Version

2.0

Power System Graph Converter

Getting Started

Authors:

Svetlana V. Poroseva

Lana A. Hoover

Kazi S. Ahmed

Contact:

poroseva@unm.edu	Email
1 (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).

To compare with the previous software version, the PSGC version 2.0 allows the User to incorporate information about the link capacitance into product outputs. Also, one can disintegrate a network (system topology) 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:

Jar File

Executable 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**.

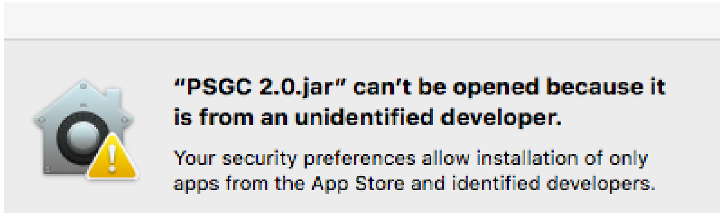


Figure 1: Security Warning

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.

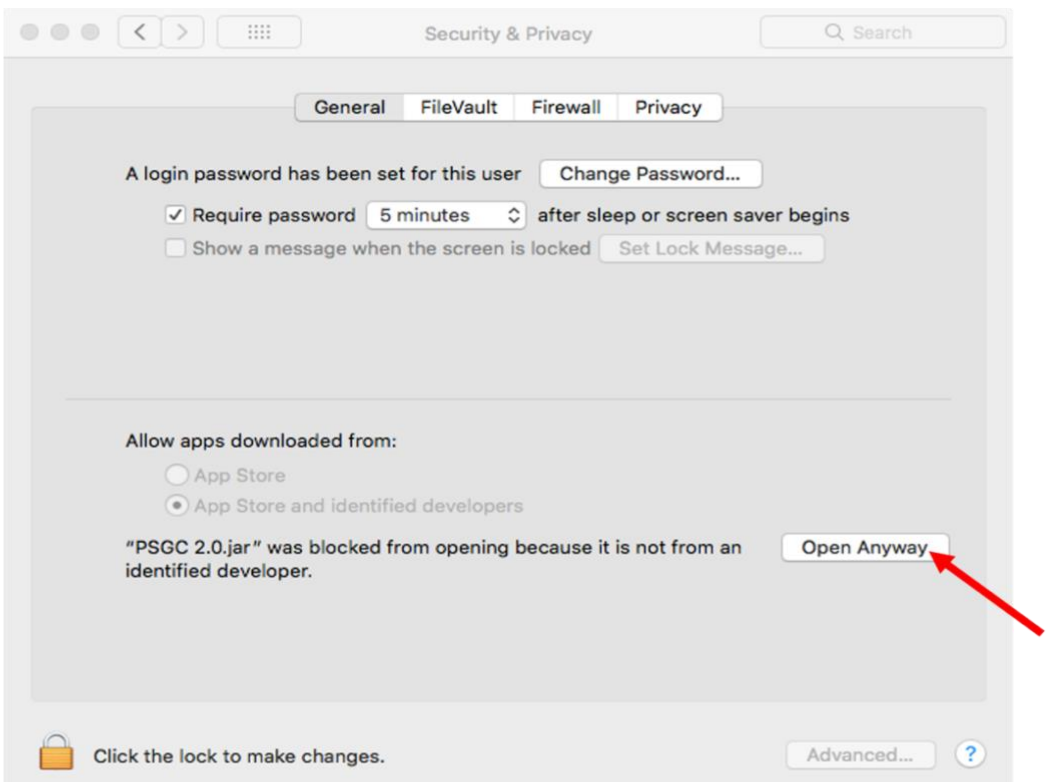
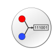


Figure 2: Security & Privacy in Mac

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.

