

# Appendix D

## *P*Spice for Windows

There are several computer software packages, such as *Spice*, *Mathcad*, *Quattro*, *MATLAB*, and *Maple*, which can be used for circuit analysis. The most popular is *Spice*, which stands for *Simulation Program with Integrated-Circuit Emphasis*. *Spice* was developed at the Department of Electrical and Computer Engineering at the University of California at Berkeley in the 1970s for mainframe computers. Since then about 20 versions have been developed. *P*Spice, a version of *Spice* for personal computers, was developed by MicroSim Corporation in California and made available in 1984 and later by OrCAD and cadence. *P*Spice has been made available in different operating systems (DOS, Windows, Unix, etc.).

If you do not have access to *P*Spice, you can find out how to obtain a free student copy by going to the text website ([www.mhhe.com/alexander](http://www.mhhe.com/alexander)). The instructions and examples in this appendix were developed for version 9.1, but also work for later versions.

Assuming that you are using Windows and have the *P*Spice software installed in your computer, you can access *P*Spice by clicking the Start icon on the left-hand corner of your PC; drag the cursor to Programs, to *P*Spice students and to *Schematics*, and then click as shown in Fig. D.1.

The objective of this appendix is to provide a short tutorial on using the Windows-based *P*Spice on an IBM PC or equivalent.

*P*Spice can analyze up to roughly 130 elements and 100 nodes. It is capable of performing three major types of circuit analysis: dc analysis, transient analysis, and ac analysis. In addition, it can also perform transfer function analysis, Fourier analysis, and operating point analysis. The circuit can contain resistors, inductors, capacitors, independent and dependent voltage and current sources, op amps, transformers, transmission lines, and semiconductor devices.

We will assume that you are familiar with using the Microsoft Windows operating system and that *P*Spice for Windows is already installed in your computer. As with any standard Windows application, *P*Spice provides an on-line help system.

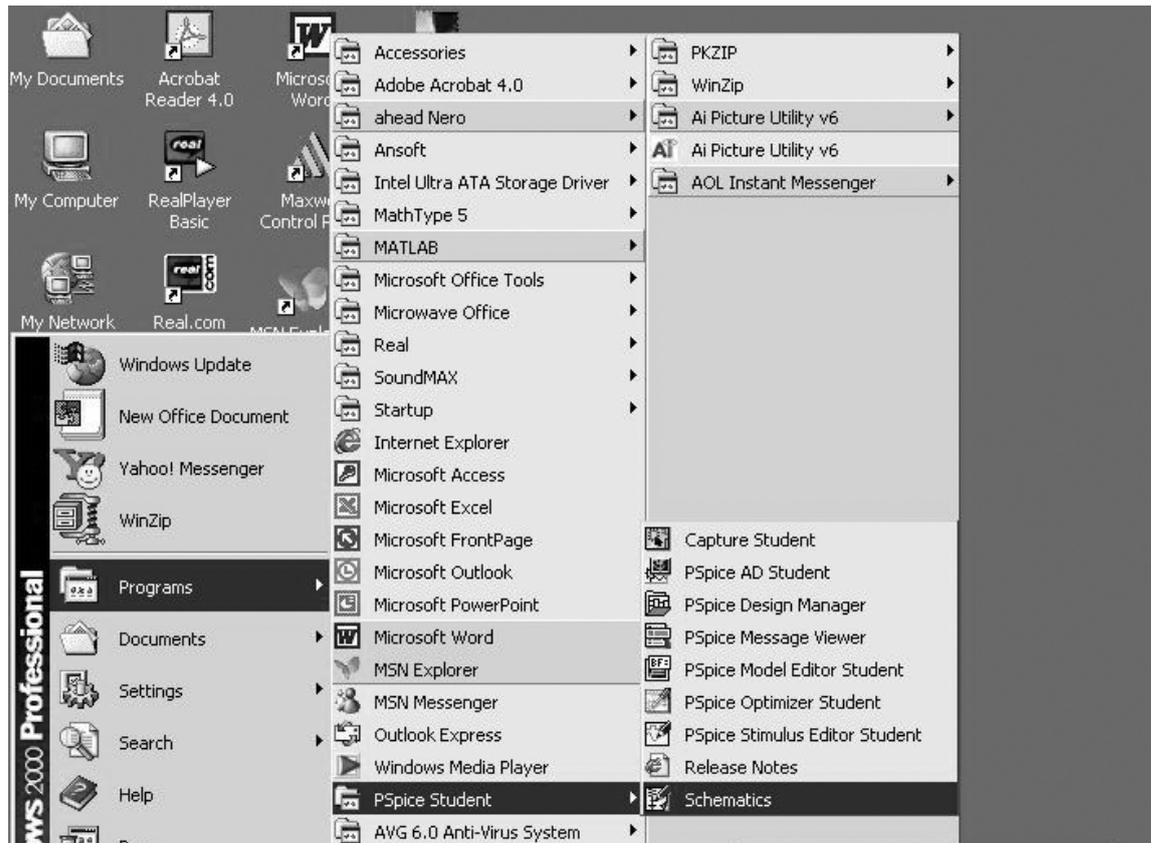
If you need help on any topic at any level, click Help, click Help Topics, click Search, and type in the topic.

### D.1 Design Center for Windows

In earlier versions of *P*Spice prior to Windows 95, *P*Spice for Windows is formally known as the *MicroSim Design Center*, which is a computer environment for simulating electric circuits. The *Design Center for Windows* includes the following programs:

*Schematics*: This program is a graphical editor used to draw the circuit to be simulated on the screen. It allows the user to enter

The student version of *P*Spice can be obtained free of charge.



**Figure D.1**  
Accessing *PSpice* on Windows.

the components, wire the components together to form the circuit, and specify the type of analysis to be performed.

*PSpice*: This program simulates the circuit created using *Schematics*. By simulation, we mean a method of analysis described in a program by which a circuit is represented by mathematical models of the components comprising the circuit.

*Orcad PSpice*: This program provides a graphic display of the output generated by the *PSpice* program. It can be used to observe any voltage or current in the circuit.

One may think of *Schematics* as the computer breadboard for setting up the circuit topology, *PSpice* as the simulator (performing the computation), and *Orcad PSpice* as the oscilloscope. Using the *Schematics* program is perhaps the hardest part of circuit simulation using *PSpice*. The next section covers the essential skills needed to operate the *Schematics*.

## D.2 Creating a Circuit

For a circuit to be analyzed by *PSpice*, we must take three steps: (1) create the circuit, (2) simulate it, and (3) print or plot the results. In this section, we learn how to create the circuit using the *Schematics* program.

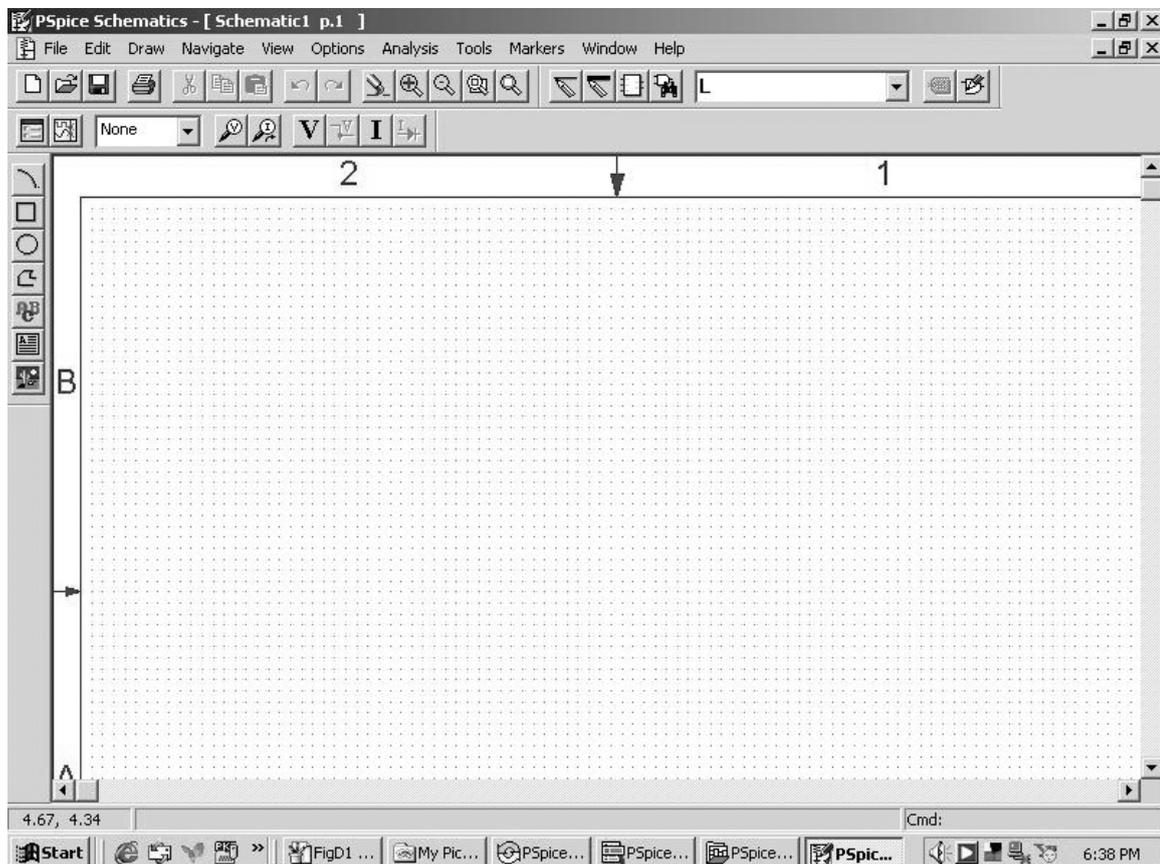
Before we discuss how to use the *Schematics* capture, we need to know how to use the mouse to select an object and perform an action. One uses the mouse in *Schematics* in conjunction with the keyboard to

carry out various instructions. Throughout this text, we will use the following terms to represent actions to be performed by the mouse:

- **CLICKL** : click the left button once to select an item.
- **CLICKR** : click the right button once to abort a mode.
- **DCLICKL** : double-click the left button to edit a selection or end a mode.
- **DCLICKR** : double-click the right button to repeat an action.
- **CLICKLH** : click the left button, hold down, and move the mouse to drag a selected item. Release the left button after the item has been placed.
- **DRAG** : drag the mouse (without clicking) to move an item.

