CASTLE Help and Documentation

USER MANUAL

Raymond Allen

This page is intentionally left blank.
Remove this text from the manual template if you want it completely blank.

This page is intentionally left blank.
Remove this text from the manual
template if you want it completely blank.

Introduction to CASTLE

1 Introduction to CASTLE

CASTLE (Circuit Analysis and Simulation with Transmission-Line Emphasis) is a circuit code that bridges the gap between transmission-line codes, such as Bertha, and MNA (modified nodal analysis) codes such as SPICE. CASTLE currently only does transient analysis of a circuit, but other types of analysis may be added in the future.

See CASTLE Main Window 12 for help using the various tabs and buttons on the CASTLE main window.

CASTLE is a relatively new circuit code that is a hybrid of transmission-line (TL) and modified nodal analysis (MNA) circuit codes for transient analysis. The CASTLE program runs on the Windows platform and comprises a graphical schematic entry code, a simulator code, and an analysis and graphing code. The purpose of CASTLE is to provide a modern, graphical schematic entry interface for TL based circuits and also a simulator with the speed benefits of TL codes but with the flexible circuit topology allowed in MNA codes.

For details on how the CASTLE circuit solver works, see CASTLE Simulator Details 44.

For more information, see Comparison with TL Codes 6

Units 7 in CASTLE follow the convention of SPICE (unless otherwise stated)

Note: At this time, the analysis and graphing portions of the code are very limited. Basic graphs can be generated, but CASTLE currently relies on external programs (such as Stella 8) for analysis. A text file output option was recently added so that users without access to Stella can output simulation results to other programs (such as Origin or Excel).

1.1 Comparison with TL Codes

CASTLE combines the simplicity and speed advantages of TL codes with the free topology of MNA codes. CASTLE is a MNA code in principle, but inductors and capacitors can either be represented as TL or lumped elements. CASTLE uses a fixed time step, like most TL codes (most MNA codes use a variable time step). In this way, the output of CASTLE will EXACTLY match that of any circuit that can be represented in a TL code. Also, specialized routines written to control the impedance of an element in a TL code can be included in CASTLE with minimal modification. CASTLE intelligently utilizes the isolated aspect of TL elements to subdivide a large nodal matrix into smaller matrices. Only those submatrices that contain time varying impedances need to be inverted at every time step. In this way, the solution speed of CASTLE can rival that of TL codes. CASTLE also includes current and voltage sources, which are sometimes difficult to implement in TL codes. The graphical schematic entry module of CASTLE provides a frontend similar to that of many professional MNA codes (but still a work in progress!). But, schematic entry is improved over MNA codes with respect to TL elements because the ground node does not need to be specified for simple connections between TL elements.

TL codes generally represent all circuit elements as either transmission-lines of variable but quantized time lengths or as resistors. Inductors and capacitors can be represented by TL elements of unit time length (sometimes capacitors are two units long). Also, TL codes generally allow only simple connections between elements, which restrict the possible topology of the circuit. TL codes take fixed time steps during simulation. The time step chosen is sometimes based on the minimum common quantized length of the TL elements or the speed at which time dependent models change. Simulation in TL codes is very efficient because the ends of each element are time isolated from each other so and the end of each TL element forms an isolated subcircuit where only other connected TL ends and connected resistors need be included. Because of the usual topology limitations, only certain connections at TL ends are allowed and so the solution to the each subcircuit can be calculated in advance without the need for matrix inversion. Also, a subcircuit’s solution need only be recalculated for those time steps during which a resistance or impedance value is changed. TL codes are very well suited to many pulsed-power circuits where most circuit elements are represented very well by TL elements or resistors and generally have a simple topology.

MNA codes have no topology limitation. But, solution involves inverting a large matrix with one column and row for each node and voltage source in the circuit. Also, the time step is variable and iteration is used within each time step. MNA codes often have difficulty with pulsed-power circuits because of the rapid switching often involved. It is possible that the simulation will fail because the matrix solution fails to converge. Note that TL elements can be represented in MNA codes and that simple circuits involving TL elements and fixed resistors will have identical solutions to that of TL codes (and CASTLE). Inductor and capacitor models are created from their differential equations. Schematic entry of circuits with many TL elements in MNA codes is generally difficult because the ground node must be specified at every connection. Also, the simulation speed is often very slow compared to TL codes.

1.2 Units

Within CASTLE, units follow the same convention as SPICE (except for some special devices where units other units are explicitly shown):

The default units are Volts, Amps, Ohms, Farads, Henries, Watts, etc.
Values may be entered decimal form (e.g., 0.00245) or exponential form (e.g., 2.45e-3).
Since Castle is not case sensitive 2.45e-3 is the same as 2.45E-3.

You can also use some unit prefix abbreviations, such as "m" for "milli" and therefore write 2.45e-3 as 2.45m.

The following scale abbreviations are recognized (case insensitive):

Abbreviation	Factor	Unit Prefix
F	E-15	femto
P	E-12	pico
N	E-9	nano
U	E-6	micro
M	E-3	milli
K	E+3	kilo
MEG	E+6	mega
G	E+9	giga
T	E+12	tera

Notes:
- Do not leave any space between number and unit prefix abbreviation (write "2.45m" and not "2.45 m").
- Be careful with "M" because "M" and "MEG" are often confused.
- Letters after the first letter and unrecognized letters are ignored. For example, you can write "2.45uF" for a capacitors value, but the "F" character is simply ignored by the simulator.

1.3 Stella

Stella is a data acquisition and processing code created and maintained by Naval Research Laboratory. It is not currently available to the public, however CASTLE has options for using Stella traces as input and output.

1.4 Notes

A Note about TL Inductors

TL inductors are treated as single timestep long transmission-line elements inside the simulator. But, only two terminals are provided for the schematic symbol. This is because both outer conductors are internally treated as ground. It may be that this could cause an issue in some circuit topology, but no topology has yet been found where this was a problem. But, if you happen to find a problem topology, just use a regular transmission-line element to represent the inductor.

Variable Capacitor

A variable capacitor is a new addition to the available parts. It can either follow a user supplied polynomial function (as in SPICE) or one or two known capacitor types.

Custom Parts

It is relatively easy to program new, custom parts. Several parts have already been transferred from the transmission-line code, Bertha, into CASTLE. Several are not provided at this time because they are still being tested. Future versions may have the ability for user generated parts based on script files.

Iteration

A few parts, such as the semiconductor diode, require iteration within a time-step. It is possible for such iteration to fail to converge to a stable result before proceeding to the next time step.

This page is intentionally left blank.
Remove this text from the manual
template if you want it completely blank.

CASTLE Main Window

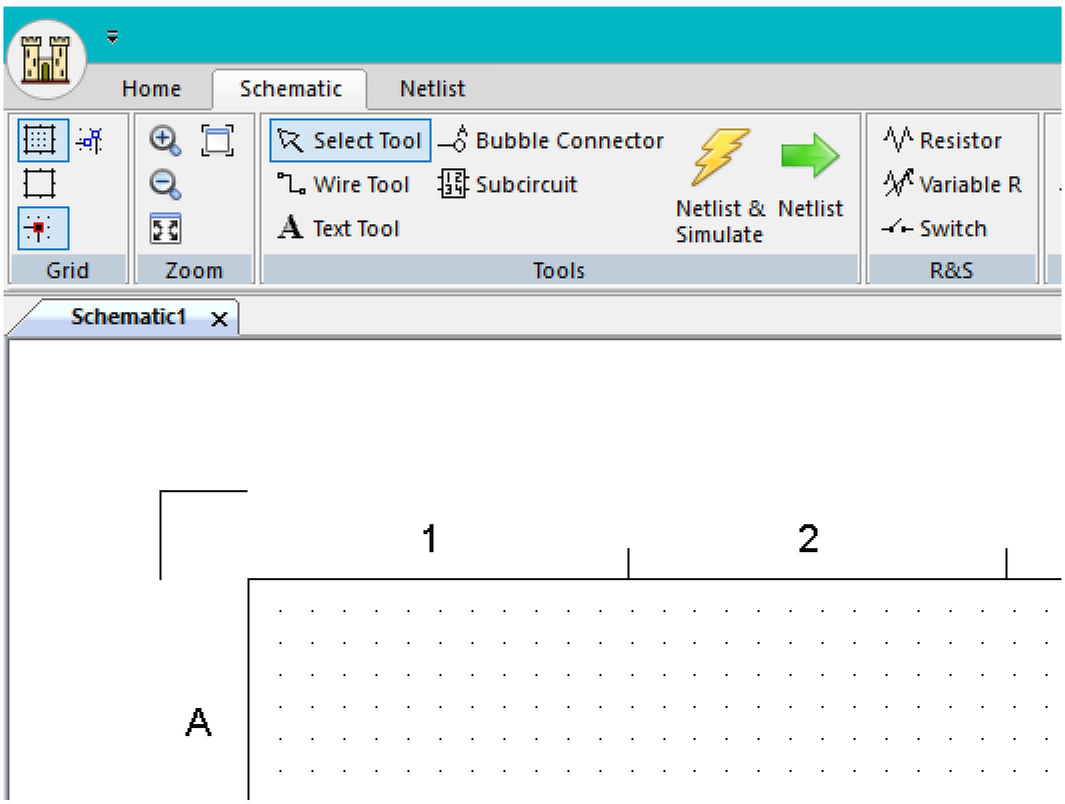
Main Window

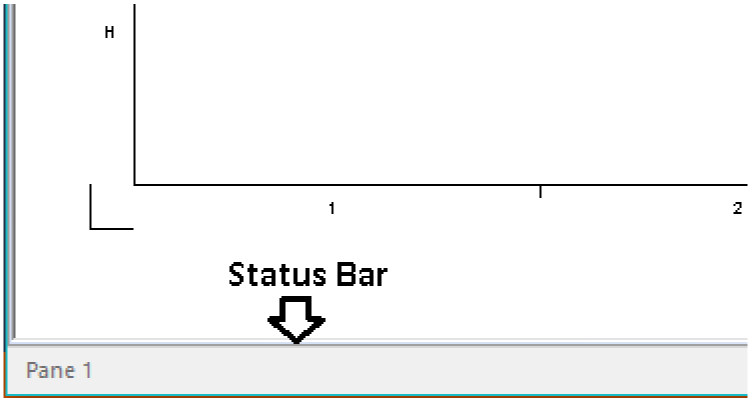
2 CASTLE Main Window

Go to Overview 12 to see a click-able image of the controls.
The Main Window consists of the Ribbon Bar 13 at the top, the Document Pane 15 in the middle and the Status Bar 15 along the bottom.

