Symbulator

A symbolic simulator of linear electric circuits

Symbulator 9

Python / SymPy

Roberto Perez-Franco

Contents

Introduction **6**

What is Symbulator 6

Why use Symbulator? 6

Get Symbulator 7

Built-in examples 8

Reading and running side by side 8

Direct current analysis **9**

1.1 Run a direct current analysis 9

1.2 How to describe a resistor 10

1.3 What answers do you get 11

1.4 How to describe a voltage source 12

1.5 A numerical DC simulation, step by step 13

1.6 Compare it with your own work 17

1.7 Instructive numerical examples, solved 18

Symbolic circuits and Expert Mode **32**

2.1 About symbolic problems 32

2.2 Numerical from symbolic, with **Solve** 34

2.3 Numerical from symbolic, in **Expert Mode** 36

2.4 Instructive symbolic examples, solved 38

Current sources, conductances and dependent sources **42**

3.1 How to describe a current source 42

3.2 What about conductances? 43

3.3 How to reduce parallel resistors 44

3.4 How to describe dependent sources 47

3.5 Instructive solved examples 48

Shorts, equivalent resistance and Thévenin/Norton **80**

4.1 How to describe a short circuit . . . 80

4.2 Finding the equivalent resistance . . . 81

4.3 Finding a Thévenin / Norton equivalent . . . 83

4.4 Instructive solved examples . . . 86

Operational amplifiers **119**

5.1 How to describe an ideal operational amplifier . . . 119

5.2 Instructive Op Amp examples . . . 121

Transient analysis **153**

6.1 How to describe a capacitor . . . 153

6.2 How to describe an inductor . . . 154

6.3 Running a transient analysis . . . 154

6.4 Two useful time-savers . . . 156

6.5 Instructive solved examples . . . 157

Alternating current analysis **206**

7.1 AC analysis in Symbulator . . . 206

7.2 Solved numerical examples . . . 208

7.3 Solved numerical-from-symbolic examples . . . 217

7.4 Solved symbolic examples . . . 219

Power in AC circuits **222**

8.1 Average power . . . 222

8.2 RMS and PF . . . 223

8.3 Complex power . . . 225

Three-phase circuits **230**

9.1 About solving three-phase circuits . . . 230

9.2 Wye-Wye . . . 230

9.3 Wye-Delta . . . 236

9.4 Delta-Delta . . . 238

9.5 Delta-Wye . . . 242

Mutual inductance and ideal transformers **244**

10.1 How to describe mutual inductance 244

10.2 Instructive m problems 245

10.3 How to describe ideal transformers 248

10.4 Instructive t problems 249

10.5 Switching with coupled coils 255

Bode plots and resonance **263**

11.1 Do Bode plots 263

11.2 Resonance 268

The complex frequency domain **272**

12.1 s-domain analysis 272

12.2 The t2s and s2t shortcuts 274

12.3 Instructive FD problems 274

12.4 Transfer function problems 275

Two-ports **277**

13.1 Find two-port equivalents 277

13.2 Use two-ports as elements 278

13.3 Instructive two-port problems 280

13.4 The gain mini-tool 284

Tech Note A — Working with input files **294**

Loading input files 294

Saving your work 294

What an entry remembers 295

Tech Note B — Limiting the results to save time **296**

Asking for a few answers 296

When not to bother 297

Tech Note C — Underscores, and subscripts in your answers **298**

In an answer’s name, it is optional 298

In a symbolic value, it makes a subscript 298

Tech Note D — When a load is not the only thing attached **300**

The button 300

An example 300

Tech Note E — The SI prefixes Symbulator accepts **303**

Exact, and why that matters 303

Prefixes in the answers, not just the values 304

Tech Note F — Every control in the Settings card **305**

Rounding 305

Display 305

AC power convention 306

Roll the credits **307**

Who made Symbulator 307

Acknowledgements 309

Problem credits 310

Praise received 311

Licence 312

Index **313**

Introduction

Learn about Symbulator and how it can help you focus on learning how circuits work, instead of struggling with math. Get the latest version of Symbulator and get ready to roll.

What is Symbulator

Symbulator is an app that solves linear electric circuits symbolically and numerically, on any device with a browser. It takes numerical and symbolic values, and returns numerical and symbolic results. The odd-sounding name is a portmanteau of *"**symb**olic sim**ulator**"*. Symbulator is widely regarded as the best portable symbolic simulator of linear electric circuits.

What analyses can it do?

Symbulator can solve circuits using four types of analysis: direct current (DC), alternating current (AC), transient time-domain (TR), and complex frequency-domain (FD). It analyses numerical and, more importantly, symbolic linear circuits, returning the voltages in all nodes and the currents, voltage drops and power in all elements. It also finds the passive, Thévenin / Norton and two-port equivalent of a circuit, and draws time domain and Bode plots.

What elements does it accept?

Symbulator accepts elements such as resistors, inductors, mutual inductance, capacitors, independent and dependent current and voltage sources, ideal operational amplifiers, ideal transformers and six types of two-ports.

Where does it run?

Symbulator 9, released in 2026, is a port of the classic code to Python, powered by the SymPy computer algebra library. It runs anywhere Python runs: your mobile phone, tablet, laptop, a Jupyter notebook, or a browser on any computer. Version 9 solves circuits faster than versions 7 and 8.

Why use Symbulator?

Symbulator is extremely useful on the circuit theory problems engineering students face in Circuits I and II. It lets the student focus on understanding circuit analysis rather than on the mathematics of solving it.

Strengths

- A fantastic tool to solve circuits composed of ideal, linear elements.
- Shines in solving circuits with symbolic values, and symbolic results.
- Offers the absolute simplest way to define dependent sources.
- Includes ideal transformers and six different two-ports.

Limitations

- Does not simulate non-linear elements, such as diodes and transistors.
- Will not answer under-defined problems that rely on applying formulas selectively rather than solving a circuit completely.
- Is not a replacement for your brain or an excuse to not study. You must understand circuit theory to use it.

Symbulator is free!

Symbulator has always been free of charge, and always will be. *(You are welcome!)* Since 2026, Symbulator is also open source under the MIT licence. You are free to use it, change it and port it to other platforms; all the licence asks is that the copyright notice and the attribution to the original author travel with it.

Get Symbulator

You can use Symbulator 9 in two ways:

- as an online app running on a remote server, or
- as an offline app running locally on your device.

The easiest way to start is the **online app**.

To run it offline on your own device, follow the instructions at the bottom of the online app, under **Run Symbulator 9 locally**.

> CAUTION
>
> **Use at your own risk!**
>
> Every effort has gone into Symbulator, and it has no known bugs. Even so, it is provided “as is”, without warranty of any kind, express or implied, including but not limited to the warranty of fitness for a particular purpose. You use it at your own risk.

Please report any problems to help [at] symbulator [dot] com

Built-in examples

You do not have to type the tutorial's circuits. The **Built-in examples** list at the top of the app holds an entry for every simulation in every lesson; click a title to load one and solve it as if you had typed it yourself.

Working with input files (page 294) explains how to load them, save your own work, and download it to your device. It is a **technical note**, in the section of that name at the end of this tutorial: optional reading, there when you want it. Nothing in the lessons depends on it — you can follow every one of them without saving a file.

Reading and running side by side

This tutorial has you moving between the page you are reading and the app you are running. Two windows work. There is a tidier way.

The **split view** puts both in one window: the documentation on the left, the live app on the right. Drag the divider to resize them; on a phone or a narrow window they become two tabs. Both halves are the real thing, simply shown together.

The two halves are connected. On the website, every worked problem carries two links under its title:

- **Open in app ↗** loads that circuit, with its analysis and settings, ready to run.
- **Open in split view** does the same, but in the split view, on the problem you were reading.

Inside the split view, **Open in app ↗** loads the circuit into the right-hand pane instead of a new tab, and the second link is not shown.

> ⚠ WARNING
>
> **Loading a problem starts the app fresh**
>
> **Open in app ↗** reloads the app with the circuit you asked for, replacing anything you had typed there. To keep your own work, save it to an entry and download the file **before** you click the next problem. Working with input files (page 294) explains how.

A problem with more than one simulation carries a link for each, marked **DC** or **TR**. A few problems have no link, because they are worked by hand. In the PDF, the links are on the web edition at `learn.symbulator.com`; everything else on this page applies either way.

The split view keeps its address current as you move, so it can be shared where you are. Send someone:

```
learn.symbulator.com/split/?lesson=6a&entry=3
```

and they open on the same problem, with the same circuit loaded.

LESSON 1

Direct current analysis

Learn to describe a circuit in Symbulator, and how to run a *direct current* analysis. Learn how to describe a *voltage source* using **e**, and a *resistor* using **r**.

I firmly believe one learns best by *doing*, so I have written this documentation as a tutorial, where you learn Symbulator by working through a series of progressively more diverse and complex examples. In a moment, I'll teach you how to swim. But for now, let me throw you in the deep end!

1.1 Run a direct current analysis

Open Symbulator 9 — the local app if you have it installed, otherwise the **online app** in a browser. Click **Clear all inputs**, at the top right, so every input field is empty. Then type this into the **Circuit Description** box, one element per line:

> Circuit Description
>
> ```
> e,1,0,36
> r1,1,2,1'k
> r2,2,3,3'k
> r3,3,0,2'k
> ```

Scroll down to **Analysis & Settings** (if it is collapsed, open it with the blue triangle to its left). Set **Type of analysis** to *Solve circuit* and **Analysis** to *DC — direct current*. Then click **Run Symbulator**.

The button reads *Solving...* for a moment, then *Solved!*. Scroll down to **Results**, under **OUTPUTS**: there is an answer for every node and every element of the circuit.

Congratulations! You have just run your first simulation in Symbulator 9.

How does it work

Let's now go over what we just did, one piece at a time.

The two menus in **Analysis & Settings** tell Symbulator what to do (in this case, solve the circuit) and which analysis to run (in this case, in direct current). Everything else there is optional; later lessons introduce it, and until then leave it as it is.

The most important input is the description of the circuit you want to analyse. It goes in the **Circuit Description** area, as a series of lines of text, each describing one element.

Circuit Description

```
e,1,0,36
r1,1,2,1'k
r2,2,3,3'k
r3,3,0,2'k
```

Using one line per element keeps a long circuit readable, and it makes mistakes easier to find. But it is not the only option: Symbulator also accepts a colon as a separator. For example, this is the same circuit, described in one line:

Circuit Description

```
e,1,0,36:r1,1,2,1'k:r2,2,3,3'k:r3,3,0,2'k
```

To reuse this circuit later, save it as an **entry** in an **input file**. You will be asked to name both. Input files must have the `.cir` extension.

Look at the elements

Now let's examine the contents of the circuit description. Study it, and you will see it has four lines. Each line describes one element.

At the moment they may look cryptic. But by the end of this lesson you will read them with ease.

1.2 How to describe a resistor

Notice that the second, third and fourth elements in the list all start with the letter **r**. These are resistors.

> ⓘ NOTE
>
> **Describing resistors**
>
> An ideal resistor is described with four pieces of information, separated by commas: a unique name (which must start with the letter **r**), the names of its first and second node, and its value in ohms (Ω).
> For example, an ideal resistor called **r1**, between nodes **a** and **b**, of **300** Ω, is described `r1,a,b,300`

How to name resistors

Every resistor must have a name. As long as it starts with the letter **r** and is unique, it can be whatever you want. *Unique* means no other element or node has the same name.

How to name nodes

Naming the nodes is the first thing we do to solve a circuit. Every node must have a name, and node names must be unique: no other node or element may have the same one. Otherwise they can be whatever you want, with one constraint: at least one node

should be called **0** (zero). This will be considered the *ground node* of your circuit and will have, by definition, a voltage of 0 volts.

> ⚠ WARNING
>
> **You need a ground node**
>
> Every circuit must have a node called **0** to serve as reference or ground.

Values can use SI prefixes

The values of circuit elements are often given with prefixes of the International System (SI) — kilo, milli, micro. So Symbulator has a shorthand for them: an SI prefix in an element's value, preceded by an apostrophe, multiplies it by the corresponding factor, exactly.

Eleven prefixes are accepted, from peta down to atto. "SI prefixes" (in the Manual) lists them all with their factors, and covers the two where case matters — optional reading; `'k` and `'m` are the two this lesson uses.

> 💡 TIP
>
> **Prefixes welcome!**
>
> You can use SI prefixes, preceded by an apostrophe, in the values of elements.

For example, an 8 kΩ resistor can be entered as `8000` or as `8'k`, and both are exact. `8000.` and `8E3` are the same resistance approximately, which is a real difference once an answer is symbolic; "SI prefixes" (in the Manual) sets the four side by side.

1.3 What answers do you get

After the simulation in DC is complete, Symbulator fills in the **Results** section under **OUTPUTS** with every answer it worked out.

The results are arranged in groups: **Node voltages** first, one per node, and then **Results by element**, one group per element, headed by its name and its kind — **r1** *resistor*, **e** *voltage source*.

Every answer is written the way you would write it by hand, as a named quantity and its value: v_1 = 36 V, i_{r1} = 6 mA.

How answers are shown

By default, Symbulator answers *exactly*: it works symbolically, so a current comes back as 3/500 A rather than a decimal. That is what you want for symbolic results. For numerical work like the example in this lesson, decimals read better. Open the **Settings** card and set **Rounding** to *approx to n digits* with **n** = 3, then tick **Use SI prefixes in answers**. The current above then reads 6 mA instead of 3/500 A.

Ticking the prefix box moves **Rounding** off *exact* by itself, since a prefixed value is a decimal, and choosing *exact* again switches the prefixes off. Symbulator says so on screen

when it does.

In this tutorial, numerical examples assume approximate answers to three or four figures with prefixes on, and symbolic ones assume *exact*. The text says so when a change is needed.

The card holds a few more controls than these two, and the lessons reach for them as they come up. “Settings” (in the Manual) is the whole card in one place — optional reading, and nothing here waits on it.

Answer for each node

Symbulator calculates for each node a voltage with reference to ground, which is shown under **Node voltages** as `v` plus the node’s name: node **1**’s voltage is `v1`.

Answers for each resistor

Symbulator calculates for each resistor the following answers:

- The voltage drop in the resistor, defined as the voltage in the first node minus the voltage in the second node, in volts. For a resistor called **r5**, this is given in `vr5`.
- The current through the resistor, flowing from the first node towards the second, in amperes. For a resistor called **rx**, this is given in `irx`.
- The power consumed by the resistor, in watts. For a resistor called **r12**, this is given in `pr12`.

You may write the name of any of Symbulator’s answers in two ways: with or without an underscore. `ir1` and `i_r1` are the same current, `v2` and `v_2` the same voltage; capitals make no difference either. This tutorial uses the shorter spelling throughout. Underscores, and subscripts in your answers (page 298) says why both exist, and what an underscore does in a *value*, which is a different thing entirely — optional reading.

By now you should understand the description of the second, third and fourth elements in our example.

1.4 How to describe a voltage source

The first element in our example, on the other hand, started with the letter **e**. This element is a voltage source.

> NOTE
>
> **Describing voltage sources**
>
> In Symbulator, a voltage source is described with four pieces of information, separated by commas: a unique name (which must start with the letter **e**), the names of its positive and negative node, and its value in volts (V).
> For example, an ideal voltage source called **e1**, with its positive terminal in node **3**, its negative terminal in the ground node **0**, and a value of 12 volts, is described `e1,3,0,12`

How to name voltage sources

Every voltage source in your circuit must have a unique name that starts with the letter **e**, and it can be whatever you want.

Values can use SI prefixes

A voltage source's value can use SI prefixes too — they work in the values of every element.

Answers for a voltage source

For each voltage source, the following answers are calculated:

- The voltage drop in the source, defined as the voltage in the first node minus the voltage in the second, in volts. For a source called **e5**, `ve5`.
- The current through the source, flowing from the first node towards the second, in amperes. For a source called **ex**, `iex`.
- The power consumed — attention: not delivered, but consumed — by the source, in watts. For a source called **e12**, `pe12`. If we want the power delivered, we ask for the negative of this value.
- The equivalent resistance of the rest of the circuit, as seen by the source. For a source called **e2**, `re2`.

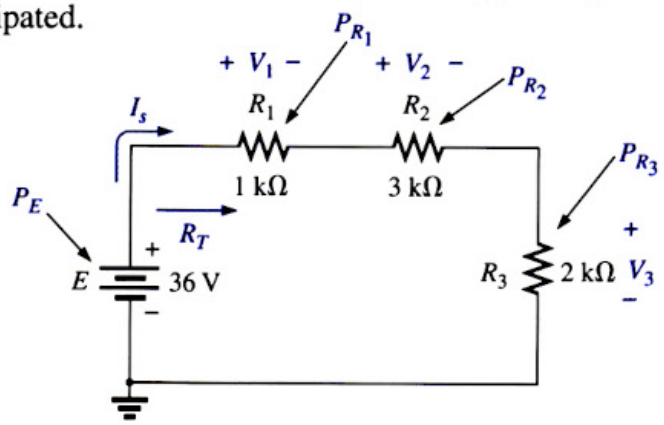
> ⓘ NOTE
>
> **For sources only**
>
> The equivalent resistance is a property of the *view from a source*, not for every element. It exists only for sources.

1.5 A numerical DC simulation, step by step

Let's go back now to the simulation we ran earlier. That simulation corresponds to the circuit given in Example 5.7 in Boylestad's *Introductory Circuit Analysis* (11ed). Moving forward, I will refer to that textbook as **B11**.

The problem statement and the circuit schematic are reproduced exactly as they appear in the textbook, as they will be for every other problem in this tutorial. Since this is for educational purposes, it is my understanding that it falls squarely within the "fair use" doctrine of copyright law. In any event, no copyright infringement is meant.

a. Determine the total resistance R_T.
b. Calculate the current I_s.
c. Determine the voltage across each resistor.
d. Find the power supplied by the battery.
e. Determine the power dissipated by each resistor.
f. Comment on whether the total power supplied equals the total power dissipated.



B11's Example 5.7

B11's Example 5.7

SOLUTION

All the values in this circuit are numbers: no element value is unknown. This is a *numerical circuit* — one where we know the numerical value of every element in it.

I will now walk you step by step through the solution. This process applies to most numerical simulations in Symbulator: (1) describe the circuit, (2) run the simulation, and (3) get the answers.

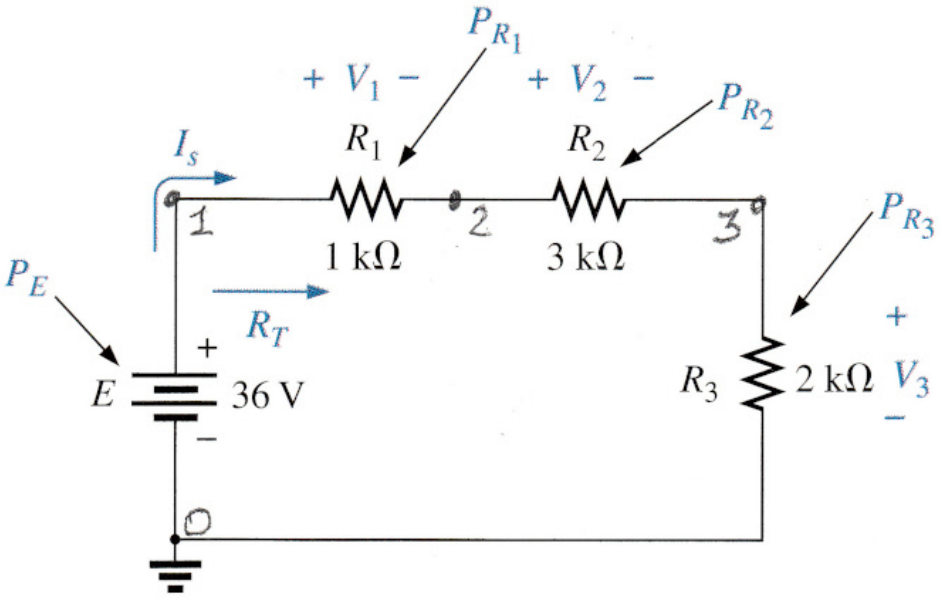
Step 1: Describe the circuit

Description starts with **naming the nodes**. As we said, you can call them anything you want, number or letter, as long as the name is unique — but one node must always be called **0** (zero), the ground node, with a voltage of 0 V. In this circuit the ground node is marked with the ground symbol. That is not always so; when it is not marked, you pick a node to serve as zero.

I labelled the nodes in this circuit, starting in the ground and moving clockwise, as 0, 1, 2 and 3. It helps me to pencil the names in the schematic itself.

> TIP
> **Write 'em down!**
> Write the names of your nodes and elements onto the circuit schematic itself, so you don't forget who's who.

After naming the nodes, I am ready to describe the elements of the circuit in Symbulator notation.



B11's Example 5.7, with the node names pencilled in

Let's start with the source: when I only have one voltage source, like here, I enjoy giving it a short name: **e**. So the voltage source is `e,1,0,36`: its name is **e**, its positive node is **1**, its negative node is **0**, and its value is 36 volts between them.

Now the resistors. I named the first **r1** and described it `r1,1,2,1'k`: its name is **r1**, its first node **1**, its second node **2**, and its value 1 kΩ. The second is `r2,2,3,3'k`, and the third `r3,3,0,2'k`.

> ⓘ NOTE
>
> **An important point about SI prefixes**
>
> In the case of kilo, we can write it as either `'k` or `'K`. But that's not the case for every other SI prefix: their case matters. For example, `'m` means milli, while `'M` means mega.

Those four element descriptions, gathered together, are the whole circuit. Put them in the **Circuit description** box, one to a line:

Circuit Description

```
e,1,0,36
r1,1,2,1'k
r2,2,3,3'k
r3,3,0,2'k
```

Step 2: Run the simulation

Under the box, choose the simulation:

- **Type of analysis** → *Solve circuit*
- **Analysis** → *DC — direct current*

Then press **Run Symbulator**.

