Version 1.0

10/17/2025

Documentation for ENCoRP

# Reference publication

The ENCoRP software is described in detail in our publication “ENCoRP: A Tool to Compute and Visualize an Electrical Network’s Conductive and Radiative Propagation”, which is openly available via IEEE Access. The paper discusses the underlying models and equations (including some derivations), numerical limitations, in addition to verifying results and offering an example of the tool applied to a hybrid wired-wireless system. You can access the paper through the link below:

<https://doi.org/10.1109/ACCESS.2025.3612839>

We would appreciate if users cite this paper when using the software.

# software requirements

## matlab

The GUI was designed in and has primarily been tested against MATLAB 2022b. Note that if exported to a standalone application, MATLAB may no longer be required. While you may attempt to run ENCoRP with alternative versions of MATLAB, various functionality may be limited, deprecated, or unavailable. Minimal testing has been performed with respect to other versions.

## Python

Although we have mainly operated with Python 3.9, we suspect any version 3.8 or higher should be compatible.

Importantly, having an instance of Python on your machine is insufficient to run ENCoRP unless it has been properly linked with MATLAB. While having undergone limited testing, the *Run\_ENCoRP* function attempts to search for and link an appropriate Python environment. You will be alerted to the success or failure of both the search and the linkage. If the link succeeds, ENCoRP will automatically launch.

## COMSOL

While MATLAB and Python are the only requirements to utilize the majority of the ENCoRP interface, COMSOL is required to generate electric and magnetic field predictions. Users do not need COMSOL if they are only interested in conductive predictions.

***Note that interfacing with COMSOL has only been tested on Linux machines.***

As of the release of this documentation, we have only used ENCoRP with COMSOL Multiphysics 6.1. Additionally, we use COMSOL LiveLink for MATLAB to transfer the network geometry and conductive predictions.

To establish communication between MATLAB and COMSOL for the purposes of ENCoRP, the following steps are required:

1. There is a file in the ENCoRP directory named “COMSOL\_Path”.  
   Edit this one-line file to match the path to the mli sub-directory in the multiphysics sub-directory of your COMSOL 6.1 installation folder. For example, the path might look like this:  
      
   /usr/local/comsol/comsol61/multiphysics/mli
2. When launching COMSOL LiveLink for MATLAB for the first time, you may be prompted to create a username and password to interact with the local server. If this is the case, ENCoRP will be unable to proceed until credentials have been created. To create credentials, attach to the screen created by ENCoRP. You can do so by opening a terminal and typing:

screen -r comsolserverENCoRP  
  
Once attached, you should be prompted by COMSOL to create a username and password via the terminal. After your information has been saved, and if you elect for the server to remember your information, you shouldn’t have to repeat these steps or enter your information again. Although unsupported beyond our own experience; since creating account information years ago, we have not had to attach to the screen nor recall the username/password.

1. Some users may find that COMSOL is unable to accept MATLAB functions as input. We have attempted to overcome this limitation by appending additional flags in the COMSOL server startup request. In particular, the following flag is automatically appended and typically enables COMSOL to accept MATLAB functions as input:  
     
   comsol mphserver **-allowexternalmatlab**While this has worked for our team, it has not been thoroughly tested. We note that in the case of failure, it is our understanding that there are other means of permitting COMSOL to accept MATLAB functions, such as through the COMSOL settings directly.

Otherwise, users may encounter other issues when using COMSOL via ENCoRP. For example, inadequate licensing will cause the ENCoRP-COMSOL communications to hang without error. This and other issues are sometimes explicitly described in the screen. We recommend reattaching as a first measure in debugging an ENCoRP-COMSOL connection issue.

We separately note that the success of the COMSOL simulation may depend on various parameters, such as domain size, as described in the publication. The COMSOL progress bar that appears may struggle or fail to converge, indicating that some simulation details may need to be reconsidered. Please refer to the publication for more information.

### Version Compatibility

| Software | Version | Compatibility | Reason |
| --- | --- | --- | --- |
| Python | < 3.8 | No | Incompatible with MATLAB 2022b, although 3.7 might work with MATLAB 2022a |
|  | 3.8 | Suspected yes |  |
|  | 3.9 | Yes |  |
|  | > 3.9 | Suspected yes |  |
| MATLAB | < 2022a | Suspected no | Loss of functionality in older versions |
|  | 2022a | Suspected yes |  |
|  | 2022b | Yes |  |
|  | > 2022b | Suspected yes |  |
| COMSOL | 6.1 | Yes |  |
|  | ≠ 6.1 | Unknown | Untested |

# Using ENCoRP

It is recommended that first time users follow the visual, guided example featured later in this document. Alternatively, a non-visual Overview of Process is also included.

Once the required software has been properly installed and linked, execute the *Run\_ENCoRP* function in MATLAB to open the ENCoRP GUI. The function will attempt to ensure correct versions and linkages before starting up the interface.

## Troubleshooting

The GUI will output messages throughout its use that serve to guide, warn, and correct the user during the creation of a network. The denominations of messages are shown below.

### Message Denominations

| Color | Type | Purpose | Effect |
| --- | --- | --- | --- |
| Green | Guidance | Indicates the steps the user should take at the present | None |
| Yellow | Warning | Indicates a selection that may be accidental or unavailable, but will not prevent the user from continuing | None |
| Red | Error | Indicates a mistake or invalid selection that must be corrected before the user can continue | Locks progress until correction |

## Overview of process

The following outlines the steps the user will need to take to navigate the various stages of the GUI.

### Geometry Definition

The interface initializes with the user in the Geometry Definition tab. To proceed, the user will need to define a legal network composed of **n** nodes and **n-1** wires. Use the subtabs *Nodes* and *Wires* to define the geometry appropriately.

In the *Nodes* subtab, the user must specify the (x,y,z) coordinate locations of the nodes in the network. By nodes, we mean transmitters, receivers, junction boxes, outlets (loads), etc.

In the *Wires* subtab, the user must specify the *Source Node* and *Destination Node* between which a wire will be created with a press of the *Add Wire* button.

Note that while redundant nodes are permitted, redundant wires are not. This can be circumnavigated by creating multiple nodes at a single location to define equivalent but different wires.

Additionally, users need not concern themselves with the directionality of the wires, which are corrected internally during model execution.

When a network has been fully connected, the *Node Definition* tab will unlock.

### Node definition

If all the controls are disabled, then the troubleshooting box in the bottom left will indicate the error with the network. Return to the *Geometry Definition* tab to make corrections. If controls are available and the text is green, proceed to defining the nodes.

To proceed to the *Wiring Definition* tab, the user must define all of the nodes. If the node is an outlet or receiver, the user must further specify the associated load type and parameters. The node types are tabulated below.

| Node Type | Description | Restrictions | Extra |
| --- | --- | --- | --- |
| Derivation Box | Derivation box/junction box. A node that often splits into multiple outgoing wires | Must have at least two wires connecting to the node |  |
| Receiver | The node at which the receiver is located | Must have exactly one in the network. Must be connected to a single wire | The user must select the load type and enter load parameters. The user will be constrained to reasonable values for the load parameters |
| Transmitter | The node at which the transmitter is located | Must have exactly one in the network. Must be connected to a single wire |  |
| Outlet | The location of a terminating or shunt load (determined internally) | Cannot branch into more than one outgoing wire | The user must select the load type and enter load parameters. The user will be constrained to reasonable values for the load parameters |

Once all nodes have been defined (successful definition is indicated by the lack of an asterisk), the *Wiring Definition* tab will become available.