2.1 Overview

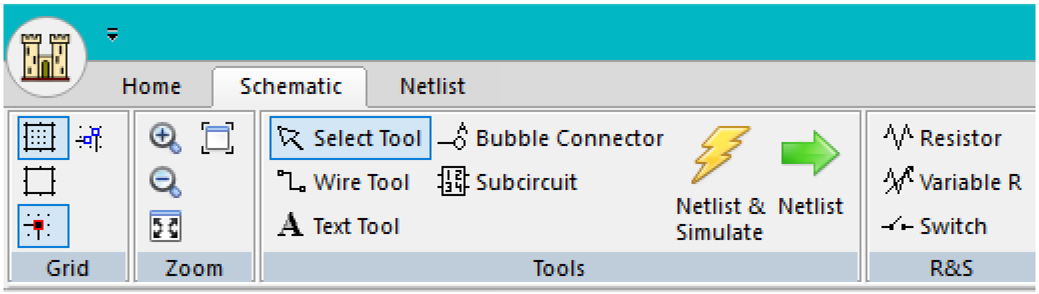
The CASTLE main window consists of three main parts, the Ribbon Bar 13 at the top, the Document Pane 15 in the middle, and the Status Bar on the bottom. The Status Bar 15 will be used in the future to provide basic information about your document. You can click on areas in the images below to identify and jump to specific help areas.





2.2 Ribbon Bar

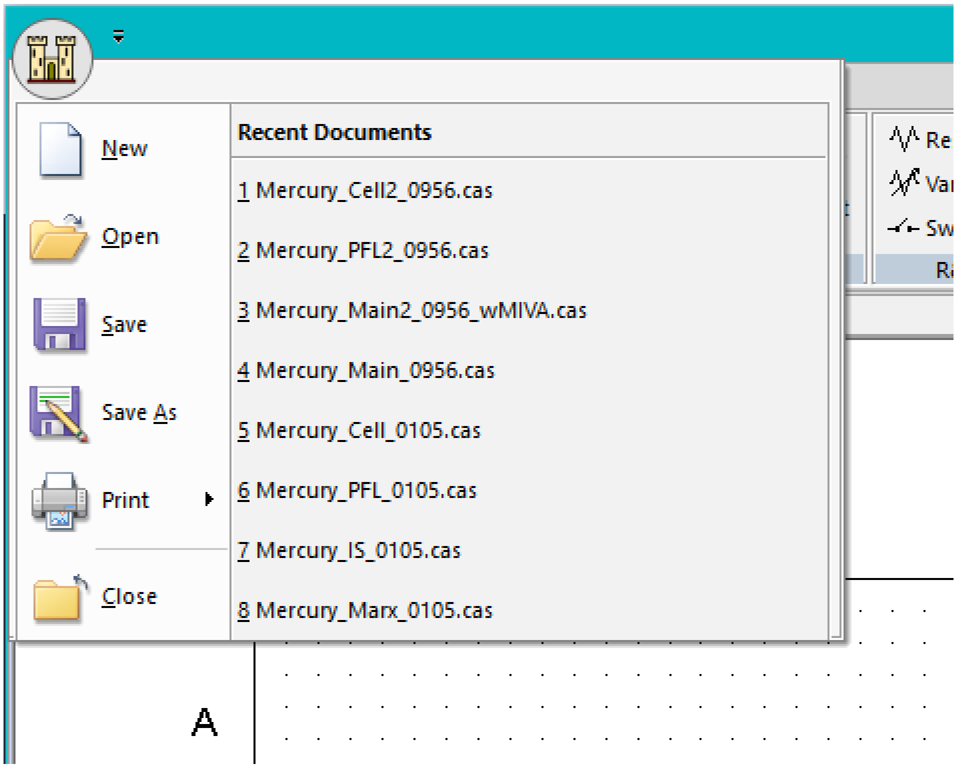
Click on the image of the Ribbon Bar below to jump to specific areas (if not already shown in this window).



The Ribbon Bar contains all the tools needed to edit, simulate, plot and save your circuit. In the top-left corner is the Application Button 13 and the Quick Access Toolbar 14.

2.2.1 Application Button

In the top-left corner is the Application button. It's vertical location spans the title bar and the ribbon tabs area. Push the Application Button to bring up a menu for saving, opening and printing the current document. It also brings up the Recent Documents list. To the right of the Application button is the Quick Access Toolbar 14.

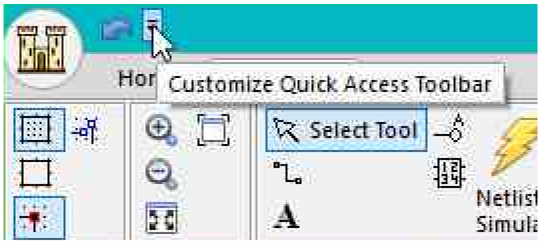


2.2.2 Quick Access Toolbar

The Quick Access Toolbar is located at the top of the CASTLE window, just right of the Application Button. You can customize this toolbar to add often used buttons.

Undo button added to Quick Access Toolbar

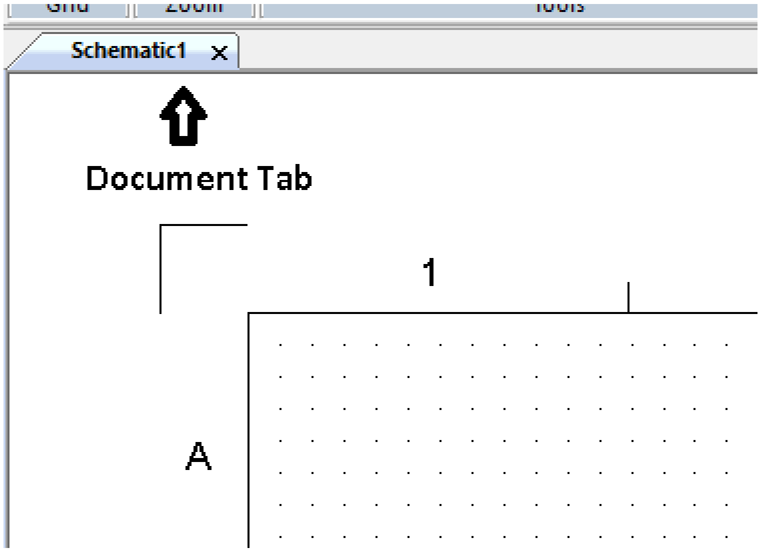
Click the black access menu button to add other buttons.



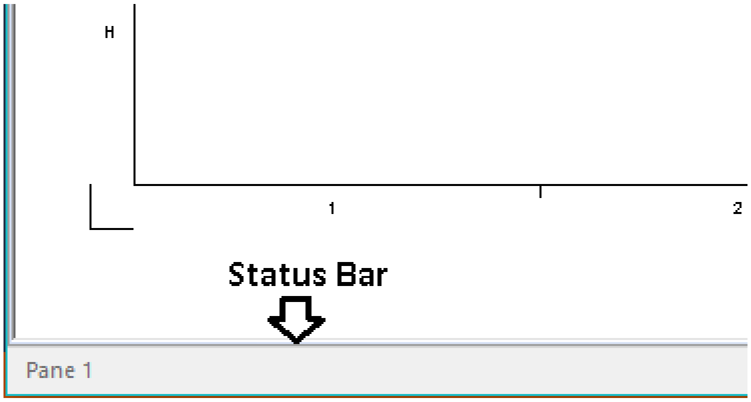
Location of menu for Quick Access Toolbar

2.3 Document Pane

The Document Pane is the main window where you edit documents. Multiple documents and be open at the same time. Each open document has it's own Document Tab.



2.4 Status Bar



The Status Bar is located along the bottom of the main window. It is currently unused. It may be used in the future to provide some information while editing documents.

This page is intentionally left blank.
Remove this text from the manual
template if you want it completely blank.

Schematic Entry

Schematic entry help

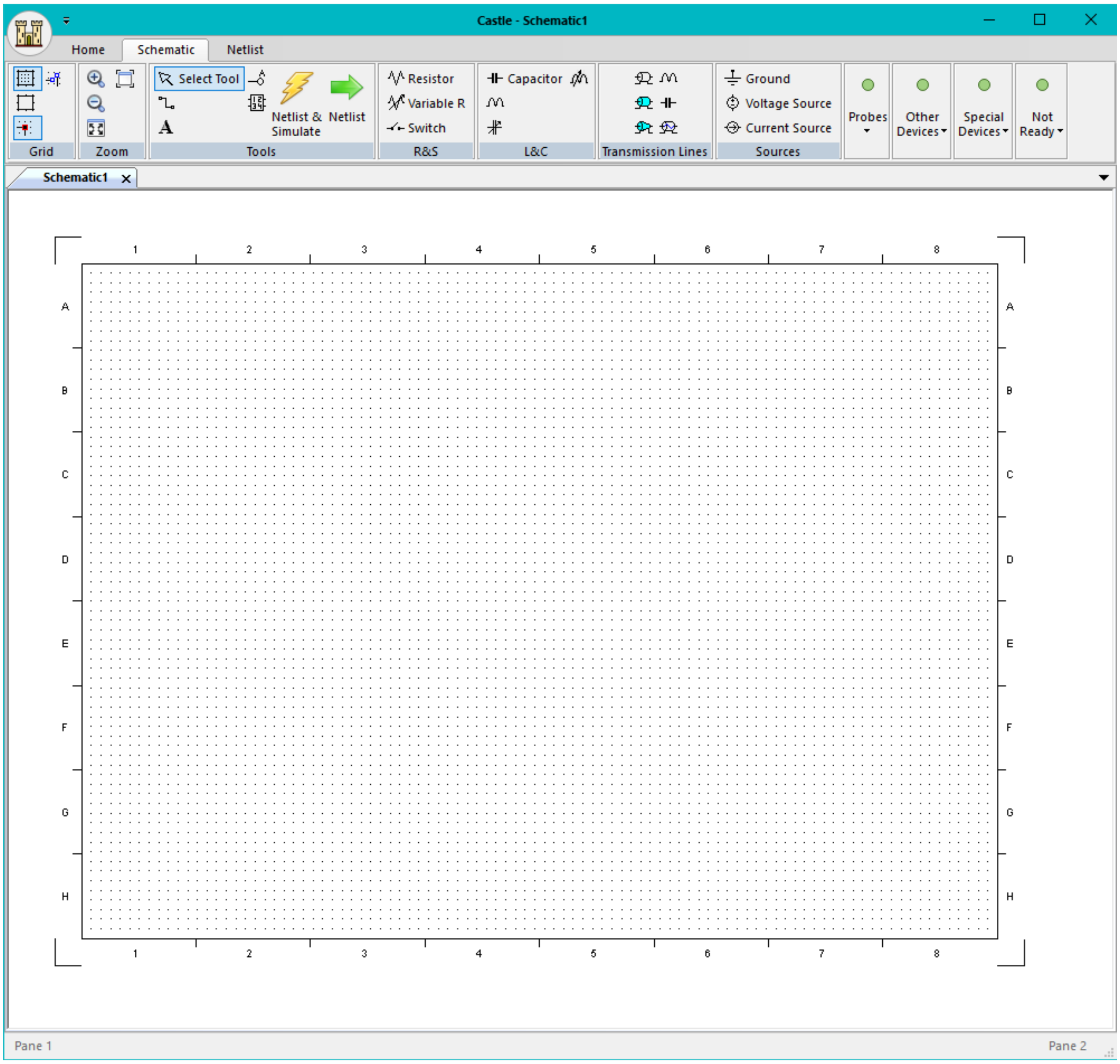
3 Schematic Entry

The Schematic Editor Window is the graphical interface where you create your circuit to be simulated. This section provides help for using the the buttons on the Schematic tab of the Ribbon Bar 13 to pick, place and edit parts such as resistors, inductors, capacitors, etc., and on using the Wire Tool to connect parts with wires.