The page fills in below — this circuit takes under a second on a laptop, a little more on a phone — and a note at the foot of **Results** says what happened: *DC analysis · 16 result(s)*, and how long it took.

For a DC analysis, these answers are as follows:

- The **voltage of each node**, so the voltages of nodes **1**, **2** and **3** are in `v1`, `v2` and `v3`, under **Node voltages**.
- The **current through each element**. The direction of the current is defined as going from the first node in the element's description to the second.
- The **voltage drop in each element**, defined as the difference in voltage between the first node in the element's description and the second, in that order.
- The **power consumed in each element**. An important point is that this is the power consumed, not the power delivered.
- Finally, **for each source**, **the equivalent resistance** of the rest of the circuit **as seen by that source**.

> ⓘ NOTE
>
> **The sign is in the details**
>
> To read the signs of your answers correctly, mind the details:
>
> - The current in the elements is defined as going through the element from the first node to the second node.
> - The voltage drop is defined as the voltage of the first node minus the voltage of the second node.
> - The power given is the power consumed. So, if you want power delivered, switch its sign.

Step 3: Get the answers

We can now answer the six questions in the problem.

Answer to question (a). The equivalent resistance as seen by the source e:

Nothing to compute: in **Results by element**, the resistance seen by the source reads `re` = **6** kΩ. Correct.

Answer to question (b). Current I_s is defined in the schematic as the current flowing through the source, in the direction that goes from node **0** to node **1**. One way to find this value is switching the sign of the current through the source, which we defined as flowing in the opposite direction:

The current through the source reads `ie` = −6 mA. The answer we want is its opposite, 6 mA.

Another way, since this is a series circuit where every element carries the same current, is to read the current through any resistor.

Answer to question (c). The voltage drop in resistor R_1 — since its polarity is defined in the schematic in the same way it is defined in our circuit description — is as follows.

The voltage drop `vr1` reads **6** V, `vr2` reads **18** V and `vr3` reads **12** V. All correct.

Answer to question (d). We know that the power consumed is given in `pe`. We get the power supplied by the source by switching the sign: 216 mW delivered.

Answer to question (e). The power consumed by the resistors is given in `pr1`, `pr2` and `pr3`: 36 mW, 108 mW and 72 mW, respectively.

Answer to question (f). Let's ask Evaluate whether the sum of the consumed power in the resistors equals the power supplied by the source:

If the resistors consume exactly what the source delivers, the four powers add up to zero. In **Evaluate**:

```
Evaluate
pr1 + pr2 + pr3 + pe
```

The answer is **0**.

This is the right answer, and concludes the solution to this, your first ever problem in Symbulator.

1.6 Compare it with your own work

You have just watched Symbulator solve a circuit, but not the way your course does it. It writes one equation for every element and one for every node, hands the whole system to its algebra engine at once, and reads the answers back. It never chooses a mesh, never draws a supernode, and never stops to ask which method would be quicker.

You have not seen that system yet, and it is worth looking at once before reading on. Open the **Settings** card and tick **Show equations**. An **Equations** card appears just above **Results**, open and listing every equation Symbulator assembled for the circuit you have just solved. You do not need to run the analysis again — the system is already there from the last solve, so the card fills in as soon as you tick the box.

Read it beside your own working and the difference is plain: one equation per element and one per node, in no particular order, with not a mesh or a supernode anywhere. That is why many of these equations look different to the ones in your notebook — and it is the reason for the card that comes next.

The **By-Hand Equations** card, found in the **TOOLS** section, does things differently. It writes out the system *you* would write — Kirchhoff's current law at each node, with a supernode where a voltage source sits between two of them, or Kirchhoff's voltage law around each mesh, with a supermesh where a current source is shared. It then solves that system on its own and checks its answers against the ones above, and tells you whether the two agree.

To use it, solve the circuit first, then open the card and press **Write the equations**. You do not have to pick a method: it works out which ones apply to your circuit, shows the one that needs fewer equations, and offers you the other if both apply. It draws your working on the circuit as well — a ring round every node whose equation is being written, a dashed line round each supernode or supermesh, and the mesh currents I_1, I_2, I_3 curling round their own loops. Every mesh current turns clockwise, as textbooks draw them, and ticking **Flip the mesh currents** turns them all the other way, which reverses

the arrows and the sign of every mesh current.

It is meant for exactly the moment when your homework answer disagrees with Symbulator's and you cannot see why. Both systems use the same names for the same quantities — **v2** is still the voltage at node 2, **ir3** is still the current through **r3** — so you can put your own page beside the card's and find the line where you and it parted company. That is usually a sign somewhere, and it is much easier to spot in one equation than in an answer.

Two things to keep in mind. Not every method suits every circuit: an op-amp has no mesh current to write, and coupled coils belong to mesh rather than to nodal — the card says so, and points you at the one that does apply. And since the **By-Hand Equations** card is a new feature, should the two ever disagree, I would trust the answers Symbulator gives above: they come from the engine that has been solving these circuits since 1999.

1.7 Instructive numerical examples, solved

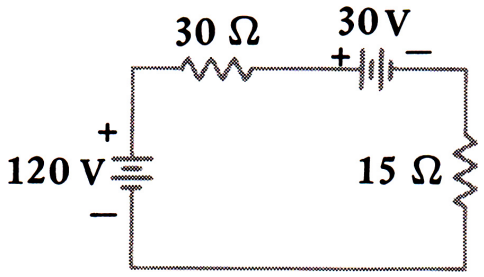
Numerical problems in DC using e and r

These practice problems are taken from several textbooks, chosen because they apply only the concepts you have learned so far. They let you practise and reinforce them before Lesson 2.

The problem below comes from Figure 1-26 (a) in Hyatt and Kemmerly's *Engineering Circuit Analysis* (5ed). Moving forward, we will refer to that textbook as **HK5**.

HK5's Figure 1-26

We are asked for the current, voltage drop and power consumed in each resistor, the power *delivered* by each voltage source, and a check that the powers in the circuit add up to zero.



Here is my solution.

I named the nodes thus: the bottom node **0**, and the top nodes, from left to right, **1**, **2** and **3**. My description of the circuit:

Circuit Description

```
e1,1,0,120
r1,1,2,30
e2,2,3,30
```

```
r2,3,0,15
```

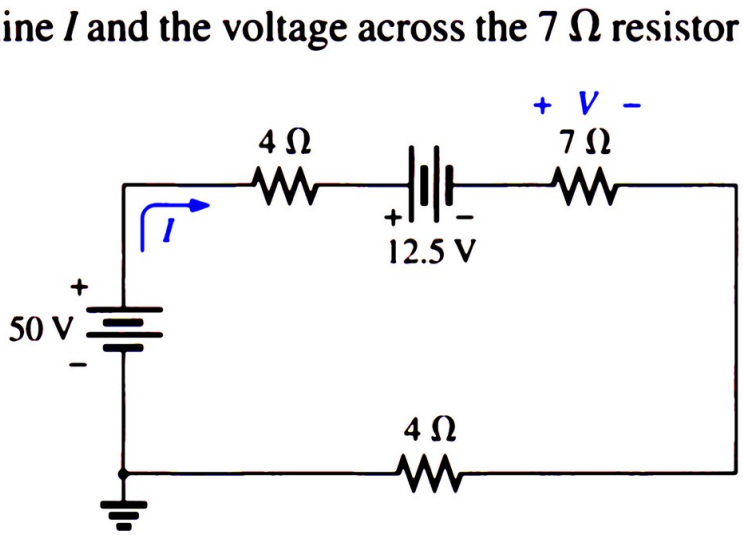
Run it, and the answers are all on the page already:

- The current through either resistor, `ir1` or, `ir2` is: **2** A
- `vr1` gives the voltage drop in the 30Ω resistor: **60** V
- `vr2` is the voltage drop in the 15Ω resistor: **30** V
- `pr1` is the power consumed in the 30Ω resistor: **120** W
- `pr2` is the power consumed in the 15Ω resistor: **60** W
- Flipping the sign of `pe1` gets the power delivered by the 120V source: **240** W
- Flipping the sign of `pe2` gets the power delivered by the 30V source: **-60** W. The source consumes 60W, so it delivers *negative* 60W.
- `pr1+pr2+pe1+pe2` is the sum of powers: **0** W. As expected.

Wasn't that easy?

B11's Example 5.20

The practice problems get progressively more complicated, so you build up your *'symbulating'* skills with confidence.



I named the nodes clockwise from the ground: **0**, **1**, **2**, **3** and **4**. My circuit description:

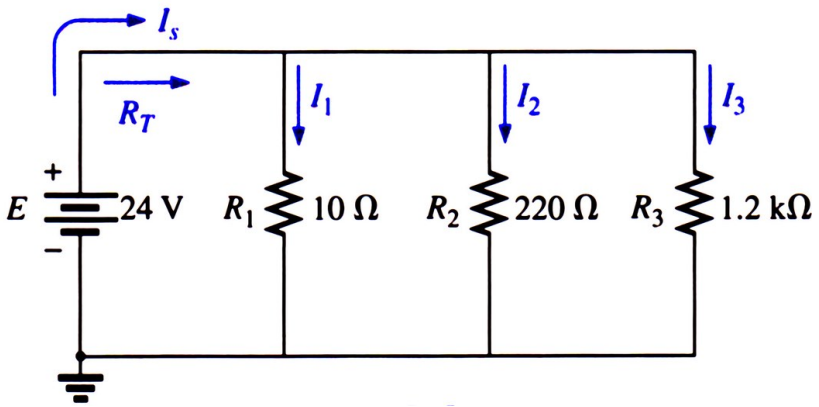
Circuit Description

```
e1,1,0,50
r1,1,2,4
e2,2,3,12.5
r2,3,4,7
r3,4,0,4
```

Run it and read the answers. `ir1` is the current I: **2.5** A. `vr2` is the voltage drop in the 7Ω resistor: **17.5** V

B11's Example 6.13

a. Find the total resistance.
b. Calculate the source current.
c. Determine the current through each branch.



Since the ground is the bottom node, I named it **0**. I named the top node **1**.

Circuit Description

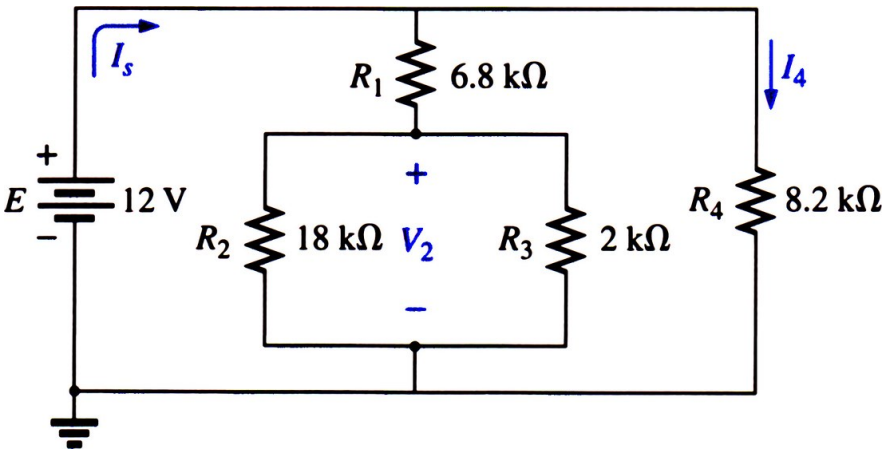
```
e,1,0,24
r1,1,0,10
r2,1,0,220
r3,1,0,1.2'k
```

- `re` is the total resistance: **9.49** Ω
- Flipping the sign of `ie` gets us the source current: **2.53** A
- `ir1` is I_1: **2.4** A, `ir2` is I_2: **109** mA, and `ir3` is I_3: **20** mA.

B11's Example 7.2

Determine I_4, I_S and V_2.

My solution: I named the top node **1**, and the other **2**:



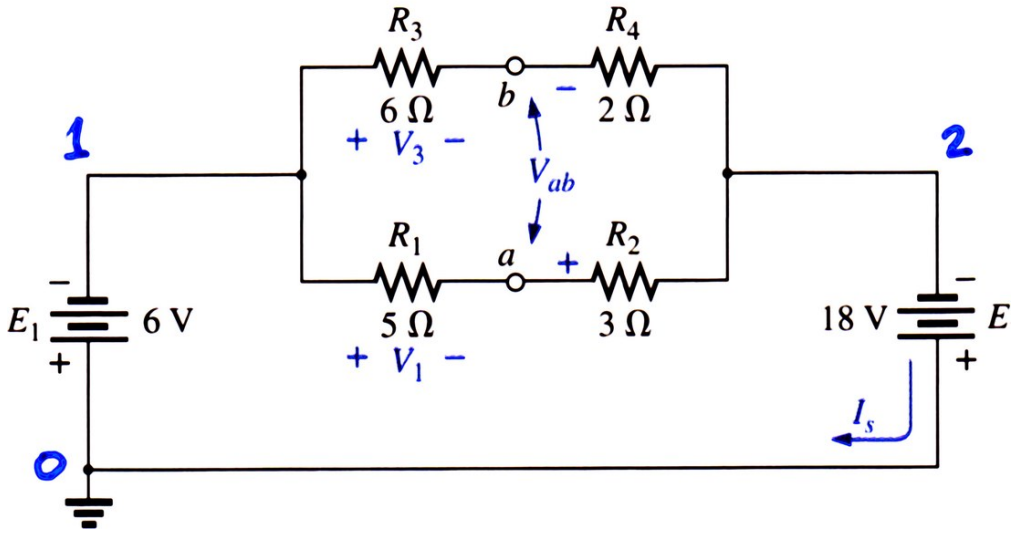
Circuit Description

```
e,1,0,12
r1,1,2,6.8'k
r2,2,0,18'k
r3,2,0,2'k
r4,1,0,8.2'k
```

Answers: `v2` is **2.51** V, the opposite of `ie` (i.e. I_S) is **2.86** mA and `ir4` is **1.46** mA.

B11's Example 7.7

a. Find the voltages V_1, V_3, and V_{ab} for the network
b. Calculate the source current I_s.



I have labelled the node names I used. My solution:

Circuit Description

```
e1,0,1,6
e2,0,2,18
r1,1,a,5
r2,a,2,3
r3,1,b,6
r4,b,2,2
```

Answers: `vr1` is **7.5** V, `vr3` is **9** V. For V_{ba}, `vb-va` is **-1.5** V. For I_S, the opposite of `ie2` is **3** A.

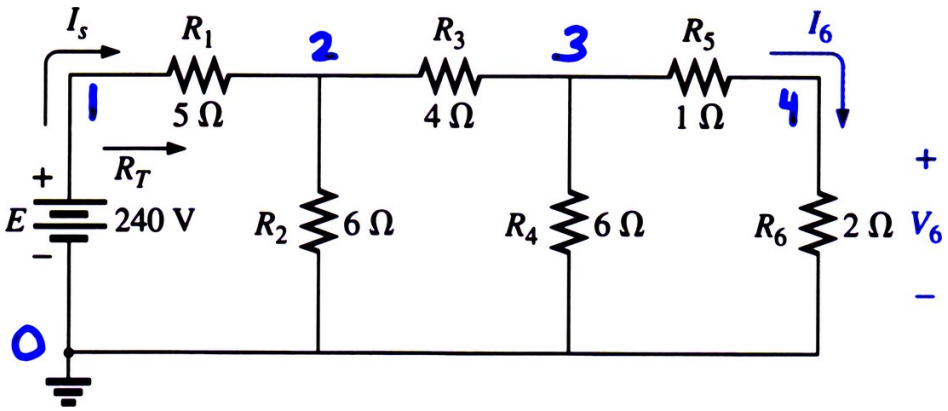
B11's Figure 7.32

Determine I_6 and V_6.

My solution:

Circuit Description

```
e,1,0,240
r1,1,2,5
```

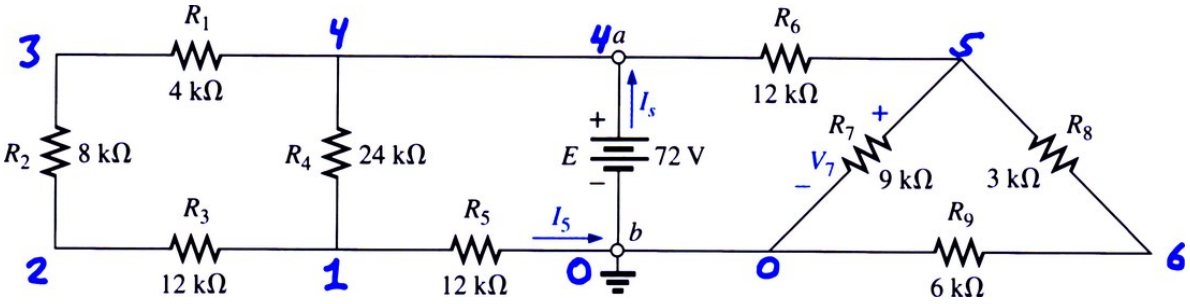



```
r2,2,0,6
r3,2,3,4
r4,3,0,6
r5,3,4,1
r6,4,0,2
```

Answers: `ir6` is **10** A, and `vr6` is **20** V.

B11's Example 7.10

Calculate the indicated currents and voltages.



My solution:

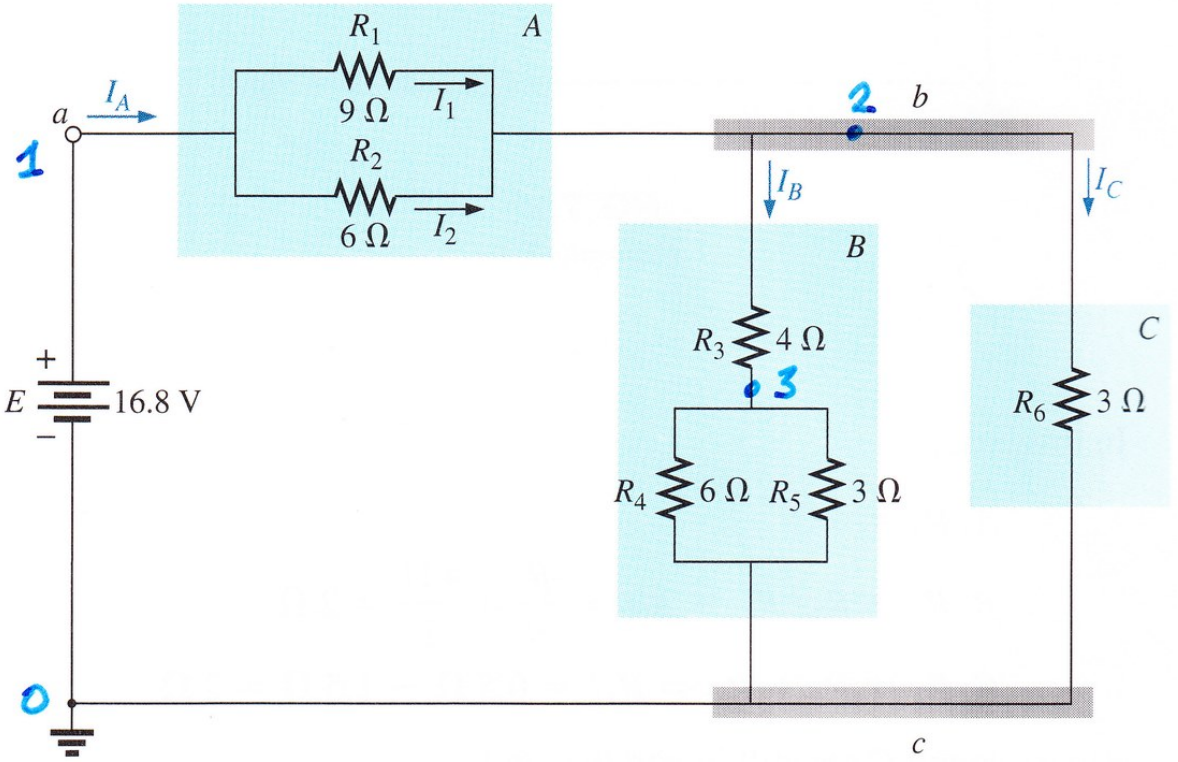
Circuit Description

```
r2,2,3,8'k
r1,3,4,4'k
r3,1,2,12'k
r4,1,4,24'k
r5,1,0,12'k
e,4,0,72
r6,4,5,12'k
r7,5,0,9'k
r8,5,6,3'k
r9,0,6,6'k
```

Answers: `ir5` is **3** mA, the opposite of `ie` (i.e. I_S) is **7.36** mA, and `vr7` is **19.6** V.

B11's Example 7.4

Determine the currents I_1, I_2, I_A, I_B and I_C, and the voltage drop in areas A, B and C.



My solution:

Circuit Description

```
e,1,0,16.8
r1,1,2,9
r2,1,2,6
r3,2,3,4
r4,3,0,6
r5,3,0,3
r6,2,0,3
```

Current I_1 is found via `ir1` = **1.2** A, I_2 via `ir2` = **1.8** A, I_A, via the opposite of `ie` = **3** A, I_B via `ir3` = **1** A and I_C via `ir6` = **2** A. The voltage drop in area A is `vr1` = **10.8** V; in both B and C it is `v2` = **6** V.

