, then its boundary condition will be $\frac{\partial \phi}{\partial y} = 0$, since the *y*-axis is perpendicular to the *x*-axis.

The Poisson equation can be written in one, two, or three dimensions, as follows:

$$\nabla \cdot \left(\varepsilon \left(y \right) \nabla \phi \left(y \right) \right) = 0 \quad \textbf{(one-dimensional Poisson Equation)} \qquad \textbf{(3.3)}$$

$$\nabla \cdot \left(\varepsilon \begin{pmatrix} x \\ y \end{pmatrix} \nabla \phi \begin{pmatrix} x \\ y \end{pmatrix} \right) = 0 \quad \textbf{(two-dimensional Poisson Equation)} \qquad \textbf{(3.4)}$$

$$\nabla \cdot \left(\varepsilon \begin{pmatrix} x \\ y \\ z \end{pmatrix} \nabla \phi \begin{pmatrix} x \\ y \\ z \end{pmatrix} \right) = 0 \quad \textbf{(three-dimensional Poisson Equation)} \qquad \textbf{(3.5)}$$

Each of these equations requires specific boundary conditions in order to have a finite solution.

Considering figure 3-1, the one dimensional case may be represented by constant *x* and *z* positions with a varying *y* position. The two dimensional case is represented by a constant *z* position and varying *x* and *y* positions. In three dimensions, the *x*, *y*, and *z* positions vary.

In this project, we aim to solve the Poisson equation for ϕ, the potential at an arbitrary point on the MOS capacitor, which can in turn be used to solve for the capacitance of the device and for its electric field.

4 Numerical Modeling

4.1 Meshes Used for Numerical Approximation

Using a numerical model means that we will only be able to solve ϕ for a certain number of points on the MOS structure. This rectangular grid is known as a **mesh** and each of its points is known as a **meshpoint**. The number of meshpoints is represented in this work by *n* subscripted by the appropriate direction. Meshes can be created in one, two, or three dimensions. The distance between each point and its neighbor is called the **spacing**. In some cases, it is the same in all directions and between all points. This is called a **square mesh**, and the spacing may be represented by Δ since it is always constant. In a **uniform mesh**, the separation distance is constant in each direction, but is not necessarily the same in all directions; thus, three separate variables must be used: Δ_x, Δ_y, and Δ_z for the *x*, *y*, and *z* directions, respectively. In a **nonuniform mesh**, the separation distance between the meshpoints is determined by a function of the location along the current axis and can be represented by the functions $\Delta_x(x)$, $\Delta_y(y)$, and $\Delta_z(z)$ for the *x*, *y*, and *z* directions, respectively. These functions will be described in more detail in section 4.3.1. In this work the square and nonuniform meshes are used predominantly. For examples of meshes, see the solutions for potential in section 6.1.

4.2 Numerical Model for Uniform and Square Meshes

4.2.1 General Solution

In order to solve the numerical approximation for uniform and square meshes, we have chosen to use the **central difference approximation**. For each dimension (*x*, *y*, and/or *z*), we must simplify the Poisson equation (3.2) into a first derivative in the form $\frac{\partial \xi(r)}{\partial r}$, where ξ is some function, *r* is the direction *x*, *y*, or *z*, and Δ_r is the value representing the separation distance in the *r* direction.

We use the Taylor approximation[15] to solve for $\xi(r+\Delta_r)$ and $\xi(r-\Delta_r)$:

$$\xi(r+\Delta_r)=\xi(r)+\frac{\partial \xi(r)}{\partial r}\Delta_r+\frac{1}{2!}\frac{\partial^2 \xi(r)}{\partial r^2}\Delta_r^2+\frac{1}{3!}\frac{\partial^3 \xi(r)}{\partial r^3}\Delta_r^3+O\left(\Delta_r^4\right) \tag{4.1}$$

and

$$\xi(r-\Delta_r)=\xi(r)-\frac{\partial \xi(r)}{\partial r}\Delta_r+\frac{1}{2!}\frac{\partial^2 \xi(r)}{\partial r^2}\Delta_r^2-\frac{1}{3!}\frac{\partial^3 \xi(r)}{\partial r^3}\Delta_r^3+O\left(\Delta_r^4\right) \tag{4.2}$$

We now subtract equation (4.2) from equation (4.1), solve for $\frac{\partial \xi(r)}{\partial r}$, and neglect the error term *O*:

$$\frac{\partial \xi(r)}{\partial r}=\frac{\xi(r+\Delta_r)-\xi(r-\Delta_r)}{2\Delta_r} \tag{4.3}$$

To allow for the storage of fewer meshpoints, we divide each of the quantities in the right hand side of the equation by 2:

$$\frac{\partial \xi(r)}{\partial r} = \frac{\xi\left(r + \frac{\Delta_r}{2}\right) - \xi\left(r - \frac{\Delta_r}{2}\right)}{\Delta_r}$$

(First derivative approximation formula for uniform meshes) (4.4)

4.2.2 One-Dimensional Square/Uniform Mesh

To be consistent with figure 3-1, we will choose our variable dimension to be *y*. Note that the square and uniform meshes are the same here since there is only one dimension. First, we rewrite the one-dimensional Poisson equation (3.3):

$$\nabla \cdot \left(\varepsilon(y) \nabla \phi(y)\right) = 0 \tag{4.5}$$

Evaluating the inner gradient (∇) operator, we derive that

$$\nabla \cdot \left(\left(\varepsilon(y) \left(\frac{\partial \phi(y)}{\partial y} \right) \right) \mathbf{j} \right) = 0 \tag{4.6}$$

Now we simplify the outer divergence, which is a dot product, resulting in a scalar equation:

$$\frac{\partial \left(\varepsilon(y) \left(\frac{\partial \phi(y)}{\partial y} \right) \right)}{\partial y} = 0 \tag{4.7}$$

We apply equation (4.4) on the inner derivative, setting $r = y$, $\xi = \phi$, and $\Delta_r = \Delta$ (square mesh):

$$\frac{\partial \left(\varepsilon(y) \left(\frac{\phi(y + \Delta/2) - \phi(y + \Delta/2)}{\Delta} \right) \right)}{\partial y} = 0 \tag{4.8}$$

Expanding,

$$\frac{\partial \left(\frac{\varepsilon(y)\phi(y + \Delta/2) - \varepsilon(y)\phi(y - \Delta/2)}{\Delta} \right)}{\partial y} = 0 \tag{4.9}$$

Now we apply equation (4.4) to the outer derivative, setting $r = y$, $\xi = \frac{\varepsilon(y)\phi(y + \Delta/2) - \varepsilon(y)\phi(y - \Delta/2)}{\Delta} = 0$, and $\Delta_r = \Delta$:

$$\frac{\left(\frac{\varepsilon(y+\Delta/2)\phi(y+\Delta/2+\Delta/2)-\varepsilon(y+\Delta/2)\phi(y-\Delta/2+\Delta/2)}{\Delta}-\frac{\varepsilon(y-\Delta/2)\phi(y+\Delta/2-\Delta/2)-\varepsilon(y-\Delta/2)\phi(y-\Delta/2-\Delta/2)}{\Delta}\right)}{\Delta}=0 \qquad (4.10)$$

which readily simplifies to

$$\varepsilon(y-\Delta/2)\phi(y-\Delta)-\big(\varepsilon(y+\Delta/2)+\varepsilon(y-\Delta/2)\big)\phi(y)+\varepsilon(y+\Delta/2)\phi(y+\Delta)=0$$

(one dimensional square mesh numerical model) (4.11)

Notice that since we have a square mesh, all of the Δ expressions in the denominator have cancelled.

This form is especially convenient because it can readily be written in matrix format in the discrete mesh of n_y meshpoints and solved using linear algebra methods (discussed in more depth in section 5). Note that in some cases 0 terms may be replaced by numbers, such as when implementing a nonuniform mesh.

$$\begin{array}{c} y_{pos}=0 \\ \\ \\ \vdots \\ \\ \\ y_{pos}=\Delta n_y \end{array}
\underbrace{\begin{bmatrix}
\text{Boundary condition} & 0 & 0 & 0 & 0 & 0 & 0 \\
\ddots & \ddots & \ddots & 0 & 0 & 0 & 0 \\
0 & \ddots & \ddots & \ddots & 0 & 0 & 0 \\
0 & 0 & \varepsilon(y-\Delta/2) & -\begin{pmatrix}\varepsilon(y+\Delta/2)\\+\varepsilon(y-\Delta/2)\end{pmatrix} & \varepsilon(y+\Delta/2) & 0 & 0 \\
0 & 0 & 0 & \ddots & \ddots & \ddots & 0 \\
0 & 0 & 0 & 0 & \ddots & \ddots & \ddots \\
0 & 0 & 0 & 0 & 0 & 0 & \text{Boundary condition}
\end{bmatrix}}_{\text{Coefficients of }\phi\text{: }n_y\text{ rows by }n_y\text{ columns}}
\underbrace{\begin{bmatrix} \phi(0) \\ \vdots \\ \phi(y-\Delta) \\ \phi(y) \\ \phi(y+\Delta) \\ \vdots \\ \phi(\Delta n_y) \end{bmatrix}}_{\phi\text{: }n_y\text{ elements}}
=
\underbrace{\begin{bmatrix} \text{Boundary condition} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \text{Boundary condition} \end{bmatrix}}_{n_y\text{ elements}}$$

(one dimensional matrix equation) (4.12)

4.2.3 Two-Dimensional Square Mesh

Extending the one-dimensional solution derived above to two dimensions is straightforward. Recall equation (3.4), the two dimensional version of the Poisson equation:

$$\nabla\cdot\left(\varepsilon\begin{pmatrix}x\\y\end{pmatrix}\nabla\phi\begin{pmatrix}x\\y\end{pmatrix}\right)=0 \qquad (4.13)$$