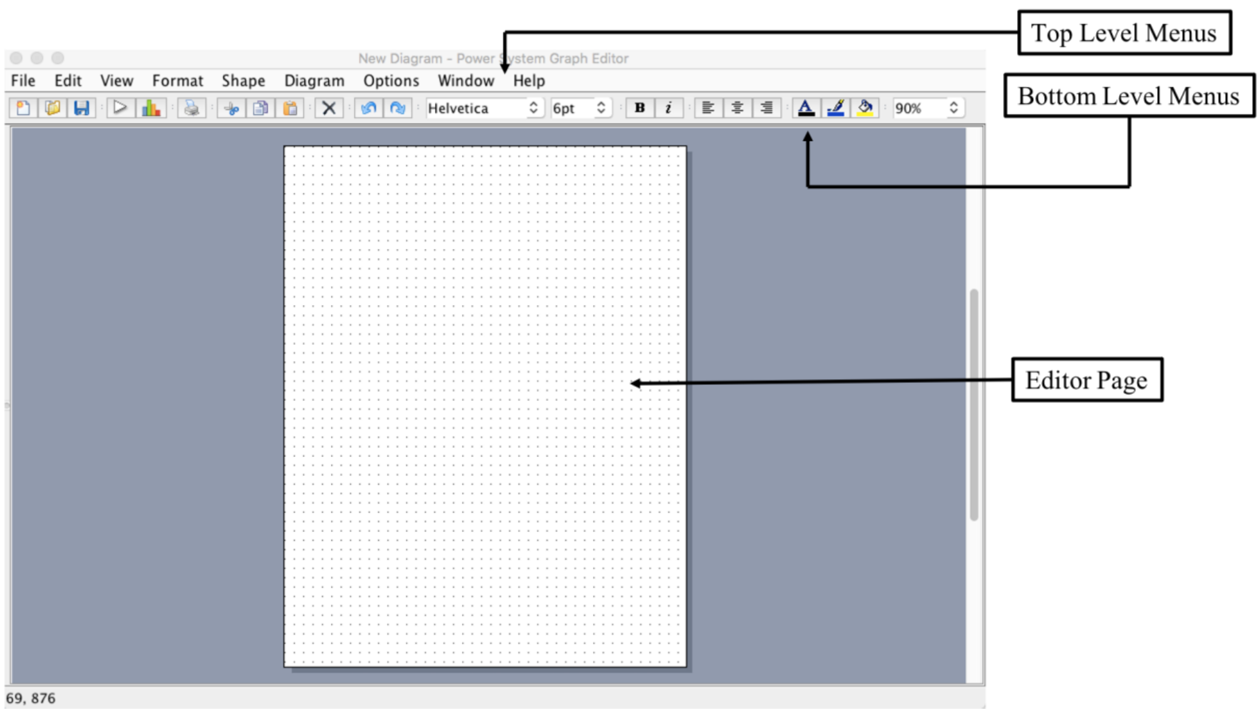


Figure 3: The PSGC interface (Default Style)

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. 4.

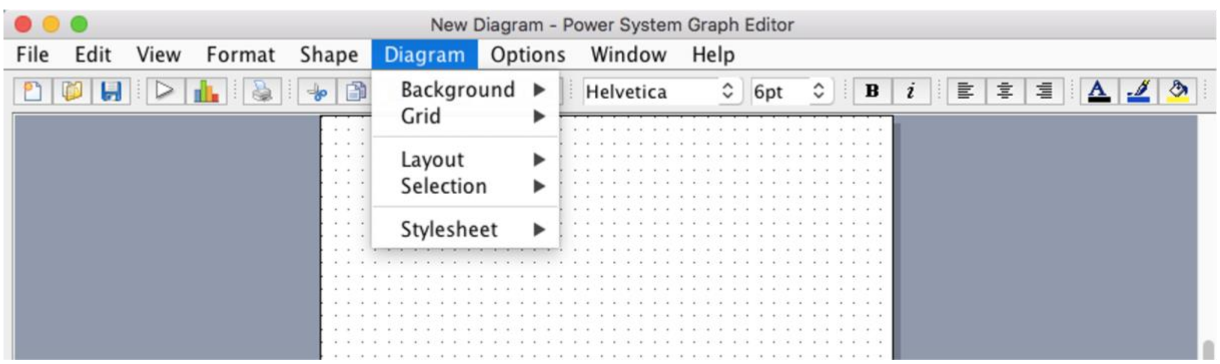


Figure 4: Top Level Menus

Bottom Level Menus

Under the Top-Level Menus, there are twenty-three options with icons called the Bottom Level Menus. 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 using the Links-only Representation 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. 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 survivability analysis. The User may find other uses for the Common Graph outputs as well.

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.

Links-only 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 is 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 the generator to the power system in parallel, 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 the load in parallel to the 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. It may represent several elements in the power system connected in series. Several horizontal links in parallel can be between any two junctions.

The *Junction* input is the only type of nodes used for drawing a power system in the Links-only Representation. However, they are not included in the Matrix and Adjacency List outputs in this network representation. This is because in the survivability analysis, where this network representation is of importance, 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 available 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. 5. 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* to draw the first junction of a network. After that, the Vertical Link options will also become available.

