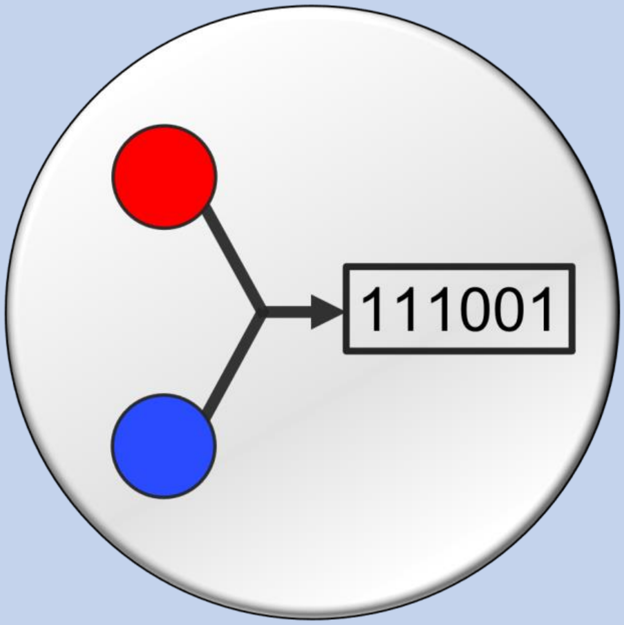
Version

2.0

Power System Graph Converter



Getting Started

Authors:

Svetlana V. Poroseva

Lana A. Hoover

Kazi S. Ahmed

Contact:

poroseva@unm.edu	Email
505-277-1493	Phone
MSC01 1150 Department of Mechanical Engineering 1 University of New Mexico Albuquerque, NM 87131-0001 U.S.A.	Mail

Version History:

March 2011	First Version	For PSGC 1.1
January 2020	Second Version	For PSGC 2.0

Developers:

Dominik Neumayr
Kazi S. Ahmed
Svetlana V. Poroseva

This current version of this software is based upon work supported in part by the National Science Foundation EPSCoR Cooperative Agreement OIA-1757207. Partial support for first version, PSGC 1.0, was provided for the software development from the Office of Naval Research (grant N00014-08-1-0080) and from the Engineering Research Center for Future Renewable Electric Energy Delivery and Management Systems supported by the National Science Foundation (grant NSF EEC-0812121) during Dr. Poroseva's affiliation with the Center for Advanced Power Systems, Florida State University, where the project was initiated.

Table of Contents

List of Figures

Introduction

What is PSGC?

PSGC stands for *Power System Graph Converter*. PSGC is a graph software designed to facilitate conversion of a network drawing into an adjacency matrix or an adjacency list. Such conversion is useful for many network analyses. However, one will find this software of particular help when analyzing large-scale networks with sources and sinks such as power systems, communication, and control systems as well as other critical infrastructures. Because initial application of this software has been power systems, thus, the software name. Terminology used in this document is also relevant to this particular application: sources are generators and sinks are loads.

The software generates non-traditional matrices/lists for the network analysis. Specifically, every network element from sources, sinks, and links between sources and sinks, is represented as a matrix (list) element in the software outputs. This is of particular importance for the network survivability analysis, where characteristics of every network element including individual links must be taken into consideration. More information about software outputs, network survivability analysis, and how software outputs contribute in reducing the analysis computational cost are provided in Neumayr & Poroseva (2011) and Poroseva (2012). One of such features available in the PSGC version 2.0 is disintegration of a network with multiple sources and sinks into smaller sub-topologies, each including a single sink (a load in a power system) connected by a single link to the network layer of sources (generators). Adjacency matrices/lists for these sub-topologies are also outputs of the software.

PSGC integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in a familiar graphical notation and in mathematical forms ready to export for further analysis.

PSGC has evolved over years with input from several developers and users. This is a tool of choice for high-productivity research and analysis.

Development and Libraries

The PSGC has been developed in **Java 11.0.1** due to Java’s robustness, ease of use, cross-platform capabilities and security features. The following libraries were used in the process:

- **JGraph 1.3.1** - for creating a graphical interface
- **MxGraph** – for modeling and designing graphs
- **Poi-4.1.0** – for converting a Java list into a matrix with comma-separated values
- **Commons-math3-3.6.1** – for performing basic math calculations
- **Commons-collecttions4-4.3** – for using Java collections like List, Map.
- **Commons-compress-1.18** – for compressing images
- **Commons-codec-1.12** – for decoding an image into components

The PSGC Structure

PSGC consists of three main parts as described below.

Development Environment. This is a set of tools that help the User to use the PSGC functions and files. Many of these tools are graphical User interfaces such as the PSGC desktop shortcut, the editor, and menus for viewing files and help.

Graphics. PSGC has tools for drawing power systems as graphs, saving the graphs and outputs, adding power inputs, single load disintegration and so on as described in the PSGC Operation section. The software provides two options to represent a power system: by Links-only (see for details Poroseva, 2012) and as a common graph as described in the **Outputs of PSGC** section.

The PSGC Functions. PSGC has a collection of computational algorithms ranging from algorithms common for a graphical software such as graphical component sequencing, positioning, and drawing, to algorithms more specific for this application such as conversion of a power system drawing into an adjacency matrix/list, export of an adjacency matrix/list into the file system, assignment of power inputs to generators and loads and capacitances to links, and the power system topology disintegration into sub-topologies as well as others.

PSGC Installation

Environment

The User has to install Java into the system before attempting the PSGC software installation. The minimum requirement is the Java version **jre1.8.0_241**.

Download the Software

The user can download the software **here.** Alternatively, click on the following links:

Executable File

Jar File

Installation

Installation for different operating systems is different as described below.

Windows

PSGC software installation wizard will be open by double-clicking the downloaded file **psgcsetup.exe**. Users then can proceed with installation following the instructions. Later, the User can start PSGC by double-clicking the PSGC shortcut icon in the directory where PSGC has been installed.

The User must note that the ***Java SE Development Kit 15.0.1*** must be installed in order to open the application.

Other Operation Systems

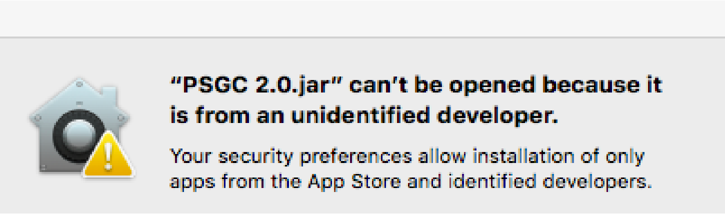
When installing the software on other OS, only the **Jar** file can be downloaded. There is a command for Linux and Mac operating systems to run **Jar** file. The command line is:

- *java -jar PSGC_2.0.jar* .

Outputs of this PSGC software will be generally saved in the same directory, where the **Jar** file is installed.