Since the mesh we are working with is still a square mesh, the spacing is represented by Δ. By applying procedures similar to those described in section 4.2.2 on both the *x* and *y* directions, we derive a similar discretized equation along the *x* and *y* axes. Thus our numerical solution can be expressed as a sum of the solutions in the *x* and *y* directions:

$$\varepsilon\begin{pmatrix} x \\ y-\Delta/2 \end{pmatrix}\phi\begin{pmatrix} x \\ y-\Delta \end{pmatrix}+\varepsilon\begin{pmatrix} x-\Delta/2 \\ y \end{pmatrix}\phi\begin{pmatrix} x-\Delta \\ y \end{pmatrix}-\left(\varepsilon\begin{pmatrix} x \\ y+\Delta/2 \end{pmatrix}+\varepsilon\begin{pmatrix} x+\Delta/2 \\ y \end{pmatrix}+\varepsilon\begin{pmatrix} x \\ y-\Delta/2 \end{pmatrix}+\varepsilon\begin{pmatrix} x-\Delta/2 \\ y \end{pmatrix}\right)\phi\begin{pmatrix} x \\ y \end{pmatrix}$$

$$+\,\varepsilon\begin{pmatrix} x+\Delta/2 \\ y \end{pmatrix}\phi\begin{pmatrix} x+\Delta \\ y \end{pmatrix}+\varepsilon\begin{pmatrix} x \\ y+\Delta/2 \end{pmatrix}\phi\begin{pmatrix} x \\ y+\Delta \end{pmatrix}=0$$

(two dimensional square mesh numerical solution) (4.14)

This equation is similarly convenient to express as a matrix. We have n_y meshpoints in the *y* direction and n_x meshpoints in the *x* direction. Using the form below, the Poisson equation can be solved using linear algebra. Again, some of the 0 terms may be replaced by numbers in some solutions. See the Appendix (page 20) for this matrix.

4.3 Numerical Solutions for Nonuniform Meshes

4.3.1 Mesh Definition

To choose the separation distance for a nonuniform mesh, we choose two values for each direction: *p* and Δ. Then the separation distance between each of the meshpoints is determined by the following series:

$$\left[\cdots \quad \Delta p^3 \quad \Delta p^2 \quad \Delta p \quad \Delta \quad \Delta \quad \Delta p \quad \Delta p^2 \quad \Delta p^3 \quad \cdots\right] \qquad (4.15)$$

In most cases, two additional meshpoints that do not conform to this series must be added at both ends of the mesh to complete the structure's desired size (see section 6.1 for more details about device sizes).

4.3.2 Numerical Modeling

4.3.2.1 Model for the points inside the air and oxide

For each dimension (*x*, *y*, and/or *z*), we discretize the second derivative in the Poisson equation (3.2). The second derivative is $\frac{\partial^2 \xi(r)}{\partial r^2}$, where ξ is some function, *r* is the direction *x*, *y*, or *z*. During discretization, we represent the nonuniform spacing in the *r* direction by $\Delta_r(r)$. First we choose two values of $\Delta_r(r)$, to be referred to as Δ_{r1} (forward spacing) and Δ_{r2} (backward spacing). Again we use the Taylor approximation and multiply the forward and backward expansions by the backward and forward spacings, respectively:

$$\Delta_{r2}\xi(r+\Delta_{r1})=\xi(r)\Delta_{r2}+\frac{\partial \xi(r)}{\partial r}\Delta_{r1}\Delta_{r2}+\frac{1}{2!}\frac{\partial^2 \xi(r)}{\partial r^2}\Delta_{r1}^2\Delta_{r2}+O\left(\Delta_{r1}^3\Delta_{r2}\right) \qquad (4.16)$$

$$\Delta_{r1}\xi\left(r-\Delta_{r2}\right)=\xi\left(r\right)\Delta_{r1}-\frac{\partial\xi\left(r\right)}{\partial r}\Delta_{r2}\Delta_{r1}+\frac{1}{2!}\frac{\partial^{2}\xi\left(r\right)}{\partial r^{2}}\Delta_{r2}^{2}\Delta_{r1}+O\left(\Delta_{r2}^{3}\Delta_{r1}\right) \tag{4.17}$$

Adding these two formulas and ignoring higher order terms,

$$\Delta_{r2}\xi\left(r+\Delta_{r1}\right)+\Delta_{r1}\xi\left(r-\Delta_{r2}\right)=\Delta_{r2}\xi\left(r\right)+\Delta_{r1}\xi\left(r\right)+\frac{\partial^{2}\xi\left(r\right)}{\partial r^{2}}\Delta_{r1}\Delta_{r2}\left(\Delta_{r1}+\Delta_{r2}\right) \tag{4.18}$$

Now we assume that $\Delta_{r1}=\Delta_{r}\left(i\right)$ and that $\Delta_{r2}=\Delta_{r}\left(i-1\right)$, where *i* is the meshpoint number. Solving for $\frac{\partial^{2}\xi\left(r\right)}{\partial r^{2}}$ and applying this assumption:

$$\frac{\partial^{2}\xi\left(r\right)}{\partial r^{2}}=\frac{2\xi\left(r-\Delta_{r}\left(r\right)\right)}{\Delta_{r}\left(r-1\right)\left(\Delta_{r}\left(r\right)+\Delta_{r}\left(r-1\right)\right)}-\frac{2\xi\left(r\right)}{\Delta_{r}\left(r\right)\Delta_{r}\left(r-1\right)}+\frac{2\xi\left(r+\Delta_{r}\left(r\right)\right)}{\Delta_{r}\left(r\right)\left(\Delta_{r}\left(r\right)+\Delta_{r}\left(r-1\right)\right)}$$

(Second derivative approximation formula for nonuniform meshes) (4.19)

4.3.2.2 Model for the interface

The above model is valid everywhere in the device except at the interface between the oxide and the air. At this part of the device, we must use a different formula that takes into account the special properties of the boundary. Assume that we take a small cylinder of volume *V* around a certain section of the interface, perpendicular to the interface. Regardless of the dimension of the simulation, our interface at $y=y_0$ is always only perpendicular to the *y*-axis, so we will solve this equation specifically for this direction.

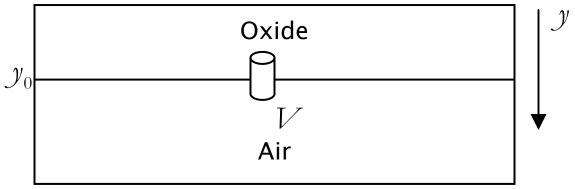


Figure 4-1 (above) Derivation of the interface equation.

Starting from the Poisson equation (3.2), we have

$$\nabla\cdot\left(\varepsilon\nabla\phi\right)=0 \tag{4.20}$$

From the physics of electricity, we know that $-\nabla\phi=E$, where *E* is the electric field. Thus, we may write this equation as

$$\nabla\cdot\left(\varepsilon E\right)=0 \tag{4.21}$$

Taking the volume integral of the cylinder, we have

$$\oint_V \nabla \cdot (\varepsilon E)\, dV = 0 \tag{4.22}$$

By Stoke’s Theorem,

$$\oint_S \varepsilon E dS = 0 \tag{4.23}$$

As the cylinder is collapsed to a radius of 0, all surface vectors cancel except at the top and bottom circle, leaving us with

$$-\varepsilon_{oxide} \nabla \phi_{y_0^+} + \varepsilon_{air} \nabla \phi_{y_0^-} = 0 \tag{4.24}$$

Which, by the definition of the derivative, evaluates to our formula:

$$\varepsilon_{oxide} \frac{\phi(y_0) - \phi(y_0 - \Delta_y(j))}{\Delta_y(j-1)} - \varepsilon_{air} \frac{\phi(y_0 + \Delta_y(j)) - \phi(y_0)}{\Delta_y(j)} = 0 \tag{4.25}$$

where *j* is the meshpoint number. To facilitate storage in the matrix, this formula may be rewritten as

$$\frac{\varepsilon_{oxide}}{\Delta_y(j-1)} \phi(y_0 - \Delta_y(j)) - \left(\frac{\varepsilon_{oxide}}{\Delta_y(j-1)} + \frac{\varepsilon_{air}}{\Delta_y(j)} \right) \phi(y_0) + \frac{\varepsilon_{air}}{\Delta_y(j)} \phi(y_0 + \Delta_y(j)) = 0$$

(approximation formula at interface for all nonuniform mesh problems) (4.26)

4.3.3 Two Dimensional Nonuniform Mesh

Recall equation (3.4), the two dimensional version of the Poisson equation:

$$\nabla \cdot \left(\varepsilon \begin{pmatrix} x \\ y \end{pmatrix} \nabla \phi \begin{pmatrix} x \\ y \end{pmatrix} \right) = 0 \quad (4.27)$$

This equation may reduce to the following form (see section 4.2.2 for derivation in one dimension):

$$\frac{\partial \left(\varepsilon \begin{pmatrix} x \\ y \end{pmatrix} \left(\frac{\partial \phi \begin{pmatrix} x \\ y \end{pmatrix}}{\partial x} \right) \right)}{\partial x} + \frac{\partial \left(\varepsilon \begin{pmatrix} x \\ y \end{pmatrix} \left(\frac{\partial \phi \begin{pmatrix} x \\ y \end{pmatrix}}{\partial y} \right) \right)}{\partial y} = 0 \tag{4.28}$$

Recall that we derived a separate formula for the interface of the device in equation (4.26). Then, for all non-interface meshpoints, $\varepsilon\begin{pmatrix}x\\y\end{pmatrix}$ may be considered constant. Therefore, we can rewrite the equation as:

$$\varepsilon\begin{pmatrix}x\\y\end{pmatrix}\frac{\partial\left(\frac{\partial\phi\begin{pmatrix}x\\y\end{pmatrix}}{\partial x}\right)}{\partial x}+\varepsilon\begin{pmatrix}x\\y\end{pmatrix}\frac{\partial\left(\frac{\partial\phi\begin{pmatrix}x\\y\end{pmatrix}}{\partial y}\right)}{\partial y}=\varepsilon\begin{pmatrix}x\\y\end{pmatrix}\left(\frac{\partial^2\phi\begin{pmatrix}x\\y\end{pmatrix}}{\partial x^2}+\frac{\partial^2\phi\begin{pmatrix}x\\y\end{pmatrix}}{\partial y^2}\right)=0 \qquad (4.29)$$

which means that we can apply equation (4.19) twice, one time for each dimension. From this application, we receive the numerical formula

$$\frac{2}{\Delta_y(j-1)\left(\Delta_y(j)+\Delta_y(j-1)\right)}\phi\begin{pmatrix}x\\y-\Delta_y(j)\end{pmatrix}+\frac{2}{\Delta_x(i-1)\left(\Delta_x(i)+\Delta_x(i-1)\right)}\phi\begin{pmatrix}x-\Delta_x(i)\\y\end{pmatrix}$$
$$-\left(\frac{2}{\Delta_x(i-1)\Delta_x(i)}+\frac{2}{\Delta_y(j-1)\Delta_y(j)}\right)\phi\begin{pmatrix}x\\y\end{pmatrix}$$
$$+\frac{2}{\Delta_x(i)\left(\Delta_x(i)+\Delta_x(i-1)\right)}\phi\begin{pmatrix}x+\Delta_x(i)\\y\end{pmatrix}+\frac{2}{\Delta_y(j)\left(\Delta_y(j)+\Delta_y(j-1)\right)}\phi\begin{pmatrix}x\\y+\Delta_y(j)\end{pmatrix}=0$$

(two dimensional nonuniform mesh numerical solution) (4.30)

where *i* is the meshpoint number in the *x* direction and *j* is the meshpoint number in the *y* direction.