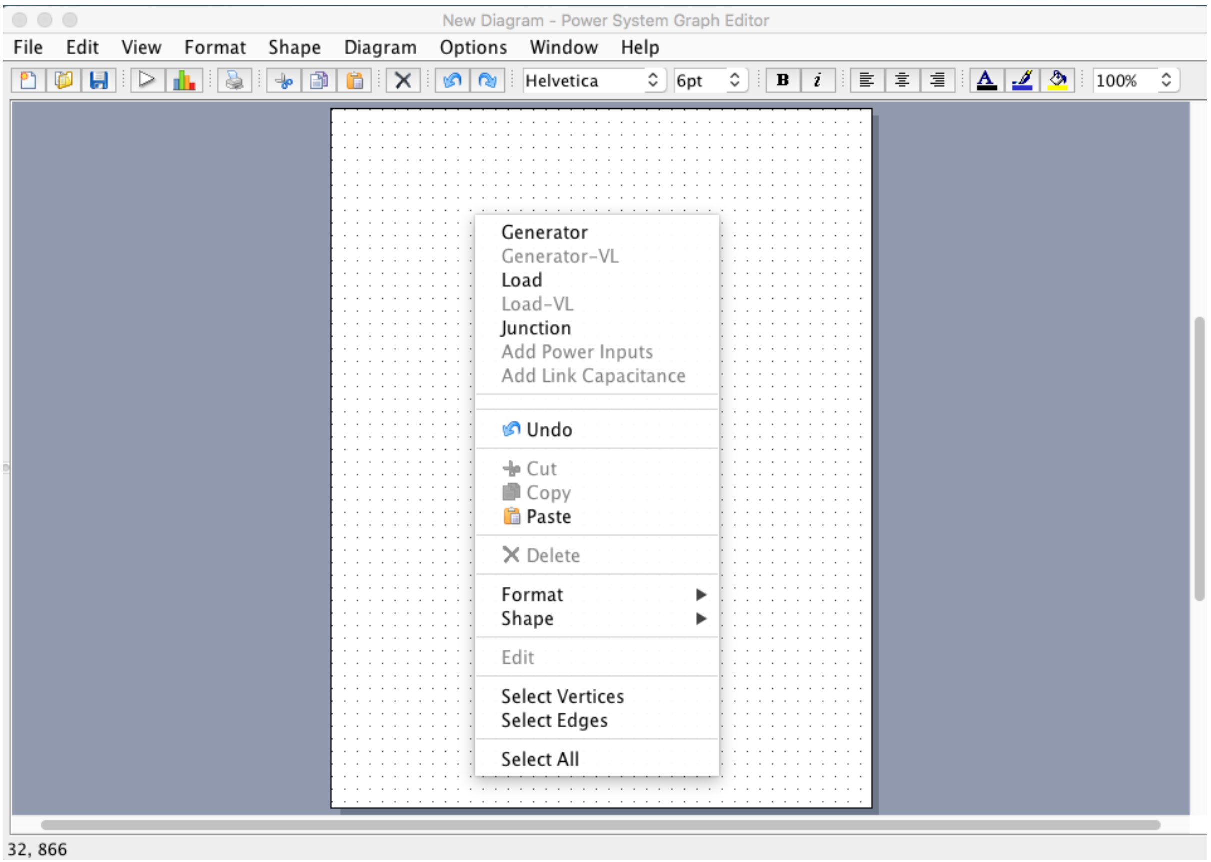


Figure 5: The Context Menu with Inputs for Drawing

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**. They are: *Power Inputs* and *Link Capacitance*.

A User can assign *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 and Saving 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 necessary.