Warnings

There are several warnings while installing the software. In OSX (Mac computer) operating system, a security pop-up will appear when attempting to run the **Jar** file as the file is not downloaded from **App Store**.



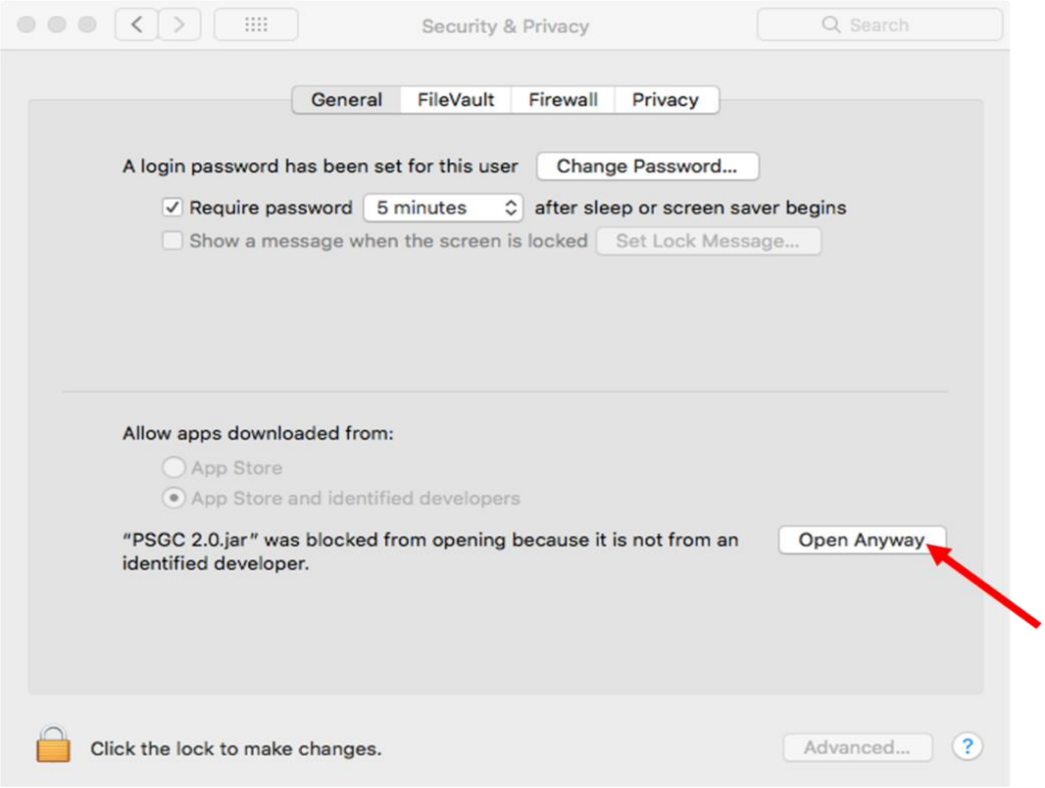
"PSGC 2.0.jar" can't be opened because it is from an unidentified developer.

Your security preferences allow installation of only apps from the App Store and identified developers.

Figure 1: Security Warning

To proceed with the PSGC installation, the User has to allow the installation of the software other than from the **App Store**. This will require performing the following steps.

Step 1. If not already done so, click on the Jar file to receive the security pop-up (Fig 1).
Step 2. Next, click on **System Preferences** on the Mac Dock
Step 3. Choose **Security & Privacy**.
Step 4. Under General, click on Open Anyway.



A login password has been set for this user

☑ Require password 5 minutes after sleep or screen saver begins

□ Show a message when the screen is locked

Allow apps downloaded from:

○ App Store

● App Store and identified developers

"PSGC 2.0.jar" was blocked from opening because it is not from an identified developer.

Click the lock to make changes.

Figure 2: Security & Privacy in Mac

Step 5. Resolve the pop-up "PSGC_2.0.jar" is from an unidentified developer. Are you sure you want to open it?"

Working with PSGC

Starting PSGC

To start using the software, double-click the software icon in the destination folder chosen during the PSGC installation. The window will open with the PSGC interface.

Quitting PSGC

To end a session, select **File -> Exit** on the software or click the cross icon on the top-right corner in the software interface.

The PSGC Interface

The PSGC interface contains tools for managing files, variables, and applications associated with the software.

There are three sections of the interface: Editor Page, Top Level Menus, and Bottom Level Menus. Figure 3 shows the default style of the software interface.

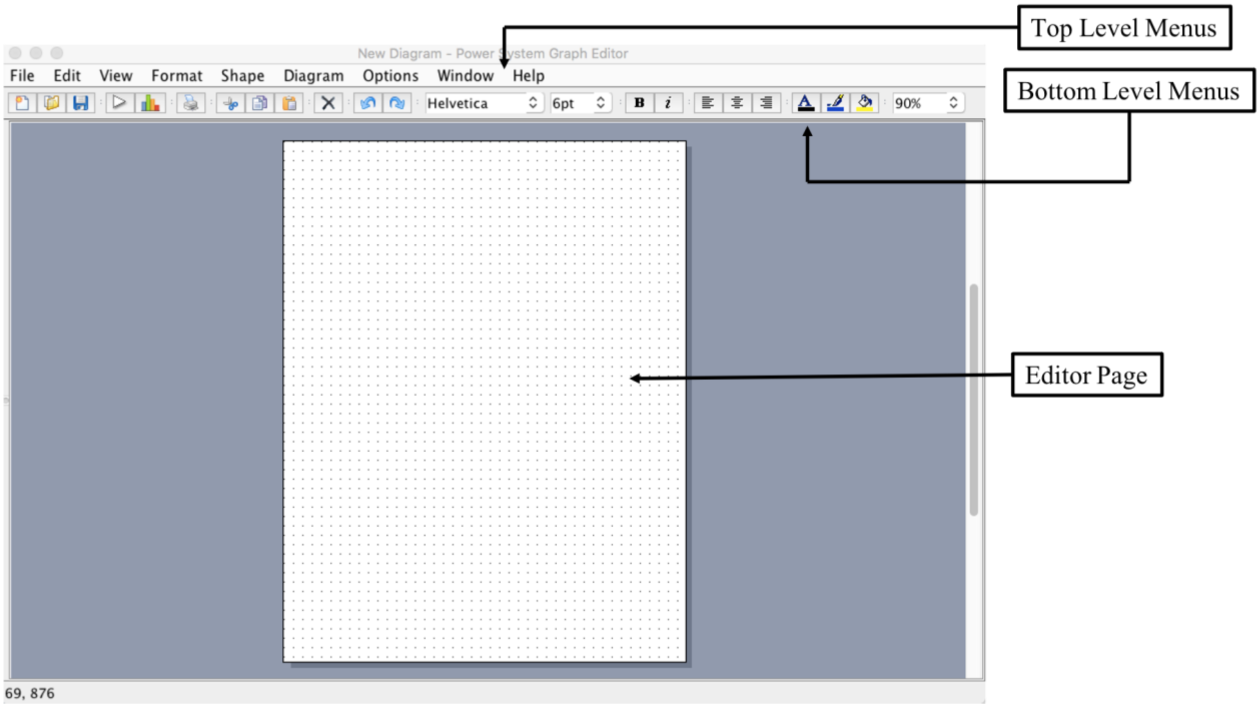


Figure 3: The PSGC interface (Default Style)

Changing the Interface style.