### Wiring definition

Similar to before, if there is any issue with the node definitions, the user will be prompted to return to the *Node Definition* tab with the specific error having been described in the troubleshooting panel.

In this section, the user is required to choose the number of conductors: 4 = three-phase or 2 = single-phase. With this selection, a corresponding wire arrangement will be enforced and the user must define the *Wire Length, Conductivity* for each conductor, *Permittivity* for each insulator*,* conductor radii (denoted *Wire Radii*), and *Separation Distance*. Note that the wire length is adjustable to account for slack that may be present in a physical network.

Rather than repeat for each conductor and cable, the user may instead opt to use the *Apply to All Wires* and *Apply to All Conductors* buttons.

Note that the wire radii and separation distance must not violate the physical geometry indicated by the wire arrangement.

Once these values have been defined, the *Run Model* tab will unlock.

### Run Model

Initially, the only available functionality will be the *Save Model* button. Doing so will unlock the simulation details panel. Define the *Frequency*, *Injection,* and *Wire Resolution* before pressing the *Simulate* button. The injection may be either voltages or currents which are injected at the transmitter onto each of the corresponding phase wires. The wire resolution corresponds to the simulation resolution along the wires. Notably, the simulation accuracy is independent of the wire resolution because the transmission along a wire is determined analytically, meaning that an increase in intermediary points does not offer any computational benefits, although it does enhance the resolution of the visualization.

Once the simulation has completed, the user will be notified in the troubleshooting panel and the *Visualize Results* tab will become available.

### Visualize Results

At this point the user can visualize the results.

Simply click on the desired frequency and the network will be illuminated accordingly. The user may press the *Phase* and *Neutral* buttons to view the various conductors, in addition to the voltage or current via the *Current* and *Voltage* buttons. Users can export the results in a topological or spatial format, as well as save graphics in the form of images or animated GIFs using the *Export Data*, *Generate PNG*, or *Generate Animated GIF* buttons, respectively.

Importantly, there is a second subtab labeled *Transmission Parameters*. This tab visualizes the various transmission parameters, including ABCD, Z, and Y, in addition to S-parameters based on user selection. Note that a characteristic impedance is required when converting to scattering parameters. Otherwise, there is functionality to view specific elements of the transmission parameters, as well as exporting the data or saving the visuals as an image.

The *E&M Fields* tab unlocks alongside the *Visualize Results* tab.

### E&M Fields

This tab features a number of parameters that will define, in conjunction with the network topology and conductive predictions, the COMSOL models to predict the electric and magnetic fields associated with the network and injection.

With the left-most entry field, the user can choose a *Frequency* of interest. Note that this only applies to the remote simulation, whereas exporting to a COMSOL session will include all frequencies. The user can also define the *Domain Bounds* for the simulation, as well as the *Quiver Density* for the quiver plot, which is the initial visualization option. The user must then choose either to compute the *Electric* or *Magnetic* field.

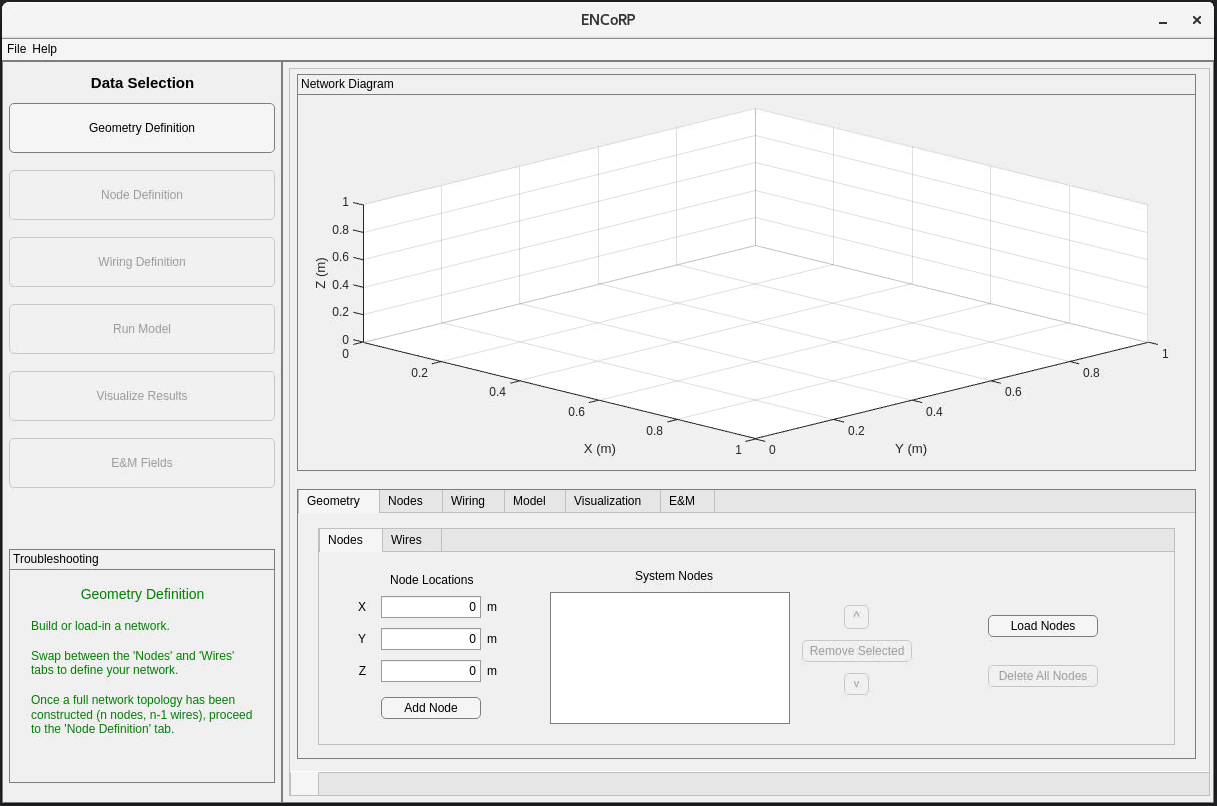
The *Simulate (Remotely)* button remotely builds and executes a COMSOL model, then displays the results in the ENCoRP interface as a quiver plot. The *Export to COMSOL Session* button will build and execute the COMSOL model, then open a COMSOL session containing the simulation setup and results for the user to display or manipulate using COMSOL functionalities.

Illustrating this process, the following pages feature a visual, guided example.

# visual Guided example

The purpose of this example is to guide first time users through building and evaluating a network.