Note: Because CASTLE is an MNA code (like Spice), you most likely need to insert at least one Ground 22 part in your schematic to tell the simulator where ground is.

Tips: *See Complete RC Example 23 and TL Circuit Example 29 for a full walk through of schematic entry and simulation. You can add text Parameters 29 to your schematic to more easily change part values. Bubble Connectors 33 and Sub-Circuits 34 can make it easier to simulate large schematics.*

When CASTLE is started, it opens with a blank, **Page Size A** (11"x8.5"), schematic. Enter your schematic using the parts from the "Schematic" tab of the ribbon bar. If you want a larger schematic sheet, go to the "Home" tab of the Ribbon Bar 13 and select the **B or C page size**. The page size can be changed at any time.



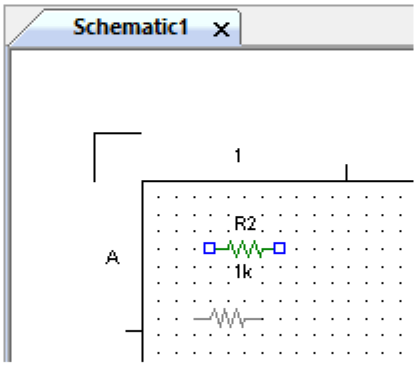
Screen shot of empty schematic created at program start (Click to expand).

The default file name for the schematic, "Schematic1", appears in the document tab. Use the Application Button 13 to bring up the save button and save your schematic with a different file name.

Note that CASTLE is programmed using Microsoft's standard MFC (Microsoft Foundation Classes) doc/view program architecture and classes so that usage of CASTLE is very similar to other familiar Windows programs. Several schematic windows can be simultaneously opened, each with it's own tab so you can easily switch between them.

Part Placement

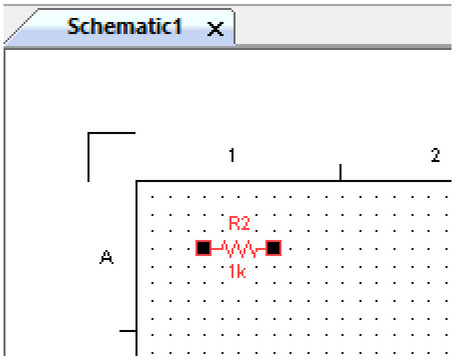
To place a part in the schematic, first click on the desired part in the Ribbon Bar. The mouse cursor will then change from an arrow to the selected part. Move this cursor to the desired position in the schematic and then left-click the mouse to place the part. To exit part placement mode, either press the escape key on the keyboard ("Esc" in top-left corner of keyboard), or select a different option from the ribbon bar (such as the Select Tool).



A resistor (R2) is placed in the schematic by left-clicking the mouse when in part placement mode. Note: Press "Esc" key to exit part placement mode.

Part Selection, Editing, Movement and Rotation

To modify a part, first it must be selected by clicking on it with the part **Select Tool** (located in the Tools section of the Schematic tab of the ribbon bar). When selected the part will turn red. Once a part is selected, you can use the Cut and Copy buttons in the "Home" tab of the ribbon bar. **Double-click** on a part to edit the part properties, such as name, value and initial conditions. See Units 7 for ways to specify a part value.



To edit a part, first select it by clicking on it with the part "Select Tool" and it will turn red.

Most parts can be **rotated** using the "r" key on the keyboard. (Note that parts rotate around their first terminal.). Parts can be **deleted** by pressing the "Delete" key on the keyboard while selected or using the Cut button on the ribbon.

The select tool can also be used to select a group of parts. To do this, hold the left mouse button down and drag to create a box around the parts to select.

Part Names

Note that CASTLE assigns a default name to new parts. CASTLE uses the first letter of the part name as a code for the part type. Therefore, you cannot change the first letter of a part name.

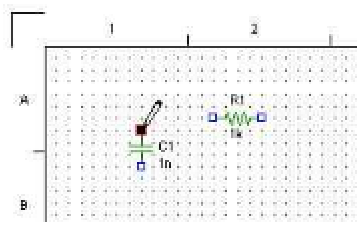
The Schematic Grid

By default placed parts are snapped to a grid and the grid is visible. Buttons in the **Grid** section of the Ribbon Bar can be used to hide the grid and remove the snap to grid feature.

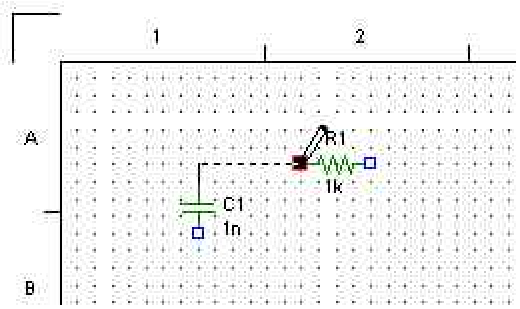
Wire Placement

Use the **Wire Drawing Tool** to connect parts with wires. The mouse cursor will change to a pen when in wire drawing mode. CASTLE assists you in connecting parts by changing the **appearance of the part nodes**. Unconnected terminals of a part (or wire) appear as empty blue squares. When the wire drawing tool is over an unconnected terminal, it will change to a black-filled, red square. Left-click on the terminal to begin drawing a wire. You can move the mouse to change the initial direction of the wire. Move the mouse to the terminal you wish to connect to and left-click again. The terminals are now connected by a black wire.

You can make several bends in your connecting wires by left-clicking at intermediate points. Press "Esc" key or a button in the Ribbon Bar to end wire placement mode. Hold the **Shift key** to force intermediate wire points to snap to the grid.



Begin drawing a wire by clicking on a terminal of a part with the Wire Drawing Tool. Note that the appearance of a terminal changes when tool is over the terminal.



Finish drawing a wire by left-clicking on the node you wish to connect to. Note: The initial direction of a wire can be changed by hovering directly in the desired direction.

See the Simple RC Example 23 for a more complete guide to schematic entry and simulation.

3.1 Keyboard Shortcuts

There are keyboard shortcuts for some functions. Some functions can only be done with the keyboard.

Keyboard functions:

Key	**Function**
Delete	Delete part (same as Cut)
r	Rotate part (only via keyboard)
w	Start wire placement
Esc	Exit current mode and enter Part Select mode
Shift	Forces wire points to snap to grid

3.2 Ground

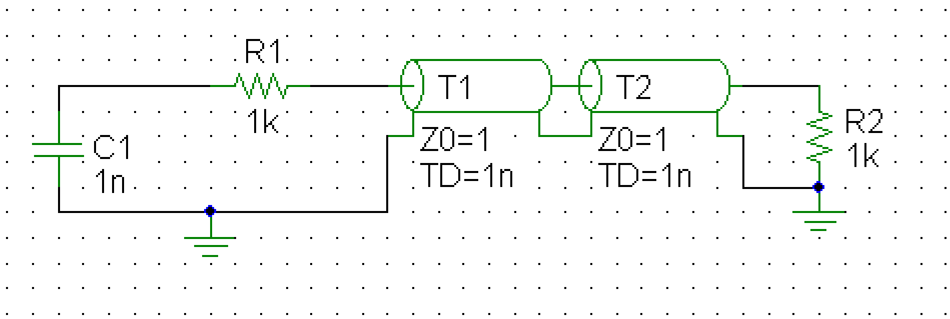
Because CASTLE is an MNA code (like SPICE), you most likely need to insert at least one Ground part in your schematic to tell the simulator where ground is.

Circuits at either end of a transmission-line are electrically isolated and therefore a ground is needed at each end.

Note: Grounds are not required when two transmission lines are joined together in a simple way as CASTLE will attempt to automatically assign ground in this case. This feature makes it easier to create schematics of circuits with many transmission line elements.

If you simulate without the required grounds, you will receive an error message that a node is floating.

Example:



Grounds are required in the circuits on the left end of T1 and the right end of T2. However, no ground is required at the junction of T1 and T2. Removing either of the grounds will result in an floating node error message when simulated.

3.3 Simple RC Example

Complete Schematic Entry and Simulation Example: Simple RC circuit

This example shows the basic method of drawing a schematic, simulating, and plotting the results for a simple RC circuit. The first step is to place the capacitor. First, click the capacitor symbol in the Ribbon Bar to switch to capacitor placement mode. At this point a capacitor and a small circle will appear in the schematic window. If you move your mouse around the schematic window the small circle and capacitor will follow the mouse. The small circle represents the mouse position and the left end of the capacitor (the positive end) will also follow the mouse, but be snapped to the grid as shown in Fig.1. Rotate the capacitor 90 degrees by pressing "R" key on your keyboard once so that the capacitor is vertical with the positive terminal on top. Note that all parts rotate about their positive terminal.

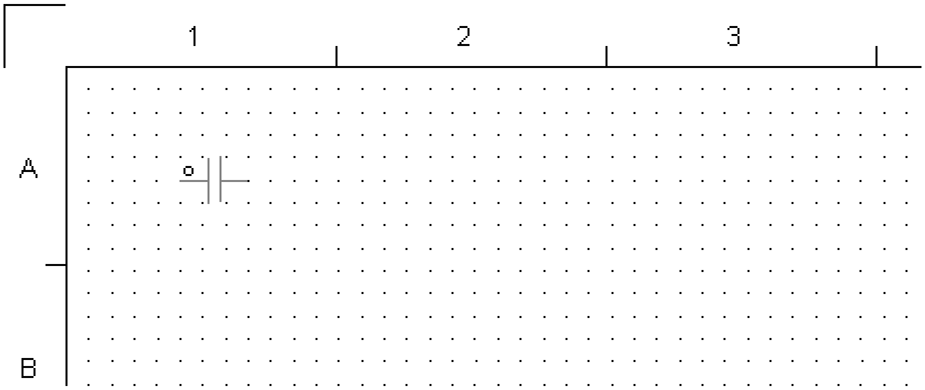


Figure.1. *Partial screenshot of Castle schematic in capacitor placement mode.*

Now, place the capacitor on the schematic by clicking the left mouse button once. Once placed, terminals will appear on the capacitor ends and default name and value labels appear. A new capacitor now follows the mouse for placement.

