Romina Hernandez

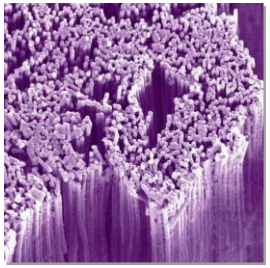
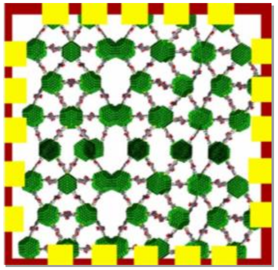
Modeling and Stimulation of Randomly Assembled Nanowire Networks

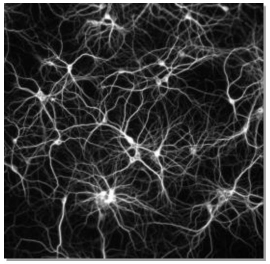
REU, August 11, 2017

Portland State University

Department of Electrical and Computer Engineering (ECE)

www.teuscher-lab.com | rehernandez@cpp.edu

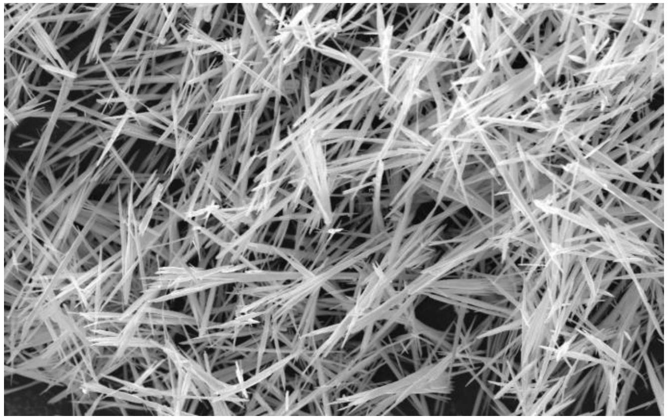


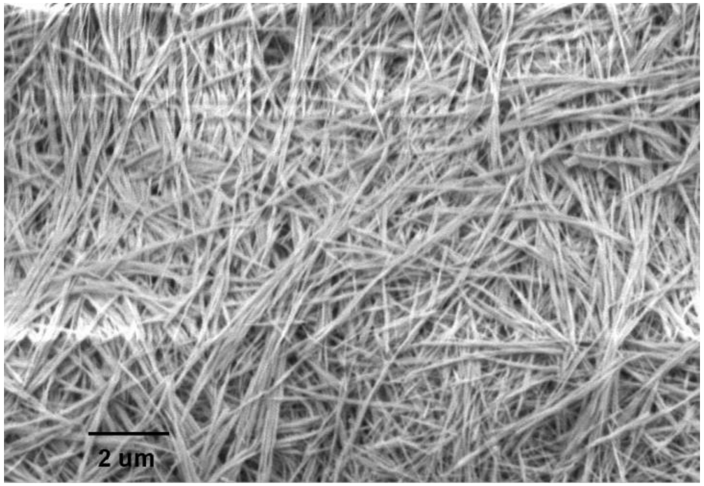
teuscher.:Lab
teuscher-lab.com
Emerging Computing Models and Technologies

Portland State UNIVERSITY

What are nanowires?

- **A nanoscale rod made of semiconducting material, used in miniature transistors and some laser applications. [7]**
- **A wire made from nanotubes. A nanowire allows a much lower current to flow compared to metal wire. [7]**
- **Nanostructures are considered the critical component in a wide range of potential nanoscale device applications [1]**





[1] Green, Joshua M., et al. "Controlled Fabrication of High-Yield CdS Nano Structures by Compartment Arrangement." 16 Apr. 2008.

[7] Teuscher, Christof et al. "The Power of Power-Laws: Or How to Save Power in SoC." IEEE, 2011. 1–6. *CrossRef*. Web. 10 Aug. 2017.

Why are nanowires important?

- **Critical component in a wide range of potential nanoscale device applications:**

1. **Performing computation (i.e., transistors) [2]**
2. **Connections (i.e., wires) [8]**
3. **Useful in microscopy [3]**
4. **Solar cells [4]**

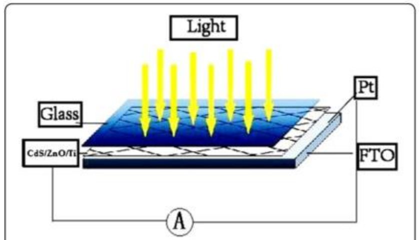


Figure 1 Schematic diagram of the CdS/ZnO/Ti nanostructured solar cell.