If acceptable versions of MATLAB and Python are available and linked, you should be looking at the following interface after executing *Run\_ENCoRP.m*:

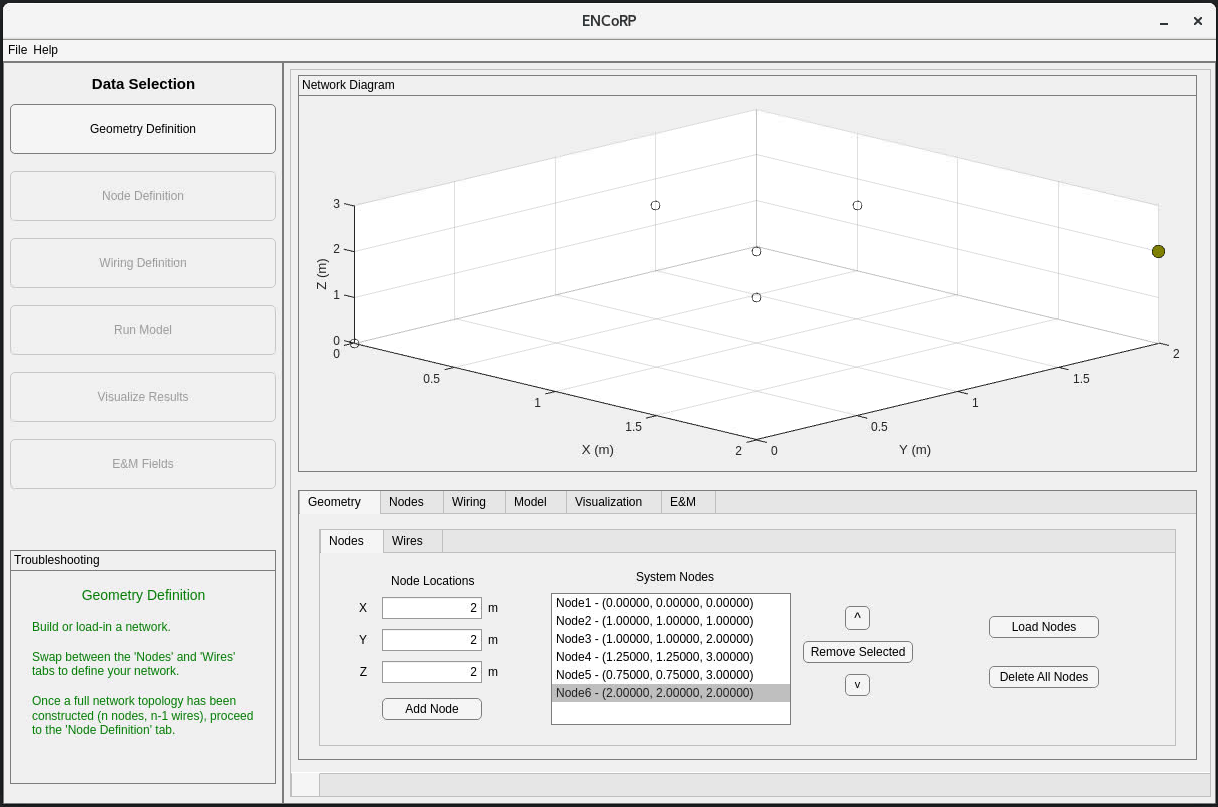


Let’s begin by making some nodes!

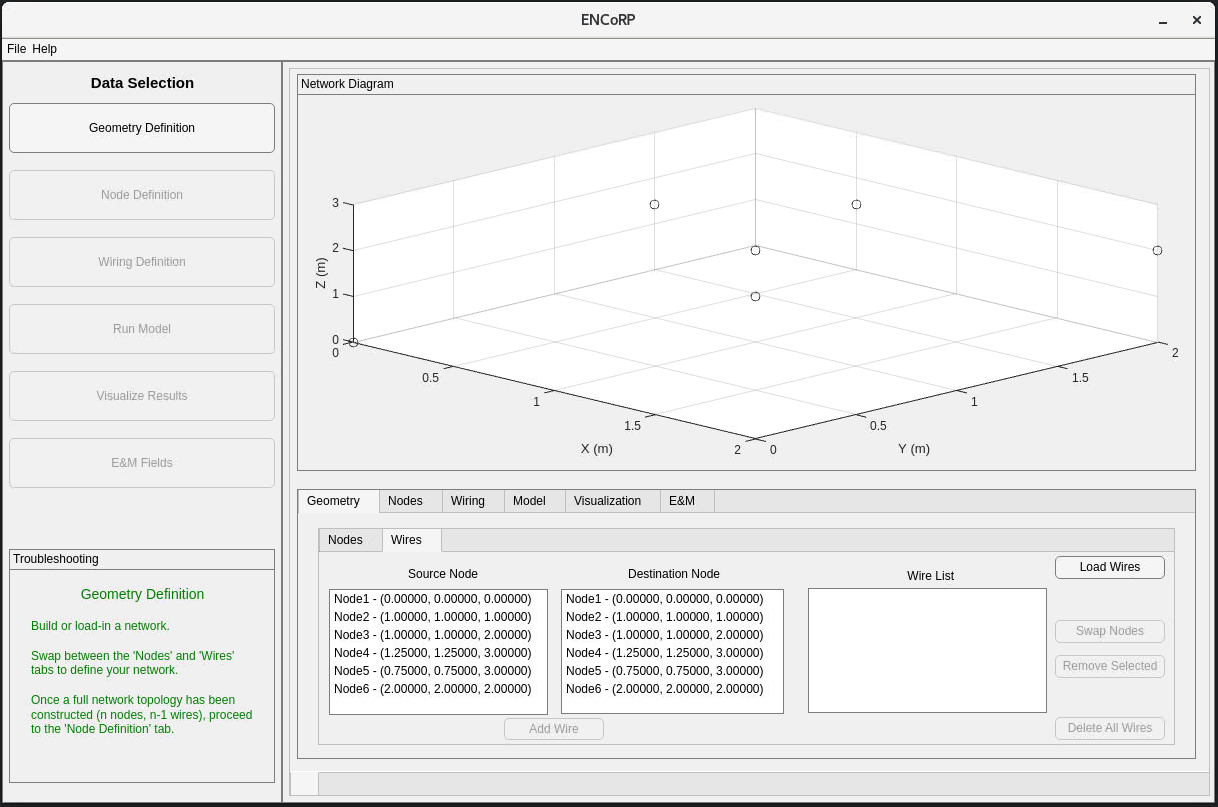
Add the following nodes to your network, in the order shown below. Although ordering does not matter, it will make things easier to follow during the example.

1. (0, 0, 0)
2. (1, 1, 1)
3. (1, 1, 2)
4. (1.25, 1.25, 3)
5. (0.75, 0.75, 3)
6. (2, 2, 2)

Your *System Nodes* box should look like the following (note, you can freely swivel the direction of the graphic at any time):



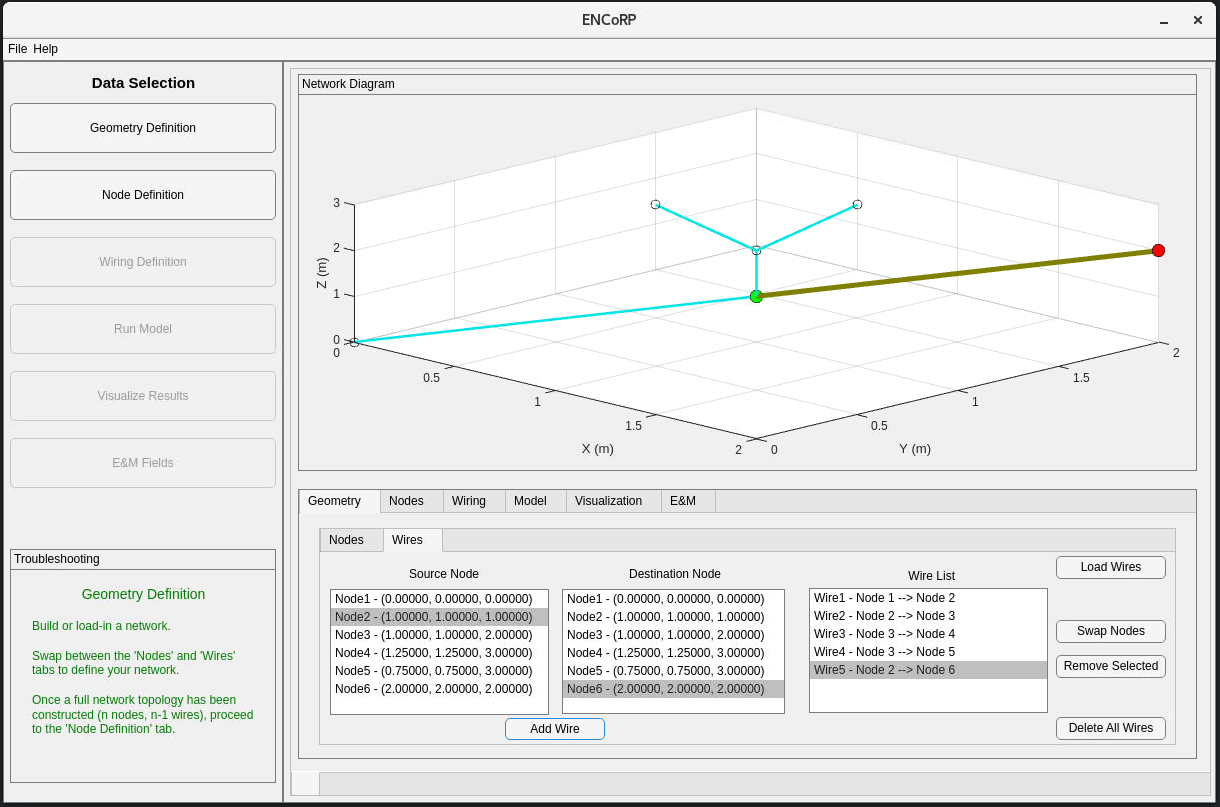
Next, click on the *Wires* tab just above *Node Locations*. Your screen should now look like this:



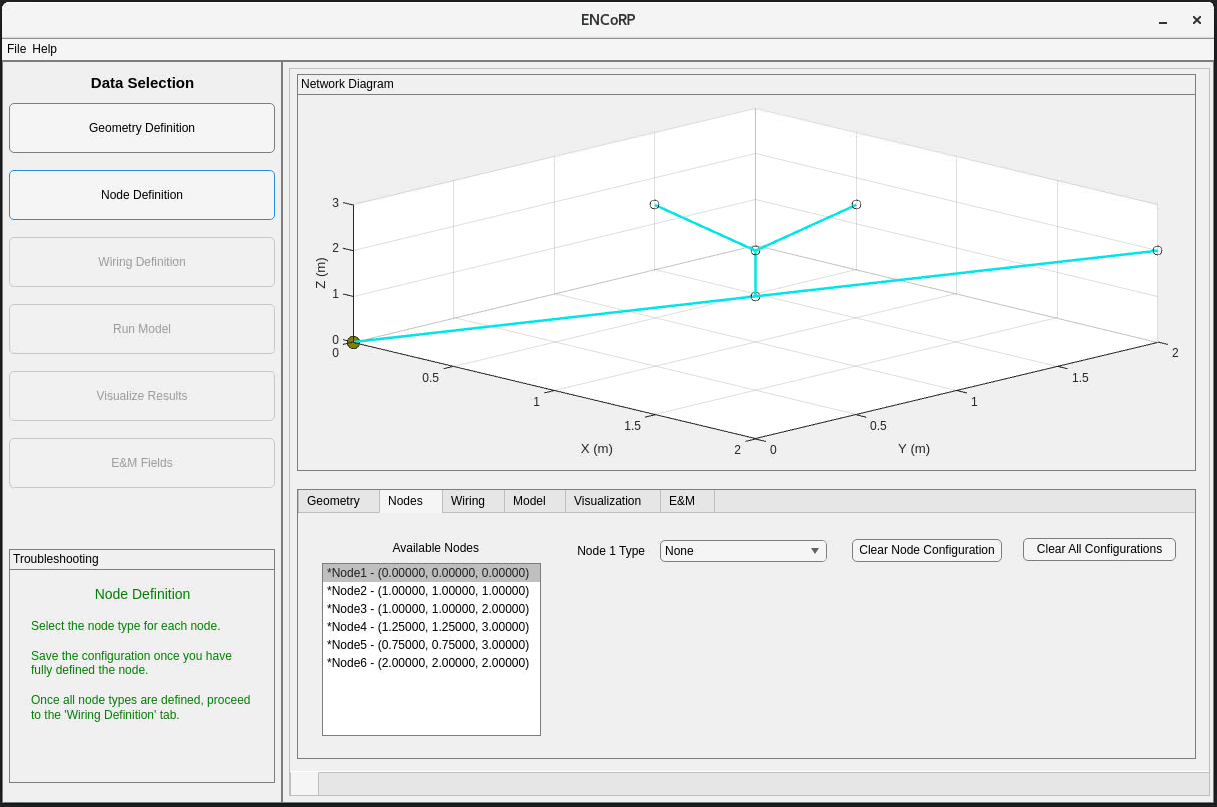
Next, select Nodes in the *Source Node* and *Destination Node* boxes and add wires in the following manner.

1. Node 1 -> Node 2
2. Node 2 -> Node 3
3. Node 3 -> Node 4
4. Node 3 -> Node 5
5. Node 2 -> Node 6

Your *Wire List* box should now be populated with 5 wires and look like this:



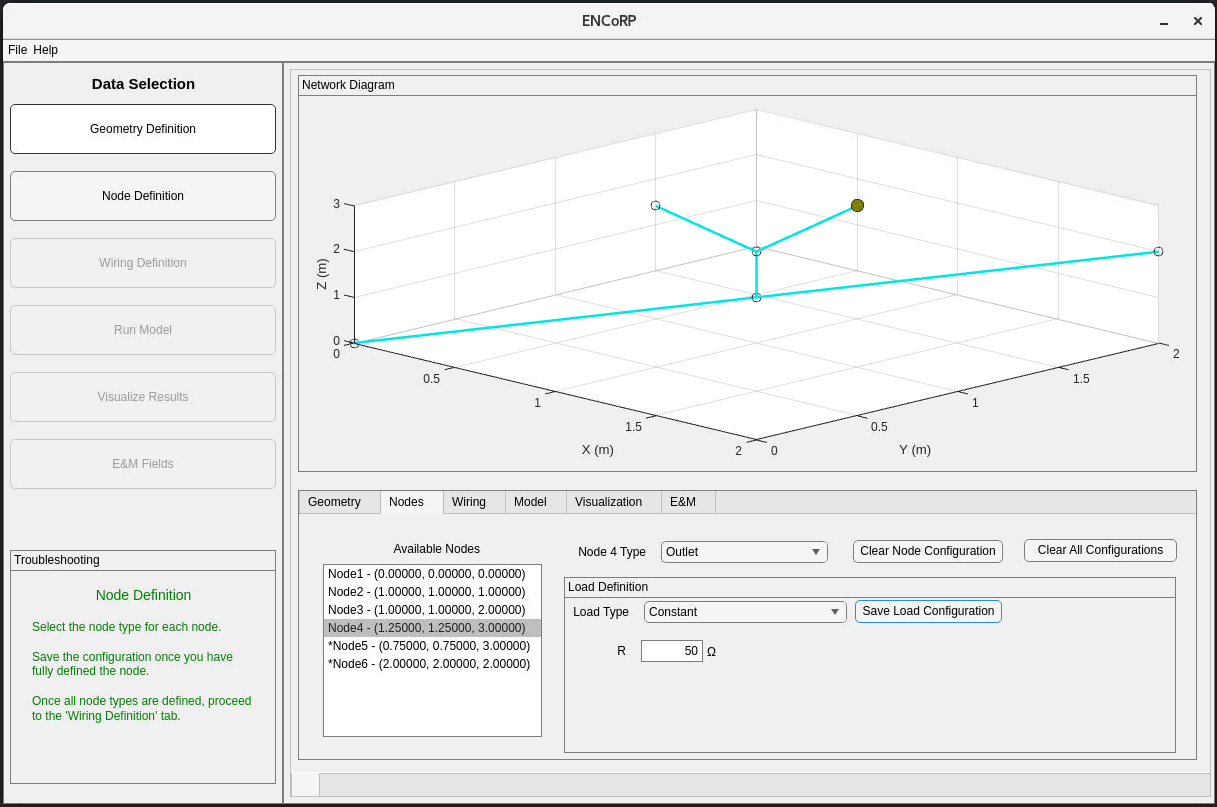
Having defined a network, lets proceed to the second tab *Node Definition* in the top left. If there is any issue with our network, error notifications would appear here and direct us back to the *Geometry Definition* tab. Your screen should now look as follows:



Let’s work through and define all the nodes using the *Node X Type* drop down menu. Define the nodes as follows.