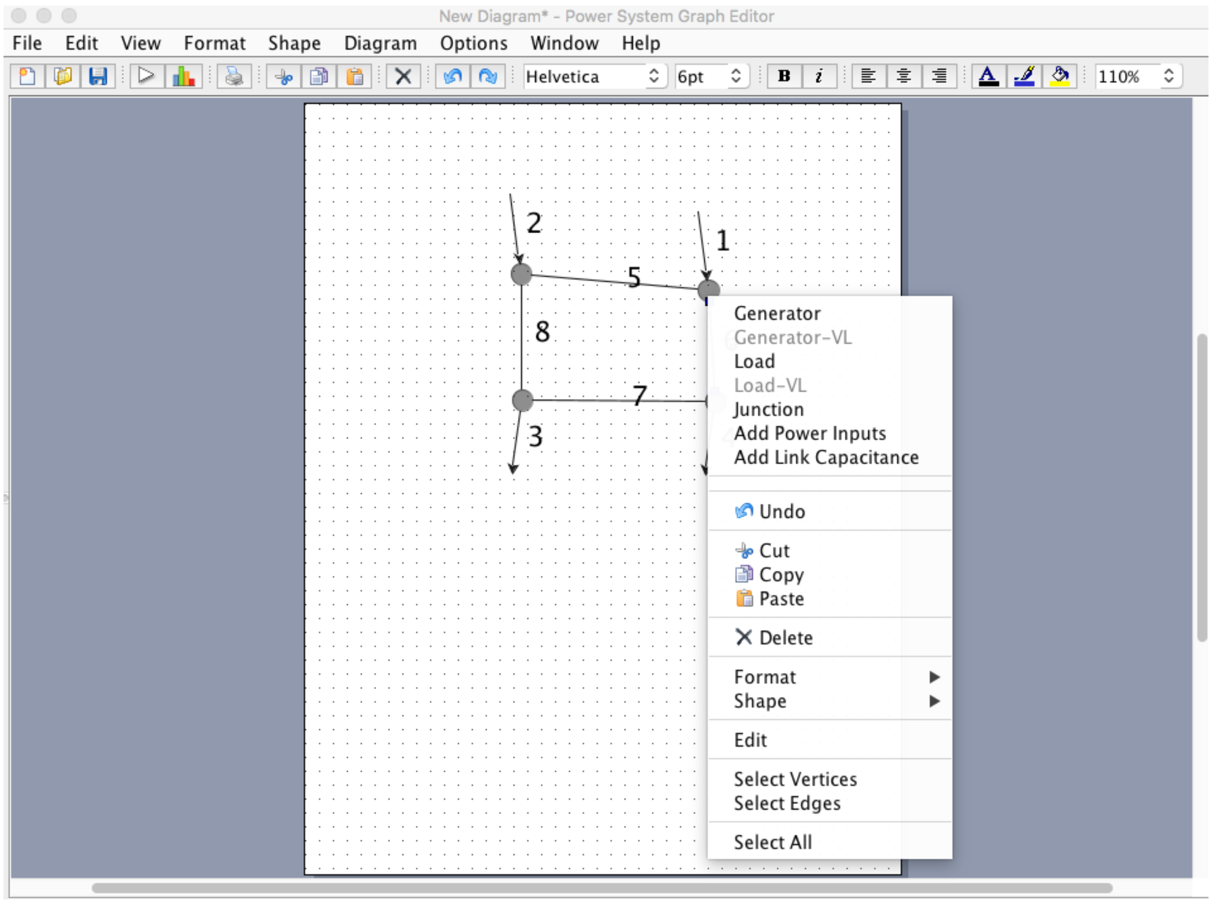


Figure 6: Context Menu with Add Power Inputs and Link Capacitance available

After entering the Secondary Inputs, the software will ask the User to save information in a file (different files will save information for different Secondary Inputs). In that file, the network will be described as a matrix with the power values written in the diagonal elements. When saving the file, the comma-separated value file option should be chosen.

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.

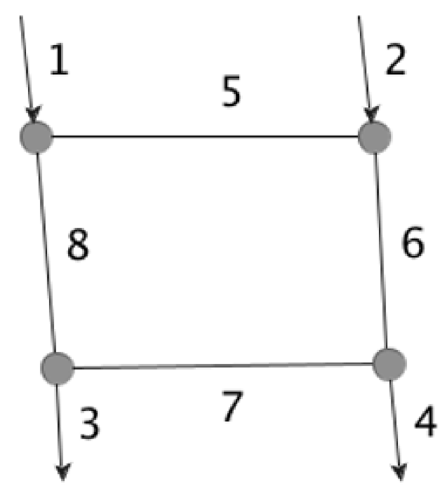
Outputs

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, they were particularly designed for the use with the Links-only representation of a network. 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 imported in 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, with the lowest numbers being assigned to the generators (or to the vertical links with generators), followed by the numbers assigned to the loads (or to the vertical links with loads). The remaining numbers are those of the horizontal links. Junctions are not included in the matrix. The connection of two elements is indicated by '1' in the location where the first element's numbered row and second element's numbered column are crossed. The matrix is symmetric. After marking all connections in the matrix by '1', the rest of the matrix elements are assigned the '0' value. A diagonal element has the value of '1' to indicate a generator, '-1' to indicate a load, and '0' to show a horizonal link. Figure 7 illustrates an example topology and its corresponding Adjacency Matrix output for a network with two generators and two loads.

a)



b)

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

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. The content of these lines is illustrated in Fig. 8 using the network topology from Fig. 7 as an example. This adjacent list contains the same information as the adjacency matrix in Fig. 7 but eliminates necessity in ‘0’. Thus, it is more efficient for the use in the network computational analysis.

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

Figure 8: Adjacency List for the Topology in Fig. 7a

Line 1 gives information about the number of generators in the network. Lines 2 and 3 provide information about the numbers of loads and horizontal links in the network, respectively. In the example, there are 2 source links containing generators (Line 1), 2 sink links containing loads (Line 2), and 4 horizontal links (Line 3). The number of total links, $\boldsymbol{N} = 8$, is the sum of the first three rows.

Information about connection of the network elements (links in the Links-only representation of a network) is provided in lines 4 and 5. This will be described using the example from Fig. 8.

The first position in Line 4 indicates how many links are adjacent to Link 1 and where information about these links is located in Line 5. For the topology shown in Fig. 7, there are 2 links adjacent to Line 1: Links 5 and 8. Thus, there is a value of 2 in the first position in Line 4 and the values of 5 and 8 in the two first positions in Line 5.

The roles of the following positions in Line 4 are similar with an adjustment to positions already taken in Line 5. Specifically, the second position in Line 4 gives the number of links adjacent to Link 2, with this number being calculated as the given number minus the number in the previous, (first) position in Line 4. For the example in Fig. 8, it gives 4 – 2 = 2. Thus, there are two lines adjacent to Line 2 and information about them is located in Line 5 in the positions from 3 to 4. Positions 3 and 4 in Line 5 indicated that these are Links 5 and 6.

In a similar fashion, information from the following positions in Lines 4 and 5 can be processed. Here, we will consider as an example one more position in Line 4, position 5, which gives a value of 12. This means, that Link 5 in Fig. 7 has 12 – 8 = 4 adjacent links; this information is located in Line 5 in positions 9, 10, 11, and 12 and are the values of 1, 2, 6, and 8, respectively. They correspond to Links 1, 2, 6, and 8 in Fig. 7.

The total number of positions in Line 4 corresponds to the total number of links in the topology represented by the Links-only or the total number of elements in the other representation (Common Graph).

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

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

Disintegration of a System Topology into Sub-topologies

To reduce the computational cost of the system analysis, it is beneficial [2] to disintegrate a topology with multiple generators and multiple loads into a set of sub-topologies, with each sub-topology connecting a single load to the same set of multiple generators. This can be achieved by applying the option Disintegration in Sub-topologies, located in File in the Top-Level Menu. By clicking this option, various files will be created to use in further analysis.

Files named as *Sub_* (*where the star will be substituted with the load label in the original topology) contain the adjacency matrix for specific individual loads.