When the term “click” is used, it means that you quickly press and release the **left** mouse button. To select an item requires **CLICKL**, while to perform an action requires **DCLICKL**. Also, to avoid writing “click” several times, the menu to be clicked will be highlighted in bold. For example, “click **Draw**, click **Get New Part**” will be written as **Draw/Get New Part**. Of course, we can always press the <Esc> key to abort any action.

Assuming that you are using Windows, you can access *PsPice* by clicking the Start icon on the left-hand corner of your PC, drag the cursor to Programs, *PsPice* student; and to *Schematics*, as shown in Fig. D.1. Alternatively, you have the *PsPice* icon on your screen. **DCLICK** on it. Either way, a blank screen will appear as shown in Fig. D.2. The file



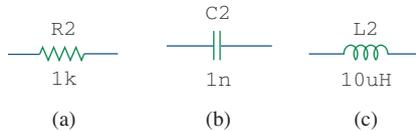
**Figure D.2**  
Schematics window.

name [*Schematic1 p.1*] next to *PSpice Schematics* is assigned to a circuit which is yet to be saved. You can change it by pulling down the *File* menu.

To create a circuit using *Schematics* requires three steps: (1) placing the parts or components of the circuit, (2) wiring the parts together to form the circuit, and (3) changing attributes of the parts.

### Step 1: Placing the Parts

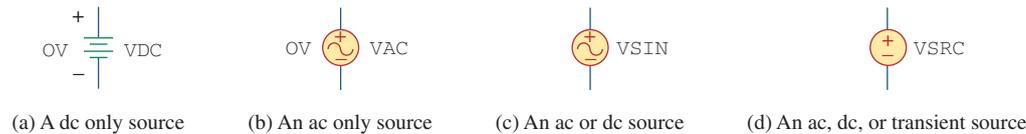
Each circuit part is retrieved by following this procedure:



**Figure D.3**

Part symbols and attributes for circuit elements: (a) a resistor, (b) a capacitor, (c) an inductor.

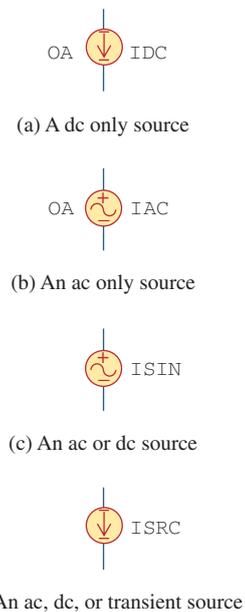
- Select **Draw/Get New Part** to pull down the Draw menu (or type <Ctrl-G>).
- Use scroll bar to select the part (or type the part name, e.g., R for resistor, in the *PartName* box). Figures D.3 to D.5 show some part names and symbols for circuit elements and independent voltage and current sources.
- Click **Place & Close** (or press <Enter>).
- **DRAG** part to the desired location on the screen.
- **CLICKR** to terminate the placement mode.



(a) A dc only source (b) An ac only source (c) An ac or dc source (d) An ac, dc, or transient source

**Figure D.4**

Part symbols and attributes for independent voltage sources.



**Figure D.5**

Part symbols and attributes for independent current sources.

Sometimes, we want to rotate a part 90°. To rotate a resistor, for example, select the part R and click **Edit/Rotate** (or type <Ctrl R>). To delete a part, **CLICKL** to select (highlight red) the part, then click **Edit/Cut** (or press <Delete>).

### Step 2: Wiring Parts Together

We complete the circuit by wiring the parts together. We first select **Draw/Wire** (or type <Ctrl-W>) to be in wiring mode. A pencil cursor will appear in place of an arrow cursor. **DRAG** the pencil cursor to the first point you want to connect and **CLICKL**. Next, **DRAG** the pencil cursor to the second point and **CLICKL** to change the dashed line to a solid line. (Only solid lines are wires.) **CLICKR** to end the wiring mode. To resume the wiring mode, press the <Space bar>. Repeat the above procedure for each connection in the circuit until all the parts are wired. The wiring is not complete without adding a ground connection (part AGND) to a schematic; *PSpice* will not operate without it. To verify that the parts are actually connected together, the *Junctions* option available in the **Options/Set Display Level** menu should be in the on position when wiring the parts. By default, the *Junctions* option is marked with a checksign (✓) in the dialog box, indicating that it is on.

Some of the connections have a black dot indicating a connection. Although it is not necessary to have a dot where a wire joins a pin,

having the dot shows the presence of a connection. To be sure a dot appears, make sure the wire overlaps the pin.

If you make a mistake, you can delete the part or wire by highlighting it (select **CLICKL**) and pressing the <Delete> key. Typing <Ctrl-L> will erase the fragments that are not really on the schematic.

### Step 3: Changing Attributes of Parts

As shown in Figs. D.3 to D.5, each component has an attribute in addition to its symbol. Attributes are the labels for parts. Each attribute consists of a *name* and its designated *value*. For example, R and VSRC are the names of resistor and voltage source (dc, ac, or transient source), while 2k and DC = +10V are the designated values of the resistor and voltage source, respectively.

As parts are placed on the screen, they are automatically assigned names by successive numbers (R1, R2, R3, etc.). Also, some parts are assigned some predetermined values. For example, all resistors are placed horizontally and assigned a value of 1 k $\Omega$ . We may need to change the attributes (names and values) of a part. Although there are several ways of changing the attributes, the following is one simple way.

To change the name R3 to RX, for example, **DCLICKL** on the text R3 to bring up the *Edit Reference Designator* dialog box of Fig. D.6(a). Type the new name RX and click the **OK** button to accept the change. The same procedure can be used to change VDC to V1 or whatever.

To change the value 1k to 10Meg, for example, **DCLICKL** on the 1k attribute (not the symbol) to open up the *Set Attribute Value* dialog box of Fig. D.6(b). Type the new value 10Meg (no space between 10 and Meg) and click the **OK** button to accept the change. Similarly, to change the default value 0V to 15kV for voltage source VDC, **DCLICKL** the symbol for VDC to bring up the *PartName* dialog box. **DCLICKL** on the DC = attribute and type 15kV in the value box. For convenience, one can express numbers with the scale factors in Table D.1. For example,  $6.6 \times 10^{-8}$  can be written as 66N or 0.066U.

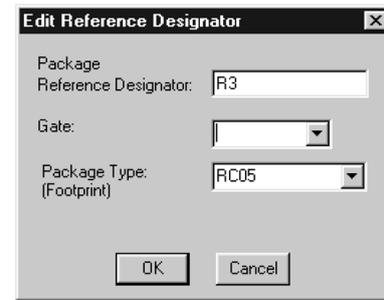
Except for the ground, which is automatically assigned node 0, every node is either given a name (or number) or is assigned one in the netlist. A node is labeled by giving a name to a wire connected to that node. **DCLICKL** the wire to open up the *Set Attribute Value* dialog box, and type the label.

To obtain a hard copy of the screen/schematic, click **File/Print/OK**. To save the schematic created, select **File/Save As** and type Filename. Click **OK** or press <Enter>. This creates a file named “filename” and saves it with extension .sch.

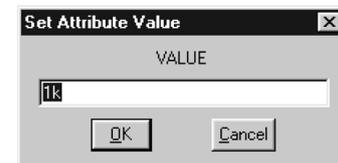
Draw the circuit in Fig. D.7 using *Schematics*.

#### Solution:

We will follow the three steps mentioned above. We begin by double-clicking the *Schematics* icon. This provides us with a blank screen as



(a)



(b)

**Figure D.6**

- (a) Changing name R3 to RX,  
(b) changing 1k to 10Meg.

A component may have several attributes; some are displayed by default. If need be, we may add more attributes for display, but we should hide unimportant attributes to avoid clutter.

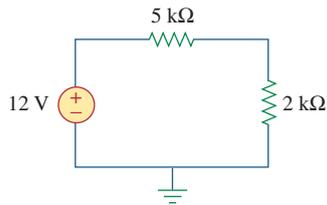
**TABLE D.1**

#### Scale factors.

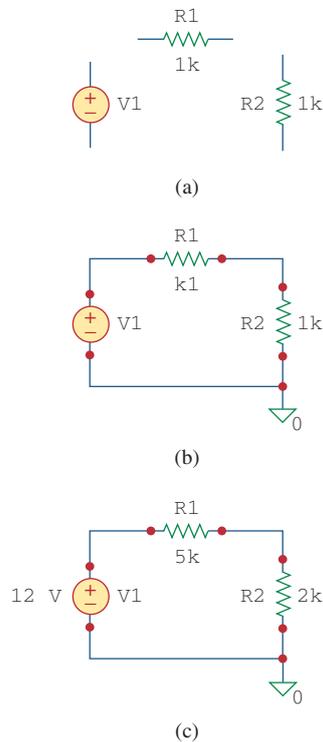
Symbol	Value	Name of suffix
T	$10^{12}$	tera
G	$10^9$	giga
MEG	$10^6$	mega
K	$10^3$	kilo
M	$10^{-3}$	milli
U	$10^{-6}$	micro
N	$10^{-9}$	nano
P	$10^{-12}$	pico
F	$10^{-15}$	femto

It is always expedient to number the nodes by numbering the wires. Otherwise, *Schematics* will label the nodes its own way, and one may not understand which node is which in the output results.