The matrix derived is similar to the one in equation (8.1), with the appropriate coefficients and one row corresponding to the interface that represents equation (4.26).

4.3.4 Three Dimensional Nonuniform Mesh

Recall equation (3.5), the two dimensional version of the Poisson equation:

$$\nabla\cdot\left(\varepsilon\begin{pmatrix}x\\y\\z\end{pmatrix}\nabla\phi\begin{pmatrix}x\\y\\z\end{pmatrix}\right)=0 \qquad (4.31)$$

We can see based on equation (4.19) and the solution derived in section 4.3.3 that the numerical solution for three dimensions may be expressed as follows:

$$\frac{2}{\Delta_y(j-1)\left(\Delta_y(j)+\Delta_y(j-1)\right)}\phi\begin{pmatrix}x\\y-\Delta_y(j)\\z\end{pmatrix}+\frac{2}{\Delta_x(i-1)\left(\Delta_x(i)+\Delta_x(i-1)\right)}\phi\begin{pmatrix}x-\Delta_x(i)\\y\\z\end{pmatrix}$$

$$+\frac{2}{\Delta_z(k-1)\left(\Delta_z(k)+\Delta_z(k-1)\right)}\phi\begin{pmatrix}x\\y\\z-\Delta_z(k)\end{pmatrix}$$

$$-\left(\frac{2}{\Delta_x(i-1)\Delta_x(i)}+\frac{2}{\Delta_y(j-1)\Delta_y(j)}+\frac{2}{\Delta_z(k-1)\Delta_z(k)}\right)\phi\begin{pmatrix}x\\y\\z\end{pmatrix}$$

$$+\frac{2}{\Delta_x(i)\left(\Delta_x(i)+\Delta_x(i-1)\right)}\phi\begin{pmatrix}x+\Delta_x(i)\\y\\z\end{pmatrix}+\frac{2}{\Delta_y(j)\left(\Delta_y(j)+\Delta_y(j-1)\right)}\phi\begin{pmatrix}x\\y+\Delta_y(j)\\z\end{pmatrix}$$

$$+\frac{2}{\Delta_z(k)\left(\Delta_z(k)+\Delta_z(k-1)\right)}\phi\begin{pmatrix}x\\y\\z+\Delta_z(k)\end{pmatrix}=0$$

(three dimensional nonuniform mesh numerical solution) (4.32)

where *i* is the meshpoint number in the *x* direction, *j* is the meshpoint number in the *y* direction, and *k* is the meshpoint number in the *z* direction.

The matrix for this case is similar to the two dimensional matrix but includes all of the terms above and has dimensions $n_z n_x n_y$ rows by $n_z n_x n_y$ columns, where n_z, n_x, and n_y represent the number of meshpoints in the *z*, *x*, and *y* directions, respectively. There is one row representing the interface that solves equation (4.26).

5 Methods of Computing the Solution

5.1 Gauss-Jordan Elimination

It is theoretically possible to solve all problems in this work for ϕ using the familiar linear algebra method of Gaus-Jordan elimination. The version of this method[16] we used requires the storage of all elements of the matrix, storage of the right hand side vector and solution vector. Each variable of size double requires a storage space of 8 bytes.[17] Given a matrix of *n* by *n* elements and a two vectors of *n* elements, the computational complexity is $O\left(n^3\right)$ and the space complexity is $(8\text{ bytes})\left(n^2+2n\right)=O\left(n^2\right)$,[16] when $n=n_y$ for the one dimensional case, $n=n_x n_y$ for the two dimensional case, and $n=n_x n_y n_z$ for the three dimensional case.

5.2 Sparse Matrix Format and Solution

5.2.1 Space-Saving Indexed Storage of Sparse Matrices

Recall from section 4 that most elements of the matrix are 0s. Especially see equations (4.12) and (8.1). Our implementation of the Gauss-Jordan elimination method stores and solves all of these zeros, and this takes up most of the time and space. Thus we use a sparse matrix format for the larger solutions. This format does not store any 0s in the off diagonals. It requires two vectors, one of variables of type unsigned long and the other of double precision values. The size of each of these vectors is all of the diagonal elements $+$ the number of all nonzero off diagonal elements $+$ 1. The right hand side and solution vectors are stored completely.

Making precise prediction of spatial complexity is somewhat harder now because we cannot readily predict the number of off-diagonal elements. However, we can use the upper bound for the number of diagonal elements plus the number of nonzero off diagonal elements, $5n_x n_y$ in two dimensions and $7n_x n_y n_z$ in three dimensions. Here, n_x, n_y, and n_z refer to the number of points in *x*, *y*, *z* directions. These bounds are derived from the number of maximum nonzero diagonals in each matrix (5 in two dimensions and 7 in three dimensions). Also, they can be derived using the number of neighbors: in the two-dimensional case, a point can have at most four neighbors, leading to five nonzero diagonals, including the point itself. For each problem, we need to store one such vector of double precision values (8 bytes) and one such vector of unsigned long values (4 bytes).[17] It follows that the total storage space for each two dimensional solution will be $5n_x n_y\left(8 \text{ bytes}\right)+5n_x n_y\left(4 \text{ bytes}\right)+n_x n_y\left(8 \text{ bytes}\right)+n_x n_y\left(8 \text{ bytes}\right)=O\left(n_x n_y\right)$ and for each three dimensional solution, it will be
$7n_x n_y n_z\left(8 \text{ bytes}\right)+7n_x n_y n_z\left(4 \text{ bytes}\right)+n_x n_y n_z\left(8 \text{ bytes}\right)+n_x n_y n_z\left(8 \text{ bytes}\right)=O\left(n_x n_y n_z\right)$.

For a detailed explanation of *how* the elements are stored, please see reference [16], page 78 (freely available at http://www.library.cornell.edu/nr/bookcpdf/c2-7.pdf [page 8 of PDF]).

5.2.2 Biconjugate Gradient Method

The biconjugate gradient method exploits the indexed storage for sparse matrices discussed above to reduce the computational complexity of the solution to $O(n)$ compared to $O\left(n^3\right)$ for the Gauss-Jordan elimination (see [16], page 83 or page 13 of the PDF referenced above). In practice, using the biconjugate gradient method is far more advantageous in terms of memory and computing time required when compared to the Gauss-Jordan elimination method in complicated examples. However, for small cases it may be easier to use the Gauss-Jordan Elimination method.

6 Results

6.1 *Computational Results: Electrostatic Potential, Electric Field, and Capacitance*

This section details various problems that were solved directly in this research. This list is not exhaustive and builds up to the addition of a nanotube to the device to represent the proposed sensor. More overarching, analytical solutions can be found in section 6.2.

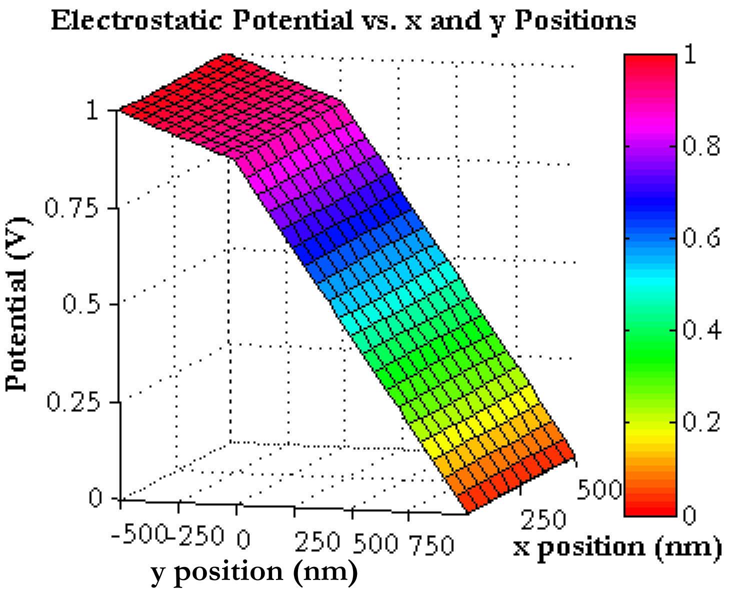


Figure 6-2 (above) Two-dimensional solution for potential.

6.1.1 One Dimensional Case Uniform Mesh

Problem statement: Solving for the one-dimensional case, we will focus on the *y*-axis, shown in figure 3-1. We choose a device that has a 500 nm oxide (silicon dioxide; $\varepsilon = 3.9$) and 1000 nm of air ($\varepsilon = 1.0$). The oxide is represented by the negative values of *y*; the air by the positive values. We use fixed boundary conditions: the left potential is set to 1 V; the right is set to 0 V. Using 151 meshpoints in the *y* direction, we obtain the solution for electrostatic potential illustrated in figure 6-1. This is not a continuous line but is instead a series of discrete points. The uniform spacing is 10 nm.