[4]

[2] Y. Li, L. Wei, X. Chen, R. Zhang, Z. Sui, Y. X. Chen, J. Jiao, and L. M. Mei, "Efficient PbS/CdS Co-sensitized Solar Cells Based on TiO2 Nanorod Arrays," Nanoscale Res Lett., 2013.

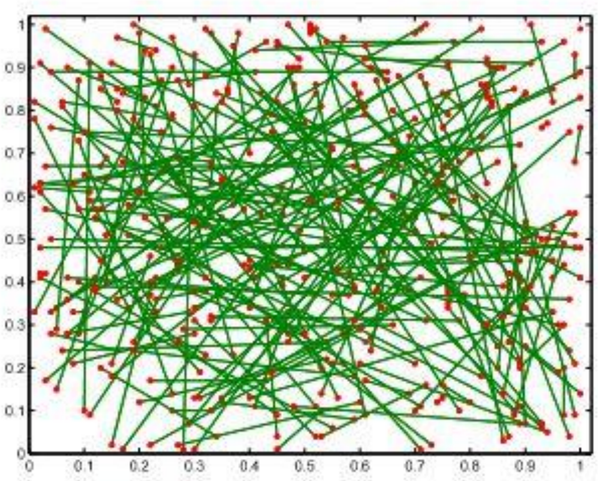
[3] J. Wu, M. Eastman, T. Gutu, M. Wyse, J. Jiao, S.-M. Kim, M. Mann, Y. Zhang and K. Teo, "Fabrication of Carbon Nanotube-based Nanodevices Using a Combination Technique of Focused Ion Beam and Plasma-enhanced Chemical Vapor Deposition," Appl. Phys. Lett., Vol.

[4] Wu, Cuncun, et al. "ZnO Nanosheet Arrays Constructed on Weaved Titanium Wire for CdS-Sensitized Solar Cells ." Nanoscale Research Letters, 11 Mar. 2014.

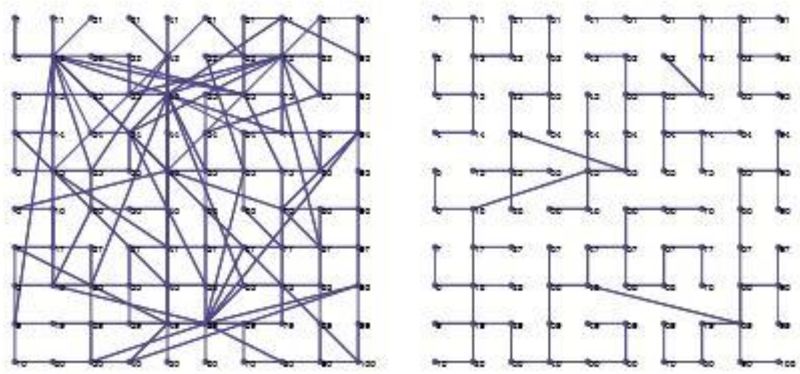
[8] Teuscher, Christof et al. "Wire Cost and Communication Analysis of Self-Assembled Interconnect Models for Networks-on-Chip." Proceedings of the 2Nd International Workshop on Network on Chip Architectures. New York, NY, USA: ACM, 2009. 83–88. ACM Digital Library. Web. 10 Aug. 2017. NoCArc '09.

Project Goal

- **The main goal of this project is to propose new models for self-assembled nanowire networks and to evaluate their performance and properties.**



[7]



[8]

[7] Teuscher, Christof et al. "The Power of Power-Laws: Or How to Save Power in SoC." IEEE, 2011. 1–6. *CrossRef*. Web. 10 Aug. 2017.
[8] Teuscher, Christof et al. "Wire Cost and Communication Analysis of Self-Assembled Interconnect Models for Networks-on-Chip." Proceedings of the 2Nd International Workshop on Network on Chip Architectures. New York, NY, USA: ACM, 2009. 83–88. ACM Digital Library. Web. 10 Aug. 2017. NoCArc '09.