### Example D.1



**Figure D.7**  
For Example D.1.



**Figure D.8**  
Creating the circuit in Fig. D.7:  
(a) placing the parts, (b) wiring the parts together, (c) changing the attributes.

a worksheet to draw the circuit on. We now take the following steps to create the circuit in Fig. D.7.

To place the voltage source, we need to:

1. Click **Draw/Get New Part** (or type <Ctrl-G>).
2. Type VSRC in the *Part Browser Basic* box.
3. Click **OK** (or type <Enter>).
4. **DRAG** the part to the desired location on the screen.
5. **CLICKL** to place VSRC and **CLICKR** to terminate placement mode.

At this point, only the voltage source V1 in Fig. D.8(a) is shown on the screen, highlighted red. To place the resistors, we need to:

1. Click **Draw/Get New Part**.
2. Type R in the *Part Browser Basic* box.
3. Click **OK**.
4. **DRAG** resistor to R1's location on the screen.
5. **CLICKL** to place R1.
6. **CLICKL** to place R2 and **CLICKR** to terminate placement mode.
7. **DRAG** R2 to its location.
8. **Edit/Rotate** (or type <Ctrl-R>) to rotate R2.

At this point, the three parts have been created as shown in Fig. D.8(a). The next step is to connect the parts by wiring. To do this:

1. Click **Draw/Wire** to be in wiring mode, indicated by the pencil cursor.
2. **DRAG** the pencil cursor to the top of V1.
3. **CLICKL** to join the wire to the top of V1.
4. **DRAG** the dotted wire to the top corner.
5. **CLICKL** to turn wire segment solid, and anchor at corner.
6. **DRAG** dotted wire to left of R1.
7. **CLICKL** to turn wire segment solid and anchor to left of R1.
8. **CLICKR** to end placement mode.

Follow the same steps to connect R1 with R2 and V1 with R2. (You can resume the wiring mode by pressing <Space bar>.) At this point, we have the circuit in Fig. D.8(b), except that the ground symbol is missing. We insert the ground by taking the following steps:

1. Click **Draw/Get New Part**.
2. Type AGND in the *Part Browser Basic* box.
3. Click **OK**.
4. **DRAG** the part to the desired location on the screen.
5. **CLICKL** to place AGND and **CLICKR** to terminate placement mode.

The last thing to be done is to change or assign values to the attributes. To assign the attribute 12V to V1, we take these steps:

1. **DCLICKL** on the V1 symbol to open up the *PartName* dialog box.
2. **DCLICKL** on the *DC =* attribute.
3. Type +12V (or simply 12) in the *Value* box.
4. Click **Save Attr**.
5. Click **OK**.

To assign 5k to R1, we follow these steps:

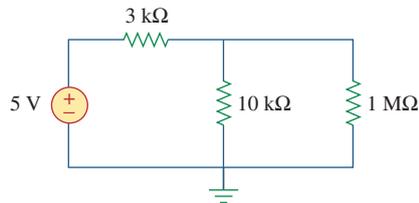
1. **DCLICKL** on 1k attribute of R1 to bring up the *Set Attribute Value* dialog box.
2. Type 5k in the *Value* box.
3. Click **OK**.

Use the same procedure in assigning value 2k to R2. Figure D.8(c) shows the final circuit.

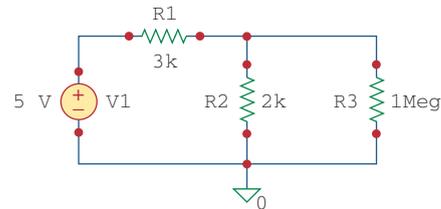
Construct the circuit in Fig. D.9 with *Schematics*.

## Practice Problem D.1

**Answer:** See the schematic Fig. D.10.



**Figure D.9**  
For Practice Prob. D.1.



**Figure D.10**  
For Practice Prob. D.1.

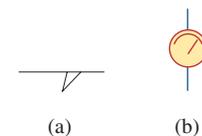
## D.3 DC Analysis

DC analysis is one of the standard analyses that we can perform using *PSpice*. Other standard analyses include transient, AC, and Fourier. Under DC analysis, there are two kinds of simulation that *PSpice* can execute: DC nodal analysis and DC sweep.

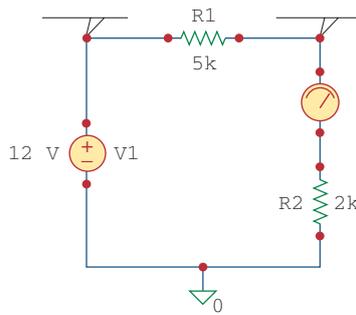
### 1. DC Nodal Analysis

*PSpice* allows dc nodal analysis to be performed on sources with an attribute of the form DC = value and provides the dc voltage at each node of the circuit and dc branch currents if required. To view dc node voltages and branch currents requires adding two kinds of additional parts, shown in Fig. D.11. The symbol VIEWPOINT is connected to each node at which the voltage is to be viewed, while the symbol IPROBE is connected in the branch where the current is to be displayed. This necessitates modifying the schematic. For example, let us consider placing voltage VIEWPOINTS and current IPROBES to the schematic in Fig. D.8(c). To add VIEWPOINTS, we take the following steps:

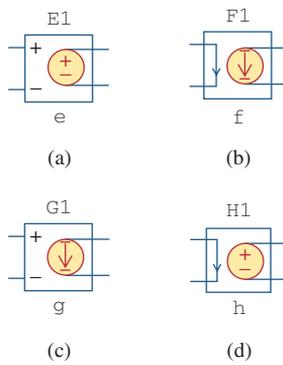
1. Click **Draw/Get New Part** (or type <Ctrl-G>).
2. Type VIEWPOINT in the *Part Browser Basic* box.
3. Click **OK** (or type <Enter>).
4. **DRAG** to locate VIEWPOINT above V1 and **CLICKL**.
5. **DRAG** to locate VIEWPOINT above R2 and **CLICKL**.
6. **CLICKR** to end placement mode.



**Figure D.11**  
Symbols for: (a) voltage VIEWPOINT,  
(b) current IPROBE.



**Figure D.12**  
Placing VIEWPOINTS and IPROBES.



**Figure D.13**

Dependent sources:

- (a) voltage-controlled voltage source (VCVS),
- (b) current-controlled current source (CCCS),
- (c) voltage-controlled current source (VCCS),
- (d) current-controlled voltage source (CCVS).

A netlist can be generated manually or automatically by *Schematics*.

There are two kinds of common errors in *PSpice*: (1) errors involving wiring of the circuit, and (2) errors that occur during simulation.

Figure D.12 shows the two voltage VIEWPOINTS. Since the IPROBE symbol must be connected in series with a branch element, we need to move R2 down by clicking and dragging R2 and the wires. Once this is done, we add IPROBE as follows:

1. Click **Draw/Get New Part** (or type <Ctrl-G>).
2. Type IPROBE in the *Part Browser Basic* box.
3. Click **OK** (or type <Enter>).
4. **DRAG** to locate IPROBE above R2 and **CLICKL**.
5. **CLICKR** to end placement mode.
6. Use wiring to join all gaps.

The schematic becomes that shown in Fig. D.12. We are ready to simulate the circuit. At this point, we must save the schematic—*PSpice* will not run without first saving the schematic to be simulated. Before learning how to run *PSpice*, note the following points:

1. There must be a reference node or ground connection (part AGND) in the schematic. Any node can be used as ground, and the voltages at other nodes will be measured with respect to the selected ground.
2. Dependent sources are found in the Parts library. Obtain them by selecting **Draw/Get New Part** and typing the part name. Figure D.13 shows the part name for each type, with the gain. *E* is a voltage-controlled voltage source with gain *e*; *F* is a current-controlled current source with gain *f*; *G* is a voltage-controlled current source with a transconductance gain *g*; and *H* is a current-controlled voltage source with transresistance gain *h*.
3. By convention, we assume in dc analysis that all capacitors are open circuits and all inductors are short circuits.

We run *PSpice* by clicking **Analysis/Simulate**. This invokes the *electric rule check* (ERC), which generates the *netlist*. The ERC performs a connectivity check on the schematic before creating the netlist. The netlist is a list describing the operational behavior of each component in the circuit and its connections. Each line in the netlist represents a single component of the circuit. The netlist can be examined by clicking **Analysis/Examine Netlist** from the *Schematics* window. If there are errors in the schematic, an *error* window will appear. Click **OK** (or type <Enter>) to display the error list. After noting the errors, exit from the error list and go back to *Schematics* to correct the errors. If no errors are found, the system automatically enters *PSpice* and performs the simulation (nodal analysis). When the analysis is complete, the program displays **Bias point calculated**, and creates the result/output file with extension *.out*.

To examine the output file, click **Analysis/Examine Output** from the *Schematics* window (or click **File/Examine Output** from the *PSpice* window). To print the output file, click **File/Print**, and to exit the output file, click **File/Exit**.

We can also examine the results of the simulation by looking at the values displayed on the VIEWPOINTS and IPROBES parts of the *schematics* after the simulation is complete. The values displayed with VIEWPOINTS and IPROBES should be the same as those in the output file.