The User can change the interface style to Metal, Nimbus, CDE/Motif, Windows, and Windows Classic from the *Window* Top Level Menu.

Top Level Menus

There are nine Top-Level Menus with different functionalities:

1. File
2. Edit
3. View
4. Format
5. Shape
6. Diagram
7. Options
8. Window
9. Help

The User can open a menu by placing the mouse cursor over the menu name. (May need to click depending on the system). This will open the list of tasks and sub-menus. Placing the mouse cursor over a sub-menu name will also provide a choice of tasks. Names of the tasks and sub-menus are self-descriptive. An example of a Top-Level Menu with sub-menus is shown in Fig, 5.

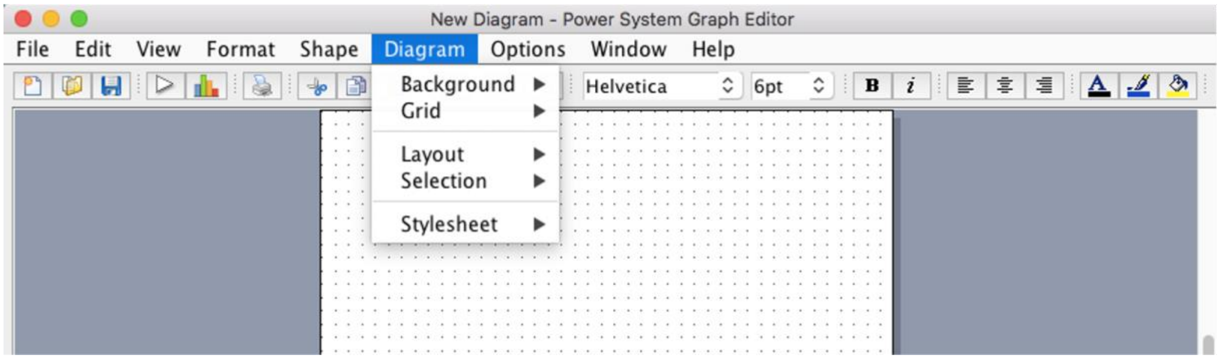


Figure 4: Top Level Menus

Bottom Level Menus

Under the Top-Level Menus, there are twenty-three options with icons called in this document the Bottom Level Menus (Fig. 4). Bottom Level Menus can perform only one task. Bottom Level Menus are:

1. New page
2. Open file
3. Save
4. Run
5. Analysis
6. Print
7. Cut
8. Copy

9. Paste
10. Delete
11. Undo
12. Redo
13. Font style
14. Font size
15. Font Bold
16. Font Italic
17. Left alignment
18. Center alignment
19. Right alignment
20. Font color
21. Underline color
22. Fill color
23. Zoom

These options are also available in the Top-Level Menus.

Primary Inputs

In PSGC, primary inputs for drawing a power system diagram depend on the User's choice of the power system representation. There are two choices: to draw a power system as a common graph with nodes and links (vertices and edges) (hereafter, the Common Graph) or by links only (hereafter, Links-only Representation).

For conducting the survivability analysis, a power system has to be drawn by links to save computational resources and to allow for assigning different properties to every individual component of a power system, including individual links connecting vital elements of the system (Poroseva, 2012). The Common Graph option is provided mainly for the purpose of producing graphics for presentations. Still, the User may find other uses for the Common Graph outputs. This level of the power system abstraction is easier to comprehend and to explain than the next level of abstraction, Links-only Representation, used in the survivability analysis.

Inputs and outputs for each representation of a power system are described below.

Common Graph Inputs

Primary inputs for the Common Graph representing a power system are the following:

- **Generator**
- **Load**
- **Junction**
- **Link**.

The *Generator* is a component of a power system, which generates and supplies power to the system. The *Load* is a component of a power system, which demands power from the system. The *Junction* is a component of a power system, where two or more links are connected (a bus in terminology traditional for the power system engineering). No power is generated or consumed in a junction. The *Junction* input is the same in both types of the power system representation used in the current software. The *Link* is a component of a power system that connects any two elements of a power system described above. All other elements existing in a real power system must be absorbed in these inputs.

By-Links Representation Inputs

When drawing a power system using links only, the inputs are:

- **Generator-VL**
- **Load-VL**
- **Junction**
- **Horizontal Link**.

In this terminology, VL stands for Vertical Links. The difference between Vertical and Horizontal Links is that the former is directed and the later are undirected.

The *Generator-VL* input is a directed link towards a junction. In a power system, such a link represents a generator and all other elements between the generator and the junction that are connected in series. If there are redundant links connecting in parallel a generator to a power system, these links have to be drawn as separate *Generator-VL* inputs. These links may be connected to the same junction or different ones reflecting a real power system layout.

The *Load-VL* input is also a directed link from a junction. In a power system, such a link represents a load and all other elements between the load and the junction that are connected in series. If there are redundant links connecting in parallel a load to a power system, these links have to be drawn as separate *Load-VL* inputs. These links may be connected to the same junction or different ones reflecting a real power system layout.

The *Horizontal Link* input is used to draw undirected links. The description of this input is equivalent to that of the *Link* in the Common Graph option.

The *Junction* input is the only type of nodes used for drawing a power system in the By-Links Representation. These inputs are not used in the survivability analysis, because a failure of such nodes can be represented by failures in adjacent links (Poroseva, 2012). The description of this input is equivalent to that of the *Junction* in the Common Graph option.

Entering Primary Inputs

To make available the Inputs of both options for drawing a power system, the User has to click the mouse right button on the dotted main page. This will activate the context menu as shown in Fig. 6. In the context menu, there are only three options are available *Generator*, *Load*, and *Junction*. All Vertical Link components are muted because to draw a *Generator-VL* or *Load-VL* a User has to first draw a *Junction*. To do so, right click on *Junction* and then Vertical Link options will become available.