MATLAB

- **MATLAB stands for “matrix laboratory”.**
- **This program, is specifically useful in this project because:**
    1. **MATLAB enables quicker computational results: for research it provides fast prototyping.**
    2. **Has many useful function toolboxes, so no need to reinvent new ones but use those to build upon what you want.**
    3. **Great plotting tools.**
    4. **Interactive matrix calculator.**

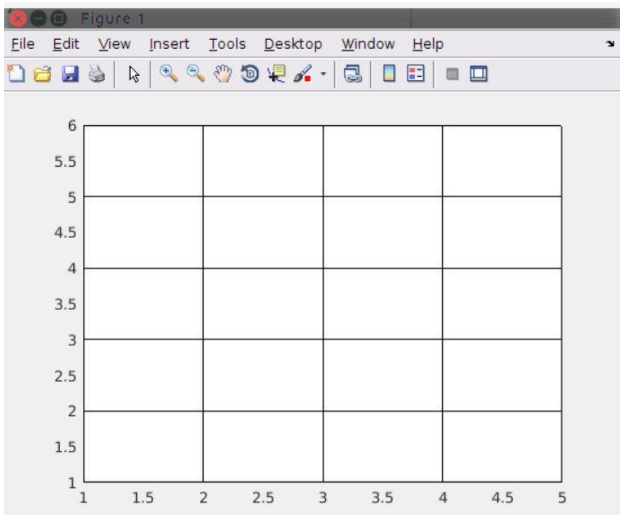
First Steps

- **Starting with MATLAB: learn all functions, inputs/outputs, variables, etc.**
- **Create step-by-step code that gives us what we want.**
- **Learn good programming skills such as keeping “visualization separate from calculations”.**
- **This will help us re use our data/codes.**

Methodology

1. **Generate grid of dimension NxM**
2. **Add one wire: plot a line segment**
3. **Generate multiple random points on the grid.**
4. **Use linear equation formula to find coordinate for two intersecting lines.**
5. **Plot function that will plot a marker on intersection given the formula.**
6. **Test functions for 2 single lines.**
7. **Create network of multiple random plotted wires.**
8. **Include formula function to determine more than one intersection at a time.**
9. **Receive all intersection points as a matrix.**

Early Code




```
%STEP 1:Grid Generator
%Generate a Grid x-by-y.
%-----------------------------------------------------------------%
function h1 = generateGrid(x,y)
close all;
grid=zeros(x+1,y+1);
hold on
h1=pcolor(grid);
%axis([0 x 0 y])
colormap white;
end
%-----------------------------------------------------------------%
%Syntax:    [handle]=gridGenerator(x coordinate, y coordinate)
%Inputs:
%   x & y coordinates.
%Outputs:
%   handle, is what it is named but when calling from the command window,
%   you can call it whatever just as long as it is being equal to the
%   function and you are giving it values from there. Point of the "handle"
%   is to be able to use it wherever.
%Example: handle=generateGrid(5,4)
%-----------------------------------------------------------------%
%Author: Romina Hernandez
%Email:minahernandez41@gmail.com
%Cal Poly Pomona
%Created: July  17
%Last Revised: July 25
```

```
%Coordinate Intersection Return Formula
%-----------------------------------------------------------------%
%This returns the coordinates of the intersections
%when the denominator is equal to 0, then there is this  {[]}, and the plot
%is not displayed.
%-----------------------------------------------------------------%
function [Px, Py] = getCoordinateFormula(x1, y1, x2, y2, x3, y3, x4, y4)
denom=(x1-x2).*(y3-y4)-(y1-y2).*(x3-x4);
Px=(((x1.*y2)-(y1.*x2)).*(x3-x4)-(x1-x2).*((x3.*y4)-(y3.*x4)))/denom;
Py=(((x1.*y2)-(y1.*x2)).*(y3-y4)-(y1-y2).*((x3.*y4)-(y3.*x4)))/denom;
if denom==0
    set(gcf, 'Visible', 'off');
    v=cell(1);
    disp(v);
end
end
%-----------------------------------------------------------------%
%Syntax:
%   [x,y value for the coordinate]=getCoord.(x-y values for 2 lines)
%Input:
%   the x's & y's for the lines on the grid.
%Output:
%   we will be getting the final coordinate, of intersection.
%Example:
%   testIntersection
%    3.5714
%    1.4286
%-----------------------------------------------------------------%
%Author: Romina Hernandez
%Email:minahernandez41@gmail.com
%Cal Poly Pomona
%Created: July  17
%Last Revised: July 25
```

```
%Point Intersection Drawn
%-----------------------------------------------------------------%
%This functin actually draws out the point where the lines intersect aka
%coordinates.
%-----------------------------------------------------------------%
function [h4] = plotIntersection(Px,Py)
plot(Px,Py,'ro', 'markersize',9,'Linewidth',2);
end
%-----------------------------------------------------------------%
%Author: Romina Hernandez
%Email:minahernandez41@gmail.com
%Cal Poly Pomona
%Created: July  17
%Last Revised: July 25
```