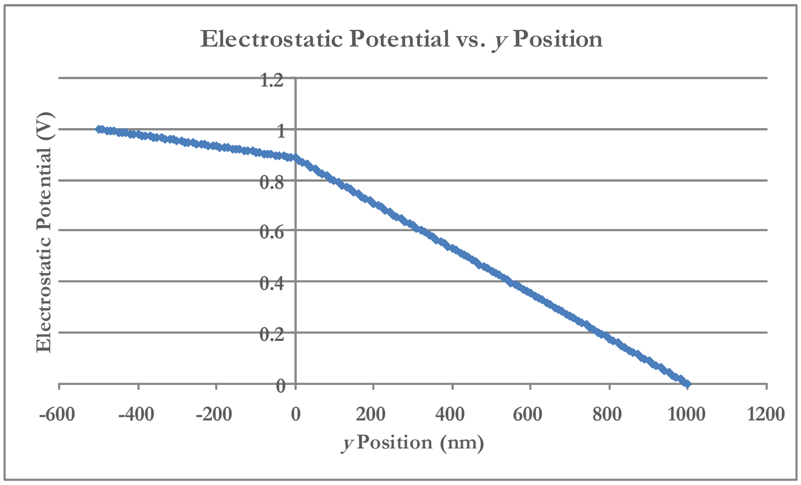


Figure 6-1 (above) One-dimensional solution for potential.

6.1.2 Two Dimensional Case Square Mesh

Problem statement: We solve the same problem as above but in two dimensions using Gauss Jordan elimination. The device is again partitioned along the *y* direction, 500 nm for silicon dioxide ($\varepsilon = 3.9$) and 1000 nm for air ($\varepsilon = 1.0$). The device width in the *x* direction is 500 nm and there is no nanotube. Our boundary conditions are constant on the *y* extremes (1 V at $y = -500$ nm and 0 V at $y = 1000$ nm) and are floating on the *x* boundaries ($\frac{\partial \phi}{\partial y} = 0$).

The number of meshpoints in the *y* direction is 31 ($n_y = 31$); in the *x* direction it is 11 ($n_x = 11$). We obtain the result for electrostatic potential illustrated in figure 6-2. We can clearly see the square mesh in this figure because the spacing between the meshpoints is 50 nm. The limitation of this solution is that the smallest nanotube that it supports is 50 nm by 50 nm, which is generally too large to represent reality. In the next examples, we will see how to solve denser problems efficiently. We use the Gauss-Jordan elimination method to solve this problem, which takes a reasonable amount of time and space.

6.1.3 Two Dimensional Solution with a Nanotube Uniform Mesh

Problem statement: We now introduce a nanotube. This nanotube will be represented by four meshpoints (20 nm by 20 nm square) of a fixed potential (0 V) at a distance of 1 nm from the interface of the oxide and air, inside the air. We will now have 151 meshpoints in the *y* direction ($n_y = 151$) and 51 meshpoints in the *x* direction ($n_x = 51$). The physical dimensions of the figure are the same as in the previous problem (see section 6.1.2), as are all boundary conditions except the bottom boundary in the *y* direction, which is changed to a floating boundary ($\frac{\partial \phi}{\partial x} = 0$). The calculated result for electrostatic potential is illustrated in figure 6-3. This result is obtained using the biconjugate gradient method.

The mesh spacing in this example is 10 nm. This creates a very smooth solution, but it still limits the size of features we can add. Using this method, the smallest nanotube that we can insert is represented by a 10 nm by 10 nm square, still quite large. Smaller tubes will be added in the following example.

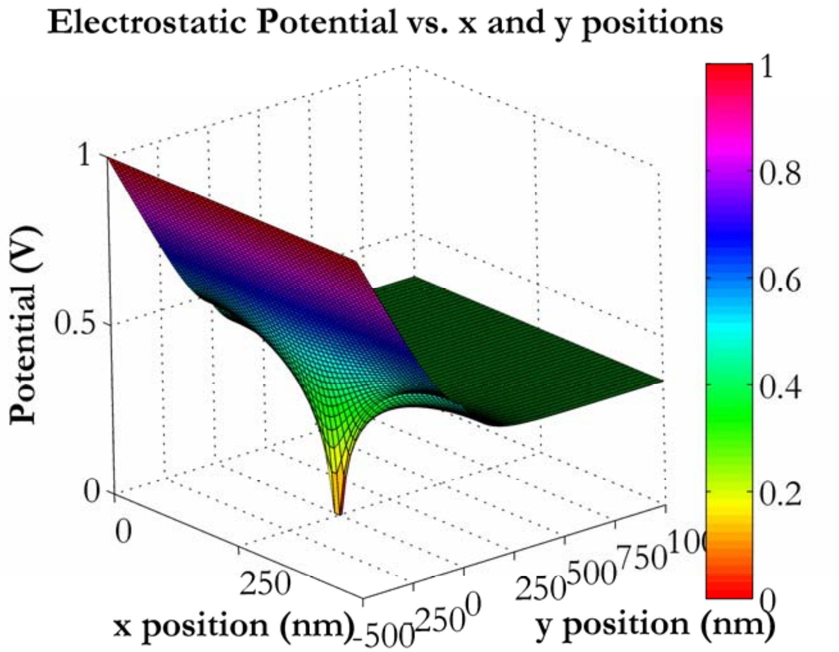


Figure 6-3 (above) Plot of the electrostatic potential of problem 6.1.3.

Capacitance: The gate capacitance of this device, with the gate and nanotube acting as the plates, was calculated to be 0.215 Picofarads/cm.

6.1.4 Two Dimensional Solution with a Nanotube Nonuniform Mesh

Problem statement: We will now add a 2 nm by 2 nm square representing a nanotube at a distance of 1 nm from the interface between the oxide and air into our device. This is

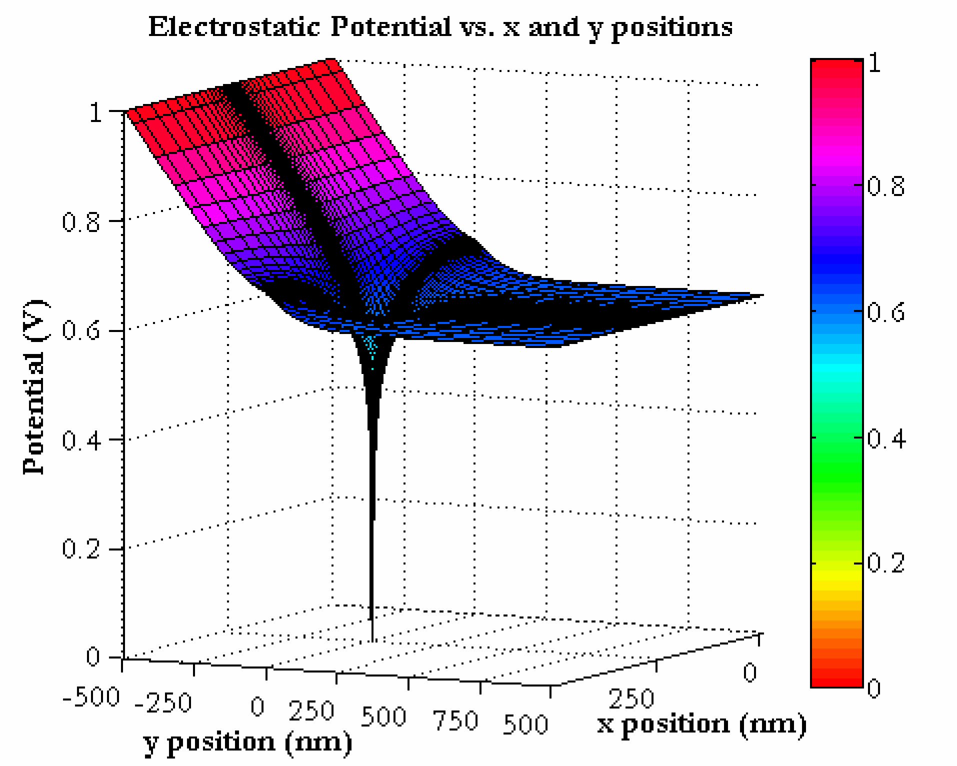


Figure 6-4 (above) Nonuniform two dimensional solution, including a 2 nm by 2 nm square representing a nanotube.

representative of the approximate size of a real nanotube (see section 2.1). The component size and boundary conditions are otherwise the same as in the previous example. We will now introduce a nonuniform mesh, centered at the nanotube. The advantage is that this allows us to have four small meshpoints representing the nanotube while not requiring such a large number of points that would make the computation inefficient. We choose $\Delta_x = \Delta_y = 1.0$ and $p_x = p_y = 1.2$ in equation (4.15) for the nonuniform mesh. The calculated potential is illustrated in figure 6-4.

The nonuniform nature of the solutions is clear from the different sizes of the squares in figure 6-4, as is the power of this method to create very small components efficiently. The small size of the nanotube can be seen by comparing the region of fixed potential to the one in figure 6-3.

This solution was completed using the biconjugate gradient method. We determine the number of meshpoints only after running the simulation, in this case, $n_x = 45 \text{ points and } n_y = 57 \text{ points}$.

Capacitance: The gate capacitance of this device was calculated to be 0.127 Picofarads/cm.

The electric field at the nanotube is seen in figure 6-5.

6.1.5 Adding a Particle to Problem 6.1.4

Problem statement: This problem is identical to Problem 6.1.4, but now we assume that a particle (1 nm by 1.2 nm) has been captured in the sensor. We will see how this impacts the electrostatic potential, electric field, and capacitance. Again, we use the biconjugate-gradient method to compute the solution.

Electrostatic Potential: Figure 6-7 is a zoomed in and rotated view of the potential near the nanotube. The particle, nanotube, and interface are pointed out in the figure. Part of the electric field is illustrated in figure 6-6. Note the difference from figure 6-5, where the particle is absent.