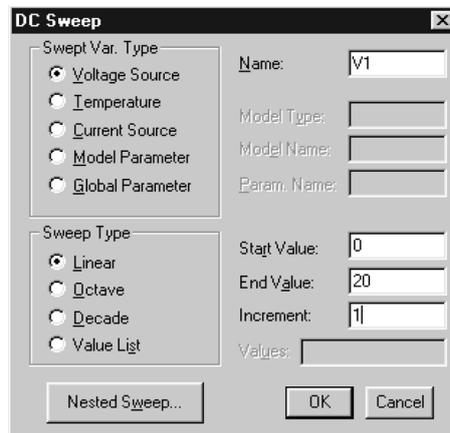
## 2. DC Sweep

DC nodal analysis allows simulation for DC sources with fixed voltages or currents. DC sweep provides more flexibility in that it allows the calculation of node voltages and branch currents of a circuit when a source is swept over a range of values. As in nodal analysis, we assume capacitors to be open circuits and inductors to be short circuits.

Suppose we desire to perform a DC sweep of voltage source V1 in Fig. D.12 from 0 to 20 volts in 1-volt increments. We proceed as follows:

1. Click **Analysis/Setup**.
2. **CLICK** **DC Sweep** button.
3. Click *Name* box and type V1.
4. Click *Start Value* box and type 0.
5. Click *End Value* box and type 20.
6. Click *Increment* box and type 1.
7. Click **OK** to end the *DC Sweep* dialog box and save parameters.
8. Click **Close** to end the *Analysis Setup* menu.

Figure D.14 shows the *DC Sweep* dialog box. Notice that the default setting is *Voltage Source* for the *Swept Var. Type*, while it is *Linear* for *Sweep Type*. If needed, other options can be selected by clicking the appropriate buttons.



**Figure D.14**  
DC sweep analysis dialog box.

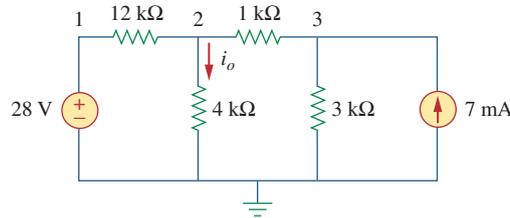
To run DC sweep analysis, click **Analysis/Simulate**. *Schematics* will create a netlist and then run *PSpice* if no errors are found. If errors are found in the schematic, check for them in the *Error List* and correct them as usual. If no errors are found, the data generated by *PSpice* is passed to *Orcad PSpice*. The *Orcad PSpice* window will appear, displaying a graph in which the X axis is by default set to the DC sweep variable and range, and the Y axis is blank for now. To display some specific plots, click **Trace/Add** in the *Orcad PSpice* menu to open the *Add Traces* dialog box. The box contains traces, which are the output variables (node voltages and branch currents) in the data file available for display. Select the traces to be displayed by clicking or typing them, and click **OK**. The selected traces will be plotted and displayed on the

screen. As many traces as you want may be added to the same plot or on different windows. Select a new window by clicking **Window/New**. To delete a trace, click the trace name in the legend of the plot to highlight it and click **Edit/Delete** (or press <Delete>).

It is important to understand how to interpret the traces. We must interpret the voltage and current variables according to the passive sign convention. As parts are initially placed horizontally in a schematic as shown typically in Fig. D.3, the left-hand terminal is named pin 1 while the right-hand terminal is pin 2. When a component (say R1) is rotated counterclockwise once, pin 2 would be on the top, since rotation is about pin 1. Therefore, if current enters through pin 2, the current I(R1) through R1 would be negative. In other words, positive current implies that the current enters through pin 1, and negative current means that the current enters through pin 2. As for voltage variables, they are always with respect to the ground. For example, V(R1:2) is the voltage (with respect to the ground) at pin 2 of resistor R1; V(V1:+) is the voltage (with respect to the ground) at the positive terminal of voltage source V1; and V(E2:1) is the voltage at pin 1 of component E2 with respect to ground, regardless of the polarity.

**Example D.2**

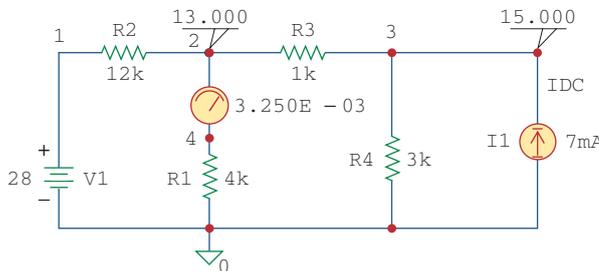
For the circuit in Fig. D.15, find the dc node voltages and the current  $i_o$ .



**Figure D.15**  
For Example D.2.

**Solution:**

We use *Schematics* to create the circuit. After saving the circuit, click **Analysis/Simulate** to simulate the circuit. We obtain the results of the dc analysis from the output file or from the **VIEWPOINT AND IPROBE** parts, as shown in Fig. D.16. The netlist file is shown in Fig. D.17. Notice that the netlist contains the name, value, and



**Figure D.16**  
For Example D.2; schematic for the circuit in Fig. D.15.

\* Schematics Netlist \*

```
V_V1      1 0 28
R_R1      0 4 4k
R_R2      1 2 12k
R_R3      2 3 1k
R_R4      0 3 3k
I_I1      0 3 DC 7mA
v_V2      2 4 0
```

**Figure D.17**  
The Netlist file for Example D.2.

connection for each element in the circuit. First example, the first line shows that the voltage source V1 has a value of 28 V and is connected between nodes 0 and 1. Figure D.18 shows the edited version of the output file. The output file also contains the Netlist file, but this was removed from Fig. D.18. From IPROBE or the output file, we obtain  $i_o$  as 3.25 mA.

```

**** 11/26/99 20:56:05 ***** NT Evaluation PSpice (Nov. 1999) *****

* C:\MSIMEV63\examd2.sch

**** CIRCUIT DESCRIPTION

*****

* Schematics Version 6.3 - April 1996
* Sat Jul 26 20:56:04 1997

**** INCLUDING examd2.als ****
* Schematics Aliases *

.ALIASES
V_V1      V1(+=1 -=0 )
R_R1      R1(1=0 2=4 )
R_R2      R2(1=1 2=2 )
R_R3      R3(1=2 2=3 )
R_R4      R4(1=0 2=3 )
I_I1      I1(+=0 -=3 )
v_V2      V2(+=2 -=4 )
__ (1=1)
__ (2=2)
__ (3=3)
.ENDALIASES

.probe

.END

NODE VOLTAGE      NODE VOLTAGE      NODE VOLTAGE      NODE VOLTAGE

( 1) 28.0000      ( 2) 13.0000      ( 3) 15.0000      ( 4) 13.0000

VOLTAGE SOURCE CURRENTS
NAME      CURRENT

V_V1      -1.250E-03
v_V2      3.250E-03

TOTAL POWER DISSIPATION 3.50E-02 WATTS

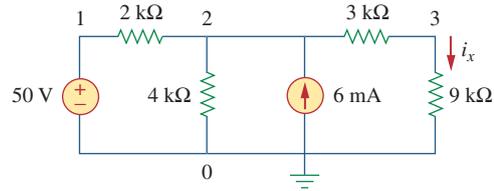
```

### Figure D.18

Output file (edited version) for Example D.2.

**Practice Problem D.2**

Use PSpice to determine the node voltages and the current  $i_x$  in the circuit of Fig. D.19.

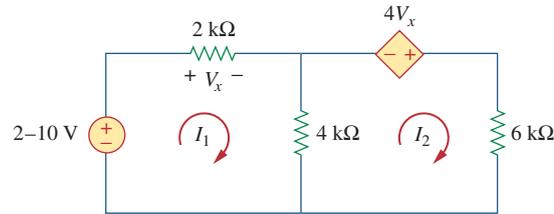


**Figure D.19**  
For Practice Prob. D.2.

**Answer:**  $V_1 = 50$ ,  $V_2 = 37.2$ ,  $V_3 = 27.9$ ,  $i_x = 3.1$  mA.

**Example D.3**

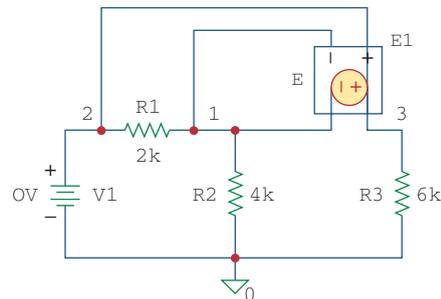
Plot  $I_1$  and  $I_2$  if the dc voltage source in Fig. D.20 is swept from 2 V to 10 V.



**Figure D.20**  
For Example D.3.

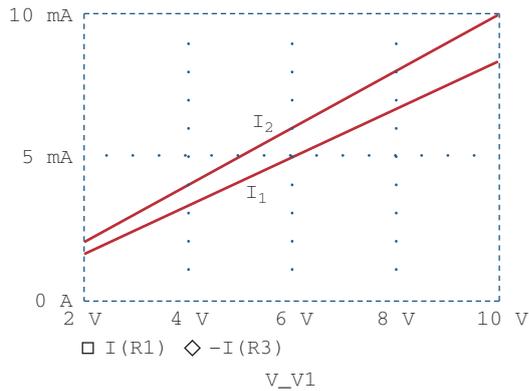
**Solution:**

We draw the schematic of the circuit and set the attributes as shown in Fig. D.21. Notice how the voltage-controlled voltage source E1 is connected. After completing the schematic, we select **Analysis/Setup** and input the start, end, and increment values as 2, 10, and 0.5, respectively. By selecting **Analysis/Simulate**, we bring up the *Orcad PSpice* window. We select **Trace/Add** and click  $I(R1)$  and  $-I(R3)$  to



**Figure D.21**  
The schematic for the circuit in Fig. D.20.

be displayed. (The negative sign is needed to make the current through R3 positive.) Figure D.22 shows the result.