Main Function: Important Code

```
%Checking Intersections between Wires
%------------------------------------------------------------------------%
function [h3,intersection] = checkWireIntersections(rows,columns,numWires)
h3=generateGrid(rows,columns);
data=generateWireData(rows,columns,numWires);
        for i=1:numWires
            x1=data(i,1);
            y1=data(i,2);
            x2=data(i,3);
            y2=data(i,4);
            addWireToFigure(x1,y1,x2,y2);
        end

intersection=[];
k=0;%counter 'k' is first equal to zero, its initial state starts at 0.
        for i=1:numWires-1
             x1=data(i,1);
             y1=data(i,2);
             x2=data(i,3);
             y2=data(i,4);
                 for j=1:numWires-i
                         x3=data(i+j,1);
                         y3=data(i+j,2);
                         x4=data(i+j,3);
                         y4=data(i+j,4);
                         k=k+1;
                        [Px,Py]=getCoordinateFormula(x1,y1,x2,y2,x3,y3,x4,y4);
         if(((x1<=Px && x2>=Px)||(x1>=Px && x2<=Px))&&(x3<=Px && x4>=Px)||(x3>=Px && x4<=Px))
                disp(Px);
         else
                 Px=0;
                 Py=0;
         end
         if(((y1<=Py && y2>=Py)||(y1>=Py && y2<=Py))&&(y3<=Py && y4>=Py)||(y3>=Py && y4<=Py))
                    disp(Py);
          else
                 Px=0;
                 Py=0;
         end
         if(Px~=0 || Py~=0)
                plotIntersection(Px,Py);
                intersection(k,:)=[Px,Py];
          end
    end
end
```