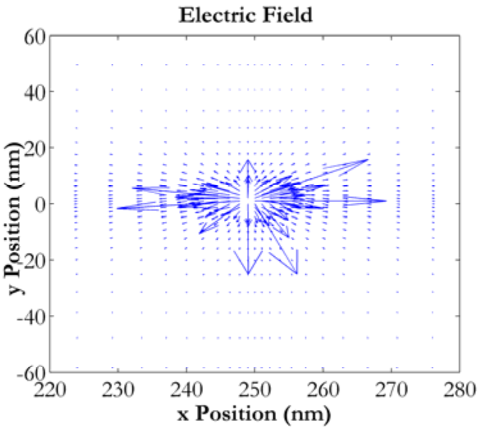


Figure 6-5 (above) Plot for the electric field of problem 6.1.4.

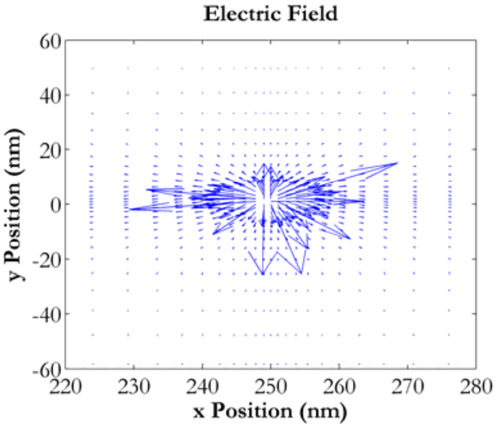


Figure 6-6 (above) Part of the plot for the electric field of problem 6.1.5. Notice the effects of the particle when compared to figure 6-5, to the left. This change leads to a difference in capacitance, which can be sensed.

Capacitance: The gate capacitance of this device was calculated to be 0.136 Picofarads/cm. This change is caused by the change in electric field.

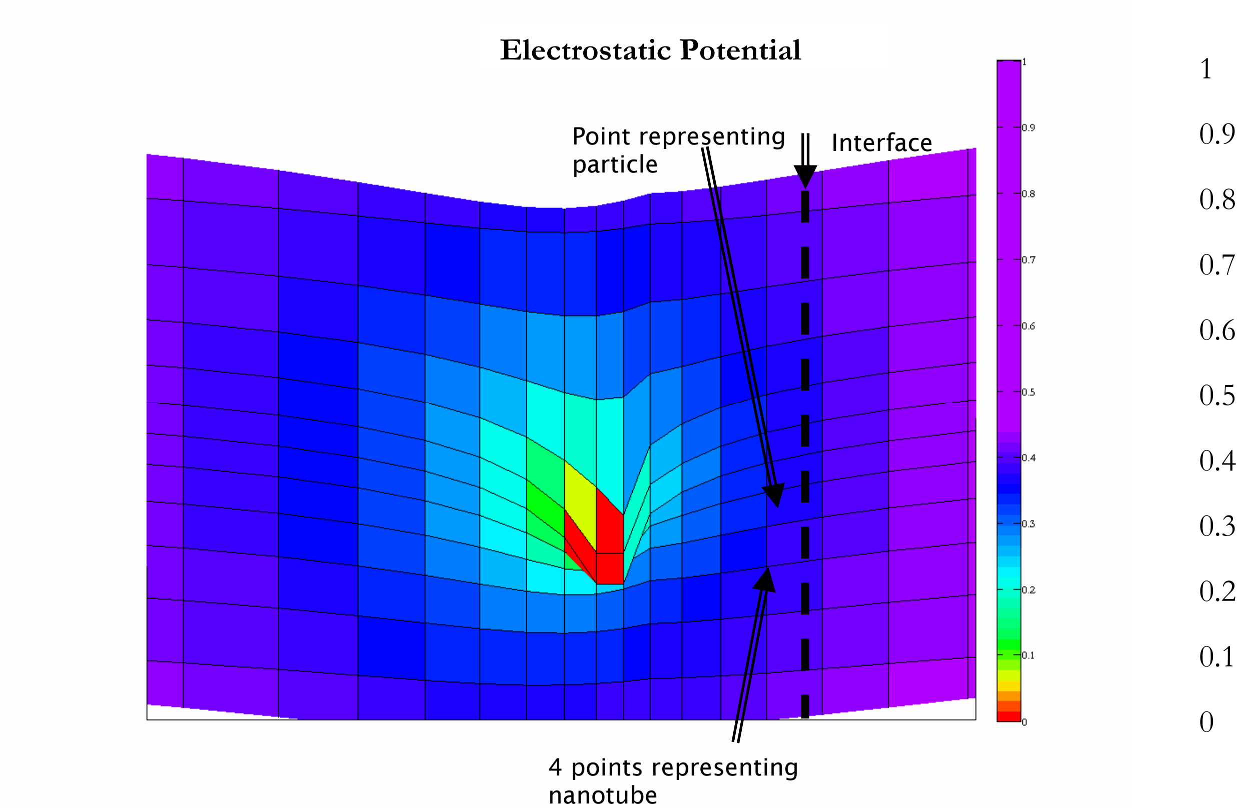
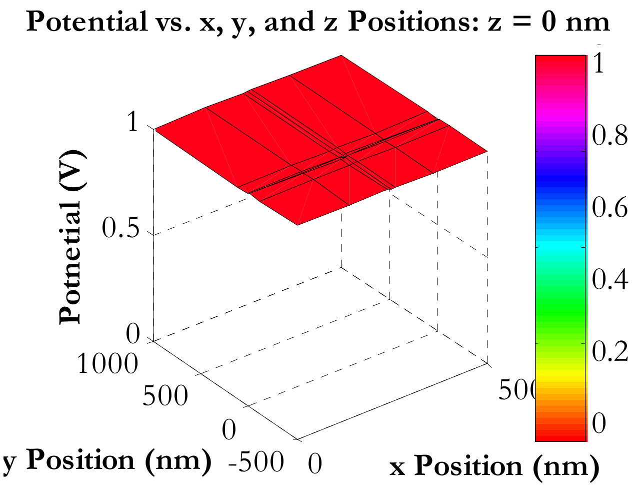
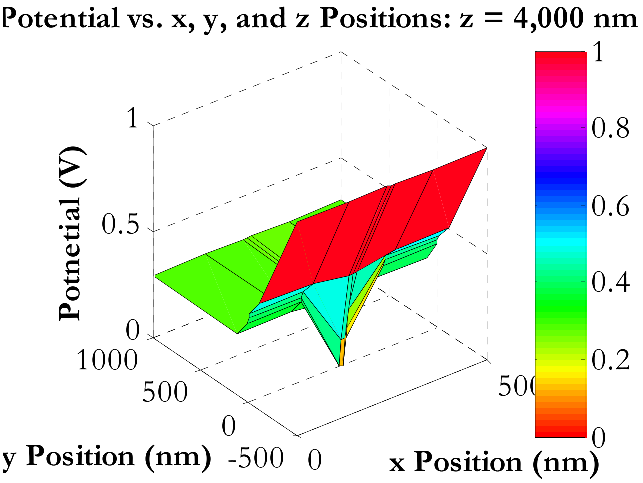


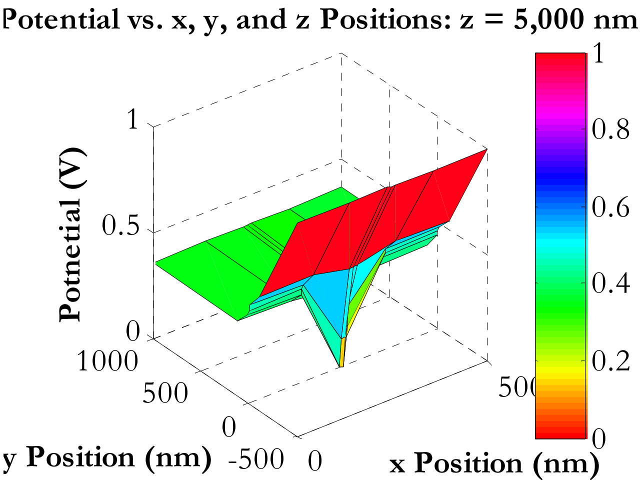
Figure 6-7 (above) Nonuniform two dimensional solution, including a 2 nm by 2 nm square representing a nanotube and a 1 nm by 1.2 nm square particle. The particle is located 1 nm from the interface.

6.1.6 Three Dimensional Solution with Nanotube Nonuniform Mesh

Problem statement: We now extend our solution into three dimensions. The width of the figure in the *x*-direction is 500 nm and its length in the *z*-direction is 10,000 nm. The nanotube added has a diameter of 2 nm by 2 nm. In the *y*-direction, the oxide is 500 nm long ($\varepsilon = 3.9$) and the air is 1000 nm long ($\varepsilon = 1.0$). Both of the *x* boundaries, the bottom *y* boundary, and a square around the nanotube in both *z* boundaries (in this case a substantial 3 meshpoints by 3 meshpoints) have floating boundary conditions. The top *y* boundary and the rest of both *z* boundaries are fixed to 1 V. The solution for potential is illustrated in figure 6-8 (below and on the next page). Note that fully drawing this solution requires four dimensions (one for each direction and one for potential). Since it is not possible to draw in this manner, we take a snapshot of the potential at every *z* meshpoint. This simulation was solved using the biconjugate gradient method. The number of meshpoints is $n_x = 7$, $n_y = 7$, and $n_z = 5$.