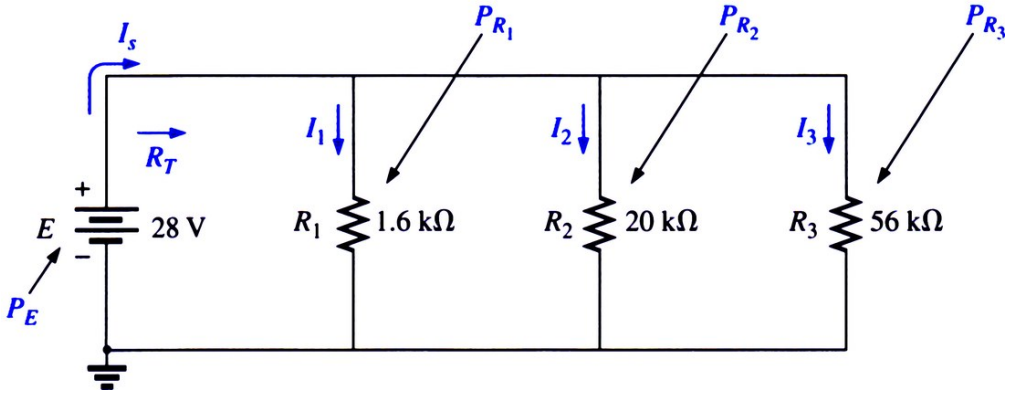
B11's Example 6.15

My solution:

Circuit Description

```
e,1,0,28
r1,1,0,1.6'k
r2,1,0,20'k
r3,1,0,56'k
```

a. Determine the total resistance R_T.
b. Find the source current and the current through each resistor.
c. Calculate the power delivered by the source.

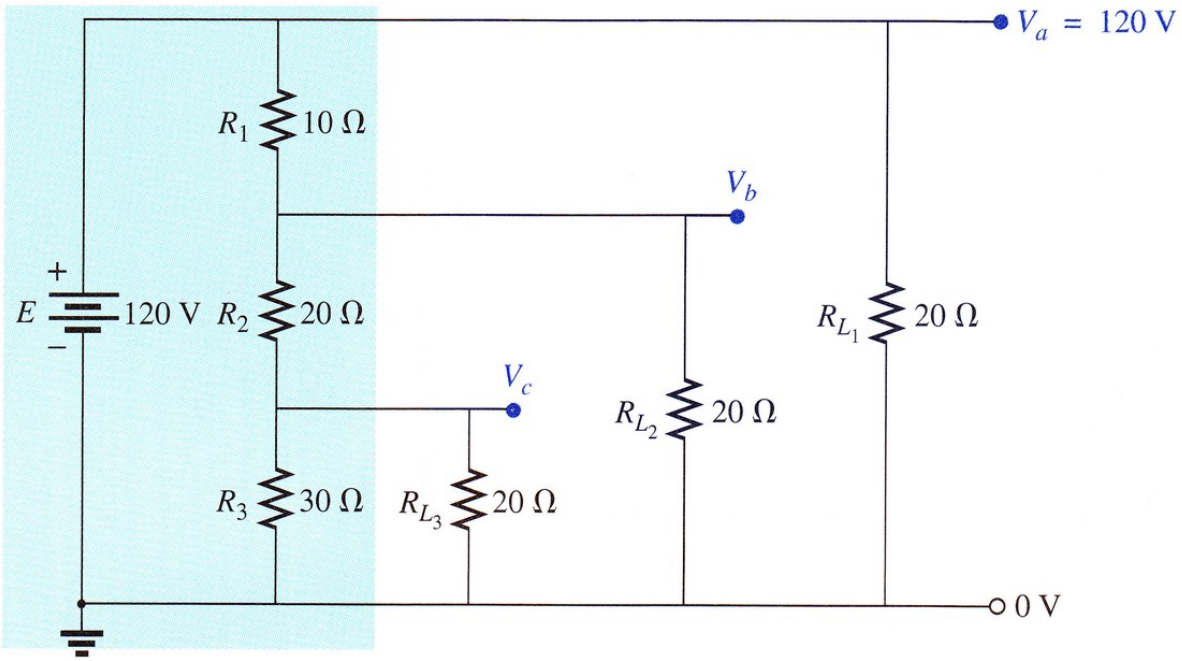


- `re` is the total resistance: **1.44** kΩ
- `ir1` is **17.5** mA, `ir2` is **1.4** mA, and `ir3` is **0.5** mA
- Flipping the sign of `pe` gets the power: **543** mW

These are the correct answers.

B11's Figure 7.40

Determine V_b and V_c.



My solution:

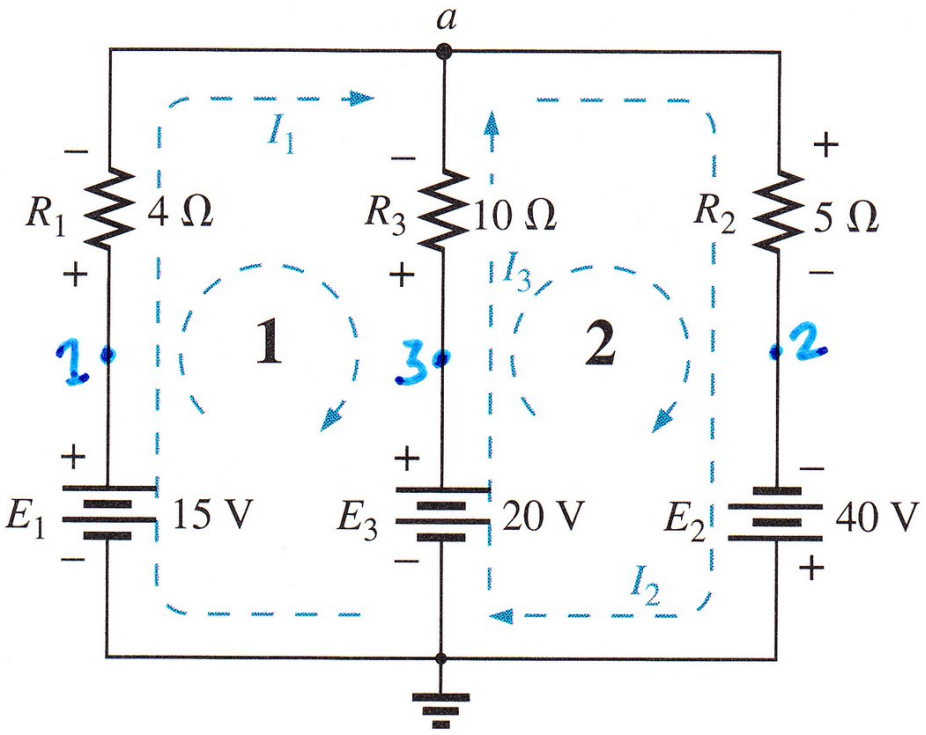
> Circuit Description
> `e,a,0,120`
> `r1,a,b,10`

```
r2,b,c,20
r3,c,0,30
rl1,a,0,20
rl2,b,0,20
rl3,c,0,20
```

Answers: `vb` is **66.21** V, and `vc` is **24.83** V.

B11's Example 8.10

Determine the current through each resistor.



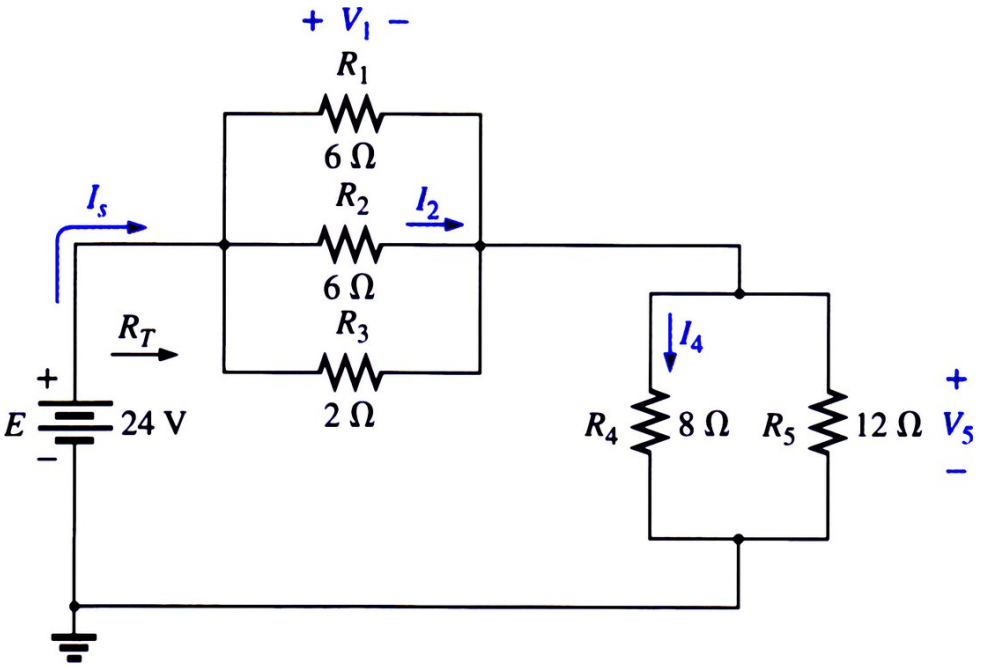
My solution:

```
Circuit Description
e1,1,0,15
r1,1,a,4
e3,3,0,20
r3,3,a,10
e2,0,2,40
r2,a,2,5
```

Reading the current through each of the three resistors, `ir1`, `ir2` and `ir3`: I_{R1} =**4.77**A, I_{R2} =**7.18**A and I_{R3} =**2.41**A. These are the correct answers.

B11's Example 7.6

My solution:



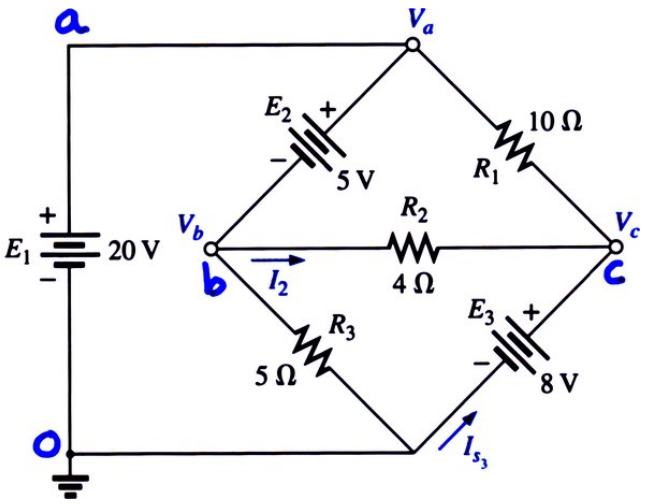
Circuit Description

```
e,1,0,24
r1,1,2,6
r2,1,2,6
r3,1,2,2
r4,2,0,8
r5,2,0,12
```

Answer: The opposite of `ie` is I_S=**4** A, `ir2` is I_2 = **.8** A, `ir4` is I_4=**2.4** A, `vr1` is V_1=**4.8** V, `vr5` is V_5=**19.2** V.

B11's Example 7.11

a. Determine voltages V_a, V_b, and V_c.
b. Find voltages V_{ac} and V_{bc}.
c. Find current I_2.
d. Find the source current I_{s_3}.



My solution:

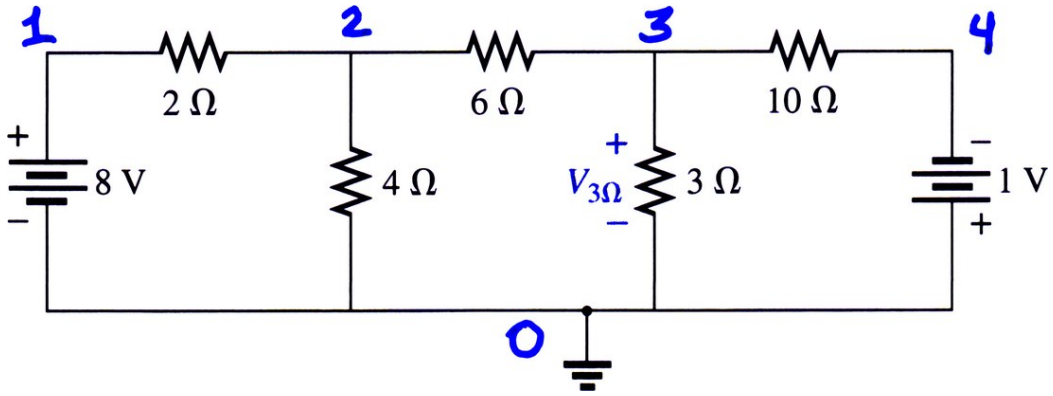
Circuit Description

```
e1,a,0,20
e2,a,b,5
e3,c,0,8
r1,a,c,10
r2,b,c,4
r3,b,0,5
```

Answers: va=**20** V, vb=**15** V, vc=**8** V, va-vc= **12** V, vb-vc=**7** V, ir2=**1.75** A, I_S via the opposite of ie3=**-2.95** A

B11's Example 8.24

Find the voltage drop in the 3Ω resistor.



My solution:

Circuit Description

```
e8,1,0,8
r2,1,2,2
r4,2,0,4
r6,2,3,6
r3,3,0,3
r10,3,4,10
e1,0,4,1
```

The voltage drop vr3 shows that $V_{3\Omega}$ is **1.1** V. This is correct.

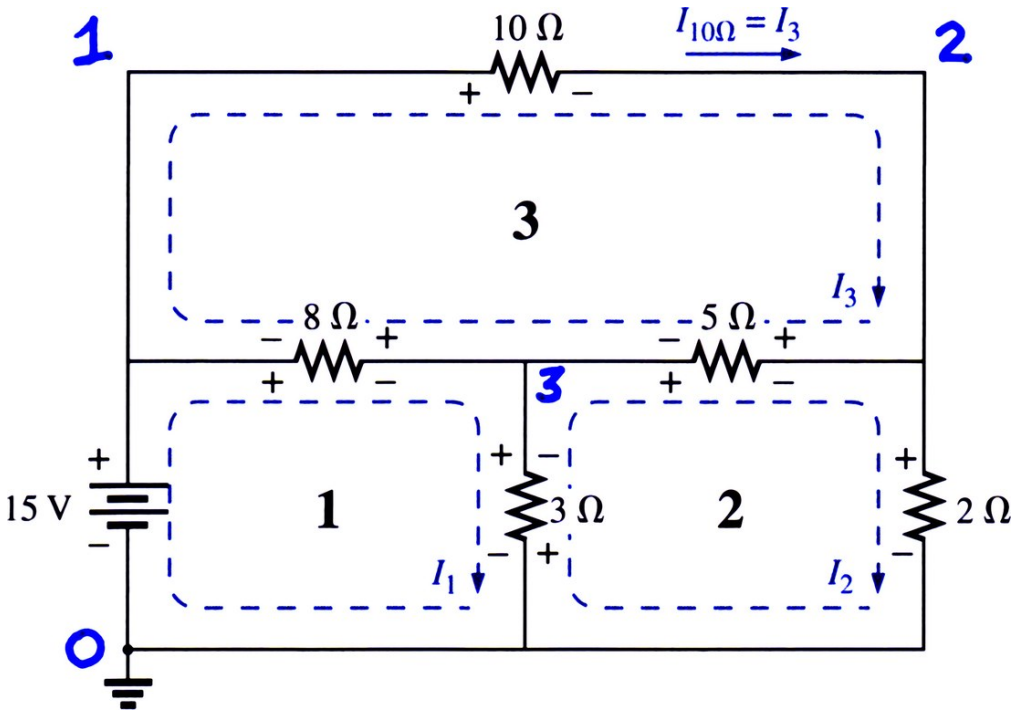
B11's Example 8.18

Find the current through the 10Ω resistor in the network shown.

My solution:

Circuit Description

```
e15,1,0,15
r10,1,2,10
r8,1,3,8
r5,3,2,5
r3,3,0,3
```




```
r2,2,0,2
```

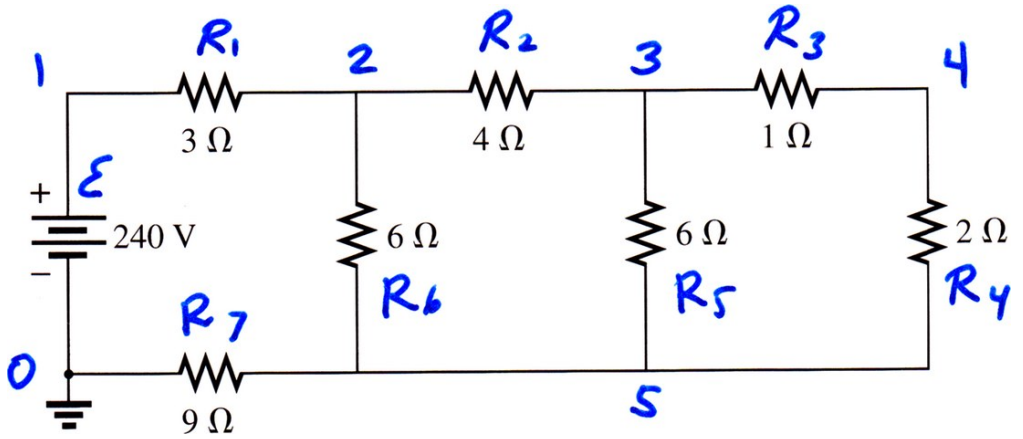
The answer you want is `ir10`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

We find that `ir10` = **1.22** A.

B11's Example 8.26

Find the voltage drop in the 2Ω resistor.



My solution is below:

Circuit Description

```
e,1,0,240
r1,1,2,3
r2,2,3,4
```

```
r3,3,4,1
r4,4,5,2
r5,3,5,6
r6,2,5,6
r7,5,0,9
```

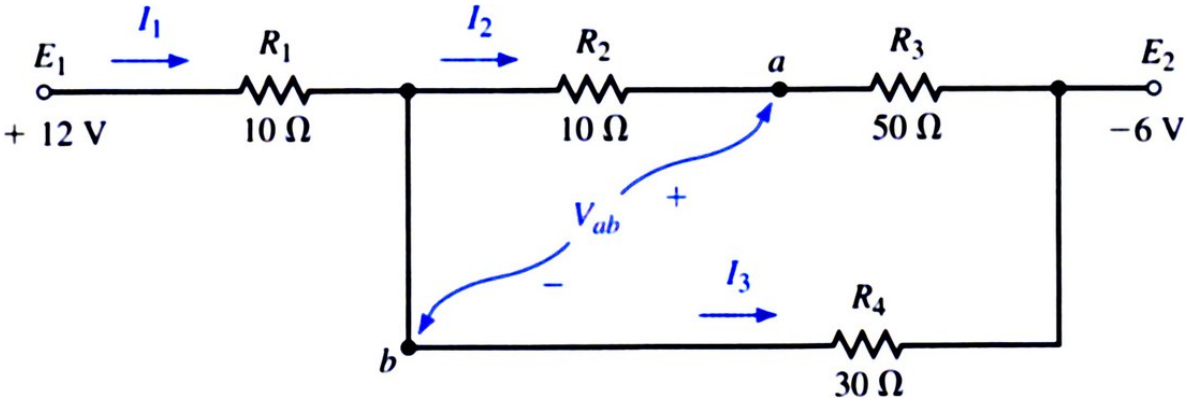
`vr4` gives the voltage drop in R_4 (the 2Ω resistor): it is **10.67** V

Circuits with 'hidden source'

Sometimes the schematics of circuits are presented in such a way that sources of voltage are not shown explicitly, yet their voltage is provided. These are what I call 'hidden source' problems. Below I offer two examples of these types of problems. Both are taken from the textbook *Circuit Analysis: Theory and Practice* (3ed) by Allan H. Robbins and Wilhelm C. Miller, to which from this point on we will refer as **RM3**.

RM3's Example 7-5 (Hidden source)

Find the indicated currents and voltages.



In my solution below, notice I introduced two sources, one for 12V and one for -6V:

Circuit Description
```
e1,1,0,12.
r1,1,b,10
r2,b,a,10
r3,a,2,50
r4,b,2,30
e2,2,0,-6
```

The answers you want are `ir1`, `ir2` and `ir4`, in **Results**.

Ask **Evaluate** for:

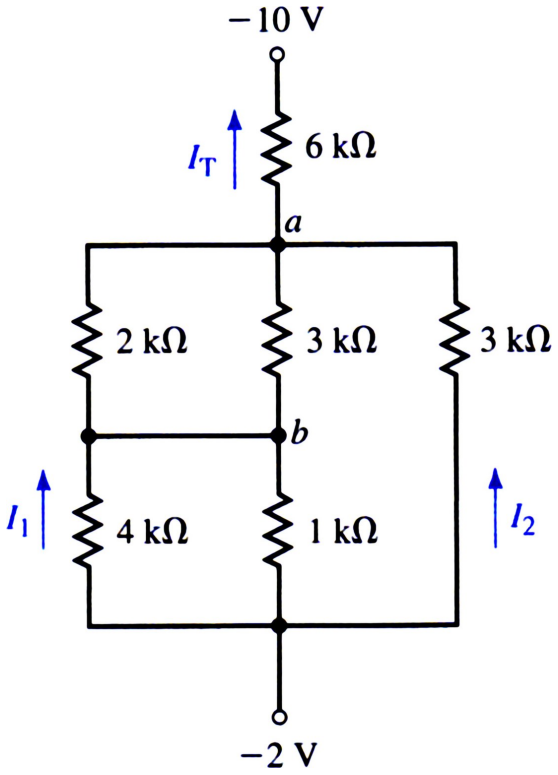
Evaluate
```
va-vb
```

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answer is I_1=**.6** A, I_2=**.2** A, I_3=**.4** A and V_{ab}=**-2** V. This is correct.

RM3's Figure 7-16 (Hidden source)

Find the total circuit resistance, and the indicated currents and voltages.



My solution:

This is how I named the nodes: the top, **1**; the bottom, **2**, and **a** and **b** as in the figure. In my solution below, notice I introduced two sources, one for -10V and one for -2V:

Circuit Description

```
r1,1,b,4'k
r2,1,a,3'k
r3,b,a,2'k
r4,b,a,3'k
r5,1,b,1'k
rt,a,2,6'k
e1,1,0,-2
e2,2,0,-10
```

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

Three answers are given directly in **Results**. The values of `irt`, `ir1` and `ir2` tell us that I_T=**1.11** mA, I_1=**.133** mA, and I_2=**.444** mA.