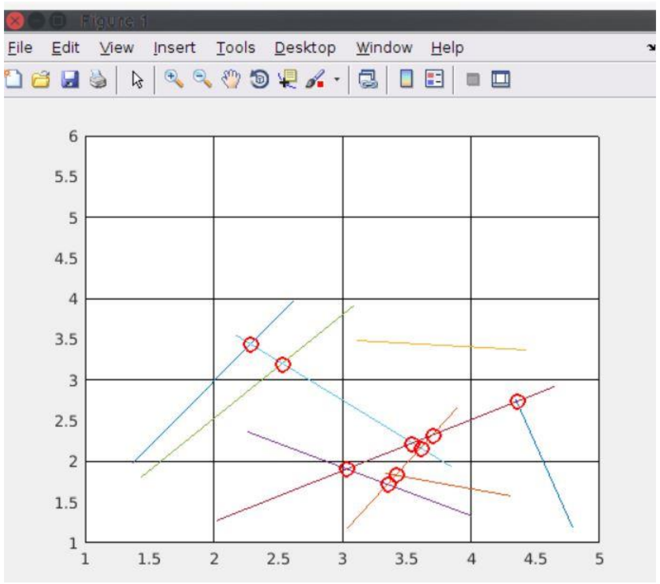
```
%Data Generator for Wires
%------------------------------------------------------------------------%
function [data] = generateWireData(rows, columns, numWires)
for i=1:numWires
        x1=1+(rows-1).*rand(1,1,1);
        y1=1+(columns-1).*rand(1,1,1);
        x2=1+(rows-1).*rand(1,1,1);
        y2=1+(columns-1).*rand(1,1,1);
        data(i,:)=[x1,y1,x2,y2];
end
end
%------------------------------------------------------------------------%
%Syntax:
%Inputs: rows, signify the y values. Columns represent the x values.
%Outputs: data, will used when finding the intersection.
%Example: data=generateWireData(6,5,7)
%data =
%
%    5.6137    4.2015    2.4297    3.1747
%    5.9239    3.8627    5.1948    2.7330
%    3.3531    3.2429    2.3455    3.9961
%    3.5194    3.5872    2.5387    1.5549
%    3.3779    2.4498    4.9406    4.1212
%    4.3426    1.5340    1.1078    3.2394
%    2.5041    4.7576    5.9045    2.1465
%------------------------------------------------------------------------%
%Author: Romina Hernandez
%Email:minahernandez41@gmail.com
%Cal Poly Pomona
%Created: July  17
```

```
%Coordinate Intersection Return Formula
%------------------------------------------------------------------------%
%This returns the coordinates of the intersections
%when the denominator is equal to 0, then there is this  {[]}, and the plot
%is not displayed.
%------------------------------------------------------------------------%
function [Px, Py] = getCoordinateFormula(x1, y1, x2, y2, x3, y3, x4, y4)
denom=(x1-x2).*(y3-y4)-(y1-y2).*(x3-x4);
Px=(((x1.*y2)-(y1.*x2)).*(x3-x4)-(x1-x2).*((x3.*y4)-(y3.*x4)))/denom;
Py=(((x1.*y2)-(y1.*x2)).*(y3-y4)-(y1-y2).*((x3.*y4)-(y3.*x4)))/denom;
if denom==0
    set(gcf, 'Visible', 'off');
    v=cell(1);
    disp(v);
end
end
%------------------------------------------------------------------------%
%Syntax:
%   [x,y value for the coordinate]=getCoord.(x-y values for 2 lines)
%Input:
%   the x's & y's for the lines on the grid.
%Output:
%   we will be getting the final coordinate, of intersection.
%Example:
%   testIntersection
%    3.5714
%    1.4286
%------------------------------------------------------------------------%
%Author: Romina Hernandez
%Email:minahernandez41@gmail.com
%Cal Poly Pomona
%Created: July  17
```