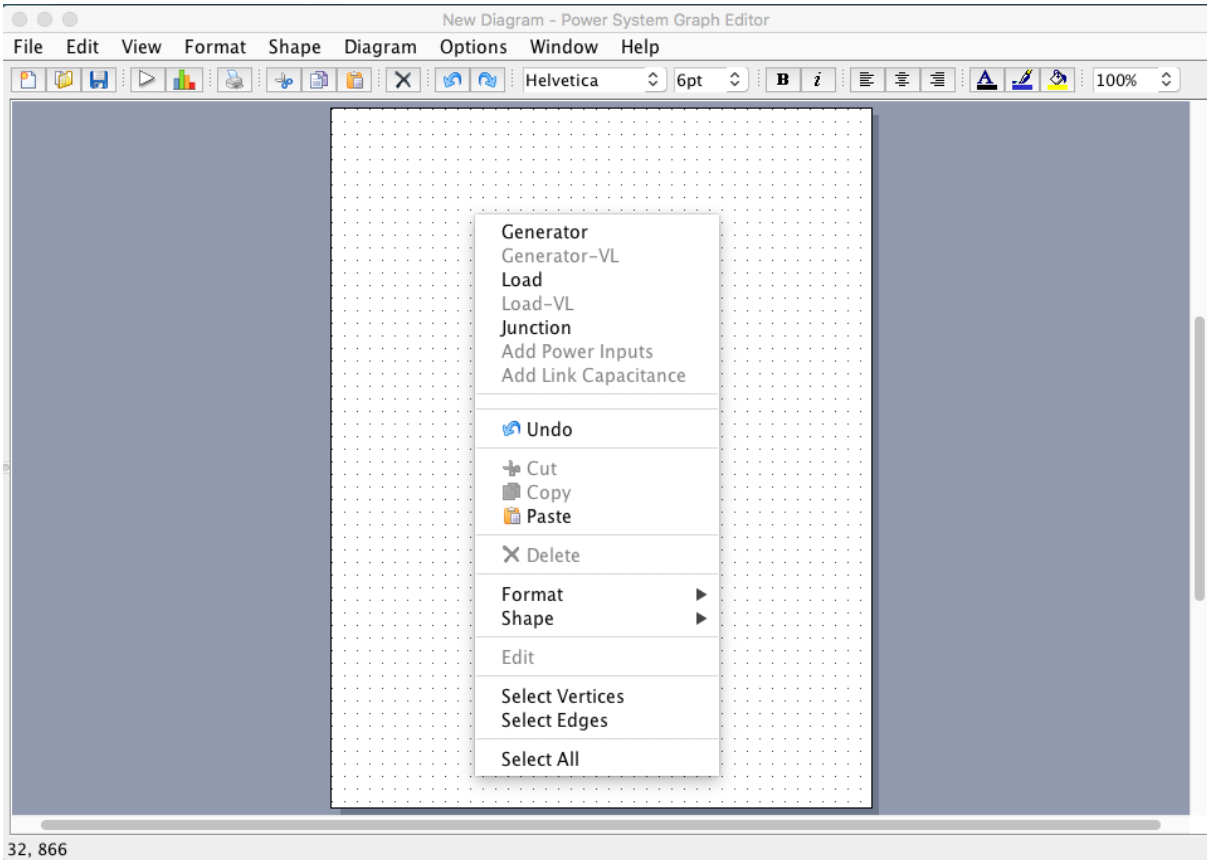


Figure 5: The Context Menu with Inputs for Drawing

Secondary Inputs

When drawing a power system diagram using the Links-only representation option, PSGC allows the User to assign secondary inputs. There are two types of secondary inputs available in **PSGC 2.0**: *Power Inputs* and *Link Capacitance*.

A User can give *Power inputs* to a Generator, a Generator-VL, a Load, or a Load-VL. By default, all power inputs are equal to 1kW. The User can change the default value to any integer greater than 1.

A User can also give inputs for each element's capacitance. The default value of capacitance is 100% assuming that power generated by all generators can be transferred through any link. The User can change the default value to any positive integer from 0 to 100.

Entering Secondary Inputs

To enter secondary inputs for a graph element, the User must first select the desired element. Next, the User will click the mouse right button to activate the context menu (Figure 6). Here, options for Add Power Inputs and Add Link Capacitance are available for the User to change inputs as required.