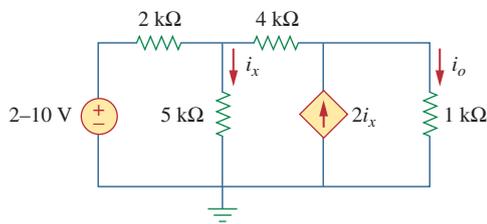


**Figure D.22**  
Plots of  $I_1$  and  $I_2$  against  $V_1$ .

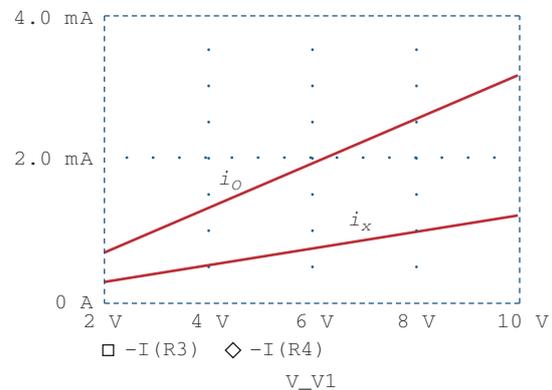
Use PSpice to obtain the plots of  $i_x$  and  $i_o$  if the dc voltage source in Fig. D.23 is swept from 2 V to 10 V.

**Practice Problem D.3**

**Answer:** The plots of  $i_x$  and  $i_o$  are displayed in Fig. D.24.



**Figure D.23**  
For Practice Prob. D.3.



**Figure D.24**  
Plots of  $i_x$  and  $i_o$  versus  $V_1$ .

**D.4 Transient Analysis**

In PSpice, transient analysis is generally used to examine the behavior of a waveform (voltage or current) as time varies. Transient analysis solves some differential equations describing a circuit and obtains voltages and currents versus time. Transient analysis is also used to obtain Fourier analysis. To perform transient analysis on a circuit using PSpice

Transient analysis is used to view the transient response of inductors and capacitors.

usually involves these steps: (1) drawing the circuit, (2) providing specifications, and (3) simulating the circuit.

### 1. Drawing the Circuit

In order to run a transient analysis on a circuit, the circuit must first be created using *Schematics* and the source must be specified. *PSpice* has several time-varying functions or sources that enhance the performance of transient analysis. Sources used in the transient analysis include:

- VSIN, ISIN: damped sinusoidal voltage or current source, e.g.,  $v(t) = 10e^{-0.2t} \sin(120\pi t - 60^\circ)$ .
- VPULSE, IPULSE: voltage or current pulse.
- VEXP, IEXP: voltage or current exponential source, e.g.,  $i(t) = 6[1 - \exp(-0.5t)]$ .
- VPWL, IPWL: piecewise linear voltage or current function, which can be used to create an arbitrary waveform.

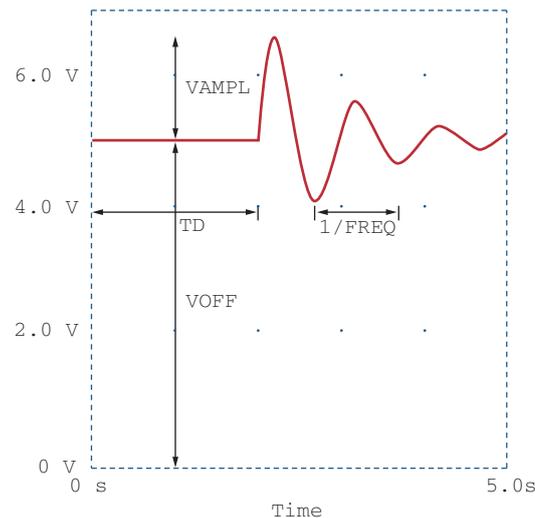
It is expedient to take a close look at these functions.

VSIN is the exponentially damped sinusoidal voltage source, for example,

$$v(t) = V_o + V_m e^{-\alpha(t-t_d)} \sin[2\pi f(t-t_d) + \phi] \quad (\text{D.1})$$

The VSIN source has the following attributes, which are illustrated in Fig. D.25 and compared with Eq. (D.1):

$$\begin{aligned} \text{VOFF} &= \text{Offset voltage, } V_o \\ \text{VAMPL} &= \text{Amplitude, } V_m \\ \text{TD} &= \text{Time delay in seconds, } t_d \\ \text{FREQ} &= \text{Frequency in Hz, } f \\ \text{DF} &= \text{Damping factor (dimensionless), } \alpha \\ \text{PHASE} &= \text{Phase in degrees, } \phi \end{aligned} \quad (\text{D.2})$$

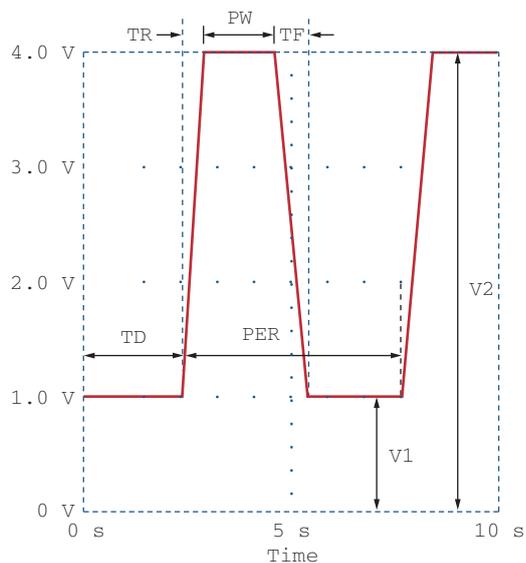


**Figure D.25**  
Sinusoidal voltage source VSIN.

Attributes TD, DF, and PHASE are set to 0 by default but can be assigned other values if necessary. What has been said about VSIN is also true for ISIN.

The VPULSE source has the following attributes, which are portrayed in Fig. D.26:

- V1 = Low voltage
  - V2 = High voltage
  - TD = Initial time delay in seconds
  - TR = Rise time in seconds
  - TF = Fall time in seconds
  - PW = Pulse width in seconds
  - PER = Period in seconds
- (D.3)**

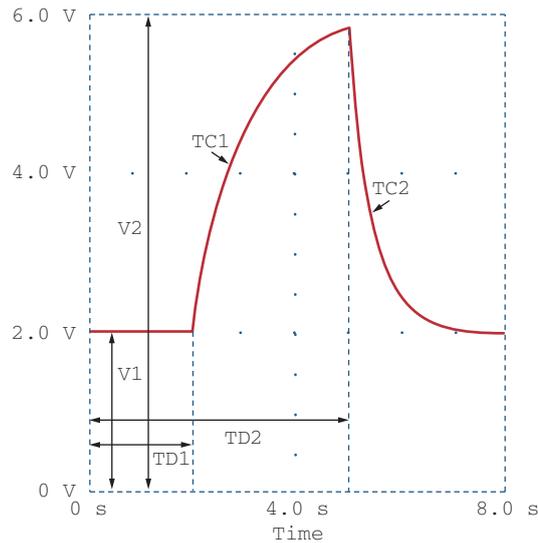


**Figure D.26**  
Pulse voltage source VPULSE.

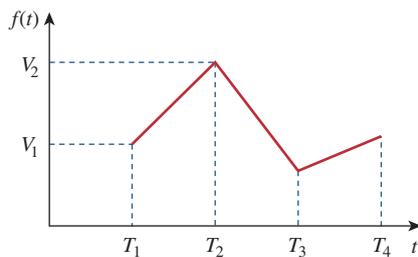
Attributes V1 and V2 must be assigned values. By default, attribute TD is assigned 0; TR and TF are assigned the *print step* value; and PW and PER are assigned the *final time* value. The values of the *print time* and *final time* are obtained as default values from the specifications provided by the user in the **Transient Analysis/Setup**, to be discussed a little later.

The exponential voltage source VEXP has the following attributes, typically illustrated in Fig. D.27:

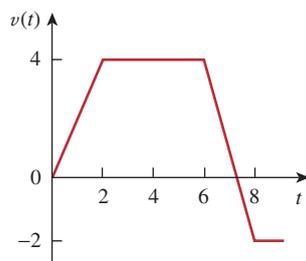
- V1 = Initial voltage
  - V2 = Final voltage
  - TD1 = Rise delay in seconds
  - TC1 = Rise time constant in seconds
  - TD2 = Fall delay in seconds
  - TC2 = Fall time in seconds
- (D.4)**



**Figure D.27**  
Exponential voltage source VEXP.



**Figure D.28**  
Piecewise linear voltage source VPWL.



**Figure D.29**  
An example of a piecewise linear voltage source VPWL.

The piecewise linear voltage source VPWL, such as shown in Fig. D.28, requires specifying pairs of  $T_N$ ,  $V_N$ , where  $V_N$  is the voltage at time  $T_N$  for  $N = 1, 2, \dots, 10$ . For example, for the function in Fig. D.29, we will need to specify the attributes  $T_1 = 0$ ,  $V_1 = 0$ ,  $T_2 = 2$ ,  $V_2 = 4$ ,  $T_3 = 6$ ,  $V_3 = 4$ , and  $T_4 = 8$ ,  $V_4 = -2$ .