But, since we only need one, exit capacitor placement mode and enter part select mode by either pressing the “Esc” button on your keyboard or selecting the “part select” tool on the parts section of the Ribbon Bar. Now, a standard mouse arrow icon follows the mouse. Use this tool for selecting parts or labels. Select the capacitor by clicking it. The capacitor will turn red when selected. It can now be rotated or moved (by dragging with the mouse). Double-click on the capacitor to bring up the edit dialog box. Change the value to 100n and change the IC (initial condition) to “IC=1”. Click the “Visible” checkbox so that the IC will be displayed on the schematic and then push “OK”. Now, your schematic should appear like that in Fig. 2. Note that all part values and units use the convention of SPICE (i.e., 100n is 100e+9, 1m = 1e-3, 1MEG=1e+6; capacitance in F, inductance in H, resistance in Ohms). (See Units 7) Note also that all fields are case insensitive in CASTLE.

Next, place a resistor to the right of the capacitor by selecting the place resistor tool from the Ribbon Bar and clicking on the schematic, just the same way as we did with the capacitor. Double-click the resistor and change it’s value to 2. Your schematic should now appear similar to that in Fig. 3.

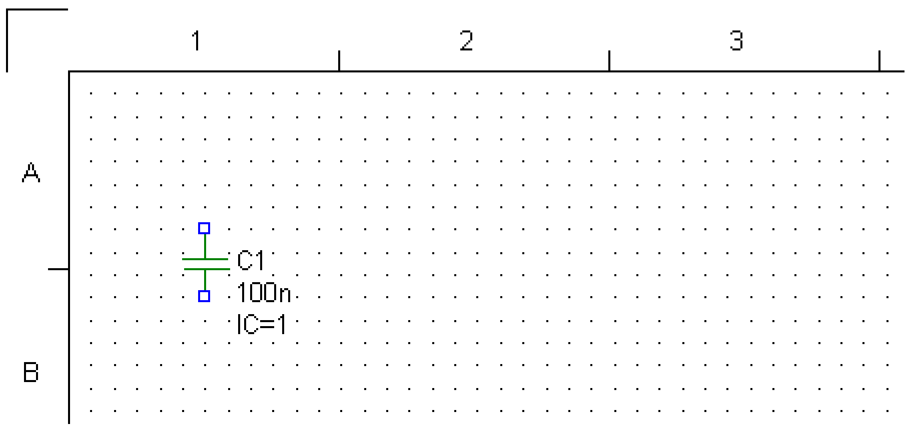


Figure 2. *Partial screenshot showing placed capacitor with value set to 100 nF and IC set to 1 V and made visible.*

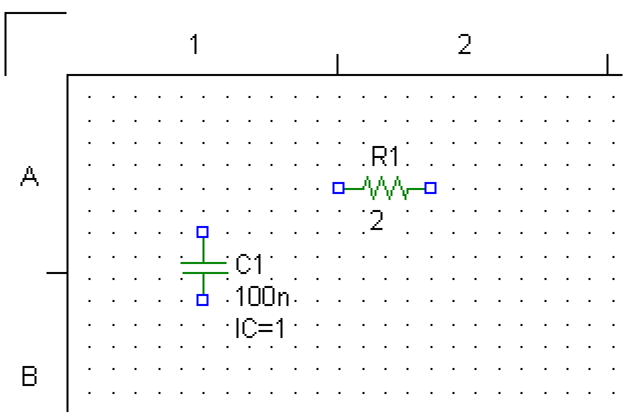


Figure 3. *Partial screenshot showing placed resistor with value set to 2 Ohms.*

Next, we need to add two wires to connect the components. Switch to wiring mode by selecting the wiring tool from the parts toolbar (fourth button from the left) or by pressing the “W” key on your keyboard. A pencil icon will now follow the mouse. Move this pencil tip to the top terminal of the capacitor until that terminal is highlighted (changes color and becomes filled in black). Now, move the mouse vertically and then to the right toward the resistor’s left terminal. You should see a solid vertical wire extending from the capacitor’s top terminal and a dashed horizontal wire to the current mouse position. Note that you can escape from wiring mode by selecting the part select tool (or any other tool). Move the mouse to highlight the left terminal of the resistor and then click the mouse to complete the wire. The schematic should now appear as in Fig. 4. Note that wired terminals are hidden. If a wire end does not connect to a part, a terminal appears at the end of the wire. You can delete a wire segment (or a part) by selecting them with the part select tool and pressing the “Delete” key on your keyboard. You can select multiple parts by dragging a box around several parts while in part select mode. Selected parts appear in red.

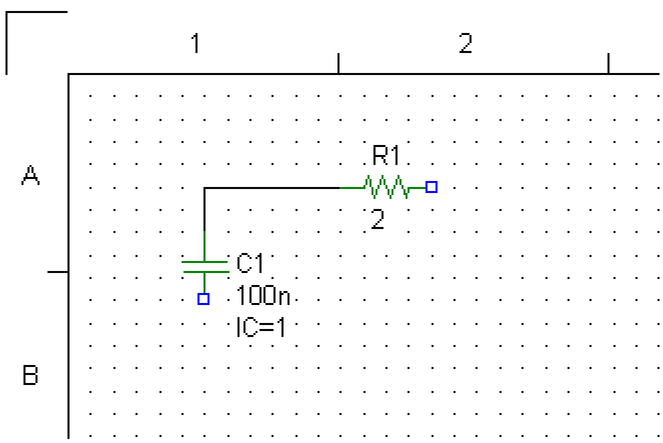


Figure 4. *Partial screenshot showing wire connecting C1 and R1.*

Next, draw a wire from the bottom terminal of the capacitor to the right terminal of the resistor using the same technique. Note that part labels can be moved by selecting them with the part select tool and then dragging them to the new location.

The final step is to place the ground node. Select the ground tool from the parts Ribbon Bar and move the mouse to the bottom terminal of the capacitor. Once over this terminal, the terminal will be highlighted. Click the left mouse button to place the ground at the bottom capacitor terminal. The ground is then placed and a node symbol appears at the junction of the capacitor terminal, wire to the resistor, and the ground’s terminal. Node symbols mark the junction of more than two terminals. Your schematic should now appear as in Fig. 5. Note that the ground terminal could also have been rotated and/or placed on the resistor’s right end or the wire between resistor and capacitor with the same affect. (Note: Ground terminals are not required at simple connections between transmission-line elements)

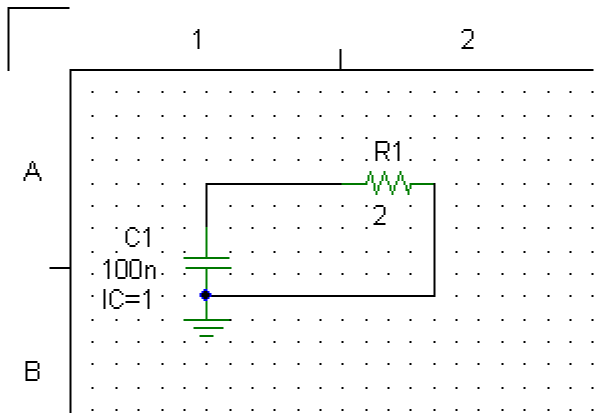


Figure 5. *Partial screenshot showing completed RC schematic.*

Netlist Generation

The schematic entry of this simple RC circuit is now complete. Before simulating you may examine the automatically generated netlist by clicking the "Netlist" button in the Ribbon Bar. The result should appear as in Fig. 6. This netlist is very similar to that used in SPICE (in fact, simple netlists are usable without modification in SPICE). This circuit contains two nodes, "N5" and "0". "0" represents the ground node and "N5" is an automated name for the connection from the positive terminals of capacitor and resistor. Note that the positive terminal is listed first, followed by negative terminal, then part value in the netlist. Note that CASTLE internally treats capacitors as single timestep long transmission-line elements with center and outer conductors tied together to provide two terminals.

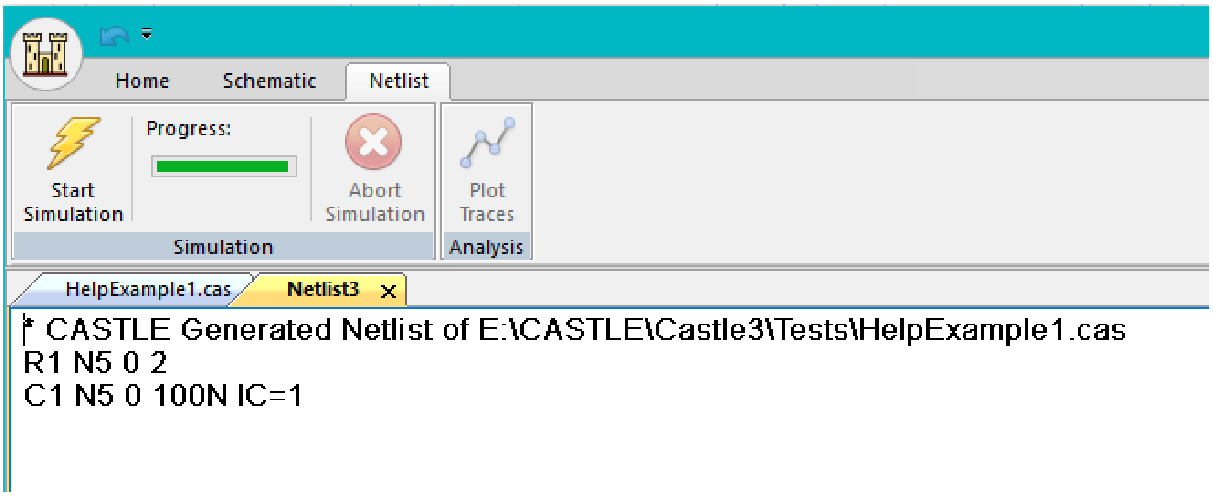


Figure 6. *Netlist for simple RC circuit.*

Before simulating, it is advised to save your schematic. Do this by selecting "Save" from the Application Button 13 popup menu. By default, CASTLE schematics have a ".cas" filename extension.

Simulate this simple RC circuit by pushing the "Start Simulation" button the simulator toolbar. The simulator dialog box appears where you can select the time step, time span and start time (note that end time = start time + time span).

Change the "Time Span" to "1000NS" and push the "Simulate" button. You should then quickly see a "Simulation completed" message. When you click "OK", the "Select Traces" dialog automatically appears. Here you select traces to plot or export.

The available traces are broken down into several categories in the top left list box, "Trace Type:". Select "Part Voltages" and "V(C1)" and "V(R1)" should appear in the top right list box, "Trace List", where they can be selected by double-clicking. Select both V(C1) and it should be added the "Selected Traces" text box. Now, click the "Plot Traces" button to see a plot of V(C1) versus time. You can zoom into a portion of the plot by holding down the "Shift" button and dragging a box over the portion to zoom to. Note that the regular zoom in and out buttons also work, but they scale the plot without changing the range of data displayed.

You can close this plot by selecting "Close" from the "File" menu or by clicking the smaller "x" in the top right corner inside the main window. Do not push the big "X" of the main window or you'll close CASTLE! You could also just leave the plot and return to the schematic by selecting the schematic window from the window list in the "Window" menu. Windows can also be cascaded or tiled.