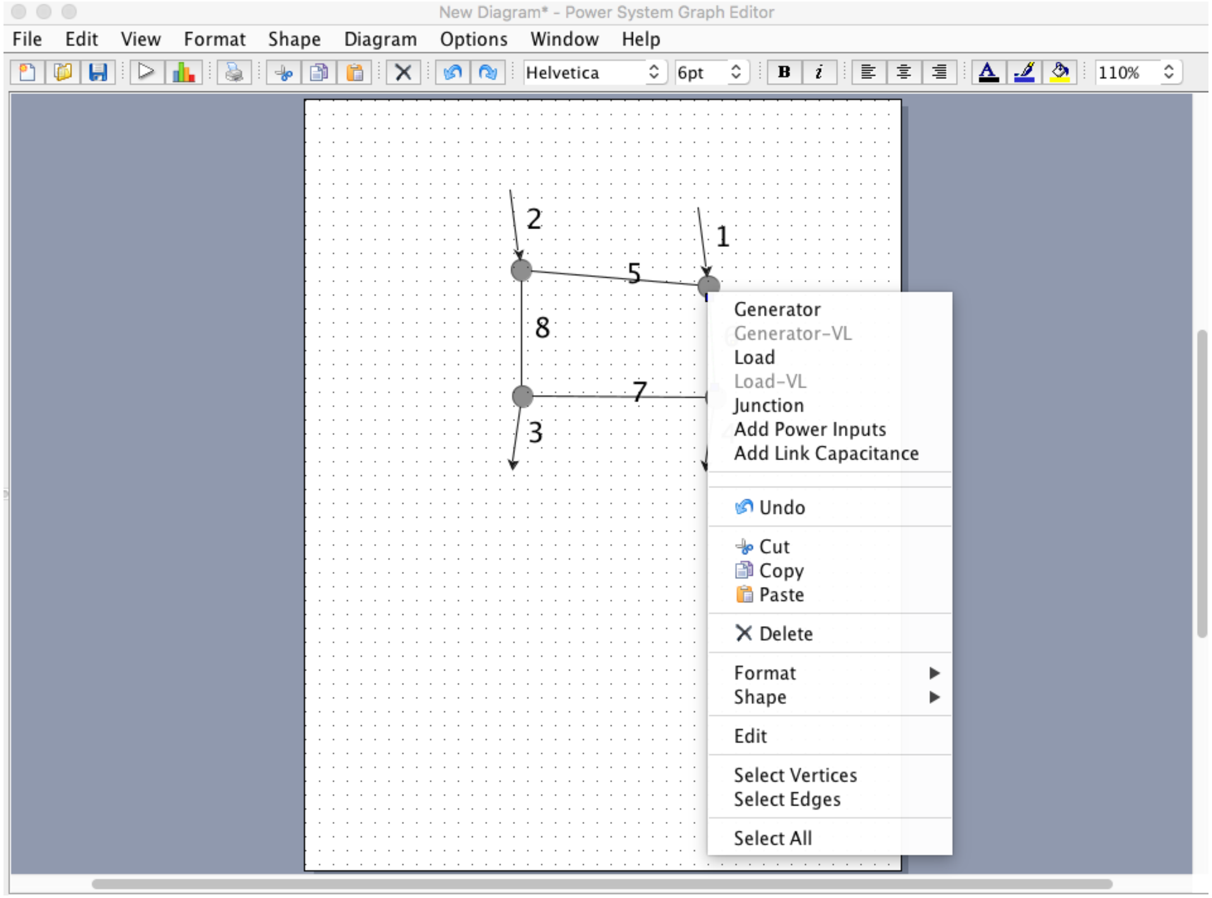


Figure 6: Context Menu with Add Power Inputs and Link Capacitance available

Drawing a Power System Diagram

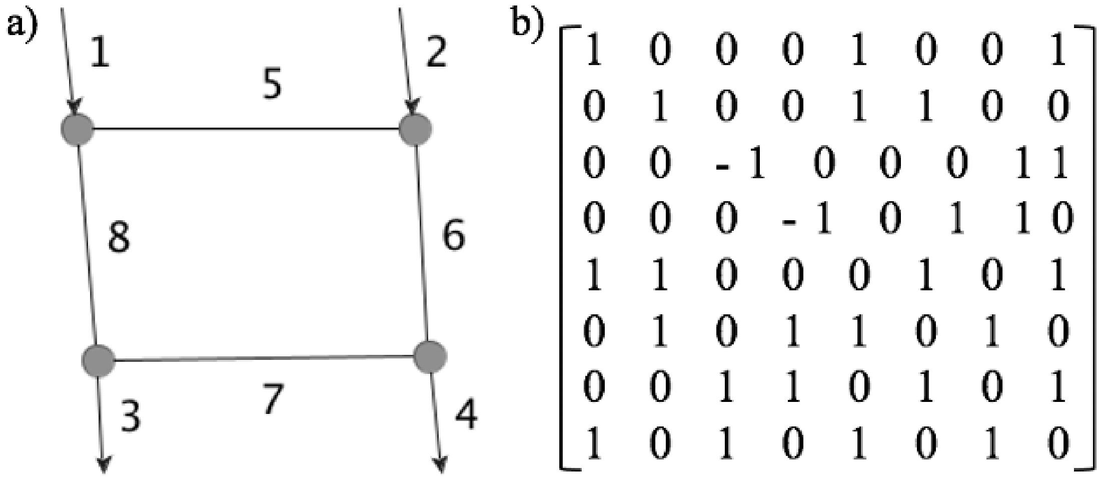
The procedure of drawing a power system diagram is described step by step in the **Power System Drawing** section of this document.

Outputs

The three types of output options available while using PSGC 2.0 are: *Matrix Export*, *Adjacency-List Export* and *Disintegration in Sub-topologies*. These options can be found under the File tab in the Top-Level Menu. They can be used for both the Common Graph and Links-only PST; however, are most applicable to the Links-only type of evaluation. An example of the outputs is available in the Example – Using PSGC Tools section of this document. Additionally, both types of PST can be saved as an image type file, which is further explained in the Exporting a Graph section of this document.

Matrix Export

Matrix Export utilizes adjacency matrices in a comma-separated value format and allows for exportation to a disk as a .csv-file. For post-processing, this file can be opened in any text editor and/or imported by another program. For the User's convenience, the export path can be specified. The adjacency matrix input is a comma-separated n x n matrix, where n is the total number of elements in the power system. The numbering follows the PST diagram, where generator numbers are the lowest, then loads are numbered, and finally, links are numbered. As with link-only representation, each element is connected to a junction. Therefore, an element is connected to another element if and only if they have a common junction or each of their connected junctions are connected through a link. Thus, if two elements are connected, then in the adjacency matrix it is marked as '1' in the first element's numbered row and the second element's numbered column and vice versa. After putting '1' in the matrix, the rest of the matrix is filled with '0', meaning no connection. The diagonal values in the matrix represent a generator using '1', a load using '-1', or link using '0'. Figure 7 illustrates an example topology and its corresponding Adjacency Matrix output for two generators and two loads.



$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Figure 7: Outputs Example Topology (a) and Corresponding Adjacency Matrix (b)

Adjacency-List Export

The structure of the adjacency list output file consists of five lines and is illustrated in the example below (Figure 8). It should be noted that this example is equivalent to the adjacency matrix in Figure 7.

$$\begin{array}{l} \textbf{Line 1} \\ \textbf{Line 2} \\ \textbf{Line 3} \\ \textbf{Line 4} \\ \\ \textbf{Line 5} \\ \\ \end{array} \left[\begin{array}{cccccccccc} 2 \\ 2 \\ 4 \\ 2 & 4 & 6 & 8 & 12 & 16 & 20 & 24 \\ 5 & 8 & 5 & 6 & 7 & 8 & 6 & 7 & 6 & 2 \\ 8 & 1 & 7 & 4 & 5 & 2 & 8 & 3 & 6 & 4 \\ 5 & 1 & 7 & 3 \end{array}\right]$$

Figure 8: Adjacency List

The first three lines are integers that identify this to be a representation of a system with 2 source links (generators), 2 sink links (loads), and 4 horizontal links. The number of total links n = 8 is then the sum of the first three rows. Representation of connections is portrayed in lines 4 and 5. Here, the links are referred to by number, and ordered such that links 1-2 are source links, links 3-4 are a sink links and links 5-8 are horizontal links.

Functionality will now be illustrated by referring to Figure 8. Line 4 is a list of indices of the comma-separated list of positive integers in line 5. It should be noted that line 4 proceeds order, describing first link 1, then link 2, and continuing on until 8.

Line 4's first value, 2, indicates that in line 5, entries 1-2 are adjacent to link 1. Meaning for link 1, links 5 and 8 are connected. Further, line 4's second value, 4, which is describing link 2, entries 3-4 in line 5 describe its connections (meaning for link 2, links 5 and 6 are connected). Continuing on to the 3rd entry in line 4, the 6 indicates that entries 5-6 in line 5 are adjacent to link 3 (links 7 and 8 are connected to link 3). Thus, all connection information for the graph can be read from the adjacency list input by referring to lines 4 and 5 in this manner.

For further clarification, when reading line 5 it is necessary to only consider entries that are not describing the previous link. For example, if line 4 entries are 8 and 12, which describe links 4 and 5, then when describing link 5, using line 5, the first 8 entries are not considered as they have been used for previous links. Therefore, only the entries past 8 and through 12 will be used (links 6, 2, 8, 1 are connected to link 5).

Like the Matrix Export output option, Adjacency-List Export uses comma-separated values as the file format and is exported as a .csv-file.

Disintegration in Sub-topologies

While using the Disintegration in Sub-topologies feature, there a multiple types of output files. Those that are named after ‘new_adjacency_matrix_for_load_[1].csv’ are the adjacency matrices of that specific load without removing unused columns and rows. Those that are named after ‘Sub_**Error! Bookmark not defined.**.csv’ are the adjacency matrices of that specific load after removing unused columns and rows. Those that are named after ‘reformed_conn_conn_list_for_load_**Error! Bookmark not defined.**.csv’ are the matrices of that specific load where each row represents the connected elements.