Potential vs. x, y, and z Positions: z = 0 nm
Potnetial (V)
1
0.5
0
1000
500
0
-500
y Position (nm)
0
50
x Position (nm)
1
0.8
0.6
0.4
0.2
0


Potential vs. x, y, and z Positions: z = 4,000 nm
Potnetial (V)
1
0.5
0
1000
500
0
-500
y Position (nm)
0
50
x Position (nm)
1
0.8
0.6
0.4
0.2
0


Potential vs. x, y, and z Positions: z = 5,000 nm
Potnetial (V)
1
0.5
0
1000
500
0
-500
y Position (nm)
0
50
x Position (nm)
1
0.8
0.6
0.4
0.2
0

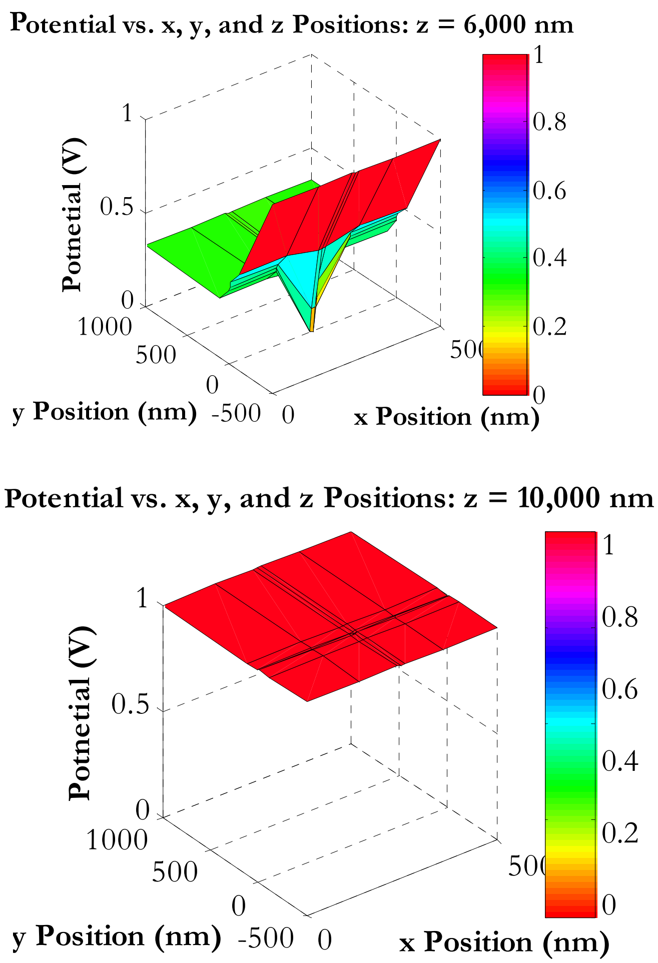


Figure 6-8 (above and on the previous page) Three dimensional, nonuniform solution with a 2 nm by 2 nm nanotube.

6.2 Analytical Results and Solution Validation

<u>6.2.1 Effect of Nanotube Potential on Gate Capacitance</u>

We found that the fixed potential of the nanotube has no impact on the gate capacitance, as illustrated by the graph in figure 6-9. We here investigated two device configurations, and varied the potential on the nanotube:

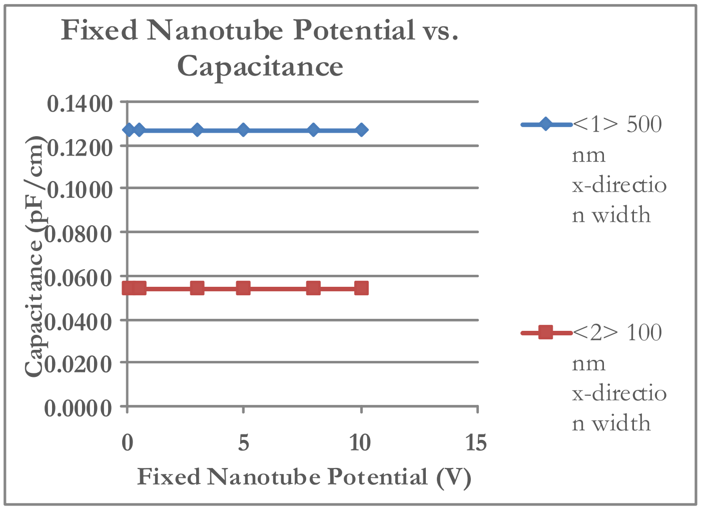


Figure 6-9 (above)

Simulations:

<1> The device examined in example 6.1.4, which has a 500 nm *x*-width.

<2> A device similar to <1>, but with a 100 nm *x*-width.

6.2.2 Effect of Gate Width (*x*-direction) on Gate Capacitance

In this example, the gate *x*-width is varied.

6.2.2.1 Devices

For all the devices below, the oxide and air have thicknesses of 500 nm and 1000 nm, and relative dielectric constants of 3.9 and 1.0, respectively. In addition, we used fixed boundary conditions (1 V) for the gate terminal, and floating boundary conditions for all other sides.

Simulated Nanotube-Embedded Sensors (nanotube size—2nm by 2nm):

<1> This is the sensor device that does not have any particles attached to it. See example 6.1.4.

<2> This is the sensor with a charged particle. Its configuration is explained in example 6.1.5. The size of the particle is always 1 nm by 1.2 nm, exactly, because it is near the center of the nonuniform mesh. This particle is always located 1 nm from the interface between the oxide and air in the *y*-direction.

<3> This is the sensor with a charged particle attached to another location. The difference between <3> and <2> is that the particle is slightly larger in effective size and is farther away from the interface.

Theoretical Nanotube Sensor Analysis (nanotube diameter—2nm):

<4> To obtain the capacitance between an infinite plane and an infinitely long tube, we used the following analytical expression:

$$C \approx \frac{2\pi\varepsilon\varepsilon_0}{\ln\left(4t/d\right)} \tag{6.1}$$

where ε is the average relative dielectric constant of the medium, *t* is the distance between the tube and the plane, and *d* is the diameter of the tube.[18] We use $\varepsilon = 3.9\times 0.85$, $t = 500\ \text{nm}$ and $d = 2\ \text{nm}$.

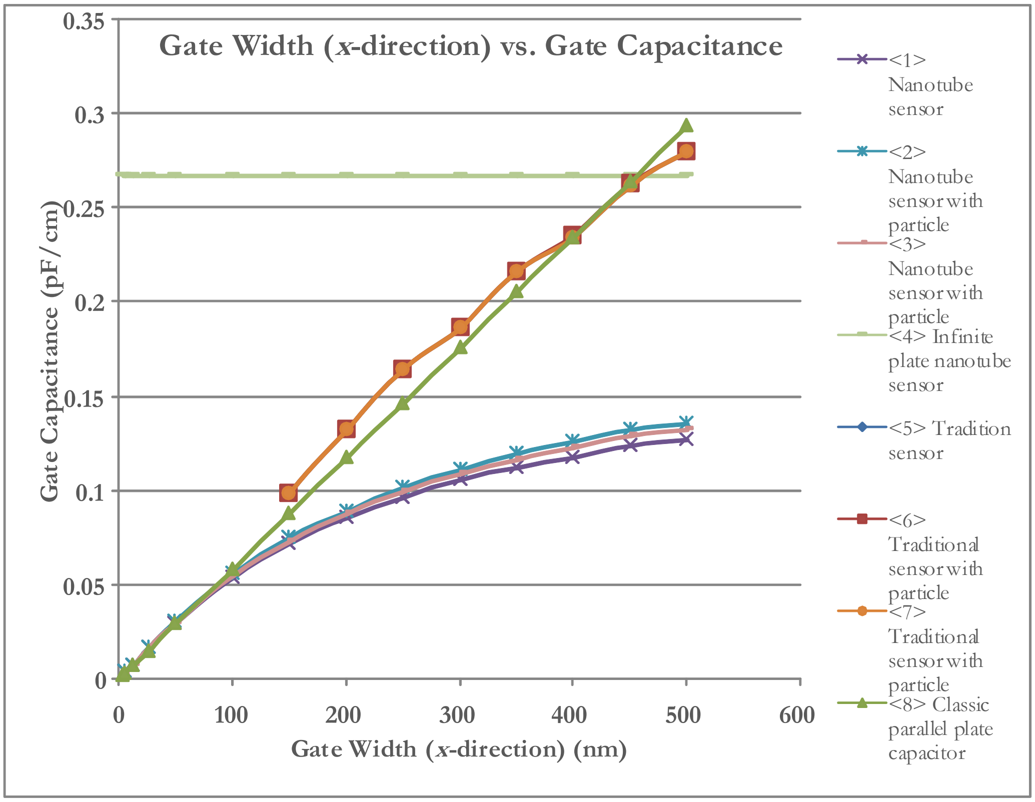


Figure 6-10 (above)

Simulated Traditional Sensors (metal plate— 129nm by 129nm):

<5> This is a sensor shown in figure 2-4(a), where we have a metal plate under the oxide instead of a nanotube. This metal plate is much bigger in size compared to a nanotube. To obtain its capacitance, we fixed its potential to a constant.

<6> We have a particle attached to the metal plate. The size of the particle is always 1 nm by 1.2 nm, exactly, because it is near the center of the nonuniform mesh. This particle is always located 1 nm from the interface between the oxide and air in the *y*-direction.

<7> This sensor also has a particle attached to the metal plate. However, the particle has a larger size and is farther away from the interface.

Theoretical Parallel Plate Capacitor Analysis:

<8> The per length capacitance of a parallel plate capacitor can be written as follows:

$$C = \varepsilon\varepsilon_0 \frac{W}{t} \tag{6.2}$$

where ε is the relative dielectric constant of the medium, *W* is the width of the plate, and *t* is the separation of the plates. We use $\varepsilon = 3.9 \times 0.85$, and $t = 500\ \text{nm}$.

6.2.2.2 Conclusions

We can make the following conclusions by looking at figure 6-10:

1. The capacitance increases as gate width increases due to higher charge accumulation.

2. In the nanotube sensor, the capacitance increases more dramatically when a particle is added, compared to the traditional sensor. It increases more with a particle closer to the interface. The line for device <5>, the traditional capacitor without a particle, is hidden by the two other lines for the traditional capacitors with particles (<6> and <7>) because the values are close to each other.

3. We see that the metal plate sensor has a higher absolute capacitance than the nanotube sensor.

4. In this graph, we have information that validates our results: (1) The lines for the traditional capacitor are relatively close to those of the classic parallel plate capacitor, as we expect; and (2) the line for the infinite nanotube sensor is the same order of magnitude as the lines of the nanotube sensors. It is expected that the value for the infinite-gate sensor is greater than that of the simulated nanotube sensors because, as we have shown, capacitance increases with gate width.

In the next section, we will examine more specifically how these results can be interpreted in light of the intended sensor application.