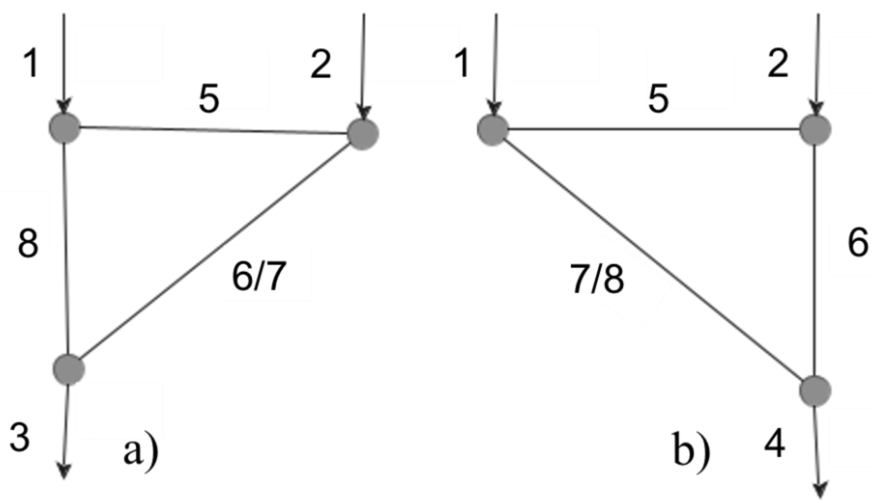


Figure 9: Sub-topology Disintegration for the Topology in Fig. 7a

Let's consider as an example, the topology in Figure 7a, which contains two loads labeled 3 and 4. After the system is disintegrated into two sub-topologies as visualized in Figs. 9a and b, respectively, there will be two files produced, named *Sub_3* and *Sub_4*. Their content is shown in Figs. 10a and b.

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

a) b)

Figure 10: Content of generated files Sub_3 (a) and Sub_4 (b) for sub-topologies in Figs. 9a and 9b, which are the result of disintegration of the topology shown in Fig. 7a.

Power System Drawing

To start drawing a power system topology in the software, the User must right-click on the dotted main page. Here, a context menu will appear (Fig. 5), from where the User can choose the system elements like such as 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 Common Graph representation, a generator appears as a green circle (the User can adjust the appearance of this and other elements as desired).

In the Links-only representation, the Generator-VL (Generator vertical link) appears as a black line directed towards a junction.

Load / Load-VL

A load is a component of a power system which consumes power from a system. In the Common Graph representation, it appears as a blue circle, which can later be adjusted in appearance.

In the Links-only representation, the Load-VL (Load vertical link) is a black line directed away from a junction.

Link / Horizontal Link

A link (or horizontal link in the Links-only representation) is a component of a power system that connects two elements of the system. The link appears as an undirected black line.

Junction

A junction is a component of a power system in which two or more links are connected. The style for the junction is a smaller gray circle.

Below, more specifics can be found on how to draw graphs in the two representations.

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 11) 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.

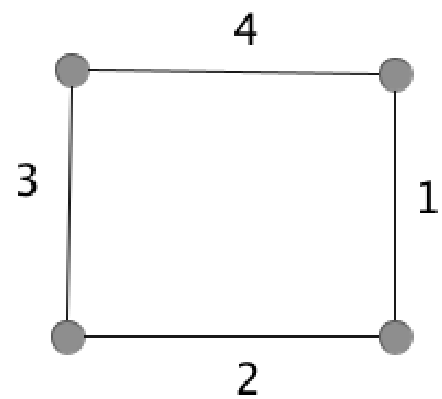


Figure 11: Junctions and Links

Step 3: To add a Generator-VL or Load-VL, simply select a junction by clicking on it, then, click the mouse right button to see the Context Menu, and choose from the Generator-VL or Load-VL options.

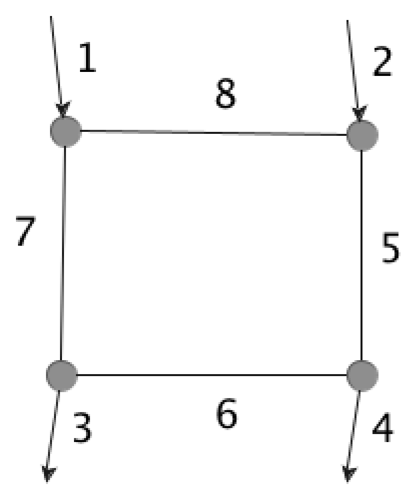


Figure 12: Topology with Generators-VL and Loads-VL

Drawing a Common Graph

Step 1: Begin by following ***Step 1*** and ***Step 2*** of the *Drawing a Links-Only Graph* description (Figure 11).

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.