1. *Node1*: *Transmitter* then press *Save Node Configuration*
   1. If done correctly, the asterisk by *Node1* in the *Available Nodes* box should disappear
2. Select *Node2* in the *Available Nodes* box
   1. *Node2*: *Derivation Box* then press *Save Node Configuration*
3. Select *Node3*, *Node3*: *Derivation Box* then press *Save Node Configuration*
4. Select *Node4*, *Node4*: *Outlet* and a panel should appear
   1. In the panel, select *Constant* in the *Load Type* drop down menu
   2. Define R = 50 Ω then press *Save Load Configuration*

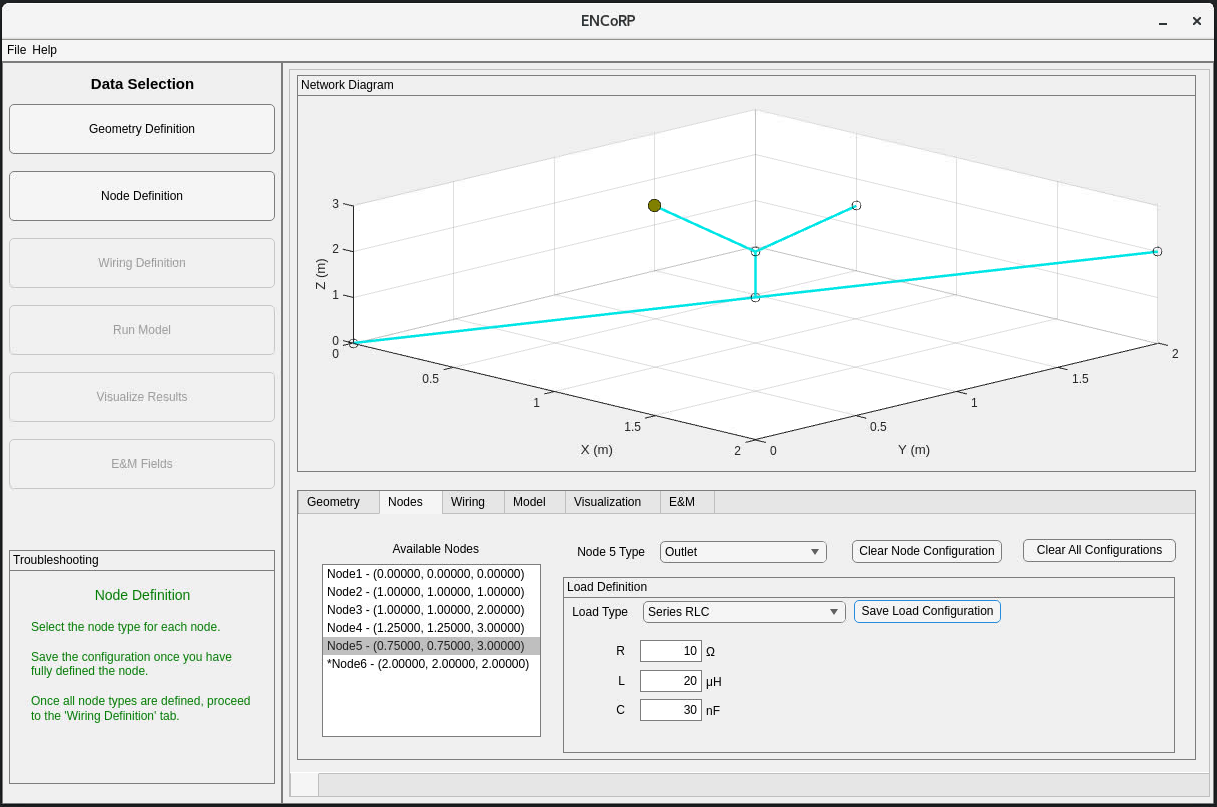
At this point, your screen should look like this:



Continuing,

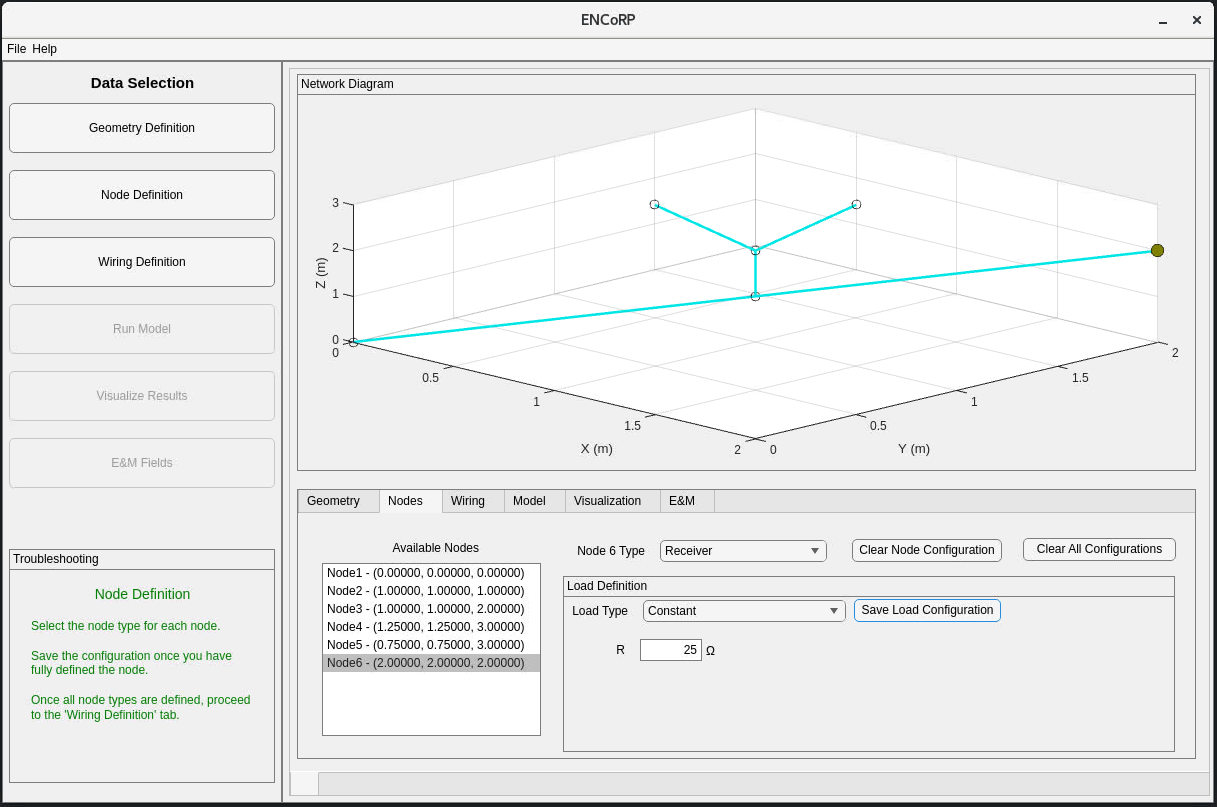
1. Select *Node5*, *Node5*: *Outlet*
   1. Pick any *Load Type* other than *Custom* and provide values for the available parameters
      1. If you’re unsure of what values to enter, hover the fields to see typical ranges
   2. *Save Load Configuration*