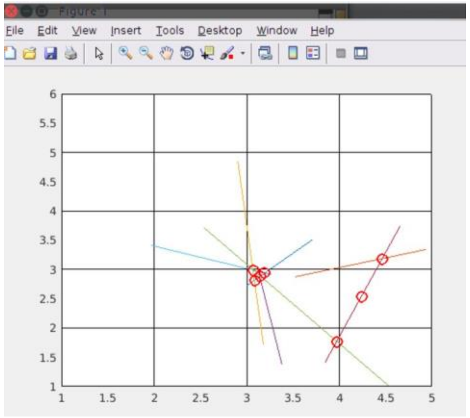
Results

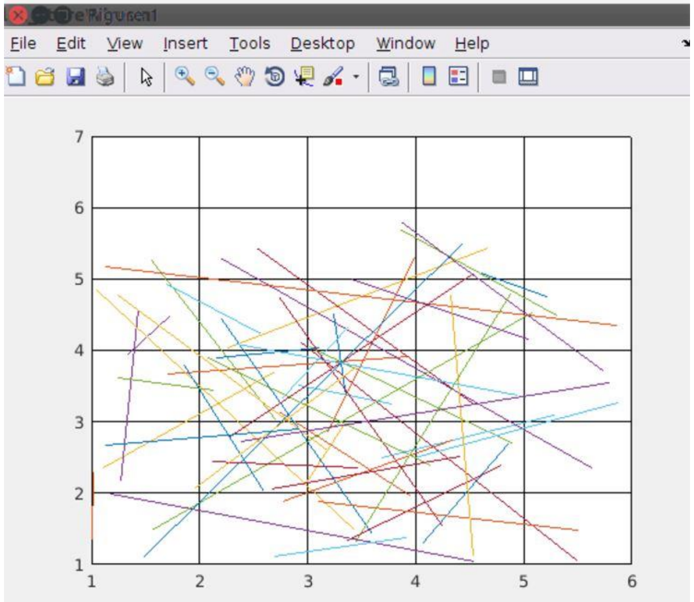
```
function [h3,intersection] = checkWireIntersections(rows,columns,numWires)
h3=generateGrid(rows,columns);
data=generateWireData(rows,columns,numWires);
for i=1:numWires
    x1=data(i,1);
    y1=data(i,2);
    x2=data(i,3);
    y2=data(i,4);
    addWireToFigure(x1,y1,x2,y2);
end
intersection=[];
k=0;
for i=1:numWires-1
     x1=data(i,1);
     y1=data(i,2);
     x2=data(i,3);
     y2=data(i,4);
     for j=1:numWires-i
         x3=data(i+j,1);
         y3=data(i+j,2);
         x4=data(i+j,3);
         y4=data(i+j,4);
         k=k+1;
 [Px,Py]=getCoordinateFormula(x1,y1,x2,y2,x3,y3,x4,y4);
         if(((x1<=Px && x2>=Px)||(x1>=Px && x2<=Px))&&(x3<=Px && x4>=Px)||(x3>=Px && x4<=Px))
                disp(Px);
         else
                 Px=0;
                 Py=0;
         end
         if(((y1<=Py && y2>=Py)||(y1>=Py && y2<=Py))&&(y3<=Py && y4>=Py)||(y3>=Py && y4<=Py))
                    disp(Py);
          else
                 Px=0;
                 Py=0;
          end

      intersection(k,:)=[Px,Py]%
      end
end
end
```

```
>> checkWireIntersections(5,4,9)
    3.1711

    1.9785

    3.4704

    2.3361

    1.8686

    1.4173

    3.2197

    2.6024

    2.1812

    3.7057

    2.5938

    3.2674

    2.0565

    2.6951

    1.9128

    1.5302

    2.0673

    1.9242
```




```
data =

    2.9673    3.9650    4.3572    3.0802
    2.7386    3.3444    1.7500    2.0486
    1.2223    1.9711    4.7747    2.7696
    4.4390    3.9454    2.7961    2.5788
    4.4171    2.7692    4.5202    1.0783
    2.6543    2.0811    3.1215    1.7882

>> |
```





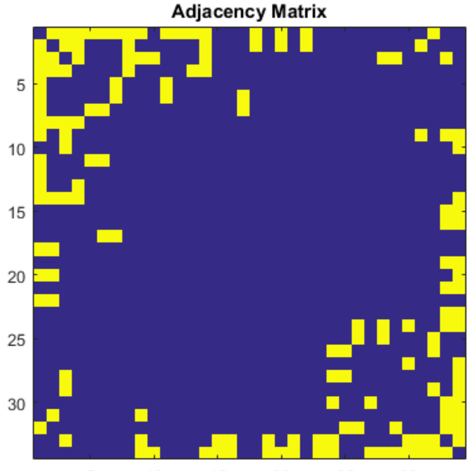
Conclusions/Future Work

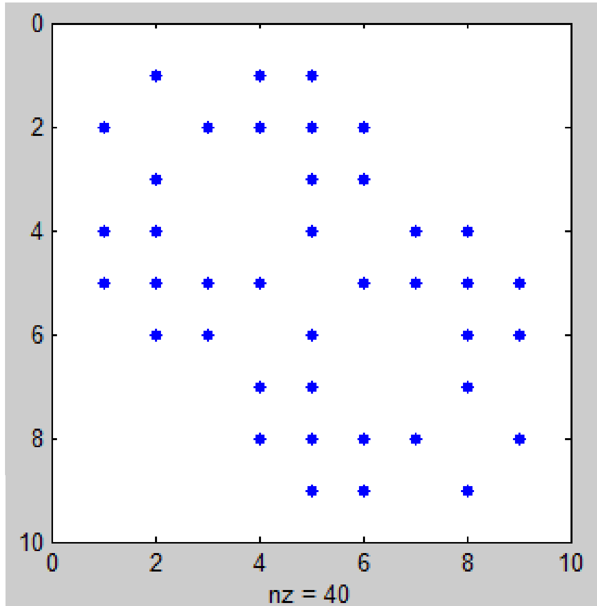
- **Approaching the nanowire modeling through the stimulation route assisted in our goal to analyze "wires".**
- **We worked with 2D plotting of wires which only allowed us to determine the efficiency of the way they interconnect through computer software.**