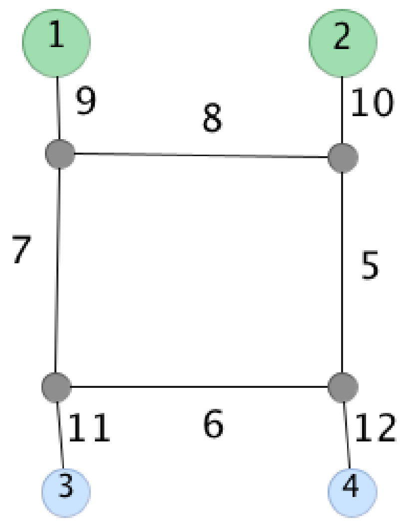


Figure 13: Completed Common Graph Topology

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

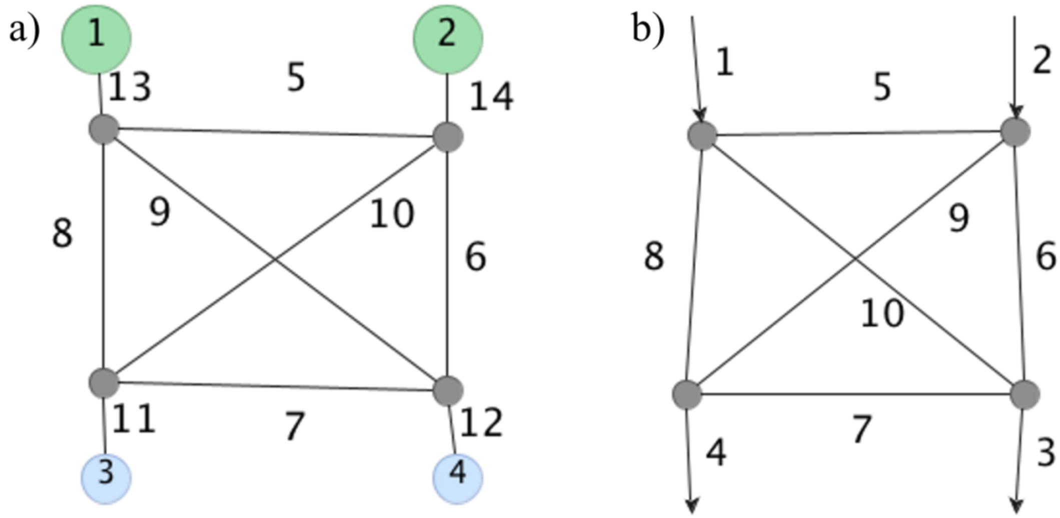


Figure 14: Power System Topology visualized as (a) Common Graph and (b) Links-only Graph.

Selecting Elements

To select an element in a drawing, the User has to double-click on it. After that, a green dashed square will appear on the element (Figure 15).

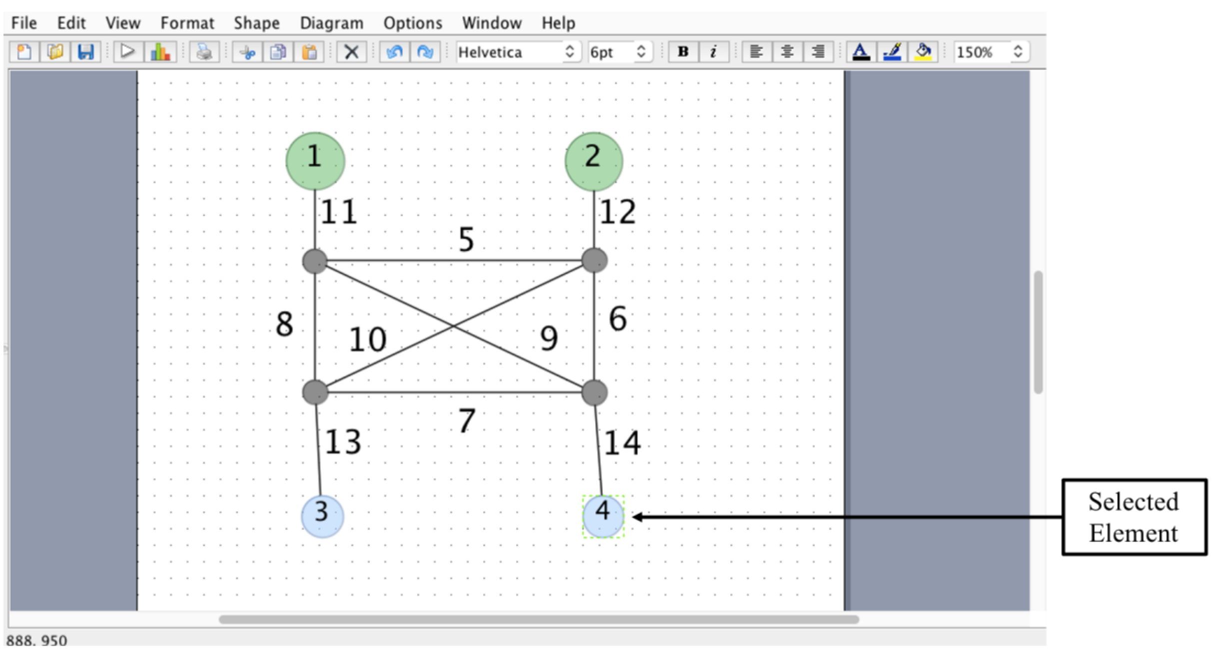


Figure 15: Illustration of Appearance of a Selected Element (green box around element 4)

The User is also able to move the element labels (numbers on a drawing) for organizational purposes by double clicking and then dragging to the desired location.