To obtain information about other sources, click **Help/Search for Help on . . .** and type in the name of the source. To add a source to the schematic, take the following steps:

1. Select **Draw/Get New Part**.
2. Type the name of the source.
3. Click **OK** and **DRAG** the symbol to the desired location.
4. **DCLICKL** the symbol of the source to open up the *PartName* dialog box.
5. For each attribute, **DCLICKL** on the attribute, enter the value, and click **Save Attr** to accept changes.
6. Click **OK** to accept new attributes.

In step 5, the attributes may not be shown on the schematic after entering their values. To display an attribute, select **Change Display/Both Name and Value** in the *PartName* dialog box.

In addition to specifying the source to be used in transient analysis, there may be need to set initial conditions on capacitors and inductors in the circuit. To do so, **DCLICKL** the part symbol to bring up the *PartName* dialog box, click **IC =** and type in the initial condition. The IC attribute allows for setting the initial conditions on a capacitor or inductor. The default value of IC is 0. The attributes of open/close switches (with part names Sw\_tClose and Sw\_tOpen) can be changed in a similar manner.

## 2. Providing Specifications

After the circuit is drawn and the source is specified with its attributes, we need to add some specifications for the transient analysis.

For example, suppose we want the analysis to run the simulation from 0 to 10 ms with a print interval of 2 ns; we enter these specifications as follows:

1. Select **Analysis/Setup/Transient** to open up the *Transient Analysis* dialog box.
2. **CLICKL** *Print Step* and type 2 ns.
3. **CLICKL** *Final Time* and type 10 ms.
4. **CLICKL** *Step Ceiling* and type 5  $\mu$ s.
5. **CLICKL** **OK/Close** to accept specifications.

These specifications control the simulation and the display of output variables. *Final Time* specifies how long the simulation should run. In other words, the simulation runs from  $t = 0$  to  $t = \text{Final Time}$ . *Print Step* refers to the time interval the print part will print out; it controls how often simulation results are written to the output file. The value of *Print Step* can be any value less than the *Final Time*, but it cannot be zero. *Step Ceiling* is the maximum time between simulation points; specifying its value is optional. By selecting 10 ms as *Final Time* and 5  $\mu$ s as *Step Ceiling*, the simulation will have a minimum of  $10 \text{ ms} / 5 \mu\text{s} = 2000$  points. When *Step Ceiling* is unspecified, PSpice selects its own internal time step—the time between simulation points. The time step is selected as large as possible to reduce simulation time. If the user has no idea of what the plot may look like, it is recommended that the value of *Step Ceiling* be unspecified. If the plot is jagged as a result of a large time step assumed by PSpice, the user may now specify a *Step Ceiling* that will smooth the plot. Keep in mind that a smaller value gives more points in the simulation but takes more time.

To obtain the Fourier component of a signal, we enable the Fourier option in the *Transient Analysis* dialog box. Chapter 17 has more on this.

### 3. Simulating the Circuit

After the circuit is drawn, the specifications for the transient analysis are given, and the circuit is saved, we are ready to simulate it. To perform transient analysis, we select **Analysis/Simulate**. If there are no errors, the *Orcad PSpice* window will automatically appear. As usual, the time axis (or X axis) is drawn but no curves are drawn yet. Select **Trace/Add** and click on the variables to be displayed.

An alternative way of displaying the results is to use *markers*. Although there are many types of markers, we will discuss only voltage and current markers. A voltage marker is used to display voltage at a node relative to ground; a current marker is for displaying current through a component pin. To place a voltage marker at a node, take the following steps while in the *Schematics* window:

1. Select **Markers/Mark Voltage/Level**.
2. **DRAG** the voltage marker to the desired node.
3. **CLICKL** to place the marker and **CLICKR** to end the placement mode.

This will cause two things to happen immediately. The voltage marker becomes part of the circuit and the appropriate node voltage is automatically displayed by the *Orcad PSpice* window when the *Schematics* is

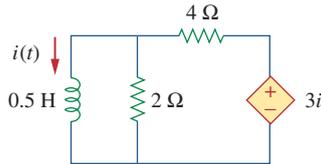
run. To place a current marker at a component pin, take the following steps in the *Schematics* window:

1. Select **Markers/Mark current into pin**.
2. **DRAG** the current marker to the desired pin.
3. **CLICKL** to place the marker and **CLICKR** to end the placement mode.

This will automatically add the current through the pin to your graph. It is important that the current marker be placed at the pin of the component; otherwise the system would reject the marker. You can place as many voltage and current markers as you want on a circuit. To remove the markers from the circuit as well as the plots from the *Orcad PSpice* window, select **Markers/Clear All** from *Schematics* window.

**Example D.4**

Assuming that  $i(0) = 10\text{A}$ , plot the zero-input response  $i(t)$  in the circuit of Fig. D.30 for  $0 < t < 4\text{ s}$  using *PSpice*.



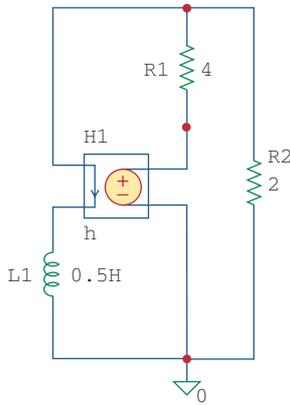
**Figure D.30**  
For Example D.4.

**Solution:**

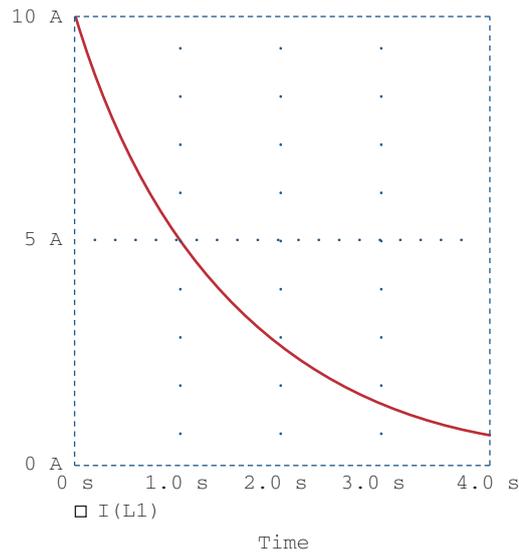
The circuit is the same as the one for Example 7.3, where we obtained the solution as

$$i(t) = 10e^{(-2/3)t}$$

For *PSpice* analysis, the schematic is in Fig. D.31, where the current-controlled source H1 has been wired to agree with the circuit in Fig. D.30. The voltage of H1 is 3 times the current through inductor L1. Therefore, for H1, we set GAIN = 3 and for the inductor L1, we set the initial condition IC = 10. Using the **Analysis/Setup/Transient** dialog box, we set *Print Step* = 0.25 s and *Final Time* = 4 s. After simulating the circuit, the output is taken as the inductor current  $i(t)$ , which is plotted in Fig. D.32.



**Figure D.31**  
The schematic of the circuit in Fig. D.30.

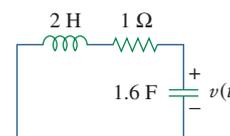


**Figure D.32**  
Output plot for Example D.4.

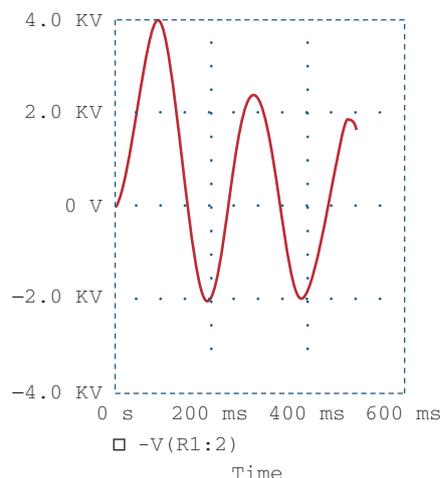
Using PSpice, plot the source-free response  $v(t)$  in the circuit of Fig. D.33, assuming that  $v(0) = 10$  V.

**Practice Problem D.4**

**Answer:** Figure D.34 shows the plot. Note that  $v(t) = 10e^{-0.25t} \cos 0.5t + 5e^{-0.25t} \sin 0.5t$  V.



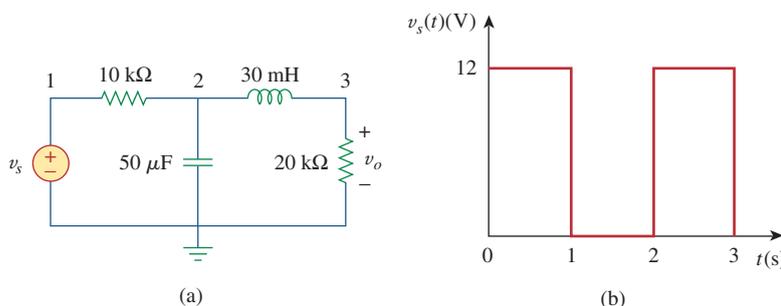
**Figure D.33**  
For Practice Prob. D.4.



**Figure D.34**  
Output plot for Practice Prob. D.4.

Plot the forced response  $v_o(t)$  in the circuit of Fig. D.35(a) for  $0 < t < 5$  s if the source voltage is as shown in Fig. D.35(b).