The other two answers can be found using **Evaluate**. The equivalent resistance, given by the expression:

$$\frac{v_1 - v_2}{I_T}$$

is found by evaluating:

> Evaluate
> `(v1-v2)/irt`

which gives a value of **7.2** kΩ.

And the value of V_{ab} is found by evaluating:

> Evaluate
> `va-vb`

which gives the value of **-0.8** V. This is correct.

LESSON 2

Symbolic circuits and Expert Mode

Learn how to solve different types of *symbolic* circuits in Symbulator. Use **Solve** in Symbulator. And explore the powerful **Expert Mode**, for the ambitious user.

Here you will learn about the types of symbolic circuits and how to solve them in Symbulator, using **Solve** when needed, and — for the more adventurous — the powerful **Expert Mode**.

2.1 About symbolic problems

The problem we saw in Lesson 1 (page 9) was a *numerical problem* because it involved a numerical circuit: from the start, we knew the values of all its elements. None were unknown.

A circuit with one or more elements whose value we do not know is a **symbolic circuit**. A *symbolic problem* is one that involves a symbolic circuit. It gives me great joy when I encounter a symbolic problem, because the ability to simulate symbolic circuits is what sets Symbulator apart from other simulators.

> ⓘ NOTE
>
> **Variables are case-sensitive; names are not**
>
> Element and node names ignore case — **R1** and **r1** are the same resistor, and so is any reference built from a name: `2*VR1`, `2*vr1` and `2*v_r1` all mean r1's voltage drop. A variable that names nothing in the circuit is different: `e,1,2,c` and `e,1,2,C` are two *different* symbolic sources, and telling Symbulator that `c = 5` says nothing about `C`. Pick one spelling for each unknown and keep it.

I find it useful to distinguish between two types of symbolic problems:

- **Purely symbolic problems** use symbolic circuits whose answers are also symbolic, given as expressions of unknown variables. Solving them in Symbulator is as straightforward as solving numerical problems.
- **Numerical-from-symbolic problems** start with a symbolic circuit but ask for numerical answers. That is possible when the problem gives us more information about the circuit.

Numerical-from-symbolic problems can be solved in two ways. If they are simple — one or two unknown values, one or two numerical answers wanted — it is quicker to simulate the circuit symbolically and then solve for the numerical answers with the **Solve** card, under the results. If they are not simple, it is easier to use **Expert Mode**.

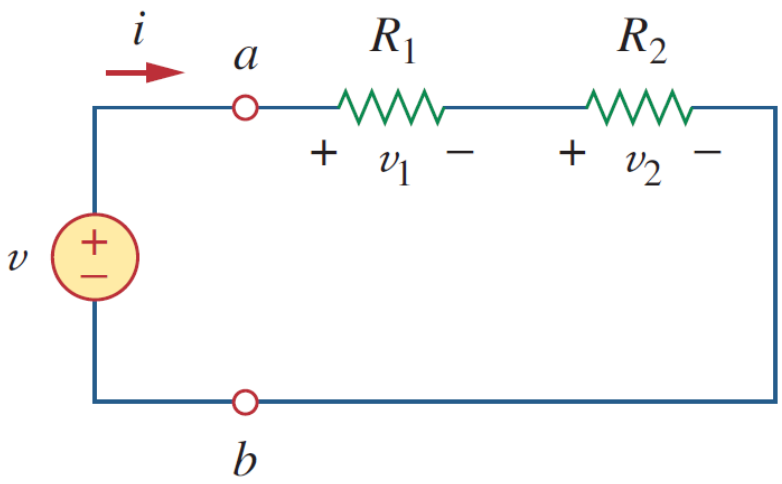
Let's see an example of each.

Solving a purely symbolic problem

Since Symbulator is natively a symbolic simulator, there is nothing special about solving a purely symbolic problem. The only limitation comes from the computing capacity of your device: while small circuits are solved almost instantly, large symbolic circuits can take longer to solve, and on top of that, they may produce expressions too large to be useful to a human.

Let's see a simple purely symbolic problem, from *Fundamentals of Electric Circuits* by Alexander and Sadiku (5th edition), from now on referred to as **AS5**.

AS5's Figure 2.29



AS5's Figure 2.29

SOLUTION

All the values in this circuit are variables: no element value is known. This is an example of a symbolic circuit. Imagine you are asked to find symbolic expressions for the current ***i***, the voltage drop in R_2, and the power consumed by R_1 and R_2. Since we are being asked to find symbolic answers from a symbolic circuit, this is a purely symbolic problem.

This is how I would describe this circuit. First, the nodes. The node called **a** in the schematic stays node **a**. The node called **b** is my ground, so it becomes node **0**. The node between the two resistors I call **c**.

Then the elements. I name the source **ev** and the resistors **r1** and **r2**, and give them variables for values: **v** for the source, **r1** and **r2** for the resistors. It is no problem that a symbolic value carries the same name as its element.

> TIP
>
> **Turn rounding back to exact for this lesson**
>
> Lesson 1 (page 9) had you set **Rounding** to *approx to n digits*. There is nothing to round here, so open **Settings** and put it back to *exact*. The **Use SI prefixes** tick

clears itself: a prefix has no place in an answer like

$$\frac{v}{r_1 + r_2}$$

Type the three elements into the box, one to a line:

Circuit Description

```
ev,a,0,v
r1,a,c,r1
r2,c,0,r2
```

Leave **Type of analysis** on *Solve circuit*, set **Analysis** to *DC — direct current*, and press **Run Symbulator**.

The answer comes back at once. To find the current i, we read `ir1` or `ir2`. To find the voltage drop on r2, we read `vr2`.

Both are expressions rather than numbers:

current through r1

$$i_{r1} = \frac{v}{r_1 + r_2}$$

voltage drop in r2

$$v_{r2} = \frac{r_2\, v}{r_1 + r_2}$$

which is the voltage divider you would have written by hand.

To find the power consumed in r2, we read `pr2`. To find the power delivered by the source, we read the *power delivered* row of the source's card, `-pev`.

$$p_{r2} = \frac{r_2\, v^2}{(r_1 + r_2)^2}$$

$$-p_{ev} = \frac{v^2}{r_1 + r_2}$$

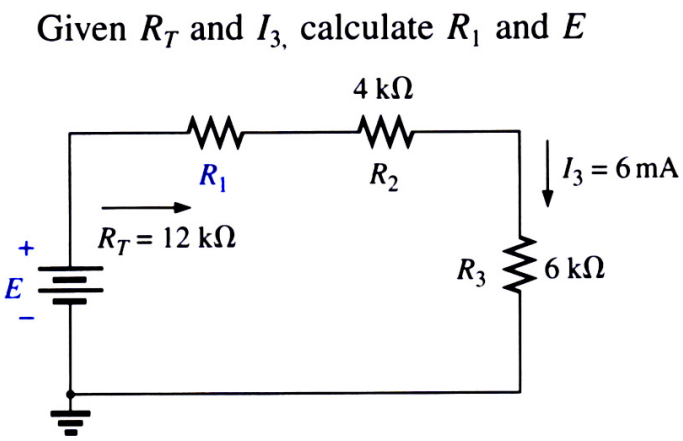
It is this ability to simulate symbolically, as if it were nothing, that puts Symbulator in a league of its own. Symbolic answers from a symbolic circuit take no extra effort.

2.2 Numerical from symbolic, with Solve

Getting numerical answers from a symbolic circuit, on the other hand, takes an extra step: **Solve**, as in this next example, or **Expert Mode**, as in the one after.

B11's Example 5.6, with Solve

SOLUTION



B11's Example 5.6

This is a very nice numerical-from-symbolic problem. We can solve it into numbers because, although it hides two values from us (the source **E** and the resistor ***R1***), it gives us in exchange two answers (the equivalent resistance ***RT*** and the current ***I3***) to solve for them with.

Since this circuit is structurally identical to **B11's Example 5.7 (page 14)** solved in Lesson 1 (page 9), we will use the same names for the nodes. The description is identical except for the values. Symbulator accepts numbers, variables or even algebraic expressions as values. Here I use **e** for source **e**, and **r1** for resistor **r1**.

Circuit Description

```
e,1,0,e
r1,1,2,r1
r2,2,3,4'k
r3,3,0,6'k
```

Run it in DC, as before.

A moment later Symbulator is done. From its symbolic answers and the answers the problem gives us, we write two equations and solve them for the two unknowns.

The problem says that ***I3*** is 6 mA. In Symbulator, ***I3*** is `ir3`, the current through resistor **r3**. Look at it and you will see an expression rather than a number, in terms of the two unknowns, **e** and **r1**:

current through r3

$$i_{r3} = \frac{e}{r_1 + 10000}$$

This is what we call a symbolic answer. The problem also says that ***RT*** is 12 kΩ; the equivalent resistance as seen by the source **e** is `re`, which is:

resistance seen by e

$$r_e = r_1 + 10000$$

We can then write two new equations and solve them for **e** and **r1**.

That is what the **Solve** card, under the results beside **Evaluate**, is for. It has two boxes.

In the first box, headed **Equation(s) to solve in terms of the results**, write the equations, using the same names the results use.

The shorthand works in these boxes too, so `re = 12'k` reads as 12000.

You can write the equations separated by `and`, like this:

> Equation(s) to solve in terms of the results
> `re = 12'k and ir3 = 6'm`

Or one equation per line, like this:

> Equation(s) to solve in terms of the results
> `re = 12'k`
> `ir3 = 6'm`

In the second box, headed **Unknown(s) to solve for**, name what you want found. You can write the unknowns separated by commas, like this:

> Unknown(s) to solve for
> `e, r1`

Or one unknown per line, like this:

> Unknown(s) to solve for
> `e`
> `r1`

Then press **Solve equations**.

An instant later we get the answers: the card lists **e** = **72.0** and **r1** = **2000.0**: a 72 V source and a 2000 Ω resistor. These are the right answers. Not many other circuit simulators allow this flexibility.

2.3 Numerical from symbolic, in Expert Mode

Symbulator's true strength is seen in numerical-from-symbolic problems like the one we solved above, when we use its **Expert Mode** of simulation. **Expert Mode** cracks these problems open in a single call, and gives fully numerical answers whenever the problem provides as many answers as it hides values.

Expert Mode takes extra equations, unknowns and conditions in three boxes. Open the **Expert Mode** box and tick **Enable Expert Mode** to see them:

- **Add equation(s)** — one or many, separated by `and` or one per line, written in the names the results use.
- **Add unknown(s)** — one or many, separated by commas or one per line.
- **Add condition(s)** — one or many, separated by `and` or one per line, for narrowing a solution down, which we will use later.

They apply to whatever analysis you run, so **Expert Mode** works the same way in DC, AC, FD and TR.

B11's Example 5.6, using ex

SOLUTION

We use the same circuit description as before, **r1** included.

The problem gives us the two extra equations: the source sees 12 kΩ, and the current through R_3 is 6 mA. They use the names the answers come back under, and the SI shorthand works here too:

Circuit Description

```
e,1,0,e
r1,1,2,r1
r2,2,3,4'k
r3,3,0,6'k
```

Add equation(s)

```
re = 12'k
ir3 = 6'm
```

Add unknown(s)

```
e, r1
```

Run it in DC. This time everything comes back numerical: the node voltages read v_1 = 72 V, v_2 = 60 V and v_3 = 36 V, and at the foot of **Results by element** are the two unknowns, listed like any other answer: **e** = **72** and **r1** = **2000**. Both are right, and the whole circuit was solved with them in one step.

The speed advantage of **Expert Mode** is not necessarily evident in this simple problem. But it does give you an idea of what **Expert Mode** is all about: you get to hand Symbulator extra information before it solves, rather than after. Had this circuit been larger, the benefit in computation time would be clear.

> ⓘ **NOTE**
>
> **What you can write an equation about**
>
> Anything the simulation reports: node voltages such as `v1`, currents `ir3`, voltage drops `vr1`, powers `pr2`, and the resistance or impedance a source sees, `re` or `ze`. The one exception is AC power (`s`, `p` and `ap` in an AC analysis), which involves a complex conjugate and cannot be solved for; Symbulator says so, and you can restate the constraint in voltages and currents.

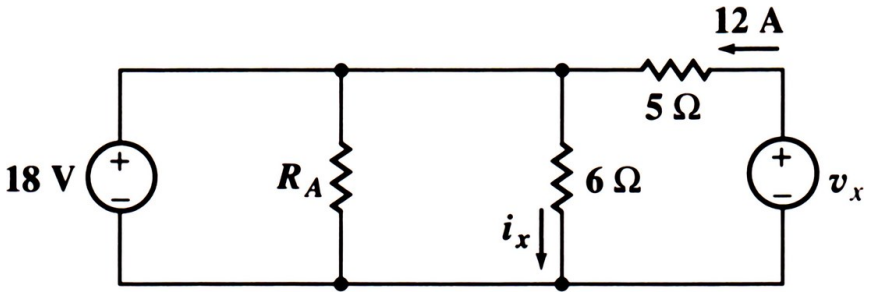
> **TIP**
>
> **Why you may get two answers**
>
> A quantity that is quadratic in an unknown, such as a power, can be satisfied by two values. Symbulator returns multiple solutions when they exist.

2.4 Instructive symbolic examples, solved

Numeric-from-symbolic, with Solve

HK5's Figure 1-24a (*solve*)

Determine i_x and v_x in the following circuit:



I named the nodes thus: bottom is **0**, top left is **1**, top right is **2**. I named the elements according to their value: this facilitates remembering who's who in the circuit. I also defined the resistors' nodes in the direction of the current indicated in the diagram.

Circuit Description

```
e18,1,0,18
ra,1,0,ra
r6,1,0,6
r5,2,1,5
evx,2,0,vx
```

Explore the answers. Since **ir5** (which we know is 12 A) is in terms of v_x, we can find v_x:

Solve it in DC, then use the **Solve** card:

Equation(s) to solve in terms of the results

```
ir5 = 12
```

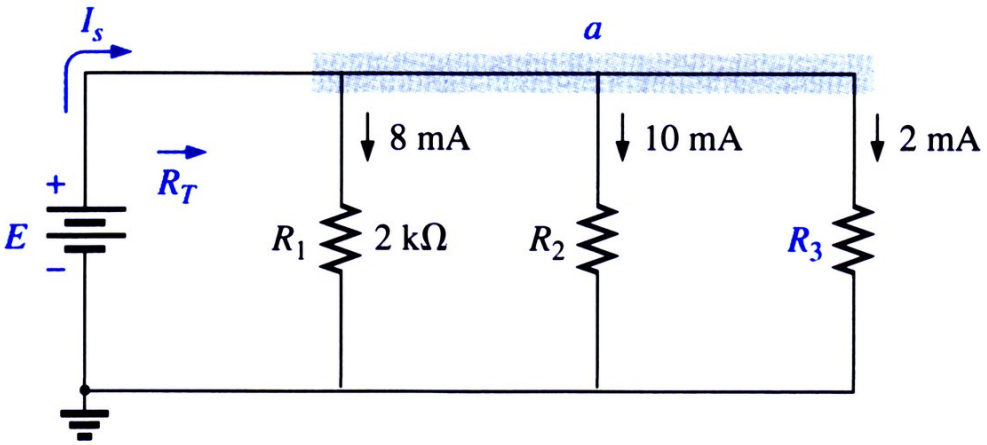
Unknown(s) to solve for

```
vx
```

v_x is **78** V, which is correct. The current `ir6` already shows that i_x is **3** A. The fact that we can find numerical answers in this problem can be quite puzzling until one realizes that ignoring the value of R_A doesn't matter: due to the circuit's structure, it is not needed to answer the two questions we have been asked.

Numeric-from-symbolic, in Expert Mode

a. Determine the source current I_s.
b. Find the source voltage E.



B11's Example 6.19 (Expert)

This problem, having three unknown element values and three known answers, is a perfect candidate for **Expert Mode**. Below is my circuit description.

Circuit Description

```
e,1,0,e
r1,1,0,2'k
r2,1,0,r2
r3,1,0,r3
```

Choose DC and, in **Expert Mode**, add these three equations:

Add equation(s)

```
ir1 = 8'm
ir2 = 10'm
ir3 = 2'm
```

Add these unknowns:

Add unknown(s)

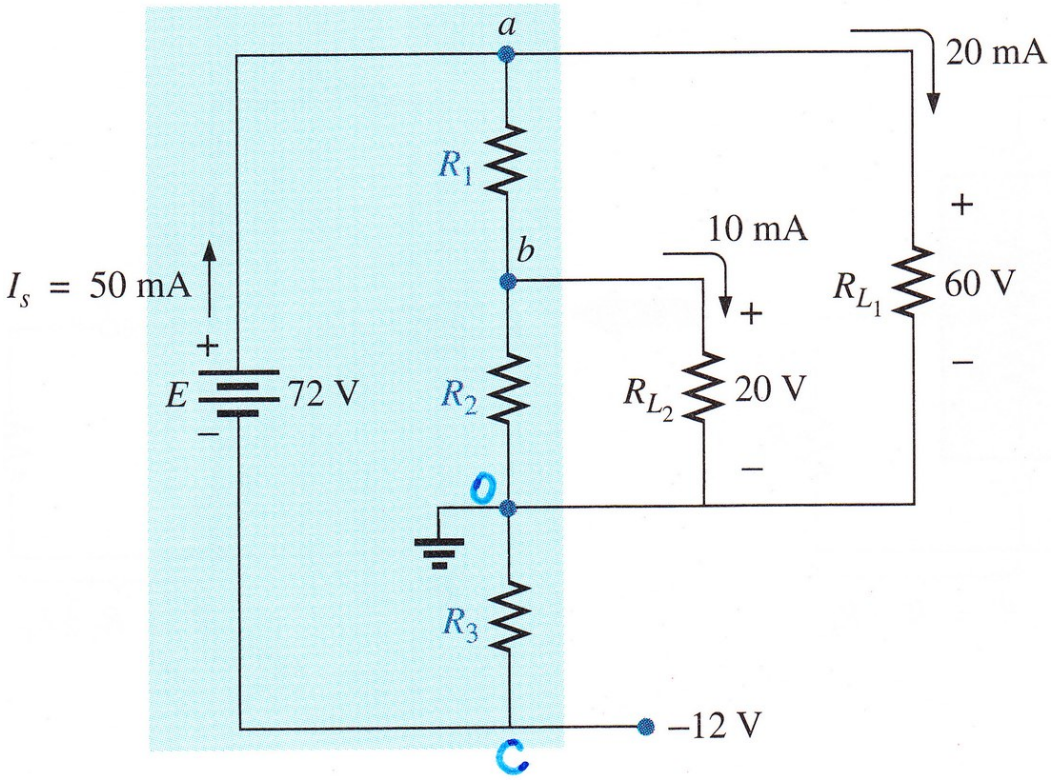
```
e, r2, r3
```

Run it. Everything comes back numerical: **e** = **16**, so E is **16** V, and `ie` = −0.02 A, so I_S is **0.02** A. Both correct. The two resistors come back too: **r2** = **1600** Ω and **r3** = **8000** Ω.

B11's Example 7.12 (Expert)

Determine R_1, R_2 and R_3 for the voltage divider supply. Can 2W resistors be used?

This problem is also perfect for **Expert Mode**, because: (a) the target is to obtain numerical values, and (b) we have N unknown element values, and in turn we are given N numerical answers. Here is how I solved it:



Circuit Description

```
e,a,c,72
r1,a,b,r1
r2,b,0,r2
r3,0,c,r3
rl1,a,0,rl1
rl2,b,0,rl2
```

Choose DC and, in **Expert Mode**, add these five equations:

Add equation(s)

```
irl1 = 20'm
vrl1 = 60
irl2 = 10'm
vrl2 = 20
-ie = 50'm
```

List these five as the unknowns to solve for:

Add unknown(s)

```
r1, r2, r3, rl1, rl2
```

Run it, and read the answers straight off the results: `r1` = **1333.3** Ω, `r2` = **1000** Ω, `r3` = **240** Ω. These are correct.

The power consumed in each resistor is given in:

> power consumed by r1
>
> $$p_{r1} = 1.2\,\mathrm{W}$$

> power consumed by r2
>
> $$p_{r2} = 0.4\,\mathrm{W}$$

> power consumed by r3
>
> $$p_{r3} = 0.6\,\mathrm{W}$$

Since all are smaller than 2W, it is possible to use 2W resistors in the design.

LESSON 3

Current sources, conductances and dependent sources

Learn how to describe a *current source* using the **j** element. Simplify *parallel resistors* using the **pr** function and its shorthand. Learn to describe *conductances* and *dependent sources*.

In this lesson you will learn how to describe a *current source* with the **j** element, and a trick to simplify *parallel resistors* with the **pr** function or its shorthand. You will also learn to describe *conductance* and *dependent sources* using elements you already know.

3.1 How to describe a current source

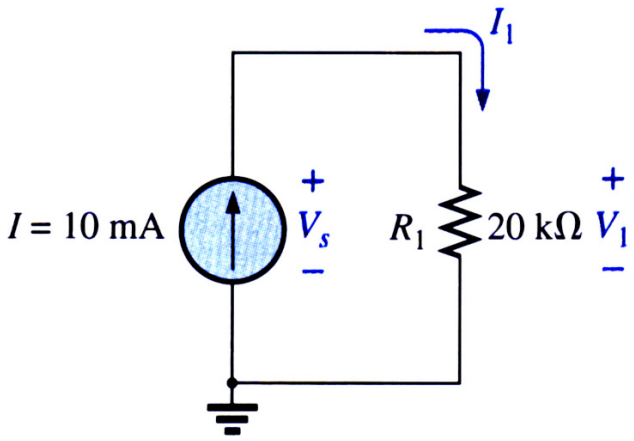
> ⓘ NOTE
>
> **Describing current sources**
>
> In Symbulator, an ideal current source is described with four pieces of information, separated by commas: a unique name (which must start with the letter **j**), the names of its first and second node, and its value in amperes (A). The value is the current flowing through the source from the first node towards the second: how much current leaves by the second node, and equally how much enters by the first. For example, an ideal current source called **j1**, between nodes **0** and **3**, carrying **5** A from the first node to the second, is described: `j1,0,3,5`

You can use SI prefixes here as well. The value of a current source is often given in milliamps; Symbulator will interpret any `'m` given in the value as a division by a thousand.