Reminder: when saving matrices, adjacency lists, power inputs etc., the User must use the comma-separated values file format or files will not save.

Topology Disintegration

Comparing the survivability of alternative topologies requires mathematical and computational tools. The development of such tools is the goal of the software. The survivability analysis of systems with complex topologies and of larger size may become computationally intractable. In a system with M components, the total number of possible combinations of faults is 2^M. In this regard, the survivability analysis may be categorized, at the very least, as an exponential time problem. To expand the applicability range of the analysis, the “selfish” algorithm was proposed (Poroseva, 2012).

[1] A number of a load

Power System Drawing

To draw a PST in the software, the User must right-click on the dotted main page. Here, a context menu will appear (Figure 5), from where Users can choose elements like junction, generator, or load.

Generator / Generator-VL

A generator is a component of a power system, which generates and supplies power to a system. In the Link-only representation Generator-VL (Generator vertical link) is black. Generator is one way directed line directed towards a junction. In general representation, a generator is a green circle. The style for a generator is elliptical, 50 percent opaque, font color is black, stroke color is dark green, fill color is light green.

Load / Load-VL

A load is a component of the power system which consumes power from a system. In link-only representation Load-VL (Load vertical link) is black. Load is one way directed line directed from a junction. In general representation, the load is a blue circle. The style for a load is elliptical, 50 percent opaque, font color is black, stroke color is dark blue, fill color is light blue.

Junction

A junction is a component of a power system that contains a generator or load of a power system. The style for the junction is elliptical, 50 percent opaque, font color is black, stroke color and fill color are gray.

Link / Horizontal Link

A link is a component of a power system that connects two or multiple elements of a power system. Link is a non-directed line between junctions. In general representation, the link is a blue circle. The style for the load is an elliptical straight line in shape, font color, and stroke color are black.

Drawing a Links-only Graph

Step 1: First, the User must create a junction by clicking on the junction option. A junction will then appear at the cursor point on the main page.

Step 2: Next, after creating the desired number of junctions, the User will be able to draw links between junctions (Figure 9) by selecting a junction and then dragging the cursor to another junction. This can be done so by hovering over a junction, which will appear to have a green box around it, clicking, and then dragging.

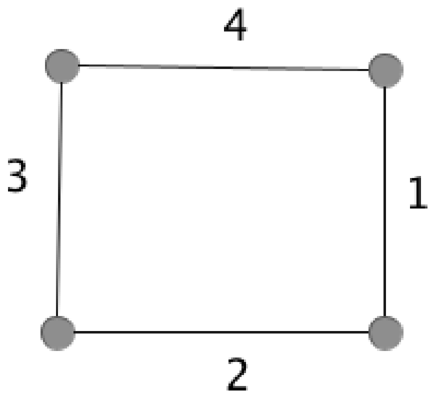


Figure 9: Junctions and Links

Step 3: To add a Generator-VL or Load-VL, simply select a junction, right click to see the Context Menu, and choose from the Generator-VL or Load-VL options.

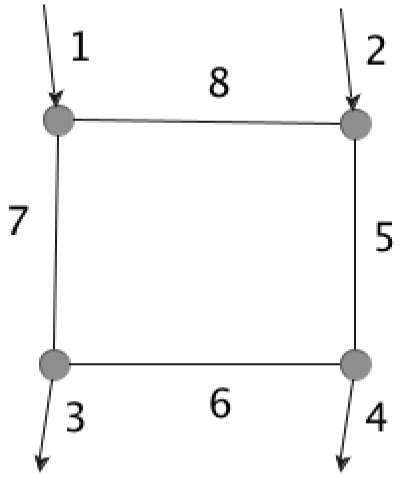


Figure 10: Topology with Generators-VL and Loads-VL

Drawing a Common Graph

Step 1: Begin by following ***Step 1*** and ***Step 2*** of the *Drawing a Links-Only Graph* description (Figure 9).

Step 2: After adding junctions and links, generators and loads can be added by using the Generator and Load options in the Context Menu.

Step 3: To complete the topology, add links to connect the generators and loads to the junctions using the same clicking and dragging method as followed in previous steps.

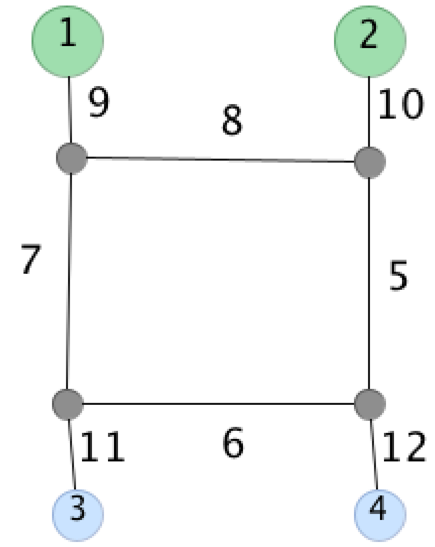


Figure 11: Completed Common Graph Topology

Below, the differences between the Common Graph and Links-Only Topology is displayed.

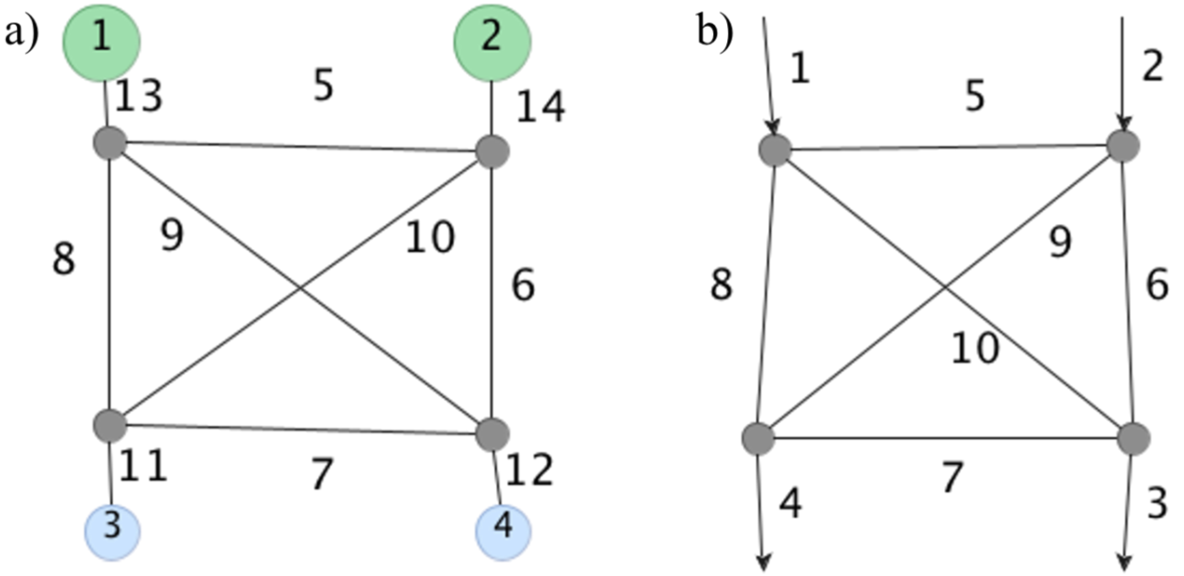


Figure 12:Power System Topology Common Graph (a) and Power System Topology Links-only Graph (b)

Selecting Elements

To select an input, the User has to double-click on it. After that, a green dashed square will appear on the element (Figure 13).