TIP: You can make further plots or export data without simulating again. Just return to the schematic and push the "Trace Select" button, which becomes usable only after the schematic has been simulated. All simulation data remains in memory until the schematic is closed.

Adding Probes to the Schematic

Voltage and current probes are not really needed as all voltages and currents can be viewed after a simulation. However, you can add explicit current and voltage probes, if desired. In the simple RC schematic previously described, we can add a voltage probe to positive terminal of the capacitor by pushing the "Voltage Probe" part button on the part toolbar (10th from the left) and then placing the probe, e.g., on the top-left bend in the wire that connects C1 and R1. Note that this bend should be highlighted when placing to ensure proper connection. The schematic should now look as shown in Fig. 7.

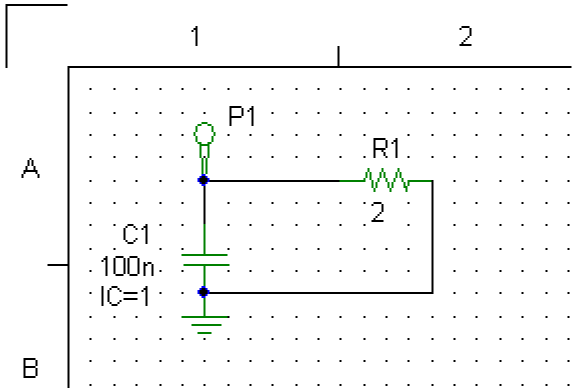


Figure 7. *Voltage probe added to simple RC circuit schematic.*

Now, you can see in the netlist that the node name for the positive ends of C1 and R1 has been renamed to “P1”, the probe name. Also, after simulating, you will see that “V(P1)” is automatically added to the “Selected Traces” box, allowing for quick plotting or saving.

We can also add a current probe to the circuit. Select and delete the wire going downward from the right end of R1. Switch to current probe placement mode by clicking the current probe button in the part toolbar (third from the right) rotate the probe, if desired, by pressing “R” once. Then place the probe where the wire was. The probe will be labeled “V1”. Note that in CASTLE (as well as SPICE) current probes are implemented as voltage sources with their voltage set to zero.

Use wires to connect the probe to the resistor and existing wire by entering wiring mode (press “W”) and drawing wires between terminals. The schematic should now look similar to Fig. 8. After simulating, “I(V1)” will be added automatically to the “Selected Traces” box.

TIP: Remember that you can use Undo/Redo to correct mistakes.

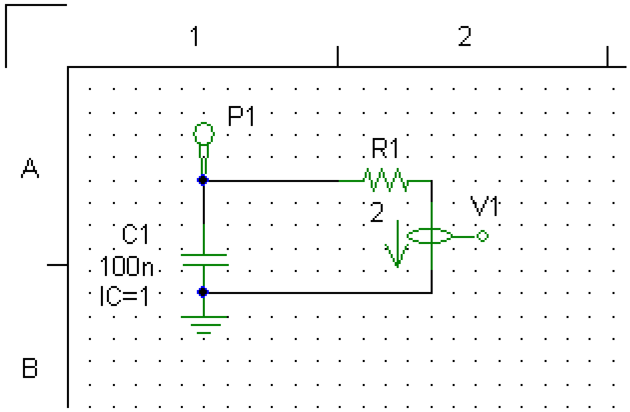


Figure 8. *Current probe added to simple RC circuit schematic.*

Adding Text Labels to the Schematic

The text tool allows you to place text labels on the schematic. Note that normal text will not affect the netlist. When you select the text tool a dialog box will prompt you to enter the desired text. You can then place this text in one or more places on the schematic. Edit text labels by double-clicking them.

Using Text Parameters

Text labels that begin with the keyword "PARAMS:" are utilized to allow easy setting of frequently changed part values or parameters. For example, with the simple RC schematic, change the value of the resistor from "2" to "RLOAD". Next, create and place a text label with the text "PARAMS: RLOAD=2". You can check the netlist to verify that R1 is assigned the value of 2. An example is shown in Fig. 9. Parameters are also used to pass values to subcircuits, described later.

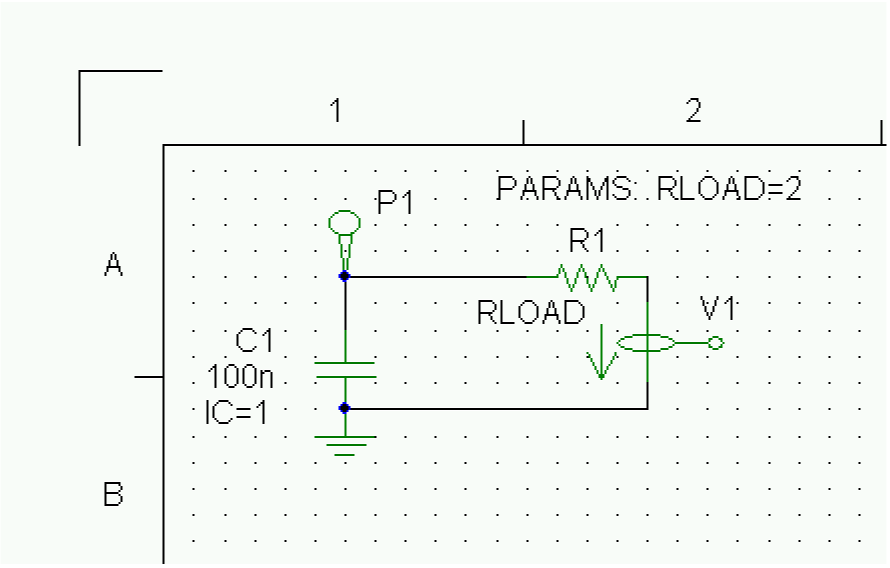


Figure 9. *Example of using text labels to set simulation parameters.*

3.4 TL Circuit Example

Transmission-Line Circuit Example: Simple Pulsed-Power Circuit

This example demonstrates how to use transmission-line elements in CASTLE.

Place three transmission-line elements on the schematic using by pushing the transmission-line placement button on the Ribbon Bar and then clicking the mouse on the schematic. Place the elements as shown in Fig. 11. Note that transmission-line elements can be rotated (using the "R" key on the keyboard) and will rotate about the left, positive (center) terminal. But, place these elements in the default orientation.

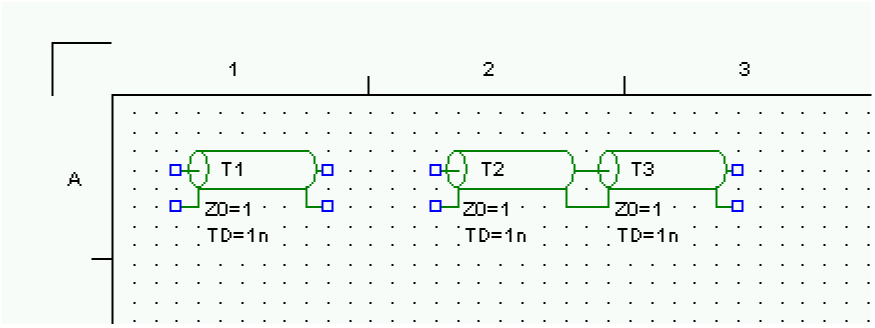


***Figure 11**. Three transmission-line elements placed on the schematic.*

Notice that the CASTLE automatically recognizes that T2 and T3 are joined and the terminals at the junction are hidden. It is not required to place a ground node at this junction.

Double-click the elements to edit their values. Change T1 to Z0= 3, TD= 25n, IC= 4MEG, and make the IC visible. Change T2 to Z0= 3, TD= 10n. Change T3 to Z0= 1.5, TD= 50n.

Place a switch element between the center terminals of T1 and T2. (the switch placement mode button is fourth from the right on the parts toolbar). Then, place a resistor at the end of T3 (resistor part button is the left button). If you want, rotate the resistor to the vertical position using the "R" button on the keyboard either before placing or after by first selecting the resistor with a mouse click. Change the resistor value to 2. You can view the switch settings by double-clicking on it. But, this example will use the default settings.

Use wires to connect center terminals of T1 and T2 to the switch ends and to connect the outer terminals of T1 and T2 to each other. Use wires to connect T3 to the resistor. The schematic should now look like Fig. 12. Note that no connection is required on the left end of T1.

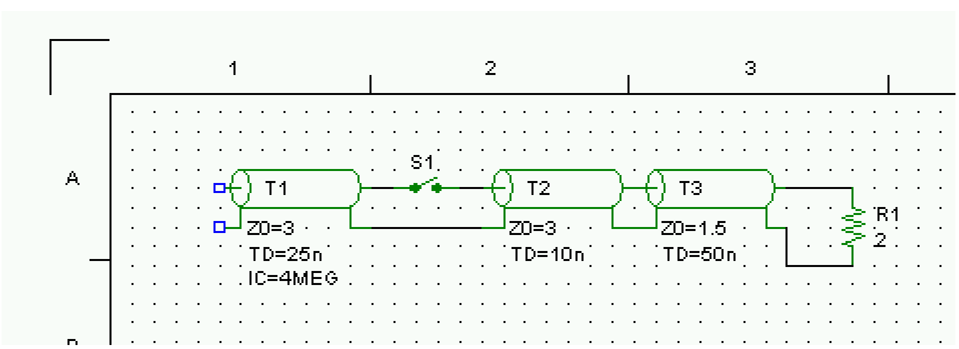


***Figure 12.** Basic transmission-line circuit wired with switch and resistor.*

Explicit grounding is required for some transmission-line junctions, but CASTLE automatically assigns ground for several types of junctions. (If desired, push the “Netlist” button to examine how ground nodes are assigned). The left end of T1 is open and CASTLE will assign ground to the left outer conductor terminal. The outer conductors of T1 and T2 are connected and neither of the center conductors attached to S1 are explicitly grounded, so CASTLE automatically assigns ground to the connected outer conductors. T2 and T3 are directly connected, so CASTLE automatically assigns ground to the connected outer conductors.

T3, however, is connected only to a parallel resistance. At this time, this type of connection requires an explicit ground. If you push the “Simulate” button at this time, you will get an error message that a node is floating. If you then were to examine the netlist, you will see that this is one of the nodes connecting T3 and R1.
Add an explicit ground terminal to the bottom terminal of R1 using the ground tool (8th button from the left on the parts toolbar). The completed schematic is shown in Fig. 13.

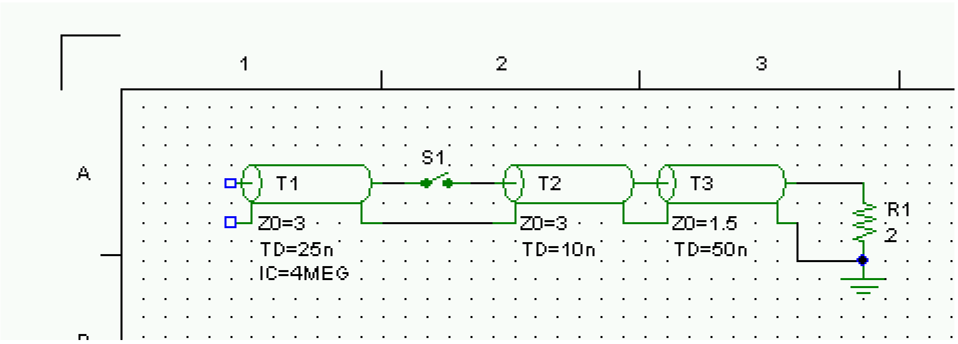


Figure 13. *Completed schematic of basic transmission-line circuit.*

You can now simulate this circuit by pressing the “Start Simulation” button on the Ribbon Bar. Change the time span to “200n” and begin the simulation by pushing the “Start Simulation” button in the simulation dialog box.