Answers for a current source

For each current source you get the same answers as for a voltage source, with the same polarity conventions: the voltage drop in it, the current through it, the power consumed by it (for the delivered power, ask for the negative), and the equivalent resistance of the rest of the circuit as seen by that source.

Let's see an example.



B11's Example 8.1

B11's Example 8.1

Given the circuit, determine the current and voltage drop in R_1.

SOLUTION

The circuit:

> Circuit Description
> ```
> j,0,1,10'm
> r1,1,0,20'k
> ```

Two lines, two elements. Leave **Type of analysis** on *Solve circuit*, set **Analysis** to *DC — direct current*, and click **Run Symbulator**. The circuit is numerical, so set **Rounding** to *approx to n digits* with **n** = 3 and tick **Use SI prefixes**, as in Lesson 1 (page 9).

We read:

> current through r1
>
> $$i_{r1} = 10 \text{ mA}$$

> voltage drop in r1
>
> $$v_{r1} = 200 \text{ V}$$

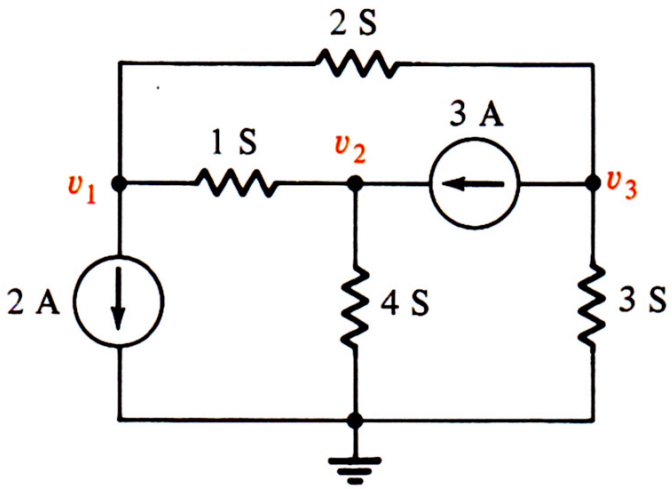
3.2 What about conductances?

Conductances are really resistors by another name. So we describe them as resistors, using the element **r**, and give as the value the inverse of the conductance: one divided by the conductance is the resistor in Ω.

The next example is from *Elementary Linear Circuit Analysis* (2ed) by Leonard S. Bobrow, which from here on I call **Bo2**.

Bo2's Example 2.2

Given the circuit, determine the voltages in the nodes.



Bo2's Example 2.2

SOLUTION

As I explained, all conductances are simulated as resistors: the 4 siemens conductance becomes a 1/4 resistor, and so on. My description:

Circuit Description

```
j10,1,0,2
r12,1,2,1
r20,2,0,1/4
r30,3,0,1/3
r13,1,3,1/2
j32,3,2,3
```

Run it in DC. The answers are under **Node voltages**:

That indicates **v1** = **–1.3** V, **v2** = **.34** V, **v3** = **–1.12** V. This is correct.

3.3 How to reduce parallel resistors

Symbulator has a function for reducing resistors (and impedances) in parallel, **pr()**. You can use it in the circuit description and in the **Evaluate** area.

To reduce three resistors in parallel with values of 10 Ω, 20 Ω and 30 Ω:

Evaluate

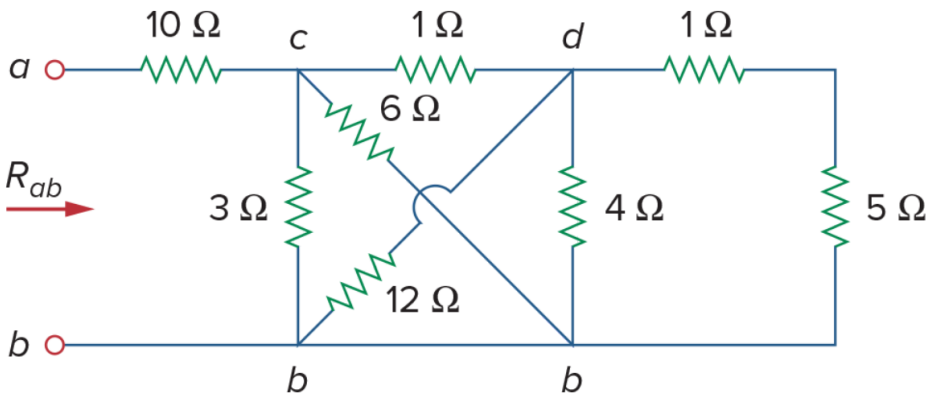
```
pr(10, 20, 30)
```

You get **60/11** Ω with **Rounding** on *exact*, or **5.45** Ω approximately. It nests as deeply as you like, as the two problems below show, and it works on symbols: `pr(r1, r2)` gives

$$\frac{r_1\, r_2}{r_1 + r_2}$$

AS7's Example 2.10

Find the equivalent resistance.



AS7's Example 2.10

SOLUTION

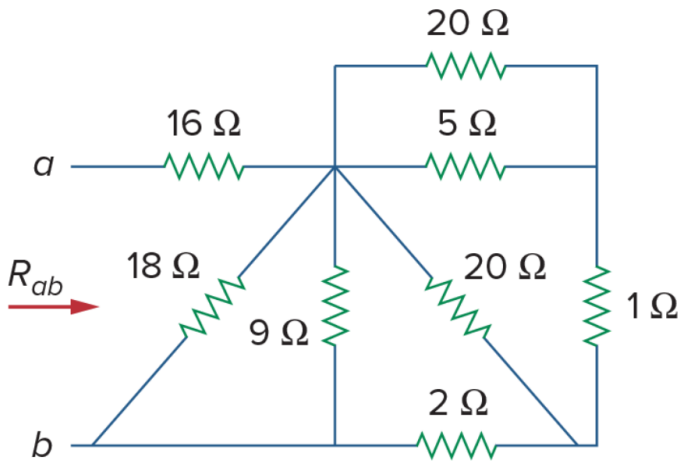
No simulation needed: the whole reduction is one nested expression in **Evaluate**, from the far end of the ladder outward:

```
Evaluate
10 + pr(3, 6, 1 + pr(12, 4, 1 + 5))
```

The answer is **11.2** Ω. This is correct.

AS7's Practice Problem 2.10

Find the equivalent resistance.



AS7's Practice Problem 2.10

SOLUTION

Again, no simulation:

Evaluate

```
16 + pr(18, 9, 2 + pr(20, 1 + pr(5, 20)))
```

The answer is **19** Ω. This is correct.

The [r,r,r...] shorthand

There is a shorthand for **pr**: put the resistor values in square brackets, separated by commas. Expressions like `[10,20,30]` or `[r1,r2,r3,r4]` are read as inputs to **pr**, both in the **Evaluate** area or in the circuit description.

When to reduce resistors

A simulation where it makes sense to use **pr** is **B11's Example 7.4 (page 23)**, which you saw in the practice problems of Lesson 1 (page 9). It makes sense to reduce R_4 and R_5 to an equivalent resistor, since we do not need to know their individual currents or power use.

The `[6,3]` below is the shorthand at work:

Circuit Description

```
e,1,0,16.8
r1,1,2,9
r2,1,2,6
r3,2,3,4
re1,3,0,[6,3]
r6,2,0,3
```

Run it in DC. The current the source delivers is the opposite of `ie`: **3** A.

An example where using **pr** makes no sense is **B11's Example 6.13 (page 20)**, because you need to know the current through each resistor.

Finally, a simulation where you can reduce part of the resistors is **B11's Example 6.22 (page 58)**. We must leave R_1 alone, because we need the current through it, but we can reduce R_2 and R_3:

Circuit Description

```
jt,0,1,12'm
r1,1,0,1'k
re,1,0,[10'k,22'k]
```

Moving forward, we will use `pr(,,)`, or its shorthand `[,,]`, in circuit descriptions whenever we feel it is appropriate.

> ⓘ NOTE
>
> **Reduce only what you don't need to see**
>
> The reason to reduce resistors at all is speed, and the cost is information: once two

resistors become one, you can no longer ask for the current through, or the power consumed by, either of them individually. Reduce only the ones the problem does not ask about.

TIP

One difference worth knowing

Where doing the reciprocal arithmetic by hand would divide by zero, Symbulator's `pr` checks for it: if any of the values you give it is exactly zero, the combination is zero, because a short circuit across a parallel network wins.

3.4 How to describe dependent sources

One of my favourite scenes in cinema comes from *The Dark Knight*: the Joker, played masterfully by Heath Ledger, is rolling on the floor of a Gotham City prison, taking a bare-knuckle beating from an ever-more-frustrated Batman. Master of the situation and laughing hysterically, the Joker says: *"You have nothing! Nothing to threaten me with!"*

Even though the movie had not been made yet, I remember feeling something similar — if less hysterical — back in 1999, when I realised that a 100% symbolic implementation let me make any element's value depend on any answer of the circuit. I could simulate voltage or current sources dependent on any voltage, current or combination of them as easily as a 12 V source.

Here is what you need to know for simulating dependent sources in Symbulator: nothing. There is nothing special to it at all. Just write the value as a function of the circuit's answers, using the answer names you know by now, and run the simulation like it's nobody's business. For example:

- if the source depends on the current through a resistor called **r1**, you define its value as `ir1`
- if the source depends on the voltage drop in a resistor called **r2**, you define its value as `vr2`
- if the source depends on the current through a short called **s3**, you define its value as `is3`
- if the source depends on the difference between the voltage of two nodes a and b, you define its value as `va-vb`

With Symbulator, instead of fearing them, you will laugh in the face of dependent sources, thinking: *"You have nothing!"* Booyah!

TIP

Dependent sources are just sources

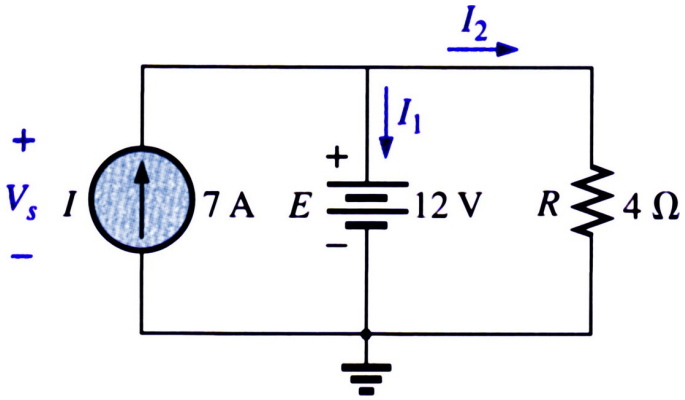
For Symbulator, dependent sources are just sources and need no special notation. State the expression that describes the value, using answer names Symbulator knows and minding their polarity, and you are off to the races.

3.5 Instructive solved examples

Circuits with E, J and R

B11's Example 8.2

Determine the values of V_S, I_1 and I_2.



Circuit Description

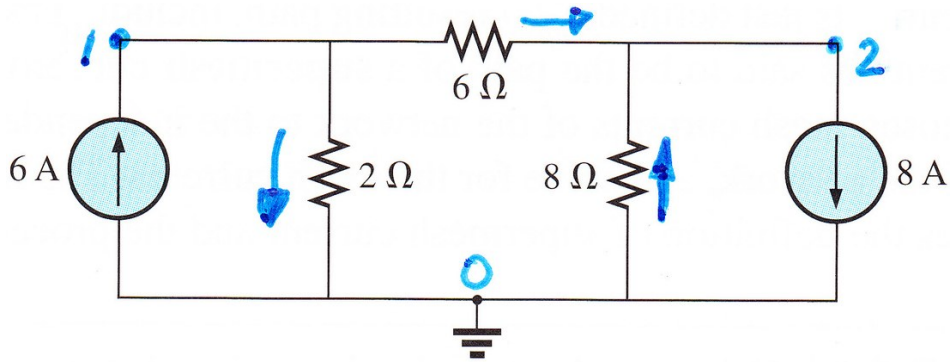
```
j,0,1,7
e,1,0,12
r1,1,0,4
```

The answers you want are `v1`, `ie` and `ir1`, in **Results**.

The answers show V_S is **12**V, I_1 is **4**A and I_2 is **3**A. All correct.

B11's Example 8.15

Determine the current through each resistor. My solution, with the direction marked in blue:



Circuit Description

```
j6,0,1,6
r2,1,0,2
r6,1,2,6
```

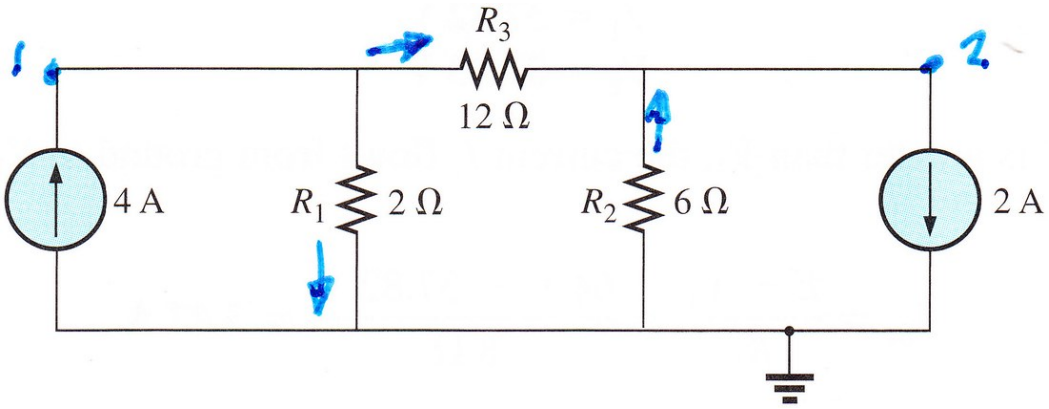
```
r8,0,2,8
j8,2,0,8
```

The answers you want are `ir2`, `ir6` and `ir8`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answer is I_{R2} = **1.25**A, I_{R6} = **4.75**A and I_{R8} = **3.25**A.

B11's Example 8.21



Determine the voltage in each node and the current through each resistor.

My solution:

Circuit Description

```
j1,0,1,4
r1,1,0,2
r3,1,2,12
r2,0,2,6
j2,2,0,2
```

The answers you want are `v1`, `v2`, `ir1`, `ir2` and `ir3`, in **Results**.

The answer is V_1=**6**V, V_2=**-6**V, I_{R1}=**3**A and I_{R2}=I_{R3}=**1**A.

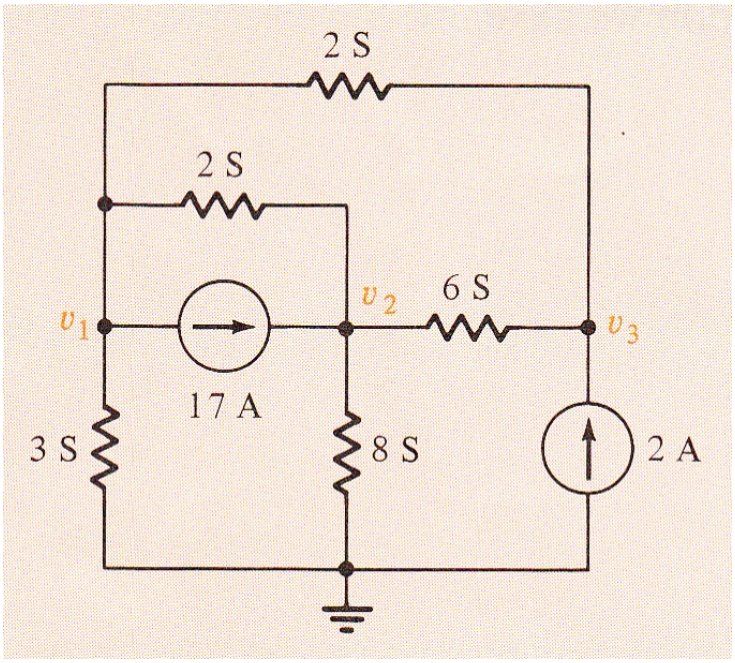
Bo2's Drill Exercise 2.2 (Conductances)

Given the circuit, determine the voltages in the nodes.

Below my solution:

Circuit Description

```
r10,1,0,1/3
r12,1,2,1/2
r13,1,3,1/2
r23,2,3,1/6
r20,2,0,1/8
j12,1,2,17
j03,0,3,2
```



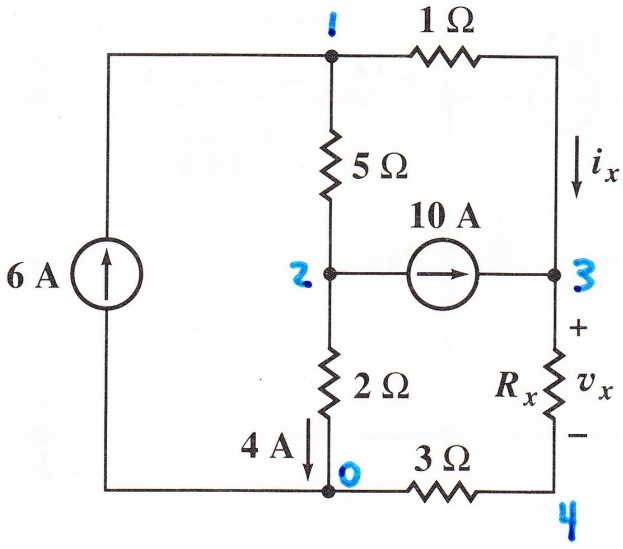
The answers you want are `v1`, `v2` and `v3`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answer is v_1=**-2**V, v_2=**1**V and v_3=**0.5**V. This is correct.

HK5's Figure 1-24b (Expert)

Determine i_x and v_x in the following circuit.



My solution:

Circuit Description

```
j6,0,1,6
r5,1,2,5
r2,2,0,2
r1,1,3,1
```

```
r3,0,4,3
j10,2,3,10
rx,3,4,rx
```

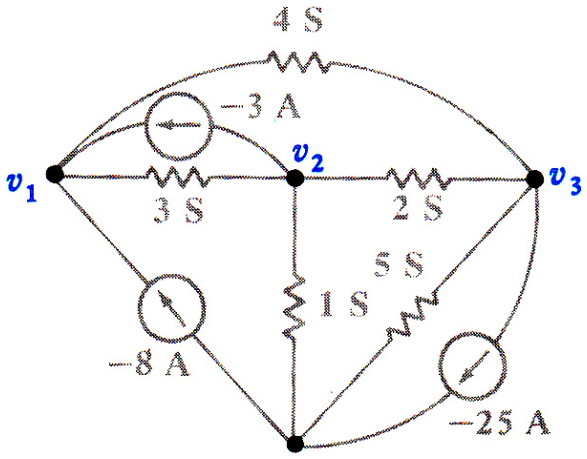
Set **Analysis** to *DC — direct current*. This one needs **Enable Expert Mode** ticked in the **Expert Mode** box; the equations and unknowns go in the fields it reveals.

The answers you want are `ir1` and `vrx`, in **Results**.

In DC, with `ir2=4` in **Add equation(s)** and `rx` in **Add unknown(s)**, run it. The answer is i_x = **-8**A and v_x = **80**V.

HK5's Example 2.2 (Conductances)

Determine the voltages in the nodes. My solution:



Circuit Description

```
j01,0,1,-8
j30,3,0,-25
j21,2,1,-3
r12,1,2,1/3
r23,2,3,1/2
r13,1,3,1/4
r20,2,0,1
r30,3,0,1/5
```

The answers you want are `v1`, `v2` and `v3`, in **Results**.

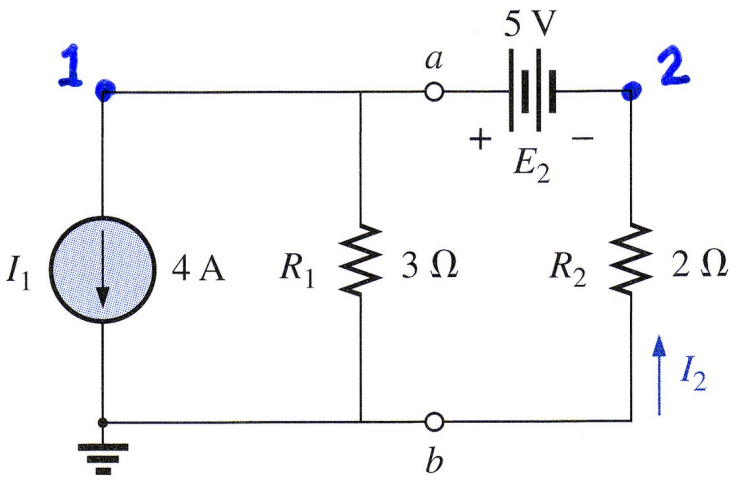
The answer is v_1=**1**V, v_2=**2**V and v_3=**3**V. This is correct.