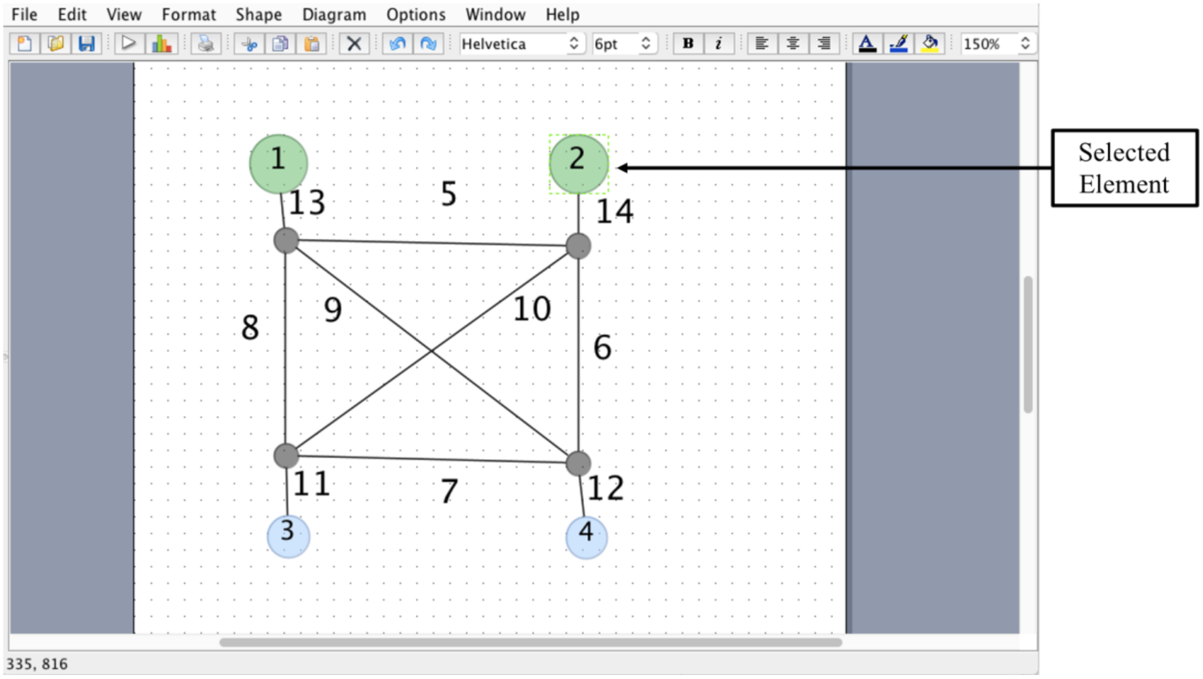


Figure 13: Selected Element

The User is also able to move the number labels for organizational purposes by clicking and then dragging to the desired location.

Specifying Group

To make a group, to cut, delete, move, etc. the User has to allow multi-select option enabled from the menu, which is by default enabled.

Levels and Titles

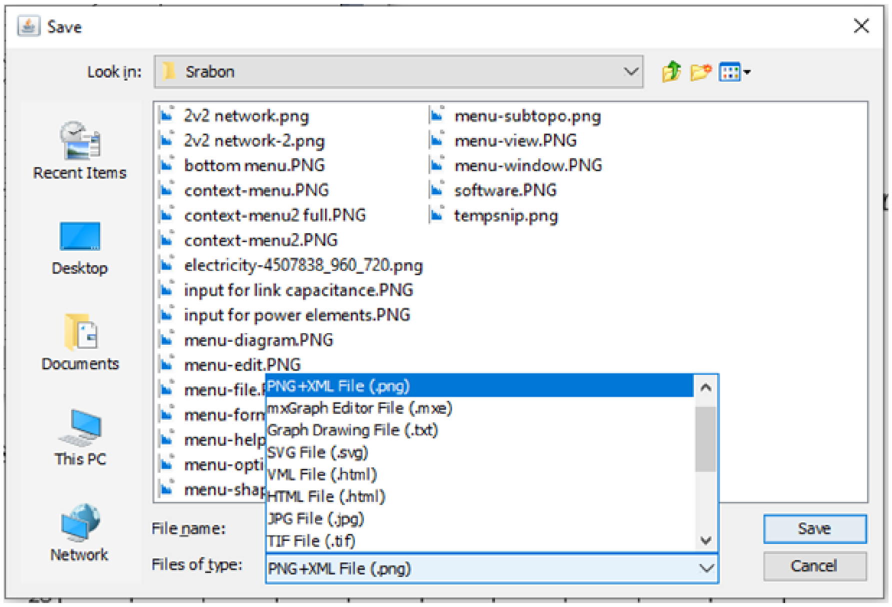
Each element, except junctions, has a label indicating its order in the matrix/list. The element order is automatically calculated and recalculated during the system drawing to ensure that the output of the drawing is a structured matrix/ list. As a result, a system can be started from any system element.

Preparing Graphs for Visualization

After finishing the drawing of topology, the User can visualize the graph into the software. For future presentations, the User can save the graph into the directory.

Exporting a Graph

Users can also export the graph from the software by selecting the *Save* or *Save as* sub-menu from the File menu (Figure 14). The graph can be saved into PNG+XML format, MxGraph Editor format, Graph Drawing format, SVG format, VML format, HTML format, JPG format, Tiff format, Tiff format, BMP format, GIF format, PNG format, and JPEG format.



Look in: Srabon

File name:

Files of type: PNG+XML File (.png)

Figure 14: Exporting Graph

Importing a Graph and Modifying an Existing Topology

To import a previously saved graph, use the File menu to select Import Diagram. For the Main Topology option, navigate to desired graph file (in exported format) and select Import Diagram. For the Sub-topology option, the User will be prompted to primarily open the main topology, and then will select the sub_x.csv files (with x being the number tag in the sub-topology name).

To modify an existing topology, the User has to open a previously saved graph file from the directory. A decoding system will decode the graph into a power system topology then the User can change the topology as necessary.