As an example, here’s a possible load you could define:



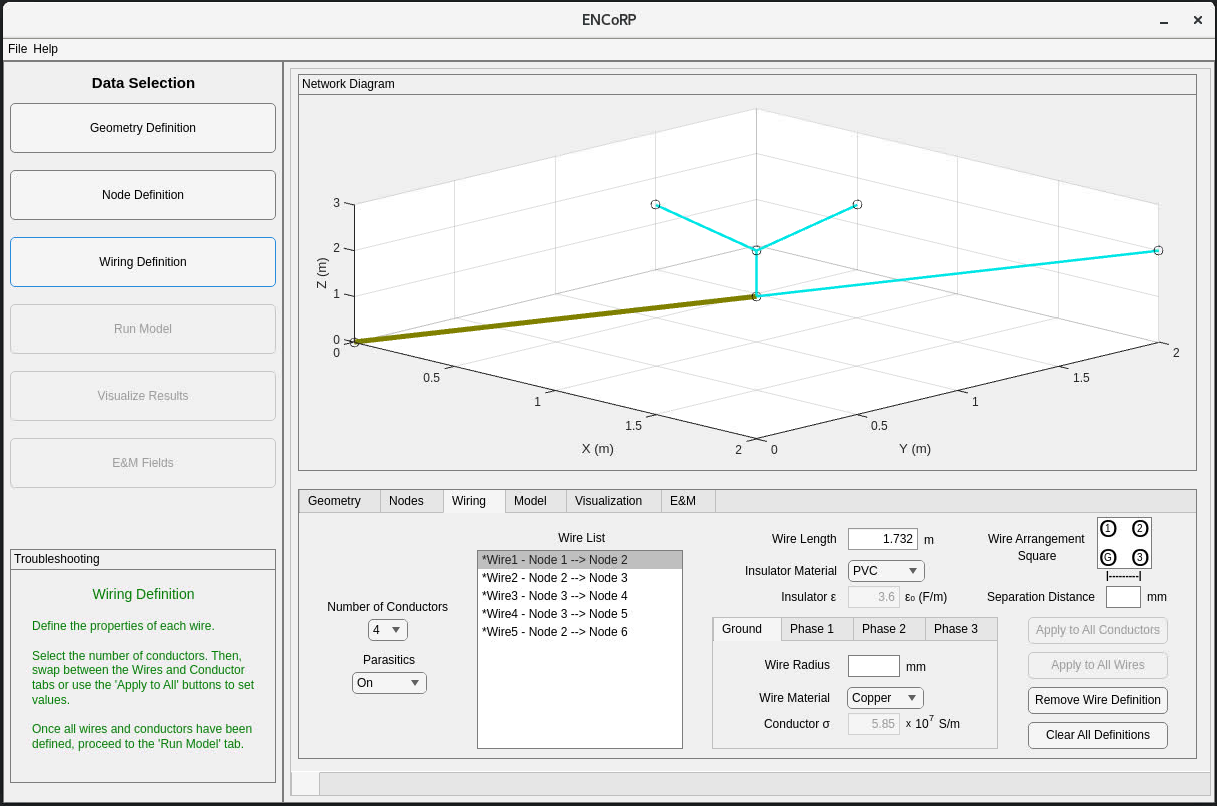
1. Select *Node6*, *Node6*: *Receiver*
   * 1. Similar to outlets, receivers are defined with a load
     2. Feel free to define the receiver load however you’d like
     3. *Save Load Configuration*

Note that your progress will be saved if you save without having defined all the load parameters. Here’s what you might be seeing after defining and saving *Node6*:



As you may have noticed, the *Wiring Definition* button in the top left is no longer inaccessible. Proceed to the *Wiring Definition* tab. If you defined any nodes improperly, the *Wiring Definition* tab panels will be locked and the troubleshooting window in the bottom left will indicate what changes need to be made to the node definitions. If this is the case, return to the *Node Definition* tab and correct accordingly.

You should now be looking at the following:

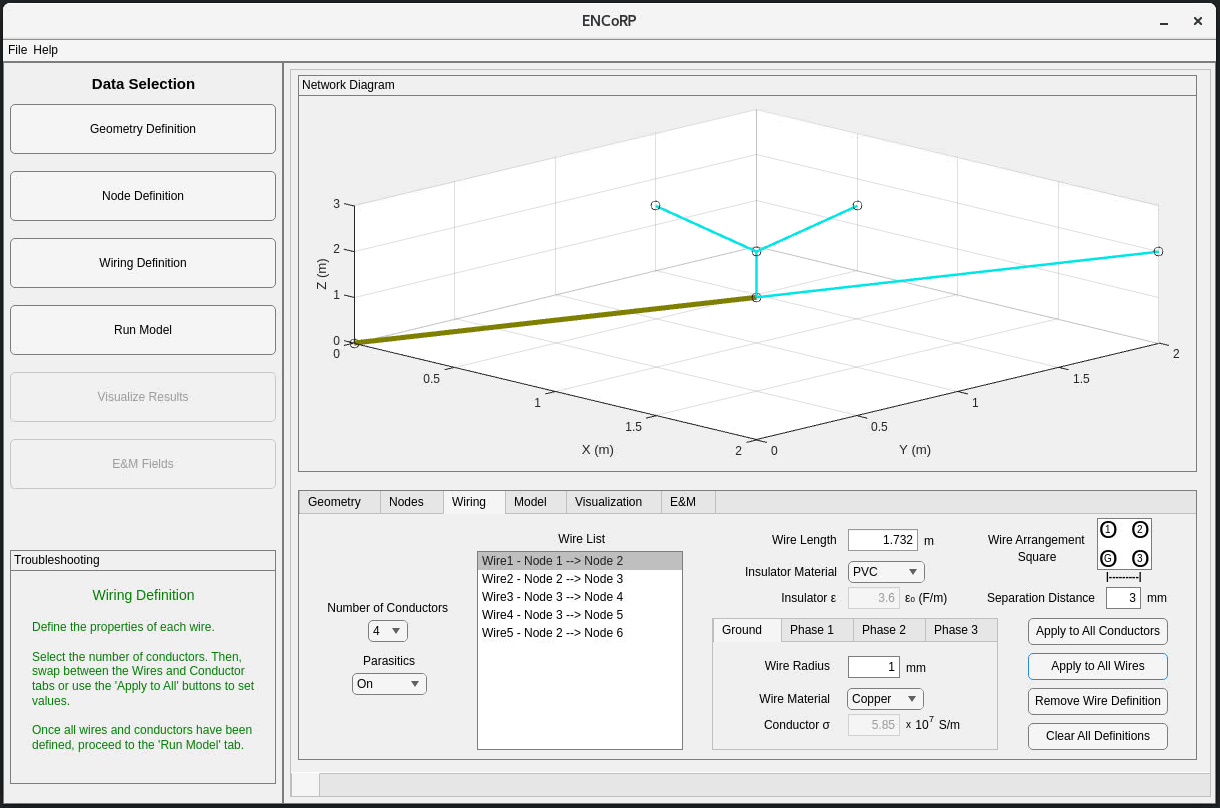


Locate the *Number of Conductors* drop down menu on the left, which is initially set to 4. This determines the phase of the network; that is, 4 = three-phase + neutral/ground and 2 = single-phase + neutral/ground. Feel free to change this to 2, although in this example we will proceed with the value set to 4.