B11's Example 8.5

Determine the current I_2 in the circuit shown. My solution:

Circuit Description

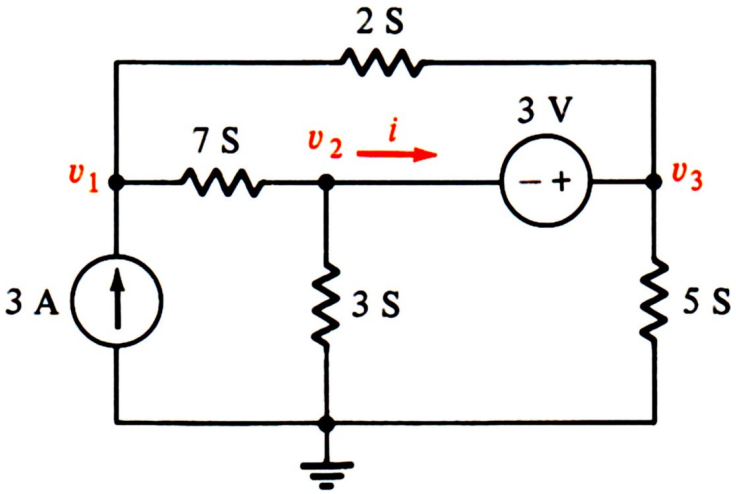
```
j1,1,0,4
r1,1,0,3
e2,1,2,5
```




```
r2,0,2,2
```

This gives us a value for I_2 of **3.4** A. This is correct.

Bo2's Example 2.5 (Conductances)



Determine the voltages in the nodes, and the current through the voltage source.

Circuit Description

```
j,0,1,3
e,3,2,3
r12,1,2,1/7
r20,2,0,1/3
r30,3,0,1/5
r13,1,3,1/2
```

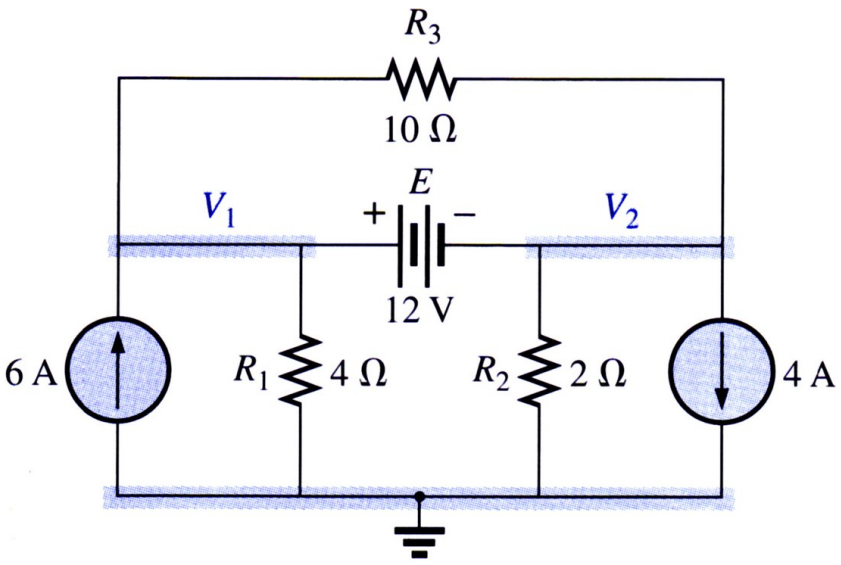
The answers you want are `v1`, `v2` and `v3`, in **Results**; ***i*** is the opposite of `ie`.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answer is v_1=**-.5**, v_2=**-1.5**, v_3=**1.5** and i=**11.5**. This is correct.

B11's Example 8.22

Determine V_1 and V_2. My solution:



Circuit Description

```
j1,0,1,6
r1,1,0,4
e,1,2,12
r3,1,2,10
r2,2,0,2
j2,2,0,4
```

The answers you want are `v1` and `v2`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answer is V_1 = **10.67**V and V_2 = **-1.33**V.

B11's Example 8.19

Determine V_1, I_1 and I_2, in the circuit:

My solution:

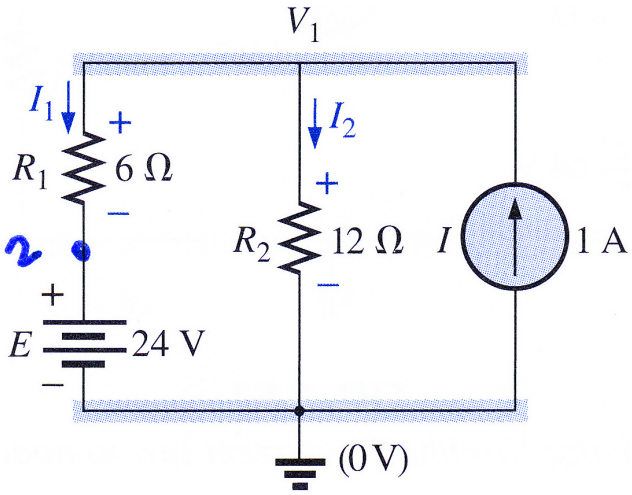
Circuit Description

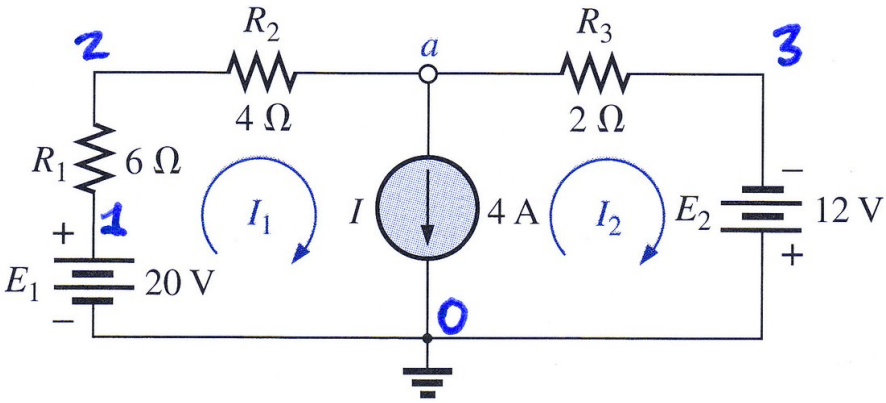
```
e,2,0,24
r1,1,2,6
r2,1,0,12
j,0,1,1
```

The answer is V_1 = **20**V, I_1 = **-.667**A and I_2 = **1.67** A.

B11's Example 8.14

Determine I_2 and I_3.





Circuit Description

```
e1,1,0,20
r1,1,2,6
r2,2,a,4
j,a,0,4
r3,a,3,2
e2,0,3,12
```

The answers you want are `ir2` and `ir3`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answers are:

current through r2

$$i_{r2} = 3.33 \text{ A}$$

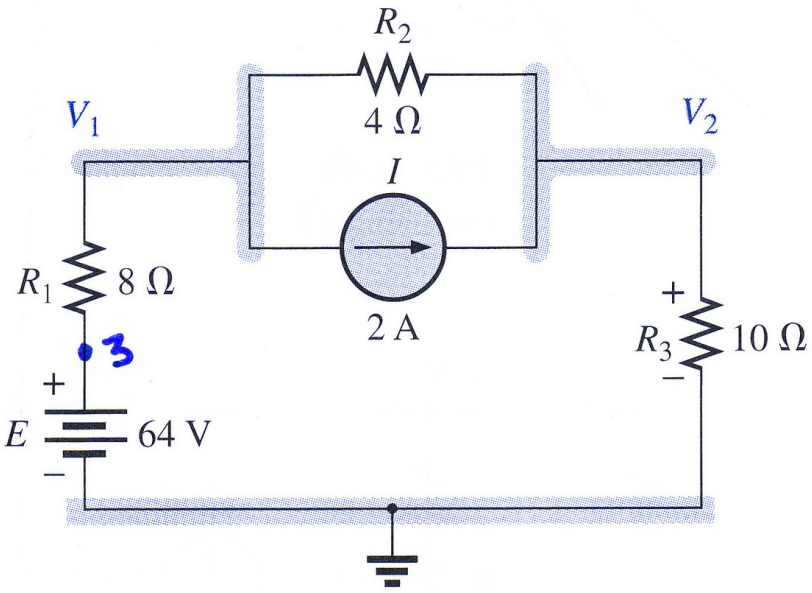
current through r3

$$i_{r3} = -.667 \text{ A}$$

This is correct.

B11's Example 8.20

Determine V_1, V_2, I_1, I_2 and I_3. My solution:



Circuit Description

```
e,3,0,64
r1,3,1,8
r2,1,2,4
j,1,2,2
r3,2,0,10
```

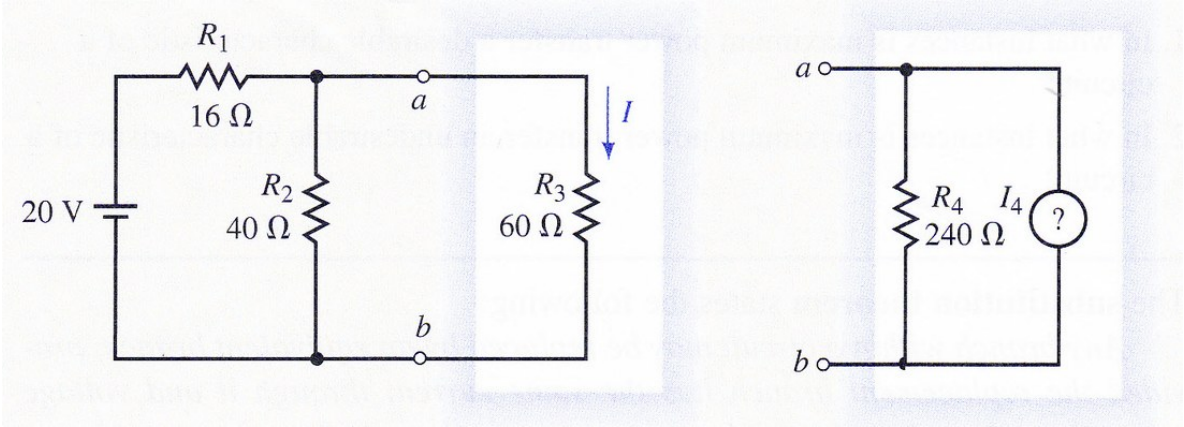
The answers you want are `v1`, `v2`, `ir1`, `ir2` and `ir3`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answer is V_1 = **37.82**V, V_2 = **32.73**V, I_1 = **3.27**A, I_2 = **1.27**A and I_3 = **3.27**A.

RM3's Example 9-12 (*solve*)

If R_3 is to be replaced with R_4 and I_4, determine the value and direction of the source.



So that we can keep the name of node **b**, we will use the top left node as reference.

First, make sure you understand what this problem is asking you to do. The idea is that, despite the change, we keep the same voltage drop and current flow between nodes **a** and **b**. We must first know what they are. So we simulate the original circuit:

Circuit Description
```
e,0,b,20
r1,0,a,16
r2,a,b,40
r3,a,b,60
```

The answer you want is `ir3`, in **Results**.

Ask **Evaluate** for:

Evaluate
```
va-vb
```

We find that the voltage drop is **12**V and the current is **0.2**A. These are the currents and voltages that we have to keep once we do the replacement. Simulate the circuit now replacing R_3 with a resistor R_4 of 240Ω and a source **j** with value I_4. Run this:

Circuit Description
```
e,0,b,20
r1,0,a,16
r2,a,b,40
r4,a,b,240
j,a,b,i4
```

Notice that both the voltage drop (given by va-vb) and the current (given by `ir4+ij`) are algebraic functions in terms of i4. Now you can find i4 solving by voltage drop:

Either equation goes in the **Solve** card:

Equation(s) to solve in terms of the results
```
va-vb = 12
```

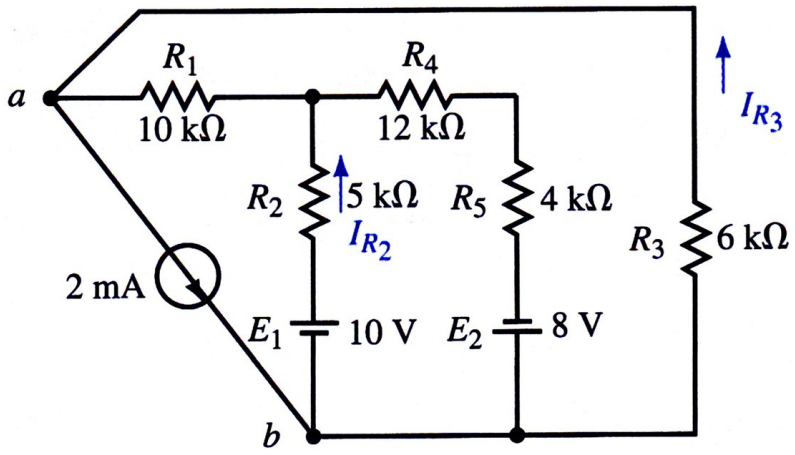
with `i4` as the unknown — or `ir4+ij = 0.2` for the same answer by current instead.

The result is the same: i4 = **.15** A. The required current source is .15A from **a** to **b**.

RM3's Example 8-13

Solve for the currents through R_2 and R_3 in the circuit shown. My solution:

Circuit Description
```
r1,a,0,10'k
r2,1,0,5'k
r3,b,a,6'k
r4,0,2,16'k
j,a,b,2'm
```




```
e1,1,b,10
e2,b,2,8
```

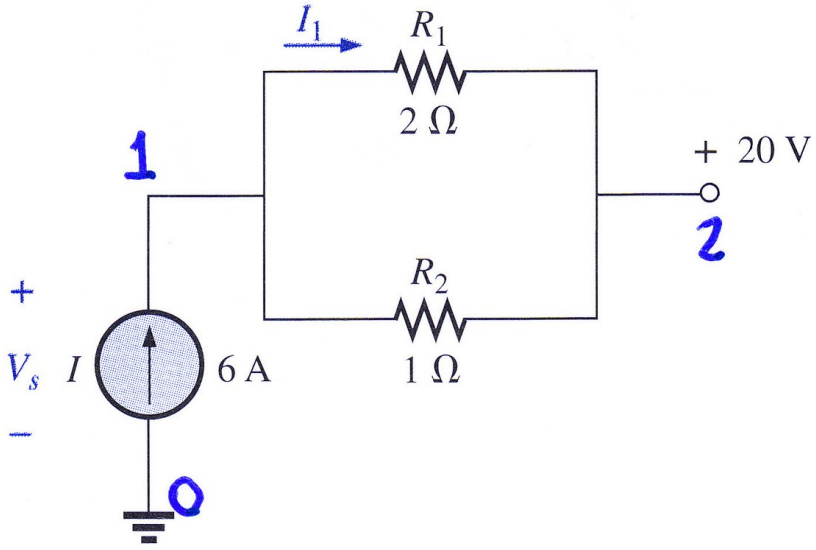
The answers you want are `ir2` and `ir3`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answer is I_{R2} = **1.54** mA and I_{R3} = **1.11** mA. This is correct.

B11's Example 6.3 (Hidden source)

Determine V_S and I_1.



The problem presents a seemingly 'hanging' node with 20V. To simulate this, imagine a 'hidden' 20V voltage source connected to the node.

Circuit Description

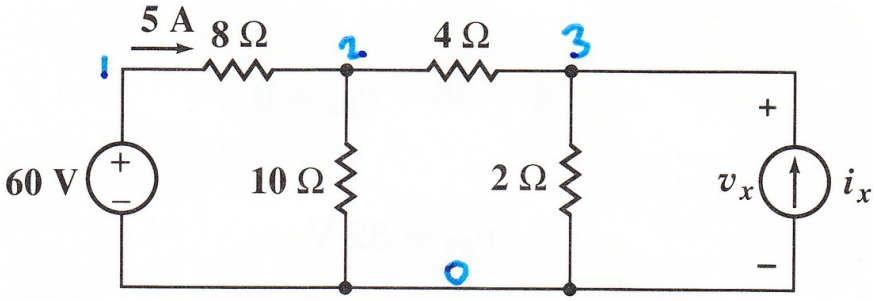
```
j,0,1,6
r1,1,2,2
r2,1,2,1
e,2,0,20
```

The answers you want are `v1` and `ir1`, in **Results**.

The answer is V_S = **24**V and I_1 = **2**A. This is correct.

HK5's Figure 1-24c (Expert)

Determine i_x and v_x in the following circuit.



Circuit Description

```
e,1,0,60
r8,1,2,8
r10,2,0,10
r4,2,3,4
r2,3,0,2
j,0,3,ix
```

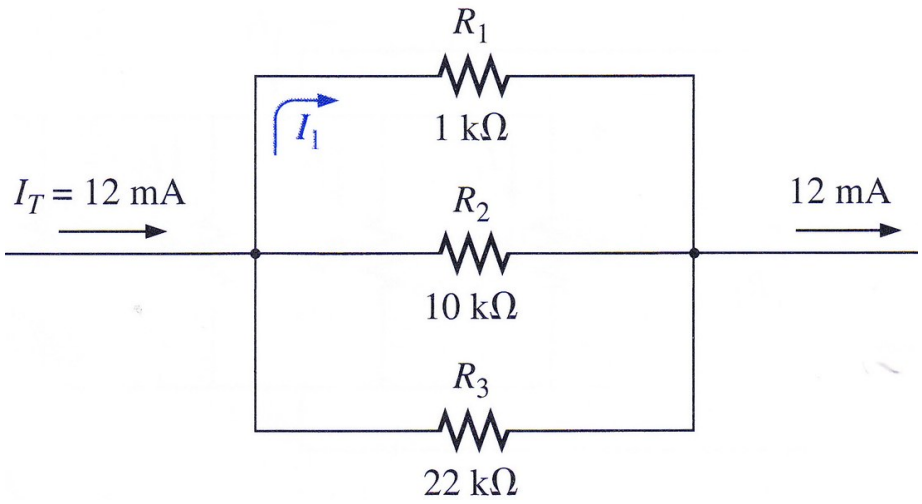
Set **Analysis** to *DC — direct current*. This one needs **Enable Expert Mode** ticked in the **Expert Mode** box; the equations and unknowns go in the fields it reveals.

The answers you want are `ix` and `v3`, in **Results**.

In DC, with `ir8=5` in **Add equation(s)** and `ix` in **Add unknown(s)**, run it. The answer is i_x = **1**A and v_x = **8**V. This is correct.

B11's Example 6.22 (Hidden source)

Determine I_1.



Although no source is shown, the circuit has a current. I use a 'hidden' current source.

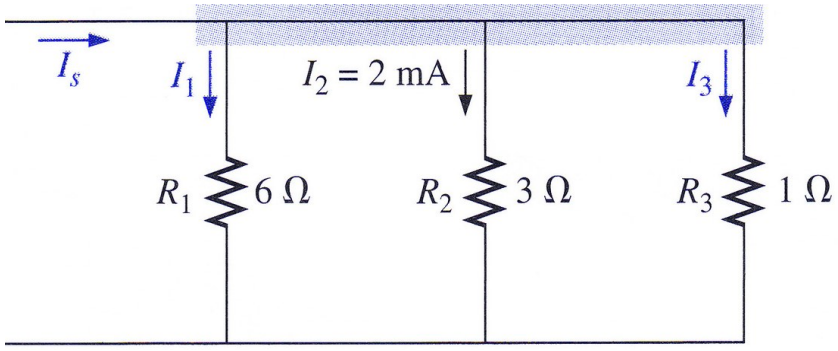
Circuit Description

```
jt,0,1,12'm
r1,1,0,1'k
r2,1,0,10'k
r3,1,0,22'k
```

`ir1` gives I_1 = **10.48** mA, which is correct.

B11's Example 6.21 (Hidden source, Expert)

Determine I_S, I_1 and I_3.



This 'hidden source' problem is perfect for **Expert Mode**. My solution:

Circuit Description

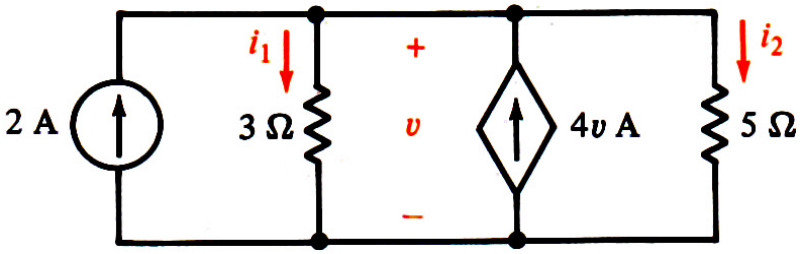
```
js,0,1,is
r1,1,0,6
r2,1,0,3
r3,1,0,1
```

Choose DC. Open **Expert Mode**, put `ir2 = 2'm` in **Add equation(s)** and `is` in **Add unknown(s)**, then **Run Symbulator**.

The answer is I_S = **9**mA, I_1 = **1**mA and I_3 = **6**mA. This is correct.

Bo2's Example 1.9

Determine I_1, I_2 and v.



This is my solution.

Circuit Description

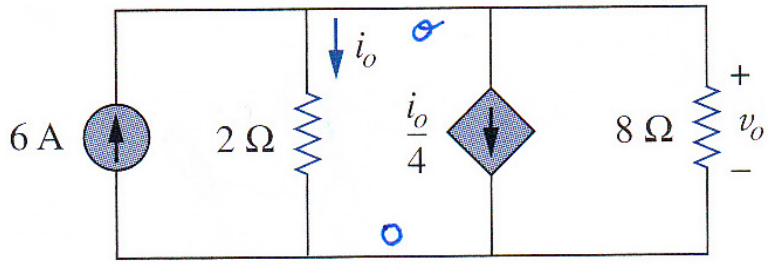
```
ji,0,1,2
r1,1,0,3
jd,0,1,4*v1
r2,1,0,5
```

The answers you want are `ir1`, `v1` and `ir2`, in **Results**.

The answer is I_1=**-5/26** A, I_2=**-3/26** A and v=**-15/26** V. This is correct.