6.2.3 Effects of Gate Width (*x*-direction) on Sensitivity

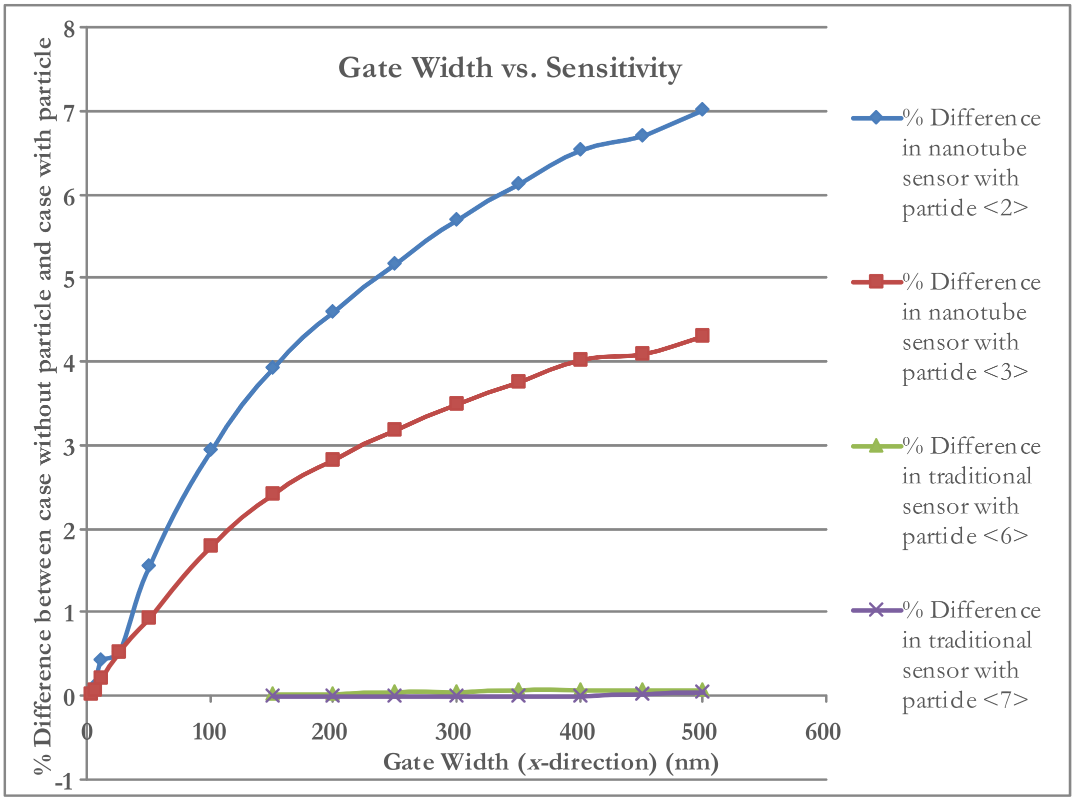


Figure 6-11 (above)

We noted above that the difference between the particle and no-particle cases appears much greater in the nanotube sensor than in the traditional sensor. Here, we measure sensitivity by determining the percent difference of the capacitance with and without a particle. We will now quantify it with the graph in figure 6-11. We see that:

1. The capacitance of nanotube sensors changes to a *much* greater extent than that of traditional sensors. **Here we show the key advantage of using the proposed sensor design:** it is much easier to detect larger changes in capacitance than smaller ones. The percent difference in the nanotube is higher by orders of magnitude than that of a traditional sensor.

2. In all cases, sensitivity increases as gate width (*x*-direction) increases.

3. For both the traditional and nanotube sensors, the percent difference in capacitance is smaller for the particle farther away from the interface.

6.2.4 Effects of Oxide Thickness (*y*-direction) on Gate Capacitance

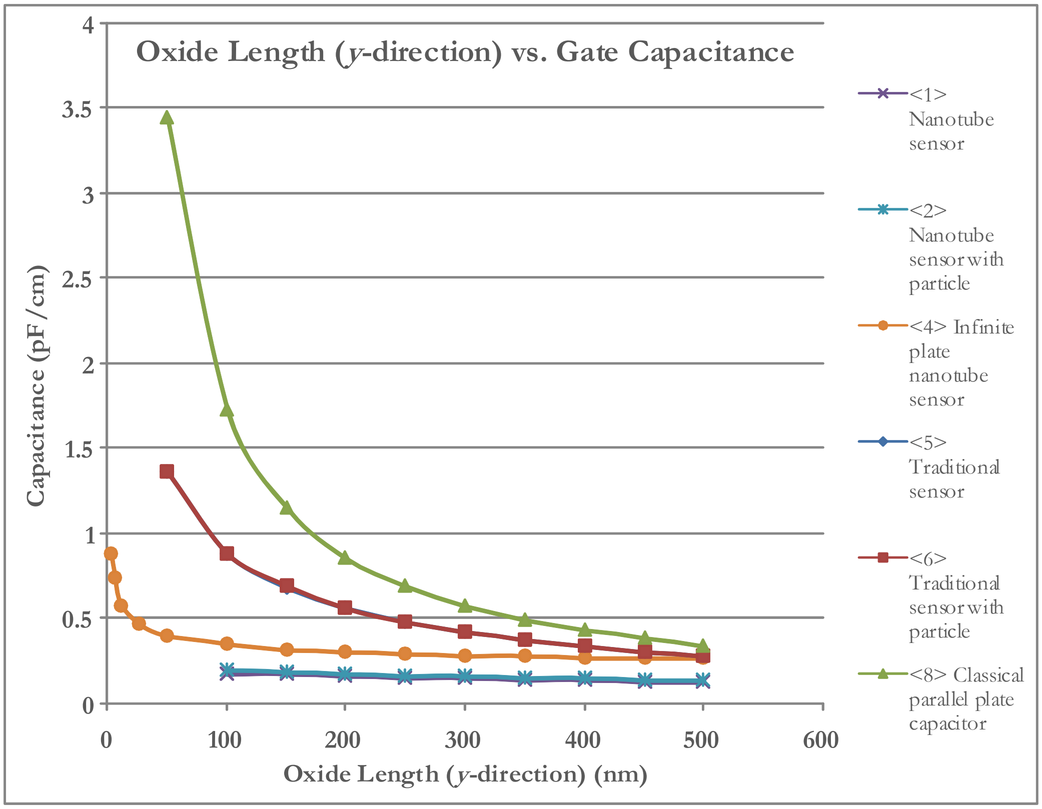


Figure 6-12 (above)

In this analysis, we will vary the oxide length (*y*-axis) and observe the effects that this has on gate capacitance.

6.2.4.1 Devices

We used the same devices as in section 6.2.2.1, except that now we vary the oxide thickness and keep the *x*-width constant at 500 nm. We also omit devices <3> and <7> in 6.2.2.1. Our results are graphed in figure 6-12.

6.2.4.2 Conclusions

1. Without exception, we see that capacitance decreases as oxide thickness increases, in agreement with theory and equation (6.2).

2. As in our previous results, the traditional sensor has a higher capacitance than the nanotube sensor.

3. As before, the only line visible for the traditional sensor is the line for device <6>. The line for device <5> is hidden underneath it since the values are close together.

4. As expected, the infinite *x*-width plate capacitor <4> has a higher capacitance than the other nanotube devices, whose *x*-width is 500 nm.

5. Again, we see a higher difference between the cases with and without the particle in the nanotube sensor than in the traditional sensor, as explained below.

6.2.5 Effect of Oxide Thickness (*y*-direction) on Sensitivity

Again we use percent difference as a measure of sensitivity.

Here we compare the change in capacitance when a particle is added to traditional and nanotube sensors in the previous example. See the graph in figure 6-13. We see that:

1. As before, the nanotube sensor far exceeds the traditional sensor in terms of percent difference in capacitance. Again, this is a major advantage of using nanotube-embedded sensors over traditional ones.

2. In both cases, the percent difference decreases as oxide thickness (*y*-axis) increases.

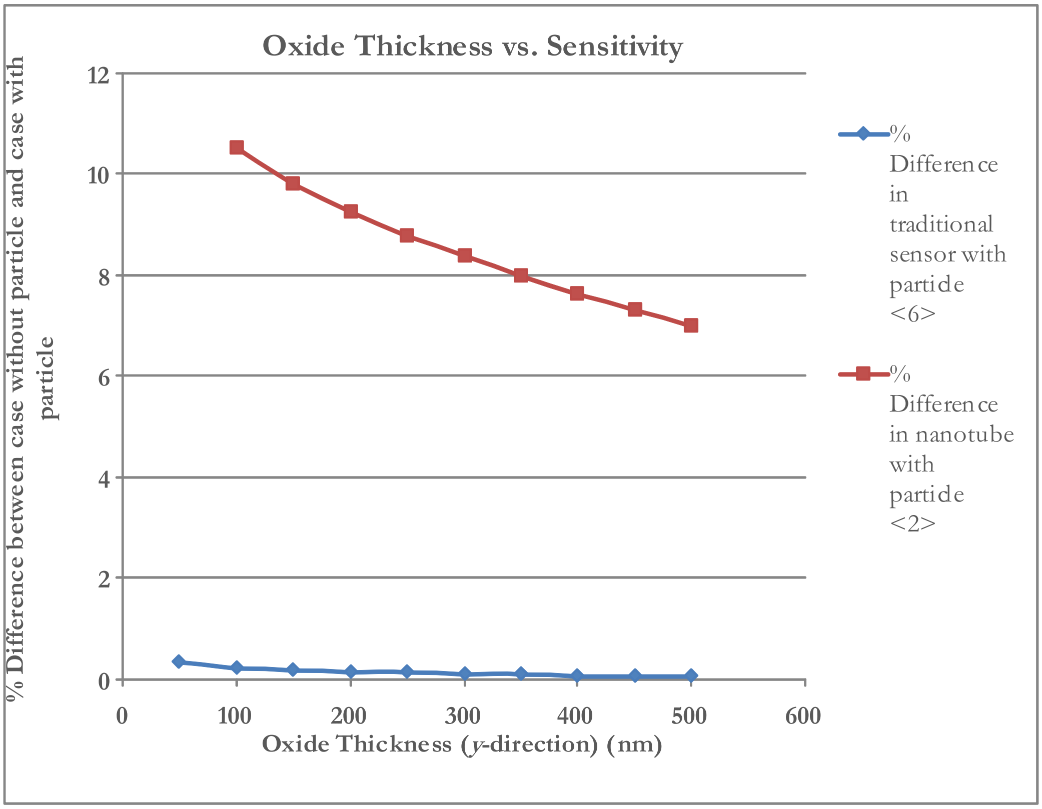


Figure 6-13 (above)

7 Discussion and Conclusions

The numerical solutions show unequivocally that the proposed design for a nanotube sensor is advantageous over the traditional design in terms of percent difference upon the insertion of a particle, which leads to a higher sensitivity. According to our results, the highest percent difference occurs in sensors which have a high gate width (*x*-direction) and a low oxide thickness (*y*-direction). This will make detection easier—even one particle (1 nm by 1.2 nm) gives a percent difference on the order of 10^0 to 10^1, which can be readily detected. This is significantly higher than the order 10^{-2} to 10^{-1} percent differences of the traditional sensors, in which case multiple particles need to enter the device before the change in capacitance is large enough to be detected.