Specifying Group

To make a group of elements, the User has to place the cursor outside the group of elements that will be combined together and then click and hold on the left mouse button while dragging the cursor, so that a highlighting rectangle will appear. All the elements within the rectangle will be combined into a group by going to **Shape** in the *Top-Level Menu* and selecting Group. This can be undone by selecting Ungroup in **Shape** or by using the Undo button at the *Bottom-Level Menu*.

Levels and Titles

Each element, except for 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. This means that generators (Generator-VLs) are in the first positions, followed by loads (Load-VLs), and then, by horizontal links. Therefore, a system drawing can be started from any element.

Options for Saving Outputs

Saving Drawings

After finishing the drawing of a topology, the User can save the graph into the desired directory by selecting the *Save* or *Save as* sub-menu from the File menu (Figure 16). 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.

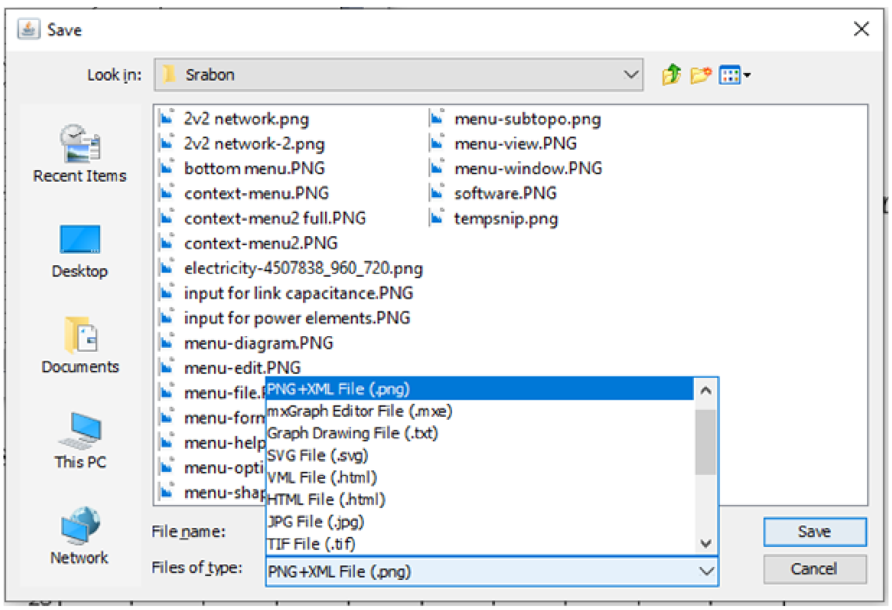


Figure 16: Exporting Graph

Saving Other Output Files

Generally, PSGC output files are saved in the directory where the software is installed. However, the User can specify where to save from the **File** menu in the *Top-Level Menu*.

Finding the Default Directory of Output Files

A quick way to view the default directory with the saved output files is by checking the top of the software (Figure 17). The default directory appears at the top of the PSGC software.