Next, let’s define the wire properties.

Typically, wires in a network share the same properties, or there are a small number of subsets of wires that share properties. While the user may define the conductors on each wire however they’d like, let’s assume for this example that all wires have the same material and cross-sectional properties.

Enter 1 for *Wire Radius* and 3 for *Separation Distance*. Then, press the *Apply to All Conductors* button followed by the *Apply to All Wires* button. This fully defines the properties for all wires and should remove the asterisks next to all wires from the *Wire List* box on the left. Your screen should now look like this:

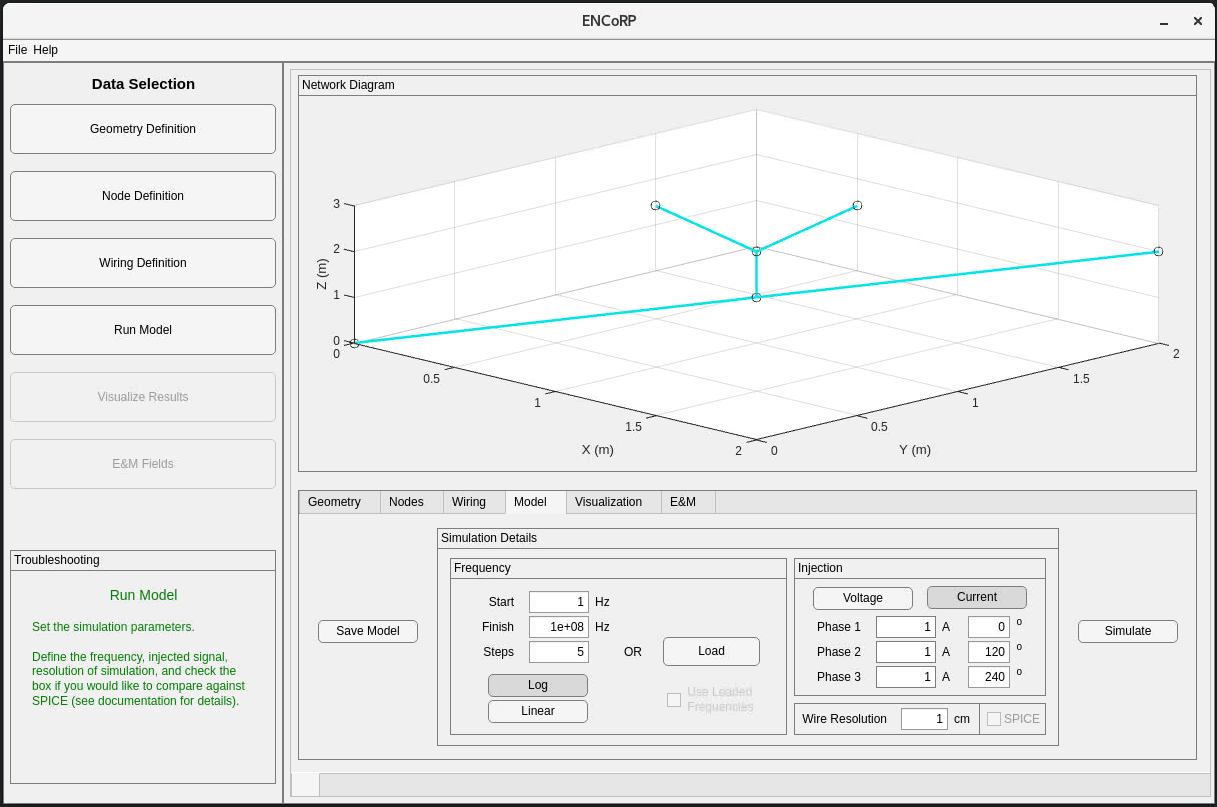


The *Run Model* tab should now be accessible. Proceed and press the only available button, *Save Model*.

You can now edit values in the *Simulation Details* panel.

Let’s start by changing the number of frequency *Steps* to 5, leaving the *Start* and *Finish* frequencies alone. Next, let’s edit the *Injection* panel by first pressing the *Current* button to inject currents. Then, we’ll enter 1 A to be injected onto each phase at angles of 0, 120, and 240, respectively.

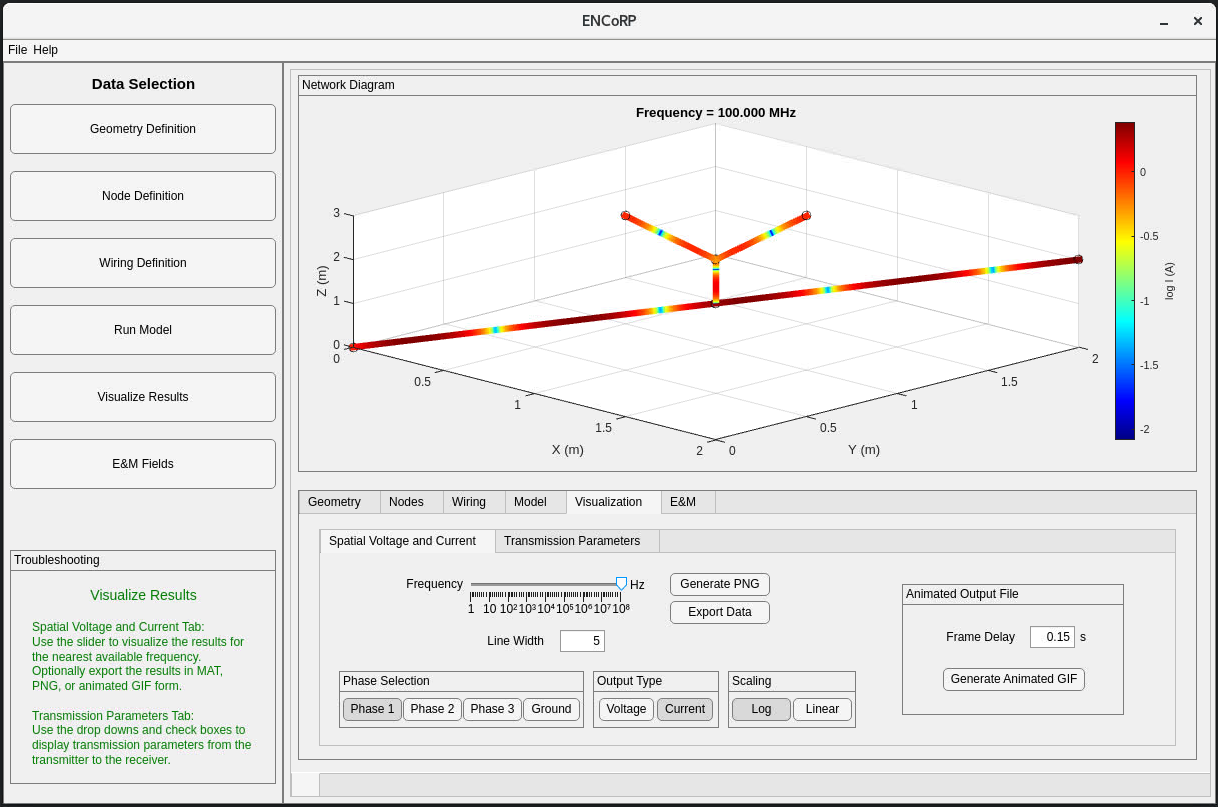
Your screen should now look like this:



Press the *Simulate* button to execute the model and unlock the *Visualize Results* tab. The troubleshooting panel in the lower left will indicate the status of the simulation.