TIP: These simple simulations will finish instantly. However, you may have more complicated simulations that can take several minutes and possibly make the computer less responsive. If this happens and you wish to stop the simulation before it is complete, push the “Abort Simulation” button in the Ribbon Bar. It may take a second to take affect, but this will end the simulation early. Note that all simulations results are still available for the completed portion. A "Progress" bar in the Ribbon Bar indicates how much of the simulation has been completed.

Once the simulation is complete you push the "Plot Traces" button in the Ribbon Bar and select which traces to plot in the trace select dialog that appears. For example, to select the current in R1 click on “Part Currents” in the left list and then “I(R1)” in the right list. You can also select transmission-line voltage, current, or waves in the left list. The last letter “R” or “L” in the trace names in the right list indicates the right or left end location of the measurement. For example, to plot the voltage at the left end of T1, select “VL(T1)”. There are two waves at each end of a transmission-line, V1 and V2. V1 refers to the wave leaving an end and V2 refers to the wave entering an end. For example, to select the wave entering the right end of T3, double-click on “V2R(T3)”.

Note: *One advantage of CASTLE over most TL codes is the unrestricted topology. In this example, you can simply connect a capacitor across S1, if desired. You could also add resistors from either end of S1 to ground. Just be aware that in more complicated junctions you may have to specify the ground node.*

3.5 Sub-Circuits

Using Bubble Connectors

For large schematics, you may want to use bubble connectors instead of drawing long wires across the schematic. Bubble connectors are also used sub-circuits (described later). In the previous example, one could separate the circuit in two parts at the T2-T3 junction using bubble connectors. To demonstrate, first select the right part of the circuit, including T3 and R1 and drag them to the right a few grid points.

TIP: You can drag multiple components by selecting the all and the dragging them (with the left mouse button held down). You can select multiple components by either dragging a box around them while in part select mode and/or by clicking on them with the "Ctrl" key held down. Selected parts will appear in red. You can also copy, cut, and paste selected components.

Next, add bubble connectors to the center connectors of the right side of T2 and the left side of T3 by pushing the bubble placement mode button on the Ribbon Bar and clicking on the desired terminals once they are highlighted. Note that bubble connectors can also be rotated using the "R" key on the keyboard. Ensure that the two bubble connectors have matching labels, such as "A", so that CASTLE will recognize them as being connected. Do not use numbers for bubble labels as these are reserved for sub-circuit pin designations. The schematic should now look like that of Fig. 14. If you view the netlist you will notice that the only change is that the common node of T2 and T3 center conductors is now called "A", same as the bubble label. If you simulate this example you will get the same result as before.

Note that grounding is not required for the outer conductors of T2 and T3 because they have a simple connection. For a more complicated connection, either explicit grounding or bubbles must be used.

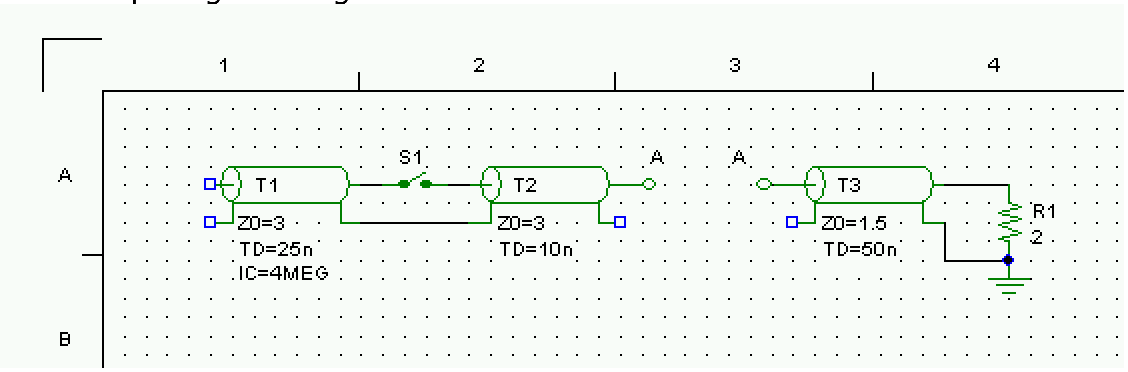


***Figure 14**. Transmission-line circuit example using bubbles to connect isolated segments.*

Sub-Circuits

Using subcircuits, you can represent a portion of one schematic with another schematic. This can be very useful when several instances of the same subcircuit in a main schematic. As an example, the same simple transmission-line example from the previous example will be split into a main schematic and a subcircuit schematic.

First, selected the entire right side of the circuit including T3, R1, and the right bubble connector. (You may want to save the current file under a different name at this point.) Click "Cut" from the "Edit" menu or push the scissors icon on the main toolbar to copy this subcircuit to the clipboard and remove it from this schematic. Next, select "New" from the "File" menu and create a new schematic. Paste the previously cut subcircuit into the new schematic. Rename the bubble connector from "A" to "sub1" (sub1, sub2, sub3, and sub4 are reserved keywords). The new subcircuit should look like that in Fig. 15. Note that the part labels may change when pasting them (T3 becomes T1 in the new schematic). This is to avoid duplicate part names within the same schematic. Just rename parts to their previous values. This in not really required though because subcircuit part labels can be the same as the main circuit without problems. Now, save this file with a new name and return to the main schematic (use the "Windows" menu to switch between schematics).

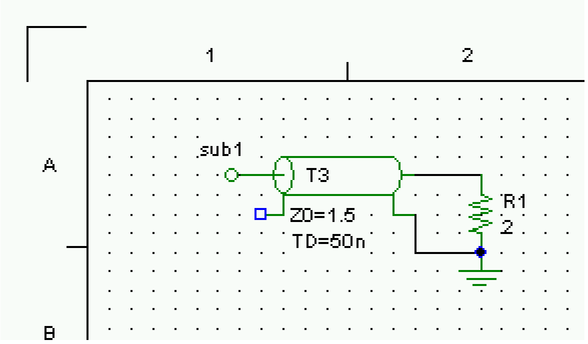


Figure 15. *An example subcircuit schematic.*

Next, delete the bubble on the main schematic and replace it with a subcircuit element (the subcircuit mode tool button is second from the right on the parts toolbar). Place the subcircuit element just to the right of T2, but leave a space between terminals. Then, draw a wire from the left, center terminal of T2 to the "1" pin of the subcircuit element, "X1". The main circuit should now appear as in Fig. 16. Note that you could have connected up the outer conductor terminal to pin 2, but this is not necessary since the ground node is common to all schematics.

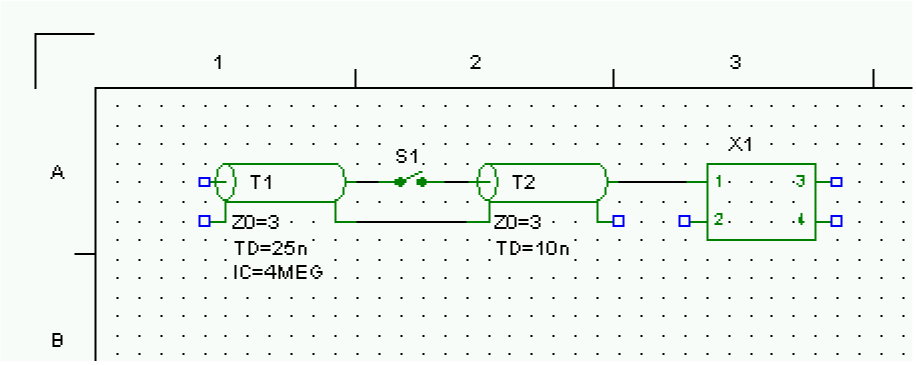


Figure 16. *An example circuit utilizing a sub-circuit element.*

Finally, you must tell the sub-circuit element which schematic file contains the sub-circuit it represents. Double-click the sub-circuit element to bring up the settings dialog. Click the "Browse" button and find the file containing the sub-circuit previously saved. Note from this dialog you can examine the sub-circuit by clicking the "Open" button and then hide it again with the "Close" button.

One can also pass parameters to the sub-circuit. This is useful when there are several instances of the same subcircuit but with some different element values. In this example, you can enter "Rload=2" in the "Parameters" text box in the sub-circuit element edit dialog and then change the resistance of the R1 in the sub-circuit from "2" to "Rload". Note that you must save the sub-circuit schematic for changes to take affect in the main circuit.

If you new "View Netlist" in the main schematic you can verify that the parameter value was passed. Each sub-circuit element will have it's own section in the netlist and it's node names with have an underscore and then the sub-circuit label as a suffix. If you then simulate this circuit and begin selecting traces, be aware that the part names for those parts in a sub-circuit will also have their labels appended by an underscore and the sub-circuit label. This is to ensure unique names when several instances of a sub-circuit are present.

Note that this example only uses 1 of 4 available nodes for the sub-circuit. And, since the ground node is common, the sub-circuit can actually share 5 nodes with the main circuit.

TIP: It is possible to have sub-circuits within sub-circuits. Also, you can pass parameters from the main circuit to these sub-sub-circuits. This can be achieved, e.g., by passing a parameter like "RLoad2=Rload" from the sub-circuit to the sub-sub-circuit and also passing "Rload=2" from the main circuit to the sub-circuit. In a similar way, you can use text parameter labels in the main schematic, e.g., "PARAMS: Rload=2", to pass parameters to sub-circuits.

This page is intentionally left blank.
Remove this text from the manual
template if you want it completely blank.