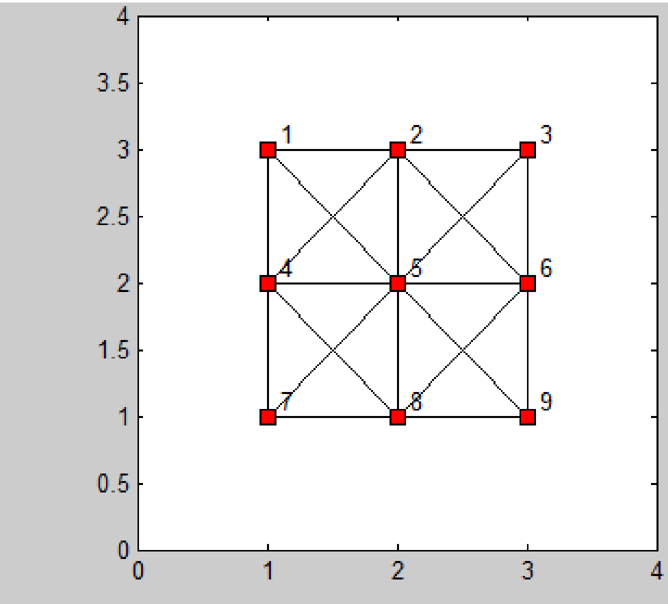
1. **Making the stimulations and models more realistic with respect to actual experiments.**