Printing the Graph

To print the graph, the User has to click the *Print* sub-menu from the File menu. Then a printing prompt will appear to set up the printing options such as the name of the printer, number of copies and so on.

Saving File Directory

PSGC output files generally save where the software is installed which called the *current directory* (Discussed in **Saving Matrices** and **Exporting the Graph**). But the User can choose where to save from the **File** menu in the *Top-Level Menu*.

Finding Current Directory

A quick way to view the current directory is by checking the top of the software (Figure 15). The current directory appears at the top of the PSGC software.

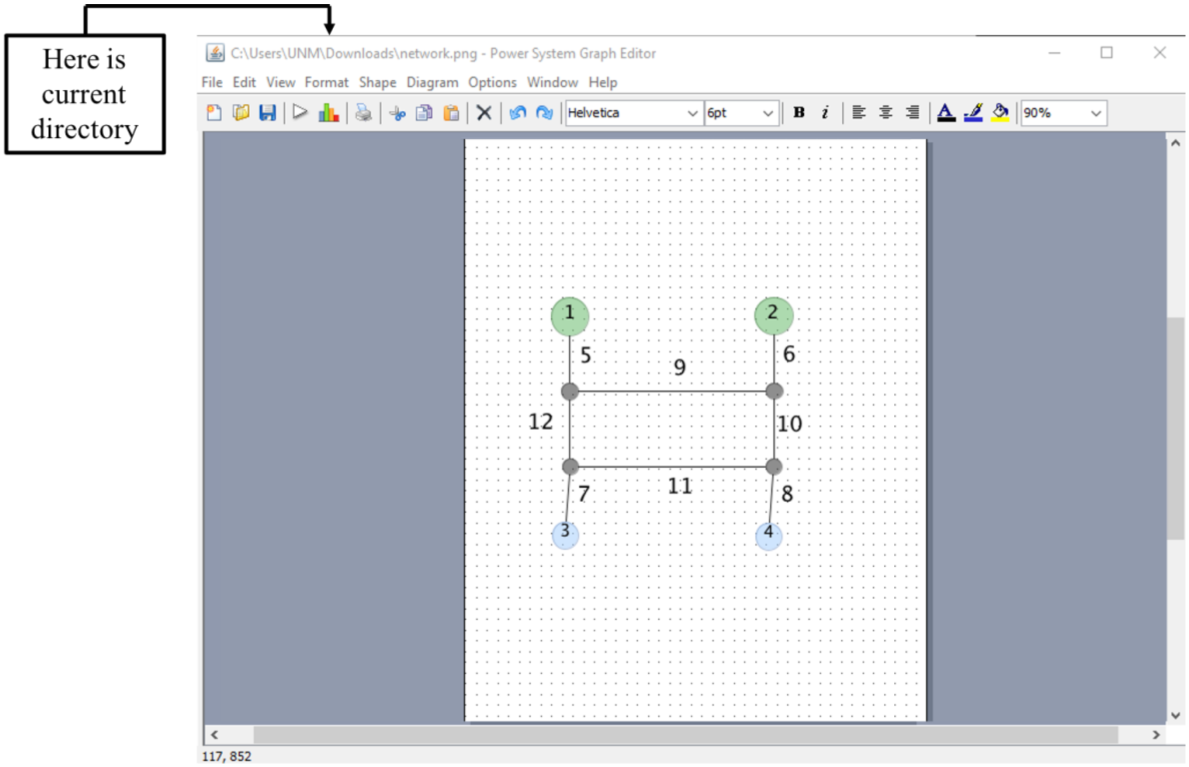

Here is current directory
C:\Users\UNM\Downloads\network.png - Power System Graph Editor
File Edit View Format Shape Diagram Options Window Help
Helvetica
6pt
90%
1
2
5
9
6
12
10
7
11
8
3
4
117, 852

Figure 15: Current Directory

Example – Using PSGC Tools

After creating a PST that contains two generators, two loads, and completely interconnected links (Figure 16), the topology results for a Links-only version will result in multiple outputs as demonstrated below.

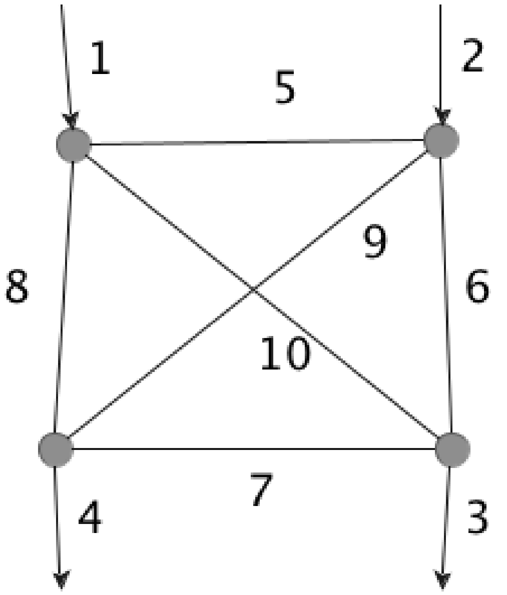


Figure 16: Using PSGC Tools Example Topology

The Matrix Export outcome would be:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Figure 17: Links-Only Matrix Example

The Adjacency-List Export outcome would be:

$$\begin{array}{l}\textbf{Line 1}\\ \textbf{Line 2}\\ \textbf{Line 3}\\ \textbf{Line 4}\\ \\ \\ \textbf{Line 5}\\ \\ \\ \end{array}\begin{bmatrix} 2 & & & & & & & & & \\ 2 & & & & & & & & & \\ 6 & & & & & & & & & \\ 3 & 6 & 9 & 12 & 18 & 24 & 30 & 36 & 42 & 48 \\ 5 & 8 & 10 & 5 & 6 & 9 & 6 & 7 & 10 & 7 \\ 8 & 9 & 6 & 9 & 2 & 8 & 10 & 1 & 7 & 10 \\ 3 & 5 & 9 & 2 & 8 & 9 & 4 & 6 & 10 & 3 \\ 5 & 10 & 1 & 7 & 9 & 4 & 5 & 6 & 2 & 7 \\ 8 & 4 & 5 & 8 & 1 & 6 & 7 & 3 & & \end{bmatrix}$$

Figure 18: Links-Only Adjacency-List Example

The Disintegration in Sub-topologies outcome is two matrices:

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix}
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & -1 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0
\end{bmatrix}$$

Figure 19: Links-Only Sub-topologies Example

References

[1] Neumayr, D., and Poroseva, S. V. (2011). On Development of Computational Tools for Evaluating System Survivability Due to Its Topology," AIAA-2011-1818, Proc. the 52st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, April 4-11, 2011.

[2] Poroseva, S. V. (2012). "Selfish" algorithm for reducing the computational cost of the network sunvivability analysis. *Optimization and Engineering, 15*(2), 381–400. doi:https://doi.org/10.1007/s11081-012-9207-1