**Example D.5**

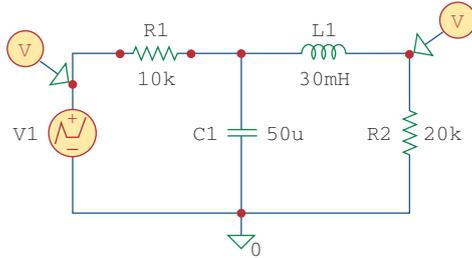


**Figure D.35**  
For Example D.5.

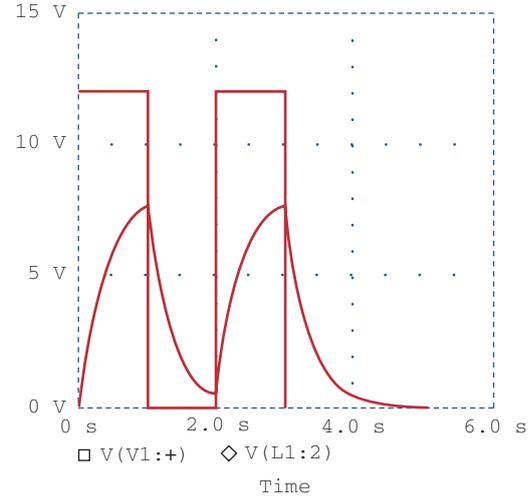
**Solution:**

We draw the circuit and set the attributes as shown in Fig. D.36. We enter in the data in Fig. D.35(b) by double-clicking the symbol of the voltage source V1 and typing in T1 = 0, V1 = 0, T2 = 1 ns, V2 = 12, T3 = 1 s, V3 = 12, T4 = 1.001 s, V4 = 0, T5 = 2 s, V5 = 0, T6 = 2.001 s, V6 = 12, T7 = 3 s, V7 = 12, T8 = 3.001 s, V8 = 0. In the **Analysis/Setup/Transient** dialog box, we set *Print Step* = 0.2 s and *Final Time* = 5 s. When the circuit is simulated and we are in the *Orcad PSpice* window, we close or minimize the window to go back to the

Schematics window. We place two voltage markers as shown in Fig. D.36 to get the plots of input  $v_s$  and output  $v_o$ . We press <Alt-Esc> to get into the *Orcad PSpice* window and obtain the plots shown in Fig. D.37.



**Figure D.36**  
The schematic of the circuit in Fig. D.35.

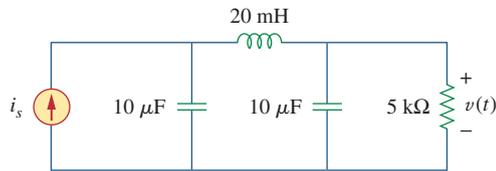


**Figure D.37**  
Output plot for Example D.5.

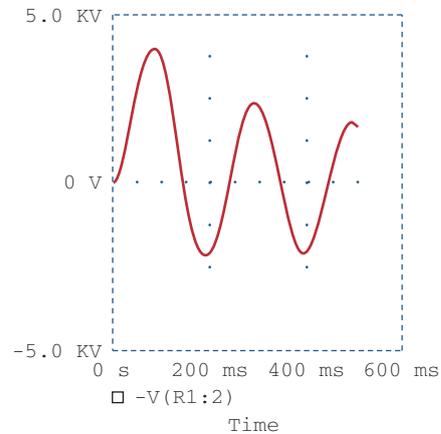
**Practice Problem D.5**

Obtain the plot of  $v(t)$  in the circuit in Fig. D.38 for  $0 < t < 0.5$  s if  $i_s = 2e^{-t} \sin 2\pi(5)t$  A.

**Answer:** See Fig. D.39.



**Figure D.38**  
For Practice Prob. D.5.



**Figure D.39**  
Output plot for Practice Prob. D.5.

**D.5 AC Analysis/Frequency Response**

Using AC sweep, *PSpice* can perform AC analysis of a circuit for a single frequency or over a range of frequencies in increments that can vary linearly, by decade, or by octave. In AC sweep, one or more sources

are swept over a range of frequencies while the voltages and currents of the circuit are calculated. Thus, we use AC sweep both for phasor analysis and for frequency response analysis: it will output Bode gain and phase plots. (Keep in mind that a phasor is a complex quantity with real and imaginary parts or with magnitude and phase.)

While transient analysis is done in the time domain, AC analysis is performed in the frequency domain. For example, if  $v_s = 10\cos(377t + 40^\circ)$ , transient analysis can be used to display  $v_s$  as a function of time, whereas AC sweep will give the magnitude as 10 and phase as  $40^\circ$ . To perform AC sweep requires taking three steps similar to those for transient analysis: (1) drawing the circuit, (2) providing specifications, and (3) simulating the circuit.

### 1. Drawing the Circuit

We first draw the circuit using *Schematics* and specify the source(s). Sources used in AC sweep are AC sources VAC and IAC. The sources and attributes are entered into the *Schematics* as stated in the previous section. For each independent source, we must specify its magnitude and phase.

### 2. Providing Specifications

Before simulating the circuit, we need to add some specifications for AC sweep. For example, suppose we want a linear sweep at frequencies 50, 100, and 150 Hz. We enter these parameters as follows:

1. Select **Analysis/Setup/AC Sweep** to open up the dialog box for AC Sweep.
2. **CLICKL** *Linear* for the *X* axis to have a linear scale.
3. Type 3 in the *Total Pts* box.
4. Type 50 in the *Start Freq* box.
5. Type 150 in the *End Freq* box.
6. **CLICKL** **OK/Close** to accept specifications.

A linear sweep implies that simulation points are spread uniformly between the starting and ending frequencies. Note that the *Start Freq* cannot be zero because 0 Hz corresponds to DC analysis. If we want the simulation to be done at a single frequency, we enter 1 in step 3 and the same frequency in steps 4 and 5. If we want the AC sweep to simulate the circuit from 1 Hz to 10 MHz at 10 points per decade, we **CLICKL** on *Decade* in step 2 to make the *X* axis logarithmic, enter 10 in the *Total Pts* box in step 3, enter 1 in the *Start Freq* box, and enter 10Meg in the *End Freq* box. Keep in mind that a decade is a factor of 10. In this case, a decade is from 1 Hz to 10 Hz, from 10 Hz to 100, from 100 to 1 kHz, and so forth.

### 3. Simulating the Circuit

After providing the necessary specifications and saving the circuit, we perform the AC sweep by selecting **Analysis/Simulate**. If no errors are encountered, the circuit is simulated. At the end of the simulation, the system displays **AC analysis finished** and creates an output file with extension .out. Also, the *Orcad PSpice* program will automatically run if there are no errors. The frequency axis (or *X* axis) is drawn but no curves are shown yet. Select **Trace/Add** from the *Orcad PSpice* menu bar and click on the variables to be displayed. We may also use

current or voltage markers to display the traces as explained in the previous section. To use advanced markers such as *vdb*, *idb*, *vphase*, *iphase*, *vreal*, and *ireal*, select **Markers/Mark Advanced**.

In case the resolution of the trace is not good enough, we may need to check the data points to see if they are enough. To do so, select **Tools/Options/Mark Data Points/OK** in the *Orcad PSpice* menu and the data points will be displayed. If necessary, we can improve the resolution by increasing the value of the entry in the **Total Pts** box in the **Analysis/Setup/AC Sweep and Noise Analysis** dialog box for AC sweep.

To generate Bode plots involves using the AC sweep and the dB command in *Orcad PSpice*.

Bode plots are separate plots of magnitude and phase versus frequency. To obtain Bode plots, it is common to use an AC source, say V1, with 1 volt magnitude and zero phase. After we have selected **Analysis/Simulate** and have the *Orcad PSpice* program running, we can display the magnitude and phase plots as mentioned above. Suppose we want to display a Bode magnitude plot of V(4). We select **Trace/Add** and type **dB(V(4))** in **Trace Command** box. **dB(V(4))** is equivalent to  $20\log(V(4))$ , and because the magnitude of V1 or V(R1:1) is unity, **dB(V(4))** actually corresponds to  $\text{dB}(V(4)/V(R1:1))$ , which is the gain. Adding the trace **dB(V(4))** will give a Bode magnitude/gain plot with the Y axis in dB.

Once a plot is obtained in the *Orcad PSpice* window, we can add labels to it for documentation purposes. To add a title to the plot, select **Edit/Modify Title** in the *Orcad PSpice* menu and type the title in the dialog box. To add a label to the Y axis, select **Plot/Y Axis Settings**, type the label, and **CLICKL OK**. Add a label to the X axis in the same manner.

As an alternative approach, we can avoid running the *Orcad PSpice* program by using *pseudocomponents* to send results to the output file. Pseudocomponents are like parts that can be inserted into a schematic as if they were circuit elements, but they do not correspond to circuit elements. We can add them to the circuit for specifying initial conditions or for output control. In fact, we have already used two pseudocomponents, VIEWPOINT and IPROBE, for DC analysis. Other important pseudocomponents and their usage are shown in Fig. D.40 and listed in Table D.2. The pseudocomponents are added to the schematic. To add a pseudocomponent, select **Draw/Get New Parts** in the *Schematics* window, select the pseudocomponent, place it at the desired location, and add the appropriate attributes as usual. Once the pseudocomponents are added to the schematic, we select **Analysis/Setup/AC Sweep** and enter the specifications for the AC sweep, and finally, select **Analysis/Simulate** to perform AC sweep. If no errors are encountered, the voltages and currents specified in the pseudocomponents will be saved in the output file. We obtain the output file by selecting **Analysis/Examine Output** in the *Schematics* window.



**Figure D.40**

Print and plot pseudocomponents.

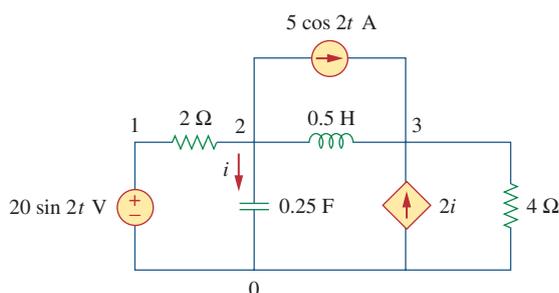
**TABLE D.2**

Print and plot pseudocomponents.