AS2's Practice Problem 2.7

Find v_o and i_o in the circuit. My solution:



Circuit Description

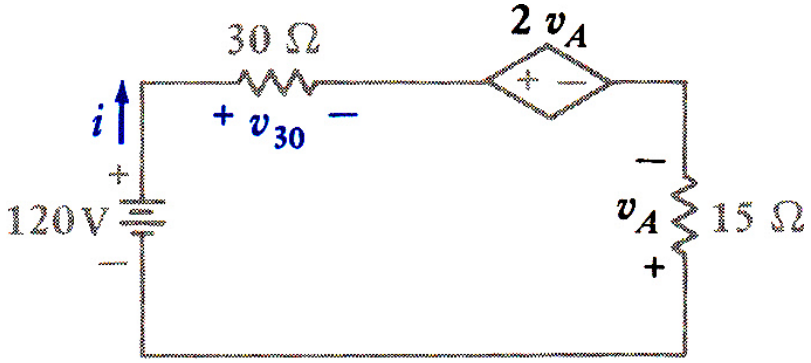
```
ji,0,o,6
ro,o,0,2
jd,o,0,iro/4
r8,o,0,8
```

The answers you want are `vo` and `iro`, in **Results**.

The answer is v_o = **8** V and i_o = **4** A. This is correct.

HK5's Example 1-3

Determine the power delivered by each source and consumed by both resistors.



Circuit Description

```
ei,1,0,120
r1,1,2,30
ed,2,3,2*vra
```

```
ra,0,3,15
```

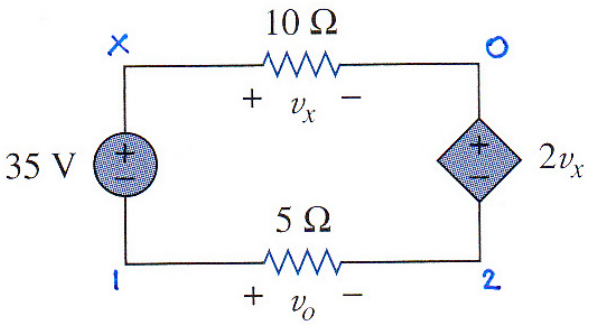
The power each source delivers is the opposite of `pei` and of `ped`. For the resistors together, ask **Evaluate** for:

Evaluate
```
pr1+pra
```

The independent source delivers **960**W, the dependent source delivers **1920**W, and the resistors consume **2880**W together. This is correct.

AS2's Practice Problem 2.6

Find v_x and v_o in the circuit. My solution:

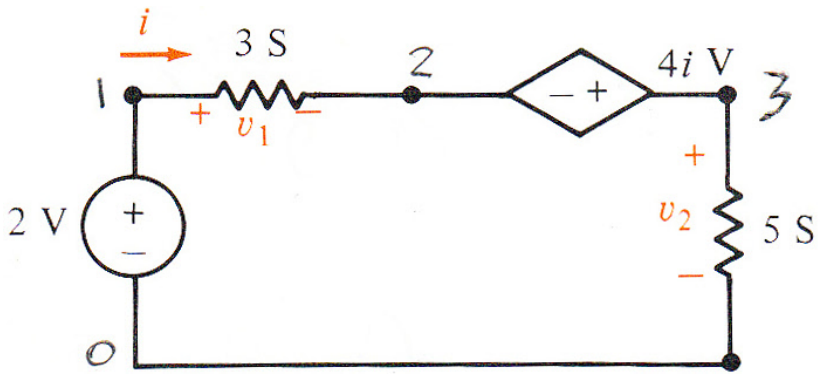


Circuit Description
```
ei,x,1,35
rx,x,0,10
ed,0,2,2*vx
ro,1,2,5
```

The answers you want are `vx` and `vro`, in **Results**.

The answer is v_x =**10** and v_o =**-5**. This is correct.

Bo2's Example 1.10



Determine v_1, v_2 and i. My solution:

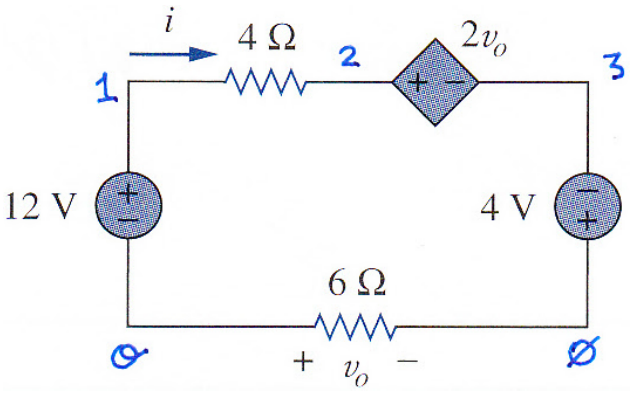
Circuit Description

```
ei,1,0,2
r1,1,2,1/3
ed,3,2,4*ir1
r2,3,0,1/5
```

The answers you want are `ir1`, `vr1` and `vr2`, in **Results**.

The answer is v_1=**-5/26** V, v_2=**-3/26** V and i=**-15/26** A. This is correct.

AS2's Example 2.6



Determine v_o and i in the circuit. My solution is shown below:

Circuit Description

```
e12,1,o,12
ri,1,2,4
ed,2,3,2*vo
e4,0,3,4
ro,o,0,6
```

The answers you want are `vo` and `iri`, in **Results**.

The answer is v_o =**48** and i = **-8**. This is correct.

HK5's Drill Problem 1.11

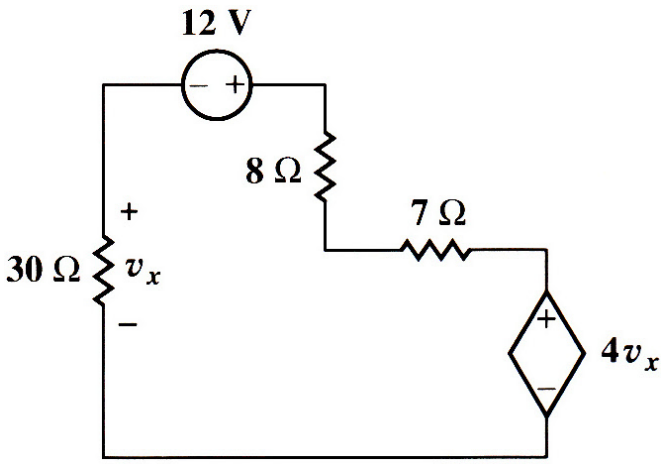
Find the power absorbed by each element in the circuit. My solution:

Circuit Description

```
r1,x,0,30
ei,1,x,12
r2,1,2,8
r3,2,3,7
ed,3,0,4*vx
```

The answers you want are `pr1`, `pr2` and `pr3` in **Results**, and for the two sources the *power delivered* row of each card, `-pei` and `-ped`.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.



The answers are:

power consumed by r1

$$p_{r1} = .768 \text{ W}$$

power delivered by ei

$$-p_{ei} = -1.92 \text{ W}$$

power consumed by r2

$$p_{r2} = .2048 \text{ W}$$

power consumed by r3

$$p_{r3} = .1792 \text{ W}$$

power delivered by ed

$$-p_{ed} = 3.072 \text{ W}$$

This is correct: the source **ei** delivers a negative power, which is to say it absorbs 1.92 W, and **ed** delivers 3.072 W.

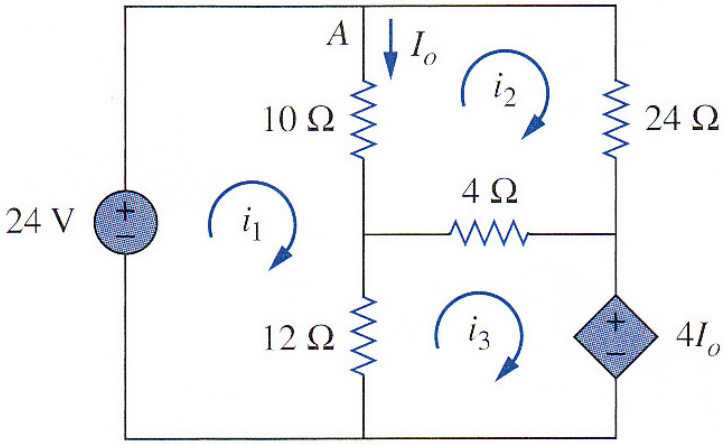
AS2's Example 3.6

Determine the value of I_o in the circuit. My solution:

Circuit Description

```
ei,a,0,24
ro,a,b,10
r12,b,0,12
r4,b,c,4
r24,a,c,24
ed,c,0,4*iro
```

The answer you want is `iro`, in **Results**.

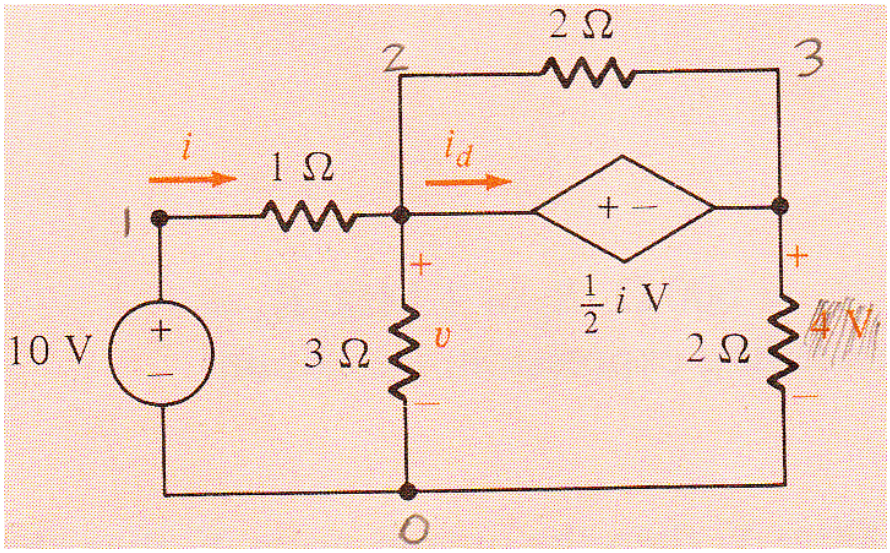


Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answer, **1.5** A, is correct.

Bo2's Drill Exercise 1.12

Determine ***i***, ***v*** and i_d.



We are given an unnecessary piece of information: the 4V drop in the 2Ω resistor.

My solution:

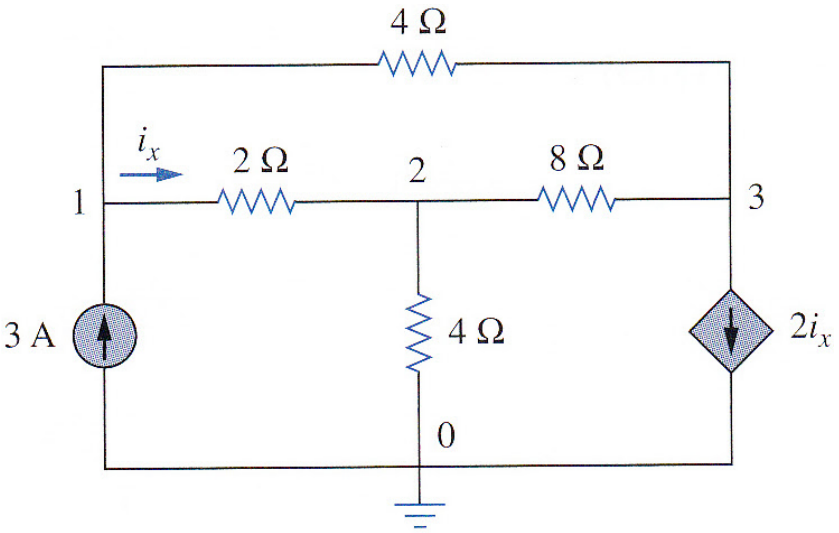
> Circuit Description
> ```
> ei,1,0,10
> r1,1,2,1
> r2,2,3,2
> r3,2,0,3
> r4,3,0,2
> ed,2,3,ir1/2
> ```

The answers you want are `ir1`, `vr3` and `ied`, in **Results**.

The answer is ***i***=**4**, ***v***=**6** and i_d=**1**. This is correct.

AS2's Example 3.2

Determine the voltages at the nodes.



My solution:

Circuit Description

```
ji,0,1,3
jd,3,0,2*ir2
r2,1,2,2
r4a,1,3,4
r8,2,3,8
r4b,2,0,4
```

The answers you want are `v1`, `v2` and `v3`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answers are:

voltage of node 1

$$v_1 = 4.8\ \text{V}$$

voltage of node 2

$$v_2 = 2.4\ \text{V}$$

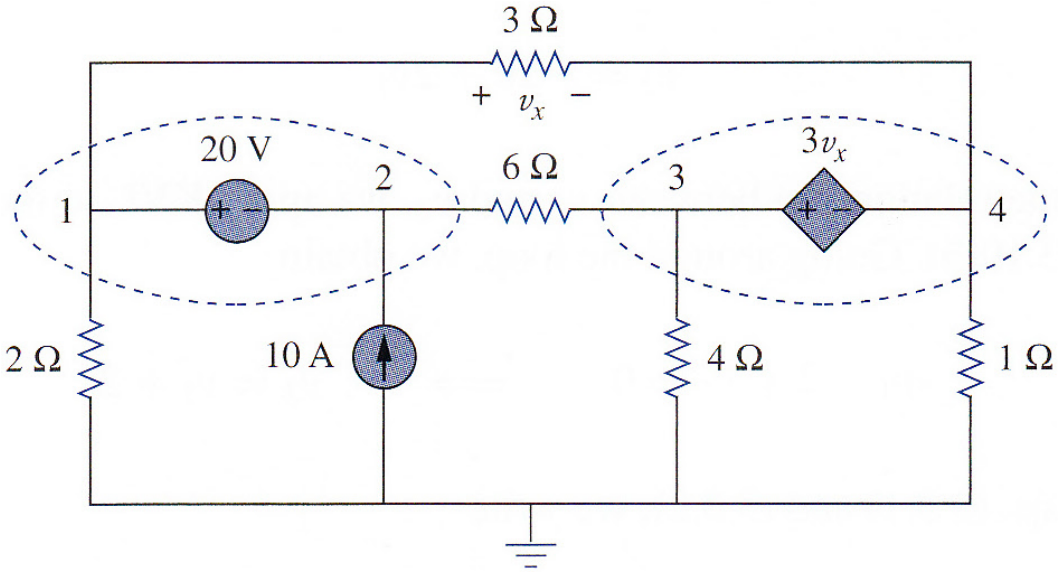
voltage of node 3

$$v_3 = -2.4\ \text{V}$$

This is correct.

AS2's Example 3.4

Find the node voltages in the circuit. My solution:



Circuit Description

```
r2,1,0,2
e1,1,2,20
j,0,2,10
r6,2,3,6
rx,1,4,3
r4,3,0,4
ed,3,4,3*vrx
r1,4,0,1
```

The answers you want are `v1`, `v2`, `v3` and `v4`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answers are:

voltage of node 1

$$v_1 = 26.67 \text{ V}$$

voltage of node 2

$$v_2 = 6.67 \text{ V}$$

voltage of node 3

$$v_3 = 173.33 \text{ V}$$

voltage of node 4

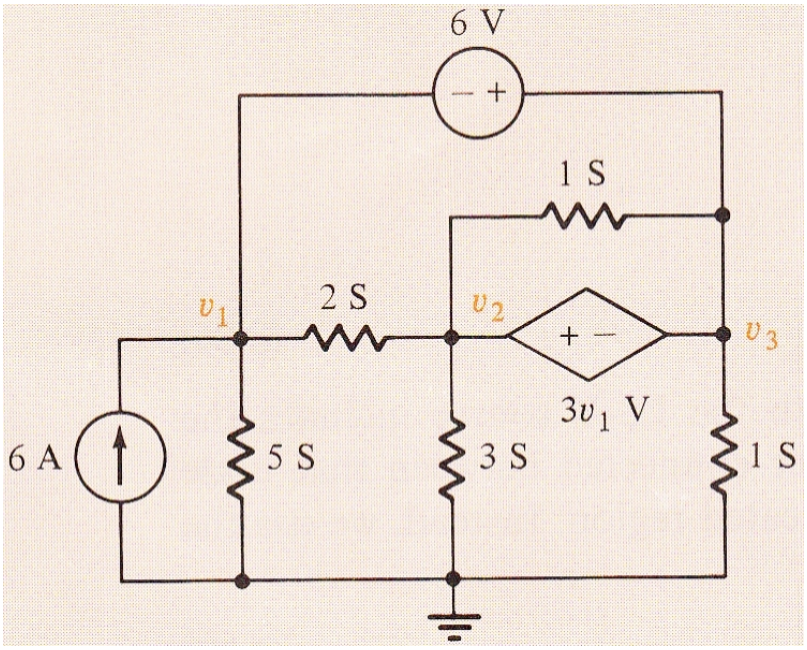
$$v_4 = -46.67 \text{ V}$$

This is correct.

Bo2's Drill Exercise 2.6

Determine the voltage in each node.

Notice that in the schematic, the resistors' values are given in siemens. Symbulator has to be fed the resistors with values in ohms. This means that, when we simulate them, we have to convert them from siemens to ohms by dividing 1 over the siemens value.



My solution:

Circuit Description

```
j,0,1,6
ei,3,1,6
ed,2,3,3*v1
r5,1,0,1/5
r2,1,2,1/2
r3,2,0,1/3
r1,2,3,1
r4,3,0,1
```

The answers you want are in **Results**:

voltage of node 1

$$v_1 = -1\ \text{V}$$

voltage of node 2

$$v_2 = 2\ \text{V}$$

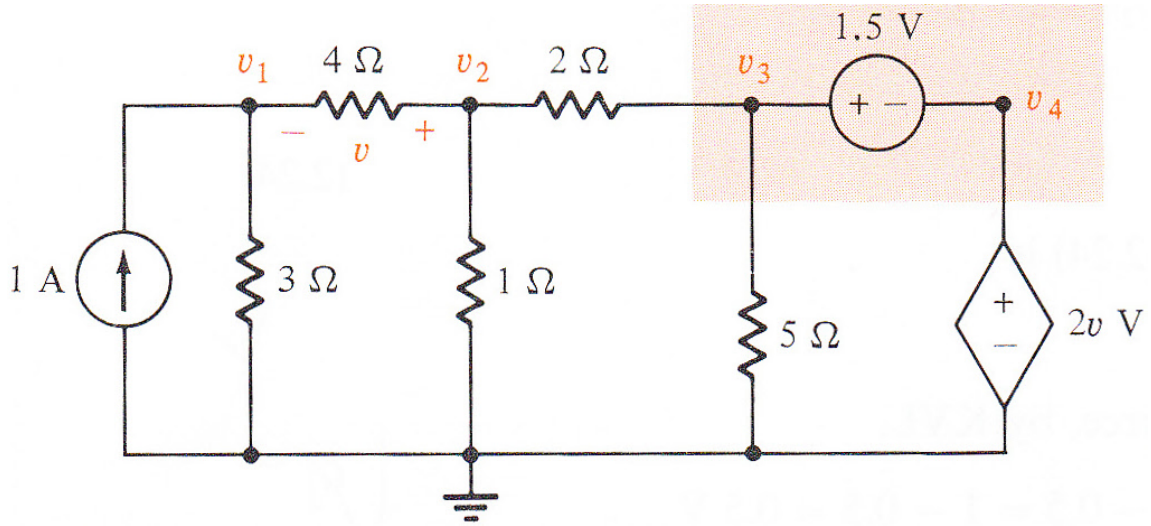
voltage of node 3

$$v_3 = 5\ \text{V}$$

This is correct.

Bo2's Example 2.7

Determine the voltages in all nodes. My solution:



Circuit Description

```
j,0,1,1
r3,0,1,3
r4,2,1,4
r1,2,0,1
r2,2,3,2
r5,3,0,5
ei,3,4,1.5
ed,4,0,2*vr4
```

The answers you want are in **Results**:

voltage of node 1

$$v_1 = 1.5\ \text{V}$$

voltage of node 2

$$v_2 = -.5\ \text{V}$$

voltage of node 3

$$v_3 = -2.5\ \text{V}$$

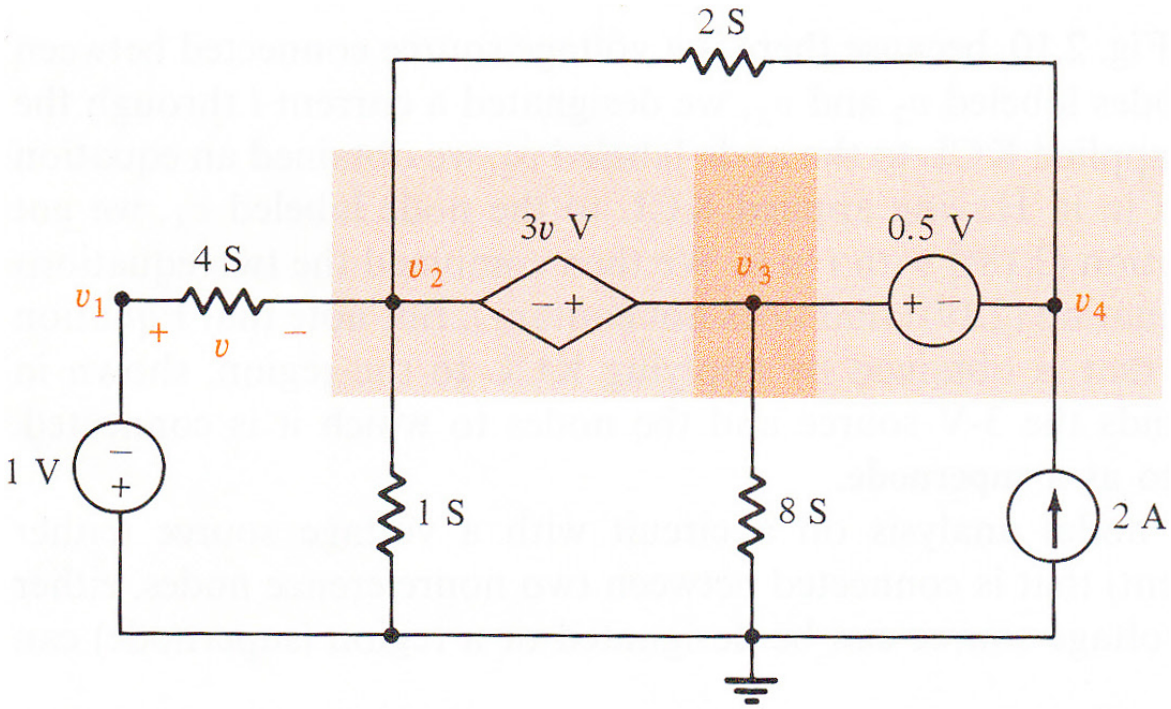
voltage of node 4

$$v_4 = -4\ \text{V}$$

This is correct.