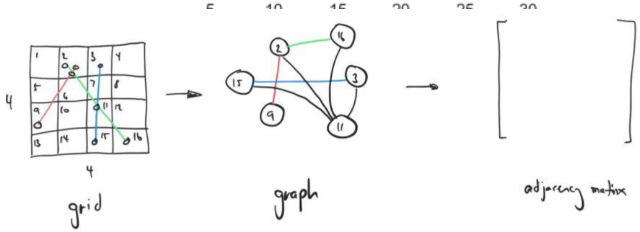
Future Work (2): Network Representation

- **Adjacency matrix.**



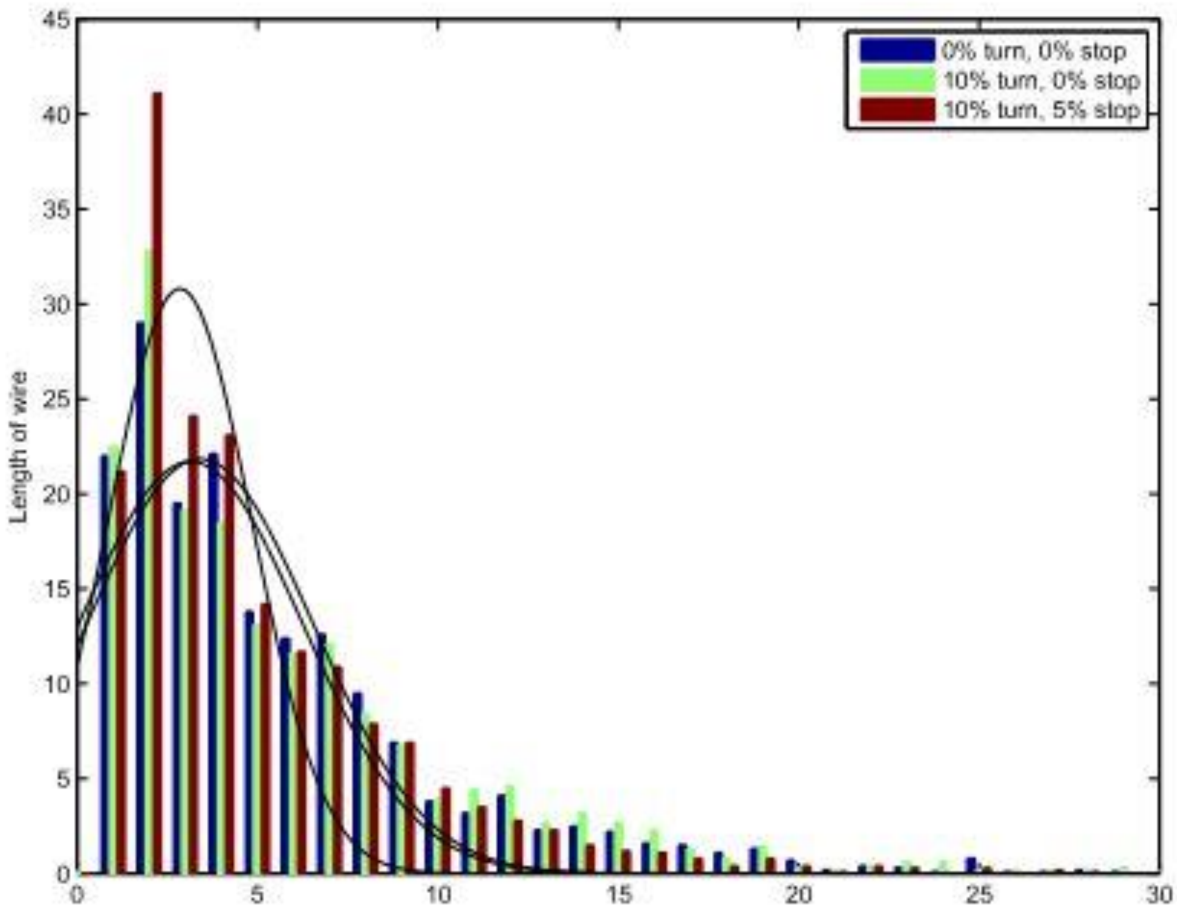






1 Determine cells where wires begin/end. These cells will become nodes of the graph.
2 Determine cells in which wires intersect. These cells will also become nodes in the graph.
3 Connect the "intersection cells" with the cells that the wires are connected to in the graph.
4 Represent the graph as an adjacency matrix.

Future Work (3): Gaussian Model



- **Translating the nanowire network that forms on the grid into a graph using the Gaussian Model to determine the distance proportional distribution for the rewiring. [7]**

[7] Teuscher, Christof et al. "The Power of Power-Laws: Or How to Save Power in SoC." IEEE, 2011. 1–6. *CrossRef*. Web. 10 Aug. 2017.

Acknowledgements

- **National Science Foundation**
- **REU Summer Program**
- **Dr. Jun Jiao**
- **Dr. Christof Teuscher**
- **Wesley Chavez**
- **Sai Kiran Chirupally**

References

[1] Green, Joshua M., et al. "Controlled Fabrication of High-Yield CdS Nano Structures by Compartment Arrangement." 16 Apr. 2008.

[2] Y. Li, L. Wei, X. Chen, R. Zhang, Z. Sui, Y. X. Chen, J. Jiao, and L. M. Mei, "Efficient PbS/CdS Co-sensitized Solar Cells Based on TiO2 Nanorod Arrays," Nanoscale Res Lett., 2013.

[3] J. Wu, M. Eastman, T. Gutu, M. Wyse, J. Jiao, S.-M. Kim, M. Mann, Y. Zhang and K. Teo, "Fabrication of Carbon Nanotube-based Nanodevices Using a Combination Technique of Focused Ion Beam and Plasma-enhanced Chemical Vapor Deposition," Appl. Phys. Lett., Vol. 91, 173122-3 (2007).

[4] Wu, Cuncun, et al. "ZnO Nanosheet Arrays Constructed on Weaved Titanium Wire for CdS-Sensitized Solar Cells ." Nanoscale Research Letters, 11 Mar. 2014.

[5] L. Lampert, A. Barnum, S. Smith, J. Conley Jr., J. Jiao, "Phase Transitions and In Situ Dynamics of Crystal Grain Formation of Alumina Nanotubes Templated by Vertically Aligned Carbon Nanotubes," RSC Adv., Vol. 5, pp. 68251-68259 (2015).

[6] L. Lampert, B. Timonen, S. Smith, B. Davidge, H. Li, J. F. Conley Jr., J. D. Singer, and J. Jiao, "Amorphous Alumina Nanowire Array Efficiently Delivers Ac-DEVD-CHO to Inhibit Apoptosis of Dendritic Cells", Chem. Comm., 50, pp. 1234-1237 (2014).

[7] Teuscher, Christof et al. "The Power of Power-Laws: Or How to Save Power in SoC." IEEE, 2011. 1–6. *CrossRef*. Web. 10 Aug. 2017.

[8] Teuscher, Christof et al. "Wire Cost and Communication Analysis of Self-Assembled Interconnect Models for Networks-on-Chip." Proceedings of the 2Nd International Workshop on Network on Chip Architectures. New York, NY, USA: ACM, 2009. 83–88. ACM Digital Library. Web. 10 Aug. 2017. NoCArc '09.