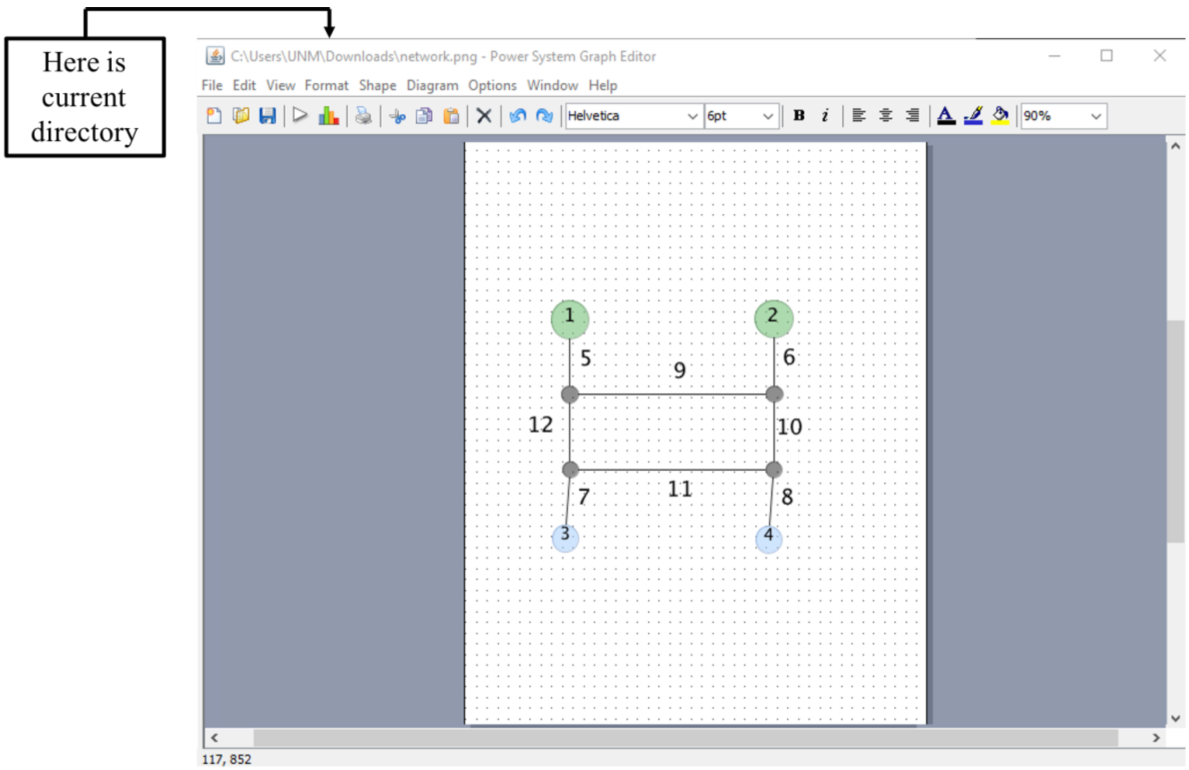


Figure 17: Illustration of location of the Default Directory

Examples of Generated Output Files

Here, we will consider as an example, a completely interconnected system with two generators and two loads. The system topology represented by Links-only is shown in Fig. 18:

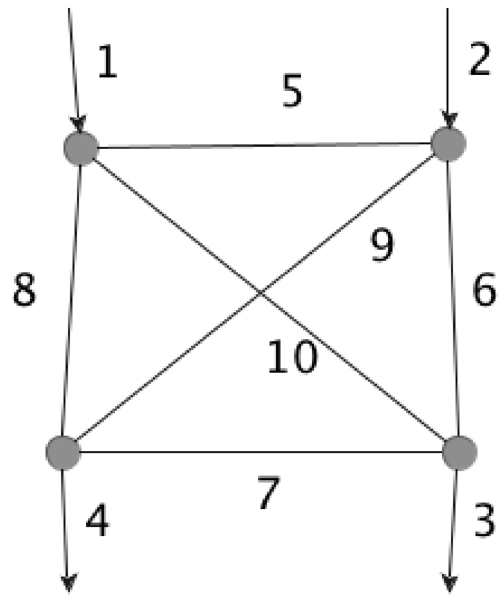


Figure 18: Completely Interconnected Topology of a System with Two Generators and Two Loads

The outputs for this topology are:

- Matrix Output (Fig. 19),
- Adjacency-List Export (Fig. 20)
- Sub-topology Disintegration - *Sub_** files (Fig. 21)

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

Figure 20: Matrix Export Output for the Topology in Fig. 18

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

Figure 19: Adjacency-List Export Output for the Topology in Fig. 18

When a topology disintegration is applied, the outputs are two matrices in files Sub_3 (Fig. 21a) and Sub_4 (Fig. 21b).

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

a)

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

b)

Figure 21: Sub-topology Disintegration for the Topology in Fig. 18

Here, the system topology represented by Links-only in Fig. 18 is shown as a Common Graph:

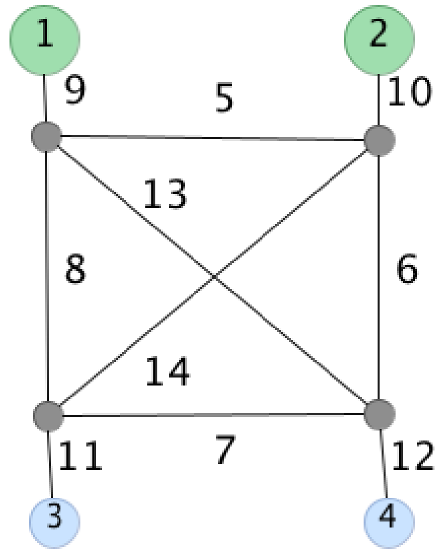


Figure 22: Common Graph Representation for the Topology in Fig. 18

Outputs for this graph representation are:

The outputs for this topology are:

- Matrix Output (Fig. 23),
- Adjacency-List Export (Fig. 24)
- Sub-topology Disintegration - *Sub_** files (Fig. 25)

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

Figure 24: Matrix Export Output for the Topology in Fig. 22

$$
\begin{array}{l l}
\textbf{Line 1} & 2 \\
\textbf{Line 2} & 2 \\
\textbf{Line 3} & 10 \\
\textbf{Line 4} & 1\ 2\ 3\ 4\ 10\ 16\ 22\ 28\ 32\ 36\ 40\ 44\ 50\ 56 \\
 & 9\ 10\ 11\ 12\ 6\ 10\ 14\ 8\ 9\ 13\ 7\ 12\ 13\ 5\ 10 \\
\textbf{Line 5} & 14\ 8\ 11\ 14\ 6\ 12\ 13\ 5\ 9\ 13\ 7\ 11\ 14\ 5\ 8 \\
 & 13\ 1\ 5\ 6\ 14\ 2\ 7\ 8\ 14\ 3\ 6\ 7\ 13\ 4\ 6\ 7\ 12 \\
 & 5\ 8\ 9\ 7\ 8\ 11\ 5\ 6\ 10
\end{array}
$$

Figure 23: Adjacency-List Export Output for the Topology in Fig. 22

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

a)

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

b)

Figure 25: Sub-topology Disintegration for the Topology in Fig. 22

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