Netlist Editor

4 Netlist Editor

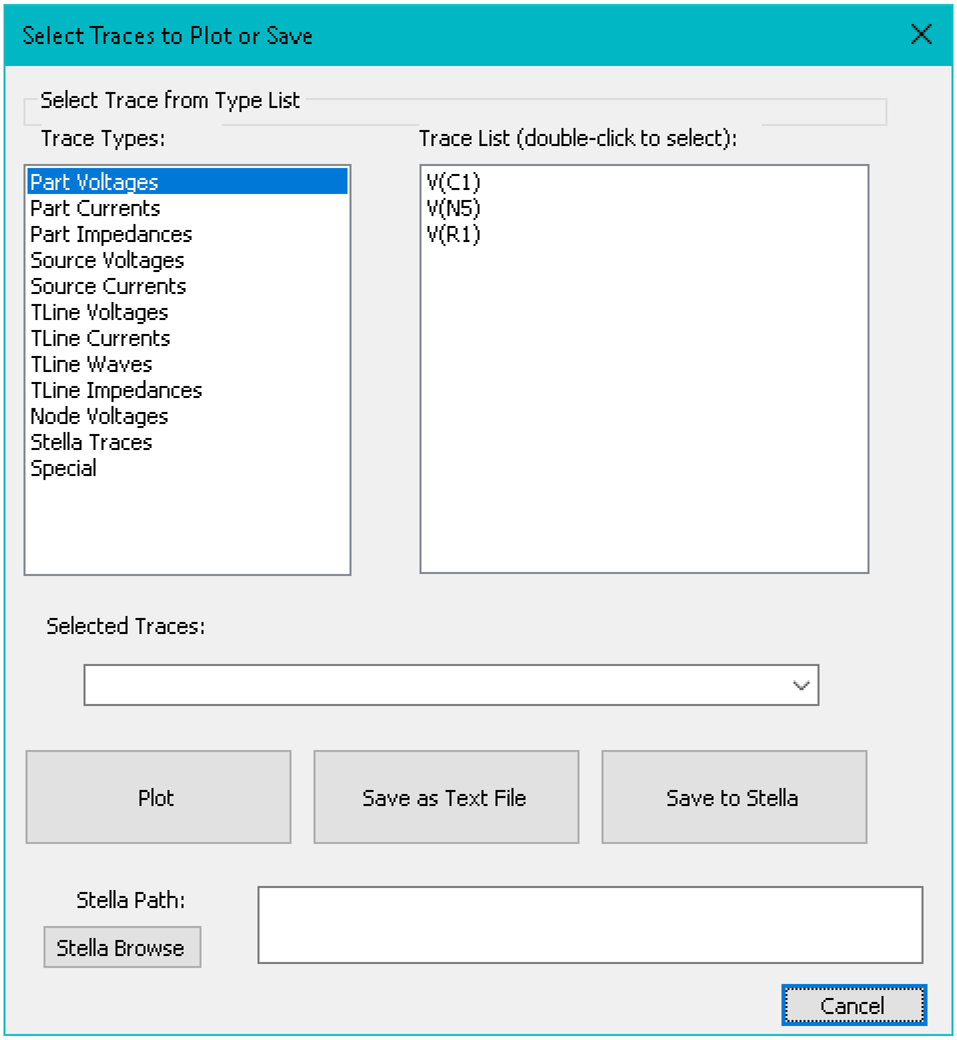
The Netlist Editor Window is created from the Schematic Editor Window 18 when either the "Netlist and Simulate" or the "Netlist" button is pushed on the Ribbon Bar 13. This window shows the text based "Netlist" created from the graphical schematic. The Netlist contains one line for every component in the circuit, along with parameters of the component and to which nodes it is connected to. The Netlist is the input to the Simulator. Although usually not necessary, it is sometimes useful to review the Netlist prior to simulation to make sure your schematic was interpreted correctly.

See Netlist Generation 26 in the Simple RC Circuit Example for some notes on the Netlist.

After simulation, the Plot Traces 38 dialog window will appear for Plotting and Saving simulation results. Later, you can use the "Plot Traces" button on the Ribbon Bar to reactive this window for a netlist that has been simulated.

4.1 Plot Traces Window

Automatically after simulation, or by pressing the "Plot Traces" button on the Ribbon Bar, the Plot Traces dialog window will appear. Select a type of trace from the left list window and all the available traces of that type will appear in the right list window. Double click on a trace in this Trace List to have it automatically added to the "Selected Traces" line. You can also manually enter the name of the traces you wish to plot in the Selected Traces Line (with a space between names). Recently plotted traces can be selected with the drop list access at the right. Next, you can Plot the Selected Traces with the Plot button or save them as text files or Stella 8 traces with the other buttons. Use the Stella Browse button to select where Stella traces will be saved.



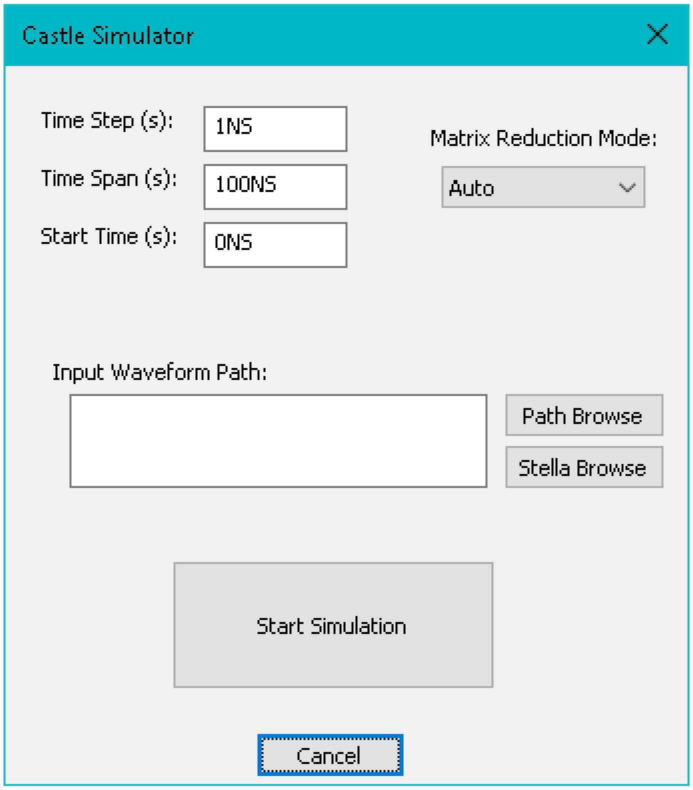
The Plot Traces dialog window.

This page is intentionally left blank.
Remove this text from the manual
template if you want it completely blank.

Simulate Window

5 Simulate Window

The "Simulate" dialog window appears when you click on "Netlist and Simulate" button from a schematic or when you click on "Simulate" button from a Netlist. Here you can specify the parameters of the simulation, such as time step, time span and start time. You can also specify the Matrix Reduction Mode used by CASTLE. If your simulation uses text or Stella traces as inputs, you can specify the input path. Finally, press "Start Simulation" to begin.



The Simulate Dialog Window

Matrix Reduction Mode

CASTLE can greatly increase simulation speed by breaking the main matrix to be solved many smaller matrices. This is done using the transit-time isolation properties of transmission lines. There are four options:

1. **Auto.** CASTLE will automatically determine the best approach.
2. **Maximal.** CASTLE will reduce the matrix as much as possible.
3. **Minimal.** CASTLE will reduce the matrix into two smaller matrices, one that includes all the time varying impedance elements and another with all the other components.
4. **None.** CASTLE will use a single matrix for all components.

CASTLE Simulator Details

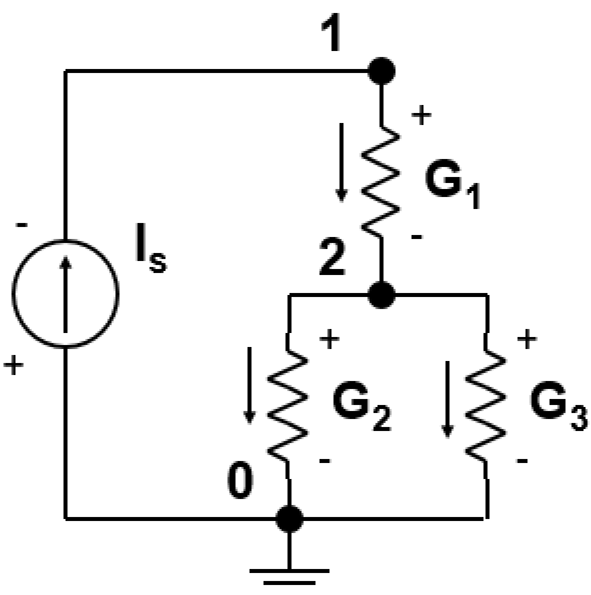
6 CASTLE Simulator Details

This section contains pages that describe the inner workings of CASTLE.

CASTLE is an MNA (Modified Nodal Analysis) based circuit solver.
To explain MNA, we first examine regular Nodal Analysis.

6.1 Nodal Analysis

Nodal Analysis solves a circuit using KCL (Kirchhoff's Current Law) on all nodes in a circuit.
Consider the circuit below:



This example circuit contains one current source, I_S, and three resistors with conductance (1/resistance) values of G_1, G_2 and G_3.
KCL dictates that the sum off all currents entering a node is zero.
The voltage at node #0 is ground, so fixed at zero. So, the #equations=#nodes-1
T
We can write equations for KCL at nodes 1 and 2 by expressing the voltages at nodes 1 and 2 as V_1 and $V_{2:}$

G1*(V1-V2) = Is
G2*V2+G3*V2-G1*(V1-V2)=0

These equations can be solved by hand to find that:

V2 = Is /(G2+G3) = Is*(R2||R3)
V1 = Is /(G2+G3) + Is /G1 = Is*(R1+(R2||R3))

where $(R_2||R_3)$ is the parallel combination of R_2 and R_3, or $R_2*R_3/(R_2+R_3)$

For solving large circuit, it is much more convenient to write the equation in matrix form.
The solution can be viewed as solving the linear equation: **GV=I**
Where, G is the admittance matrix, V is the node voltage vector and I is the current source vector.
Note: The direction of positive current is defined as entering the + terminal and leaving the - terminal.
There is one row in the G, V and I for every node in the circuit (not counting the ground node).

So, we can express the two KCL node equations from above in matrix form as:

$$\begin{bmatrix} G_1 & -G_1 \\ -G_1 & G_1+G_2+G_3 \end{bmatrix}\begin{bmatrix} V_1 \\ V_2 \end{bmatrix}=\begin{bmatrix} I_S \\ 0 \end{bmatrix}$$

This matrix can be solved to yield the same result for V_1 and V_2 given above.
See Matrix Solving for more information on how CASTLE solves the matrix.
The next section discusses Modified Nodal Analysis 45

6.2 Modified Nodal Analysis (MNA)

The problem with regular nodal analysis is that it can't handle voltage sources. MNA adds the currents through voltages to the equation as a way to solve circuits with voltage sources.