Symbol	Description
IPLOT	Plot showing branch current; symbol must be placed in series
IPRINT	Table showing branch current; symbol must be placed in series
VPLOT1	Plot showing voltages at the node to which the symbol is connected
VPLOT2	Plot showing voltage differentials between two points to which the symbol is connected
VPRINT1	Table showing voltages at the node to which the symbol is connected
VPRINT2	Table showing voltage differentials between two points to which the symbol is connected

Find current  $i$  in the circuit in Fig. D.41.

**Example D.6**



**Figure D.41**  
For Example D.6.

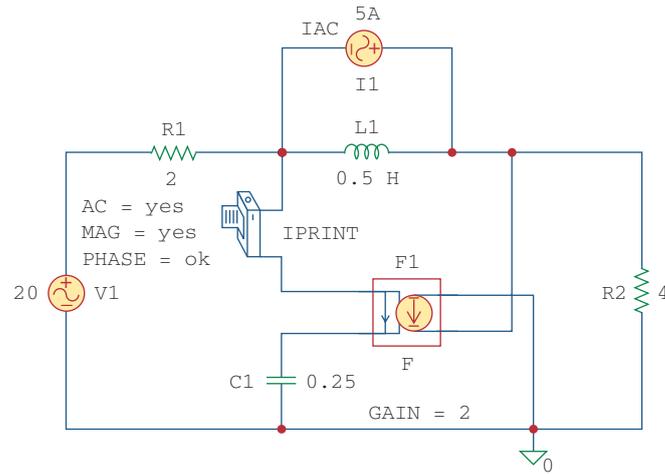
**Solution:**

Recall that  $20 \sin 2t = 20 \cos(2t - 90^\circ)$  and that  $f = \omega/2\pi = 2/2\pi = 0.31831$ . The schematic is shown in Fig. D.42. The attributes of V1 are set as  $ACMAG = 20$ ,  $ACPHASE = -90$ ; while the attributes of IAC are set as  $AC = 5$ . The current-controlled current source is connected in such a way as to conform with the original circuit in Fig. D.41; its gain is set equal to 2. The attributes of the pseudocomponent IPRINT are set as  $AC = \text{yes}$ ,  $MAG = \text{yes}$ ,  $PHASE = \text{ok}$ ,  $REAL =$ , and  $IMAG =$ . Since this is a single-frequency ac analysis, we select **Analysis/Setup/AC Sweep** and enter  $Total Pts = 1$ ,  $Start Freq = 0.31831$ , and  $Final Freq = 0.31831$ . We save the circuit and select **Analysis/Simulate** for simulation. The output file includes

```

FREQ          IM(V_PRINT3)    IP(V_PRINT3)
3.183E-01    7.906E+00        4.349E+01
    
```

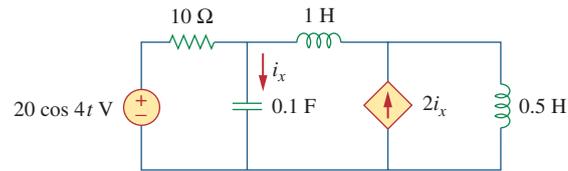
From the output file, we obtain  $I = 7.906/43.49^\circ$  A or  $i(t) = 7.906 \cos(2t + 43.49^\circ)$  A. This example is for a single-frequency ac analysis; Example D.7 is for AC sweep over a range of frequencies.



**Figure D.42**  
The schematic of the circuit in Fig. D.41.

**Practice Problem D.6**

Find  $i_x(t)$  in the circuit in Fig. D.43.



**Figure D.43**  
For Practice Prob. D.6.

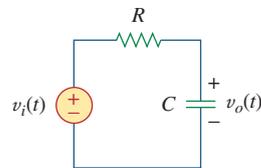
**Answer:** From the output file,  $I_x = 7.59 \angle 108.43^\circ$  or  $i_x = 7.59 \cos(4t + 108.43^\circ)$  A.

**Example D.7**

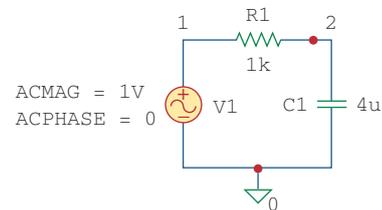
For the RC circuit shown in Fig. D.44, obtain the magnitude plot of the output voltage  $v_o$  for frequencies from 1 Hz to 10 kHz. Let  $R = 1 \text{ k}\Omega$  and  $C = 4 \mu\text{F}$ .

**Solution:**

The schematic is shown in Fig. D.45. We assume that the magnitude of V1 is 1 and its phase is zero; we enter these as the attributes of V1. We

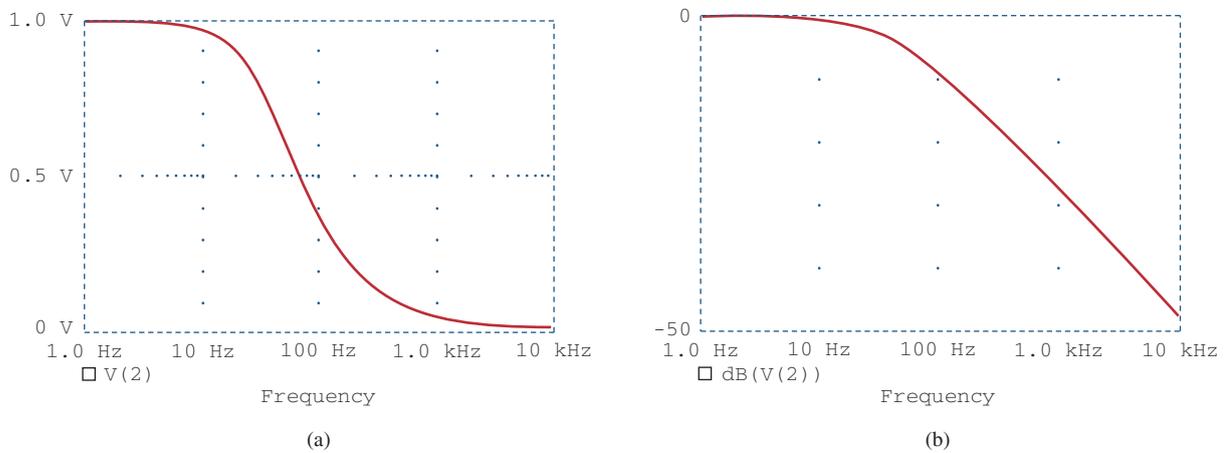


**Figure D.44**  
For Example D.7.



**Figure D.45**  
The schematic of the circuit in Fig. D.44.

also assume 10 points per decade. For the AC sweep specifications, we select **Analysis/Setup/AC Sweep** and enter 10 in the *Total Pts* box, 1 in the *Start Freq* box, and 10k in the *Final Freq* box. After saving the circuit, we select **Analysis/Simulate**. From the *Orcad PSpice* menu, we obtain the plot in Fig. D.46(a) by selecting **Traces/Add** and clicking V(2). Also, by selecting **Trace/Add** and typing dB(V(2)) in the *Trace Command* box, we obtain the Bode plot in Fig. D.46(b). The two plots in Fig. D.46 indicate that the circuit is a lowpass filter: low frequencies are passed while high frequencies are blocked by the circuit.



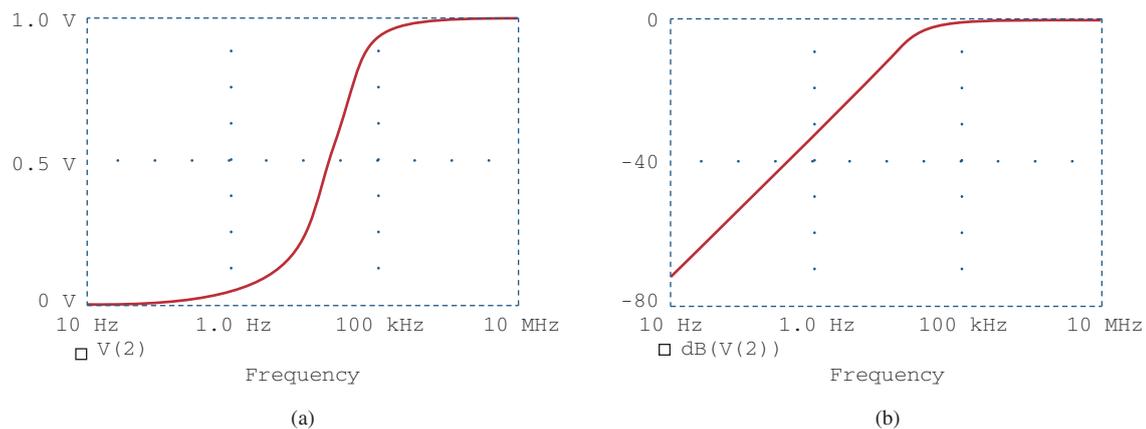
**Figure D.46**

Result of Example D.7: (a) linear, (b) Bode plot.

For the circuit in Fig. D.44, replace the capacitor  $C$  with an inductor  $L = 4$  mH and obtain the magnitude plot (both linear and Bode) for  $v_o$  for  $10 < f < 100$  MHz.

### Practice Problem D.7

**Answer:** See the plots in Fig. D.47.



**Figure D.47**

Result of Practice Prob. D.7: (a) linear plot, (b) Bode plot.