The *Visualize Results* tab has a slider to select frequency, buttons to choose which conductor and what output type (voltage or current) to display, as well as other buttons to save the visuals or to export the underlying data to a file. There’s also a *Transmission Parameters* subtab to view the point-to-point transmission parameter predictions.

Feel free to explore the various options. If you choose to view the current, log-scale, on phase 3 at 100 MHz, you should see this:



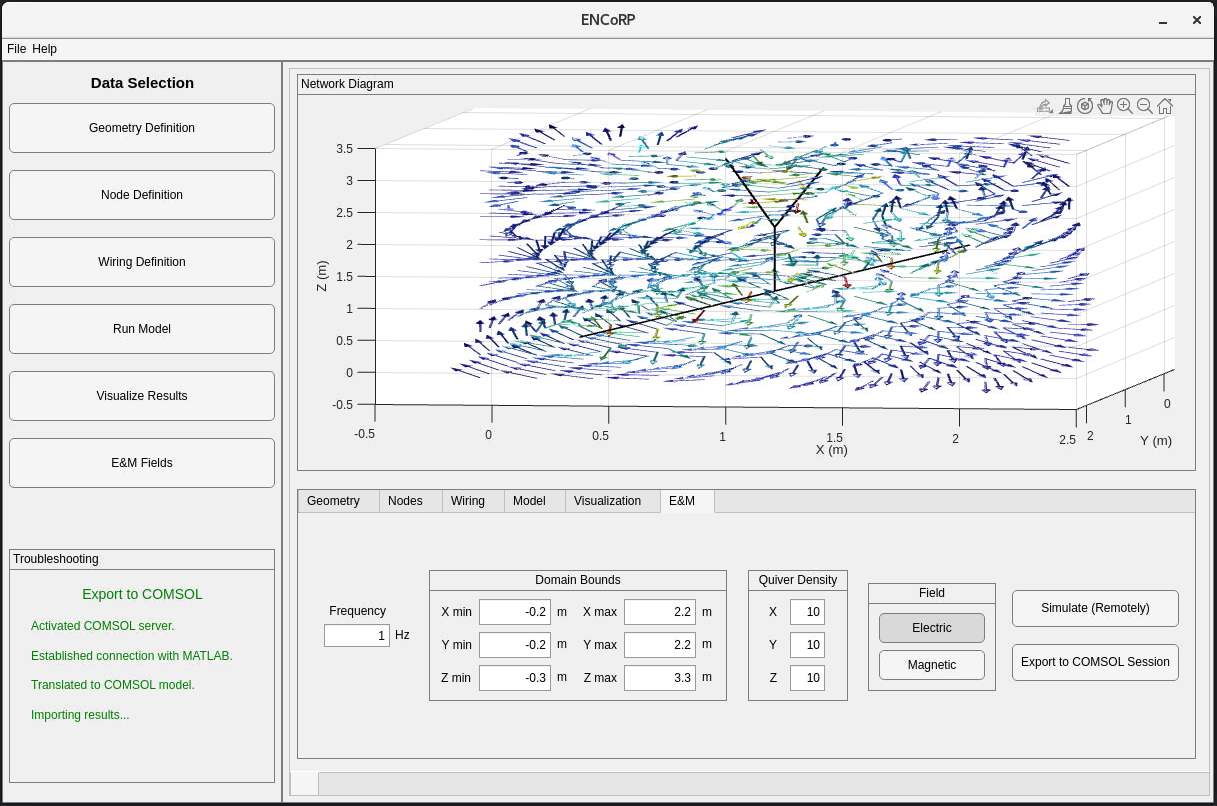
The last step in the visual, guided example takes place in the *E&M Fields* tab.

Using this portion of ENCoRP requires COMSOL.

This tab allows you to define various parameters required by COMSOL for predicting the electric and magnetic fields.

Let’s leave the values as is and press *Simulate (Remotely)*. Upon doing so, the troubleshooting panel in the lower left will provide updates as to the status of the ENCoRP-COMSOL linkage as well as the building and execution of the model in COMSOL. Note that a progress bar will appear on your screen enumerating the various mechanisms of COMSOL’s execution process.

Once completed, you should see a quiver plot in the ENCoRP figure window:



This can admittedly be a little difficult to interpret. For this reason, we can also export to COMSOL to utilize the visualizations and tools of COMSOL directly. Notably, some alternative visualization options are immediately available to the user by expanding *Results* followed by expanding *Electric/Magnetic Field Norm* andthen enabling and disabling the various options.

Unfortunately, navigating the COMSOL interface is out of scope for this example, but check out the publication to see an example network with a planar visualization of the radiating fields.

Congratulations, you have completed the visual, guided example!

# Comparison with SPICE

The *Run Model* tab features a checkbox labeled *SPICE*. This checkbox is one of a few steps required to compare the results of ENCoRP with predictions of PSpice. The steps are outlined below:

1. Check the *SPICE* checkbox: PSpice accepts R, L, C, and G as transmission line parameters. These values vary with frequency, but the *SPICE* checkbox will fix the values in ENCoRP to those defined on lines 210-213 of VI/compVInodes\_Functions.py to enable comparison with PSpice at multiple frequencies (granted the R, L, C, and G may be incorrect at other frequencies). Additionally, the *Export Data* button on the *Visualize Results* tab will now output nodal voltages and currents, which correspond to the probed voltages and currents at pins in PSpice.
2. Create and evaluate the network in PSpice: ENCoRP does not assist in building a PSpice network. It is the user’s responsibility to ensure the networks match such that the results can be meaningfully compared. When defining probes along the PSpice network for evaluation, place the voltage probe followed by the current probe then move to the next pin. If the probes are placed from transmitter to receiver in a node-based, depth-first manner, then the ENCoRP output will be directly comparable with the output of PSpice.
3. Save the PSpice output: export the output of PSpice to a CSV file. If any of the probes are out of order, this can be corrected prior to export by shifting variables around in the PSpice export interface.
4. Compare ENCoRP and PSpice: use the provided function *Compare\_ENCoRP\_with\_PSpice.m* alongside the generated data to compare the nodal voltage and current predictions. Note that you may have to modify the file or data for naming schemes to match. See the publication for an example comparison of ENCoRP and PSpice.

# Known Issues

Problem:

If you receive a warning about the inability to draw the scene (often accompanied by upside down axes), an issue may have occurred with the GPU.

Solution:

While this issue seems to only affect the visualizations; hence, you may be able to continue using ENCoRP to generate numerical results, closing and reopening MATLAB should solve the problem.

Problem:

The COMSOL server can occasionally become full, preventing further models from being created.

Solution:

This occurs when the user has generated many COMSOL models successively. Save existing models and   
 kill the COMSOL server. The next time you attempt to predict electric or magnetic fields, a new server   
 will be created. Note that operating on an existing server is much faster than creating a new server,   
 hence it’s worthwhile to not launch a server each time.

# Final Word

This software is not meant to be considered professional; rather, it is a development that was recognized as potentially being useful to the community. We have done our best to verify the correctness, but acknowledge that issues may exist. We encourage users to reach out to [encorphelp@gmail.com](mailto:encorphelp@gmail.com) with any questions or concerns.