So, the circuit equation (**GV=I)** becomes:

$$\begin{bmatrix} G & F \\ B & R \end{bmatrix}\begin{bmatrix} V \\ I \end{bmatrix}=\begin{bmatrix} C \\ E \end{bmatrix}$$

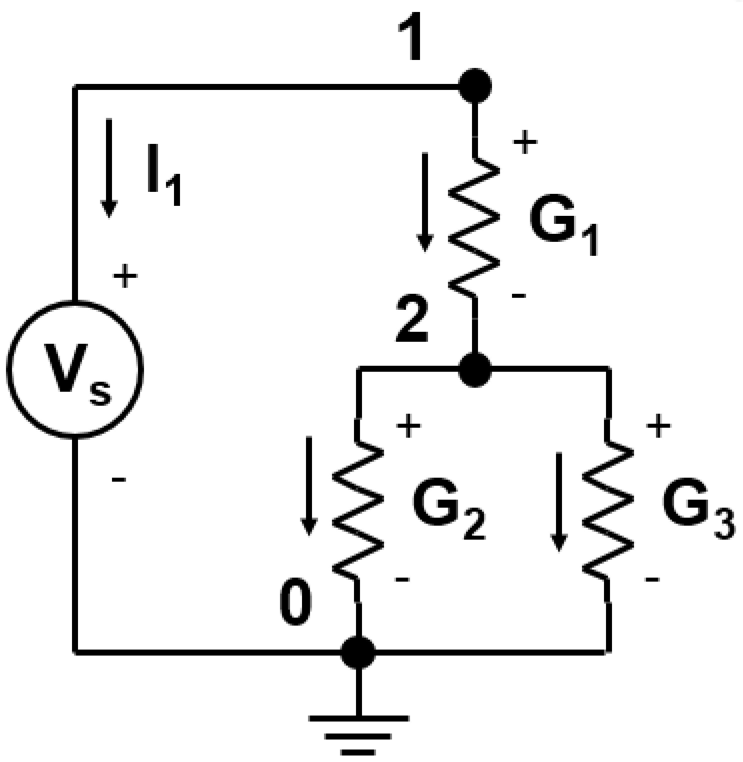
The **G** matrix is mostly the same, but we essentially add one row and one column for each voltage source in the circuit.

The **F** matrix has one column for every voltage source in the circuit with a 1 at the + terminal and a -1 at the - terminal.
The **B** matrix is the transpose of **F** and **R**==0.

The **V** vector is still the voltage at every node, but the **I** vector is now the current through each voltage source.

The **C** vector (what was I in nodal analysis[44]) is the current source vector and **E** is the voltage source vector (the voltage of each voltage source).

Example. We replace the current source from the nodal analysis example with a voltage source:



The circuit equations become:

G1*(V1-V2) +I1=0
G2*V2+G3*V2-G1*(V1-V2)=0
V1=Vs

Which can be converted to this matrix form:

$$\begin{bmatrix} G_1 & -G_1 & 1 \\ -G_1 & G_1 + G_2 + G_3 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ V_s \end{bmatrix}$$

See Matrix Solution Methods[47] for how CASTLE solves this kind of matrix.

Note that the 0 on the main diagonal (The R section of the new G matrix) is a problem that will be solved by Pivoting 48.

6.3 Matrix Solution Methods

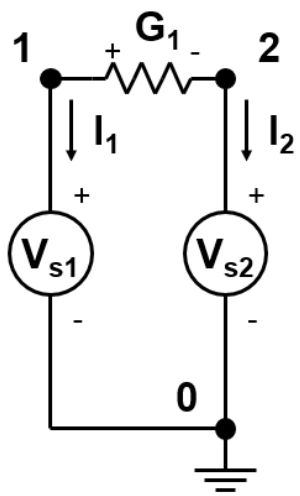
Here are a few ways to solve a linear matrix Ax=B (GV=I in this case):

- **Hard way: Matrix Inversion**
  - A-1=adj(A)/|A| → A-1Ax=A-1b → x=A-1b
- **Better way: Gaussian Elimination**
  - $$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & a'_{32} & a'_{33} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} b_1 \\ b'_2 \\ b'_3 \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \end{bmatrix}$$
- **Best way: LU Decomposition (what CASTLE uses)**
  - $Ax=b \rightarrow LUx=b$
    $\rightarrow Ux=L^{-1}b \rightarrow y=L^{-1}b$
    $\rightarrow x=U^{-1}y$

    $$U=\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \quad L=\begin{bmatrix} 1 & 0 & 0 \\ a_{21}/a_{11} & 1 & 0 \\ a_{31}/a_{11} & a'_{32}/a'_{22} & 1 \end{bmatrix}$$
  - No extra memory required using Crout's Method as L and U stored in A's memory
  - Can reuse L and U if only b changes
  - Trouble if there is a 0 on the main diagonal → Need Pivoting 48

6.3.1 Pivoting

Pivoting Required for LU Decomp



- **Second voltage source adds a zero to the main diagonal**

$$\begin{bmatrix} G_1 & -G_1 & 1 & 0 \\ -G_1 & G_1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ V_{S1} \\ V_{S2} \end{bmatrix}$$

- **Simplest pivoting scheme, *preordering*, solves this by swapping the source current equation row with the node voltage equation row of the positive terminal of the source**

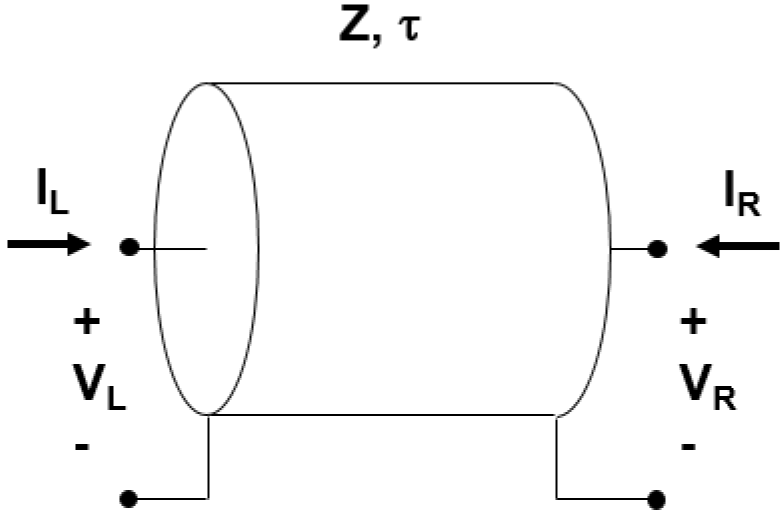
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ G_1 & -G_1 & 1 & 0 \\ -G_1 & G_1 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} V_{S1} \\ V_{S2} \\ 0 \\ 0 \end{bmatrix}$$

- **Numerical Recipes' LUDCMP routine reorders rows for maximum values on diagonal.**

6.4 Transmission-Line Models

Here we describe how CASTLE handles transmission lines.
From the outside of a transmission line we can envision an External Model that knows only the currents and voltages at each end:

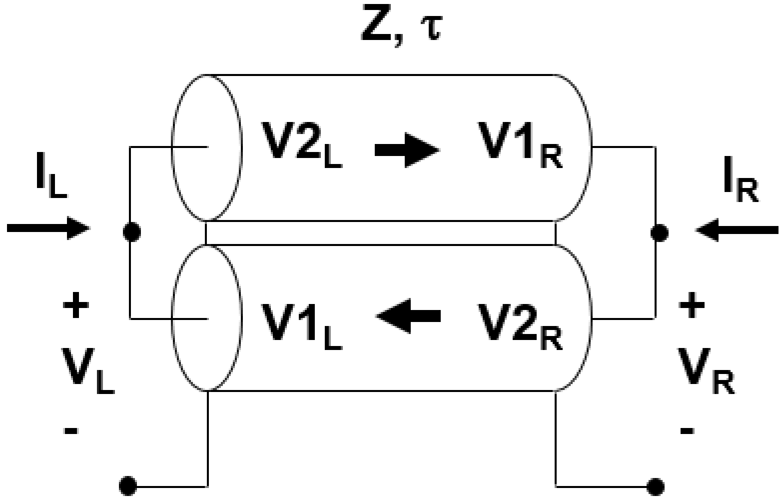
External Model



- Z is the impedance of the line
- τ is the one-way transit time
- No current flows on outer surface
- If a fast pulse is sent in the left side at $t=t_0$ and the right side is an open circuit, then $V_R[t_0+\tau]=2*V_L[t_0]$, $V_L[t_0+2*\tau]=V_L[t_0]$
- If right is a short circuit, $V_L[t_0+2*\tau]=-V_L[t_0]$

Transmission line codes (Such as Stella) work by calculating the waves going into each transmission-line end at each time step (V2) by forming equations around the waves coming out of each end of the transmission line (V1) and then solving how all the waves interact at each junction in terms of reflection and transmission coefficients:

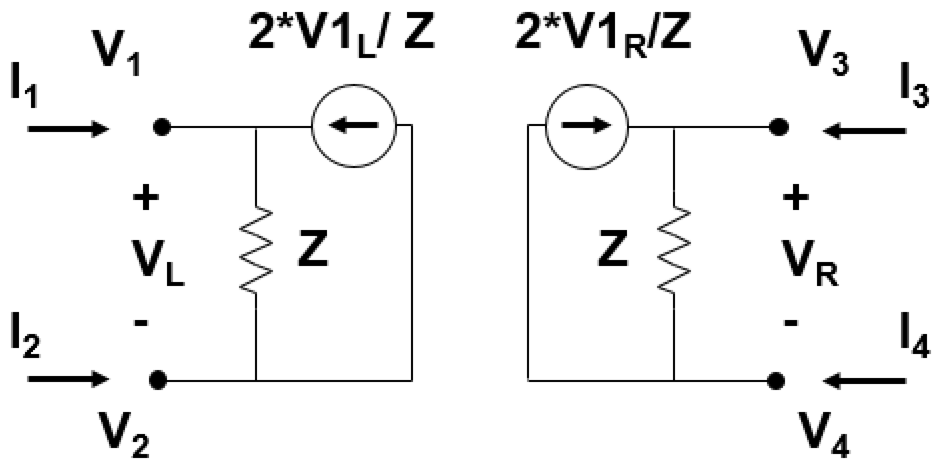
Transmission-Line Model



- V1 are the waves exiting the line
- V2 are the waves entering the line
- $V1[t] = V2[t-\tau]$
- $R = (Z_L-Z) / (Z_L+Z)$ = Reflection Coef.
- $T = 1 + R$ = Transmission Coef.
- M = portion of T_i from connected ends, i
- $V2 = R*V1 + \sum_{i=2..n} V1_i * M_i$ *($M_1=R$)*
- $V = V1 + V2$
- $I = (V2-V1) / Z = (V-2*V1) / Z$

In CASTLE, each end of a transmission line is treated as an isolated circuit. We do calculate the same V1 wave at each time step, as in transmission line codes, but we apply this to the value of a current source:

CASTLE Model



- $V1[t] = V2[t-\tau]$
  - $V1_R[t] = V2_L[t-\tau]$
  - $V1_L[t] = V2_R[t-\tau]$
- $I = (V-2*V1) / Z$
  - $I_1 = -I_2 = (V_1 - V_2)/ Z - 2*V1_L/ Z$
  - $I_3 = -I_4 = (V_3 - V_4)/Z - 2*V1_R/ Z$
- Solve **GV=I**
- $V2 = V – V1$
  - $V2_L = V_L-V1_L = V_1-V_2-V1_L$
  - $V2_R = V_R-V1_R = V_3-V_4-V1_R$

This page is intentionally left blank.
Remove this text from the manual template if you want it completely blank.

- G -

- K -

- M -

- P -

- R -

- S -