The key advantages of such a development include the following:

1. Due to the small size of its components, the nanotube sensor will inherently respond more quickly than a traditional sensor.

2. Detection of the desired particle will occur at a lower concentration and with far fewer particles.

3. Faster speed of detection will be experienced because the detector will be activated upon the entry of one particle, not many. Considering the probabilistic nature of particles entering this device, this may be a very significant advantage.

In practical terms, this might mean more time for soldiers to don a protective mask or for a building or area to be evacuated. We hope that this sensor will be used to protect and defend against needless environmental contamination and biochemical weapons and that it will be used as a shield against loss of life.

Bibliography

[1] M. R. Hillman, "Overview: cause and prevention in biowarfare and bioterrorism," *Vaccine,* vol. 20, pp. 3055-3067, Aug 2002.

[2] A. G. Robertson and L. J. Robertson, "From asps to allegations: biological warfare in history," *Mil Med,* vol. 160, pp. 369-73, Aug 1995.

[3] G. G. Harigel, "The Concept of Weapons of Mass Destruction: Chemical and Biological Weapons, Use in Warfare, Impact on Society and Environment," in *Proceedings of the Conference on Biosecurity and Bioterrorism*, Rome, Italy, 2000.

[4] R. A. Greenfield, B. R. Brown, J. B. Hutchins, J. J. Iandolo, R. Jackson, L. N. Slater, and M. S. Bronze, "Microbiological, biological, and chemical weapons of warfare and terrorism," *Am J Med Sci,* vol. 323, pp. 326-40, Jun 2002.

[5] "The Deepening Anthrax Mystery," in *New York Times* New York, 2002, p. A16.

[6] S. Iijima, "Helical Microtubules of Graphitic Carbon," *Nature,* vol. 354, pp. 56-58, Nov 1991.

[7] P. G. Collins and P. Avouris, "Nanotubes for electronics," *Scientific American,* vol. 283, pp. 62-70, Dec 2000.

[8] D. A. Neamen, *An introduction to Semiconductor devices*. Boston: McGraw-Hill, 2006.

[9] S. V. Patel, T. E. Mlsna, B. Fruhberger, E. Klaassen, S. Cemalovic, and D. R. Baselt, "Chemicapacitive microsensors for volatile organic compound detection," *Sensors and Actuators B-Chemical*, vol. 96, pp. 541-553, Dec 2003.

[10] G. Delapierre, H. Grange, B. Chambaz, and L. Destannes, "Polymer-Based Capacitive Humidity Sensor - Characteristics and Experimental Results," *Sensors and Actuators*, vol. 4, pp. 97-104, 1983.

[11] H. Shibata, M. Ito, M. Asakursa, and K. Watanabe, "A digital hygrometer using a polyimide film relative humidity sensor," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, pp. 564-569, Apr 1996.

[12] E. Fotis, "A new ammonia detector based on thin film polymer technology," *Sensors*, vol. 19, pp. 73-5, 2002.

[13] J. Guo, S. Goasguen, M. Lundstrom, and S. Datta, "Metal-insulator-semiconductor electrostatics of carbon nanotubes," *Applied Physics Letters*, vol. 81, pp. 1486-1488, Aug 2002.

[14] B. G. Streetman, *Solid state electronic devices*, 4th ed. Englewood Cliffs, N.J.: Prentice Hall, 1995.

[15] N. Goldsman, "ENEE 694: Physics and Simulation of Semiconductor Devices Class Notes," College Park, MD: University of Maryland, College Park, 1998.

[16] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical recipes in C: the art of scientific computing," 2nd ed Cambridge, New York: Cambridge University Press, 2000.

[17] "C++ Language Reference: Fundamental Types," in *Visual Studio .NET 2003 Documentation*, 7.1 ed Redmond, Washington: Microsoft Corporation, 2003.

[18] Y. F. Chen and M. S. Fuhrer, "Electric-field-dependent charge-carrier velocity in semiconducting carbon nanotubes," *Physical Review Letters*, vol. 95, Dec 2005.

Biography

TAL RUSAK is an undergraduate student majoring in Computer Science and Electrical and Computer Engineering at Cornell University. He is interested in learning through both coursework and through independent research activities and has participated in computational and theoretical projects in nanotechnology, wireless networks, biology, and the near eastern studies. He is a student member of ACM, IEEE, AIAA, and ASEE.

Appendix: 2D Matrix Equation

$$
\begin{array}{r|c|c|c|c|c|c|c|}
y_{pos}=0 & \underbrace{\begin{array}{r|c|c|c|}\cline{2-4} x_{pos}=0 & \text{B.C. }(y) & 0 & 0\\ \cline{2-4} & 0 & \ddots & 0\\ \cline{2-4} x_{pos}=\Delta n_x & 0 & 0 & \text{B.C. }(y)\\ \cline{2-4}\end{array}}_{n_x\text{ rows by }n_x\text{ columns}} & 0 & 0 & 0 & 0 & 0 & 0\\ \hline
 & \ddots & 0\cdots(n_x\text{ points})\cdots 0 & \ddots & 0 & \ddots & 0 & 0\\ \hline
y_{pos}=y & 0 & \varepsilon\begin{pmatrix}x\\ y-\Delta/2\end{pmatrix} & 0\cdots(n_x\text{ points})\cdots 0 & \underbrace{\begin{array}{r|c|c|c|c|c|}\cline{2-6} x_{pos}=0 & \text{B.C. }(x) & 0 & 0 & 0 & 0\\ \cline{2-6} & \ddots & \ddots & \ddots & 0 & 0\\ \cline{2-6} x_{pos}=x & 0 & \varepsilon\begin{pmatrix}x-\Delta/2\\ y\end{pmatrix} & -\begin{pmatrix}\varepsilon\begin{pmatrix}x\\ y+\Delta/2\end{pmatrix}+\varepsilon\begin{pmatrix}x+\Delta/2\\ y\end{pmatrix}\\ +\varepsilon\begin{pmatrix}x\\ y-\Delta/2\end{pmatrix}+\varepsilon\begin{pmatrix}x-\Delta/2\\ y\end{pmatrix}\end{pmatrix} & \varepsilon\begin{pmatrix}x+\Delta/2\\ y\end{pmatrix} & 0\\ \cline{2-6} & 0 & 0 & \ddots & \ddots & \ddots\\ \cline{2-6} x_{pos}=\Delta n_x & 0 & 0 & 0 & 0 & \text{B.C. }(x)\\ \cline{2-6}\end{array}}_{n_x\text{ rows by }n_x\text{ columns}} & 0\cdots(n_x\text{ points})\cdots 0 & \varepsilon\begin{pmatrix}x\\ y+\Delta/2\end{pmatrix} & 0\\ \hline
 & 0 & 0 & \ddots & 0 & \ddots & 0\cdots(n_x\text{ points})\cdots 0 & \ddots\\ \hline
y_{pos}=\Delta n_y & 0 & 0 & 0 & 0 & 0 & 0 & \underbrace{\begin{array}{r|c|c|c|}\cline{2-4} x_{pos}=0 & \text{B.C. }(y) & 0 & 0\\ \cline{2-4} & 0 & \ddots & 0\\ \cline{2-4} x_{pos}=\Delta n_x & 0 & 0 & \text{B.C. }(y)\\ \cline{2-4}\end{array}}_{n_x\text{ rows by }n_x\text{ columns}}
\end{array}
$$

$$
\underbrace{}_{\text{Coefficients of }\phi\text{: }n_x n_y\text{ rows by }n_x n_y\text{ columns}}
\qquad
\underbrace{\begin{bmatrix}
\phi\begin{pmatrix}0\\ 0\end{pmatrix}\\
\vdots\\
\phi\begin{pmatrix}\Delta n_x\\ 0\end{pmatrix}\\ \hline
\vdots\\ \hline
\phi\begin{pmatrix}x\\ y-\Delta\end{pmatrix}\\
\vdots\\
\phi\begin{pmatrix}x-\Delta\\ y\end{pmatrix}\\
\phi\begin{pmatrix}x\\ y\end{pmatrix}\\
\phi\begin{pmatrix}x+\Delta\\ y\end{pmatrix}\\
\phi\begin{pmatrix}x\\ y+\Delta\end{pmatrix}\\ \hline
\vdots\\ \hline
\phi\begin{pmatrix}0\\ \Delta n_y\end{pmatrix}\\
\vdots\\
\phi\begin{pmatrix}\Delta n_x\\ \Delta n_y\end{pmatrix}
\end{bmatrix}}_{\phi\text{: }n_x n_y\text{ elements}}
$$

$$
=\underbrace{\begin{bmatrix}\text{B.C.}\\ 0\\ \vdots\\ 0\\ \text{B.C.}\end{bmatrix}}_{n_x n_y\text{ elements}}
$$

B.C. means boundary condition **(two dimensional matrix equation)** **(8.1)**