Bo2's Example 2.6

Determine the voltages in all nodes.



My solution:

Circuit Description

```
e1,0,1,1
e2,3,4,.5
ed,3,2,3*vr4
j,0,4,2
r4,1,2,1/4
r1,2,0,1
r8,3,0,1/8
r2,2,4,1/2
```

The answers you want are in **Results**:

voltage of node 1

$$v_1 = -1\ \text{V}$$

voltage of node 2

$$v_2 = -2\ \text{V}$$

voltage of node 3

$$v_3 = 1\ \text{V}$$

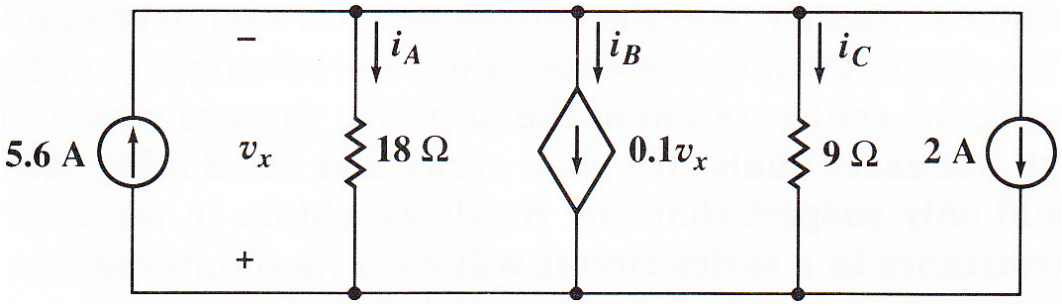
voltage of node 4

$$v_4 = .5\ \text{V}$$

This is correct.

HK5's Drill Problem 1-12

Find i_A, i_B and i_C.



My solution:

Circuit Description

```
jl,x,0,5.6
ra,0,x,18
jb,0,x,.1*vx
r9,0,x,9
jr,0,x,2
```

The answers you want are `ira`, `ijb` and `ir9`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answers are:

current through ra

$$i_{ra} = 3\ \text{A}$$

current through jb

$$i_{jb} = -5.4\ \text{A}$$

current through r9

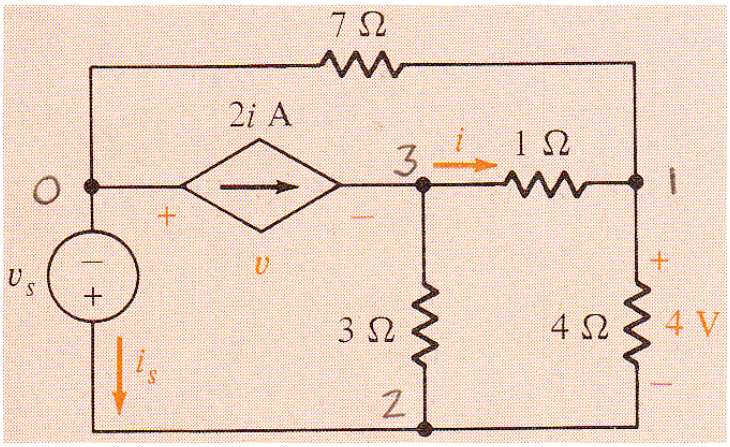
$$i_{r9} = 6\ \text{A}$$

This is correct.

Numerical-from-symbolic examples

Bo2's Drill Exercise 1.10

Determine i, v, i_s and v_s.



With one unknown value and one known solution, this problem is a job for **Expert Mode**.

Determine i, v, i_s and v_s.

Circuit Description

```
es,2,0,vs
jd,0,3,2*ir1
r7,0,1,7
r1,3,1,1
r3,3,2,3
r4,1,2,4
```

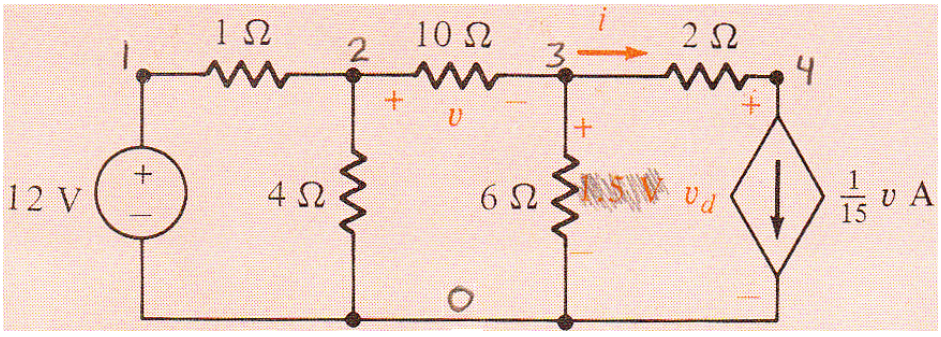
Set **Analysis** to *DC — direct current*. This one needs **Enable Expert Mode** ticked in the **Expert Mode** box; the equations and unknowns go in the fields it reveals.

The answers you want are `ir1`, `vjd` and `vs`, in **Results**; i_s is the opposite of `ies`.

In DC, add `vr4=4` to the equations and `vs` to the unknowns, and run it. The answer is i=**2**, v=**-9**, i_s=**-3** and v_s=**3**. This is correct.

Bo2's Drill Exercise 1.11

Determine i, v and v_d. (Since all element values are known, the tip we are given by the book – namely, that the voltage drop in the 6Ω resistor is 1.5V – is totally superfluous.)



Circuit Description

```
e,1,0,12
r1,1,2,1
r4,2,0,4
r10,2,3,10
r6,3,0,6
r2,3,4,2
j,4,0,vr10/15
```

The answers you want are `vr10`, `ir2` and `vj`, in **Results**.

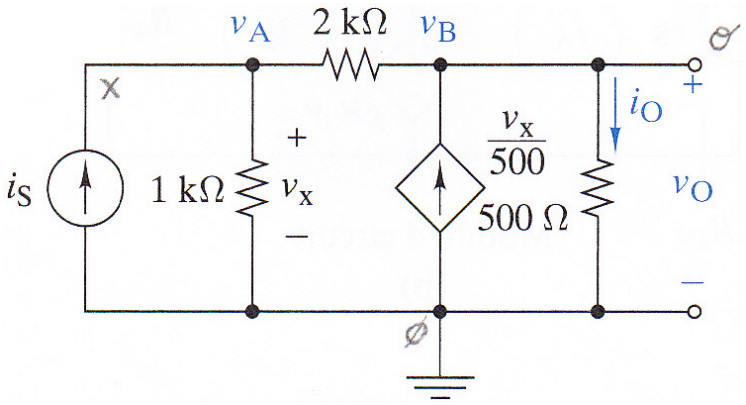
Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answer is ***i***=**.5**, ***v***=**7.5** and v_d=**.5**. This is correct.

Symbolic examples

TR5's Exercise 4.2

Find v_O and i_O in terms of i_S.



This is my solution.

Circuit Description

```
ji,0,x,is
r1,x,0,1'k
r2,x,o,2'k
jd,0,o,vx/500
ro,o,0,500
```

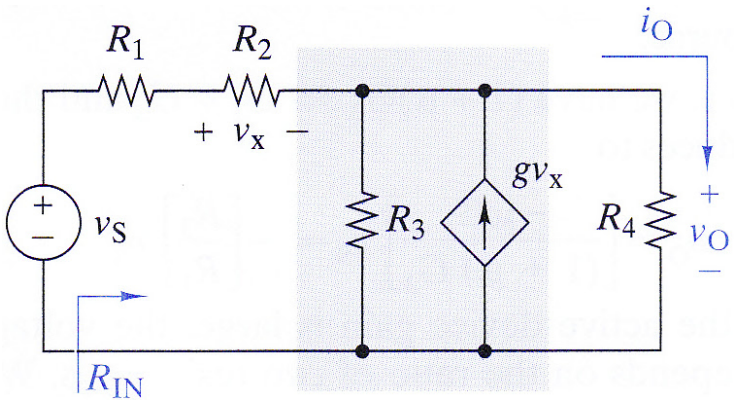
The answers you want are `vo` and `iro`, in **Results**.

The answer is v_O=*1000* i_S and i_O=*2* i_S. This is correct.

TR5's Example 4.4

Find v_O and the equivalent resistance R_{IN}, in terms of v_S, when R_1 is 50 Ω, R_2 is 1 kΩ, R_3 is 100 Ω, R_4 is 5 kΩ and g is 100 mS (i.e. 100'm).

This is my solution.



Circuit Description

```
e,1,0,vs
r1,1,2,50
r2,2,o,1'k
r3,o,0,100
r4,o,0,5'k
j,0,o,100'm*vr2
```

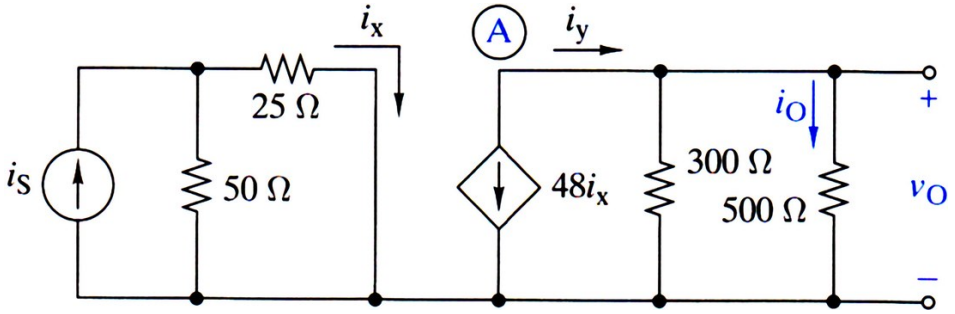
The answers you want are `vo` and `re`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

The answer is v_O=*.904* v_S and R_{IN} = *10.95 kΩ*. This is correct.

TR5's Figure 4-4

Find the voltage drop, current, and power consumed by the 500Ω resistor, and the ratio of that power to that delivered by the independent source, all in terms of i_S.



I have not labeled the nodes in the figure, so you can practice doing it. My solution:

Circuit Description

```
js,0,1,is
r50,1,0,50
rx,1,0,25
jd,o,0,48*irx
r3,o,0,300
ro,o,0,500
```

The answers you want are `iro`, `vo` and `pro`, in **Results**.

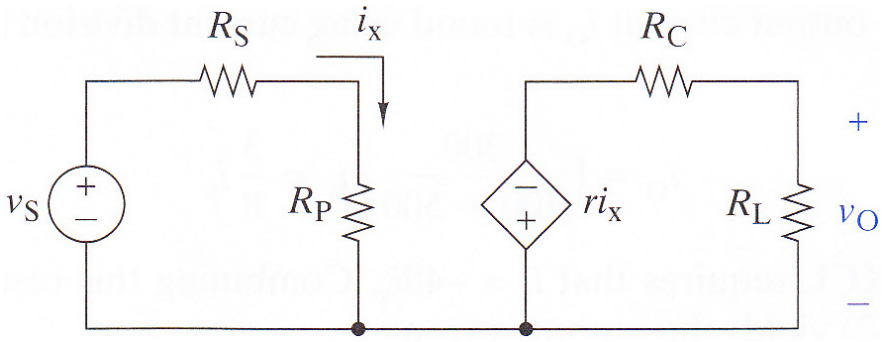
Ask **Evaluate** for:

> Evaluate
> `pro/(-pjs)`

The answers we get are correct: i_O=**-12is**, v_O=**-6000is**, p_O=**72000is**^{2}, and p_O/p_S=**4320**.

TR5's Example 4.1 (Symbolic)

Find v_O.



I did not label the nodes, so you can practice.

> Circuit Description
> ```
> ei,1,0,vs
> rs,1,2,rs
> rx,2,0,rp
> ed,0,3,r*irx
> rc,3,o,rc
> rl,o,0,rl
> ```

The answer you want is `vo`, in **Results**.

The answer, shown first below, is correct. The textbook's answer follows it.

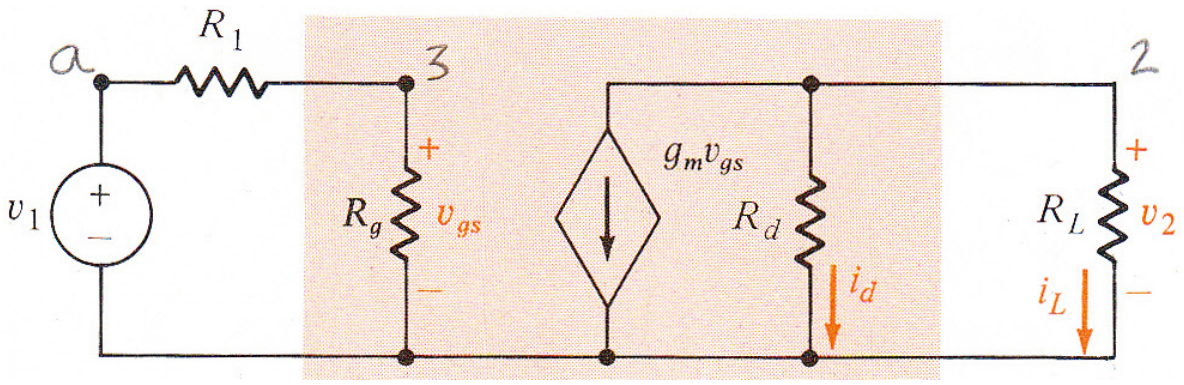
> voltage of node o
>
> $$v_o = -\frac{r\,rl\,vs}{rc\,rp + rc\,rs + rl\,rp + rl\,rs}$$

$$v_O = \left[\frac{-R_L r}{(R_S + R_P)(R_C + R_L)}\right] v_S$$

Bo2's Example 1.11 (Symbolic)

Determine v_2.

My solution. I named the source's value v_1, as in the book, so no node may be called **1**: its voltage would be `v1` too. Using r1 as a value is fine.



Circuit Description

```
e,a,0,v1
r1,a,3,r1
rg,3,0,rg
j,2,0,gm*vrg
rd,2,0,rd
rl,2,0,rl
```

The answer you want is `v2`, in **Results**.

The answer I got is shown first below (the textbook's follows it.)

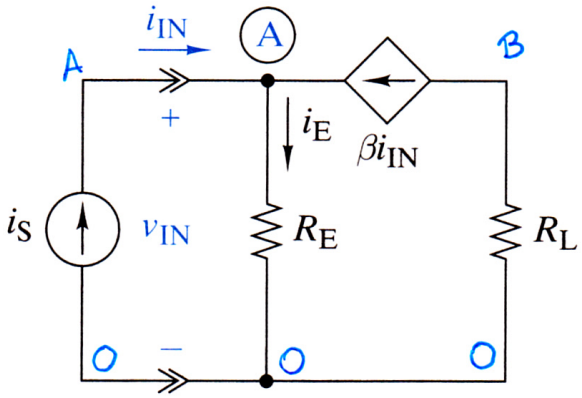
voltage of node 2

$$v_2 = -\frac{gm\,rd\,rg\,rl\,v_1}{r_1\,rd + r_1\,rl + rd\,rg + rg\,rl}$$

$$v_2 = \frac{-g_m R_d R_g R_L v_1}{(R_g + R_1)(R_L + R_d)}$$

TR5's Example 4-7 (Symbolic)

Find R_{IN}, i.e. the resistance as seen by the current source. My solution:



Circuit Description

```
ji,0,a,is
re1,a,0,re1
jd,b,a,β*is
rl,b,0,rl
```

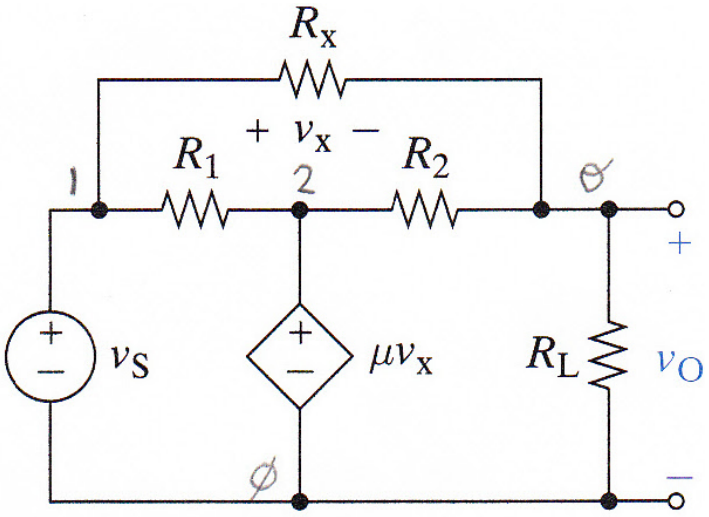
The answer you want is `rji`, in **Results**.

The answer we get — `re1*(β+1)` — is correct, as can be seen by comparing it to the textbook's answer.

$$R_{IN} = (\beta + 1)R_E$$

TR5's Exercise 4.3 (Symbolic)

Find v_o, in terms of the value in the circuit. For resistors, use their conductance value.



This is my solution. We use μ as a constant in the dependent source. It is not confused with the SI prefix for micro because the prefix has an apostrophe.

Circuit Description

```
ei,1,0,vs
ed,2,0,μ*(vrx)
r1,1,2,1/g1
r2,2,o,1/g2
rx,1,o,1/gx
rl,o,0,1/gl
```

The answer you want is `vo`, in **Results**.

The answer is correct; the textbook's follows it.

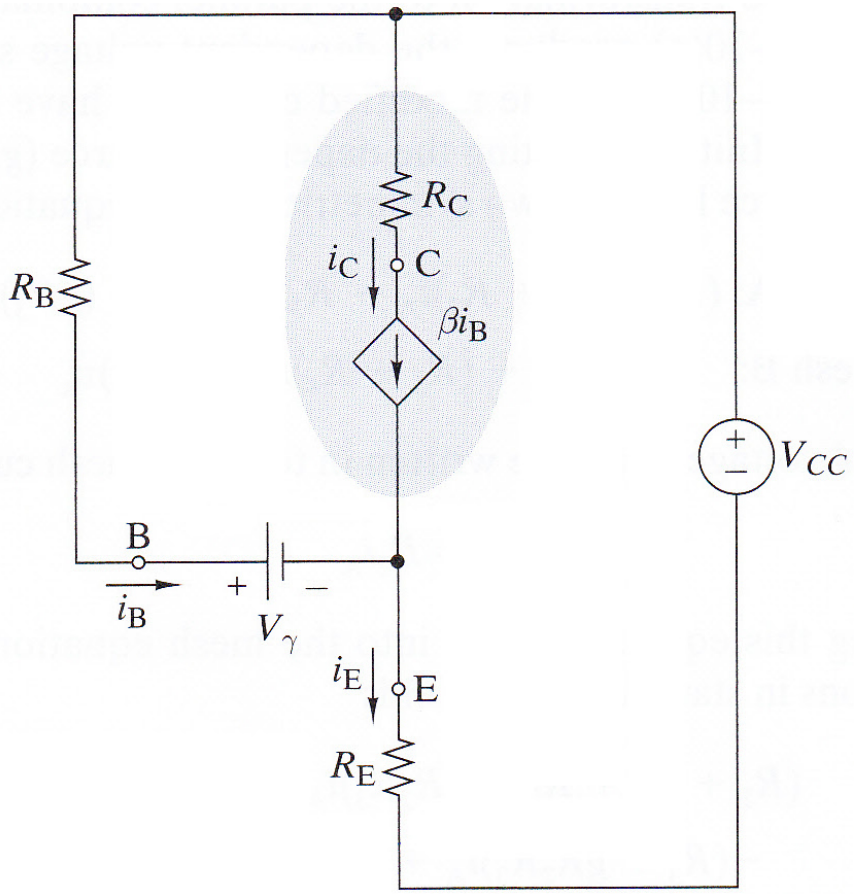
voltage of node o

$$v_o = \frac{vs\,(g_2\,\mu + gx)}{g_2\,\mu + g_2 + gl + gx}$$

$$v_O = \frac{G_x + \mu G_2}{G_x + G_L + (\mu + 1)G_2} v_S$$

TR5's Example 4.5 (Symbolic)

Find i_B.



My solution is shown below.

Circuit Description

```
e1,1,0,vcc
rb,1,b,rb
e2,e,b,vγ
re1,e,0,re1
rc,1,c,rc
j,c,e,β*irb
```

The answer you want is `irb`, in **Results**.

Compare my answer, first below, to the book's answer beneath it.

current through rb

$$i_{rb} = \frac{vcc + v\gamma}{r_{e1}\,\beta + r_{e1} + rb}$$

$$i_B = \frac{V_{CC} - V_\gamma}{R_B + (\beta + 1)R_E}$$

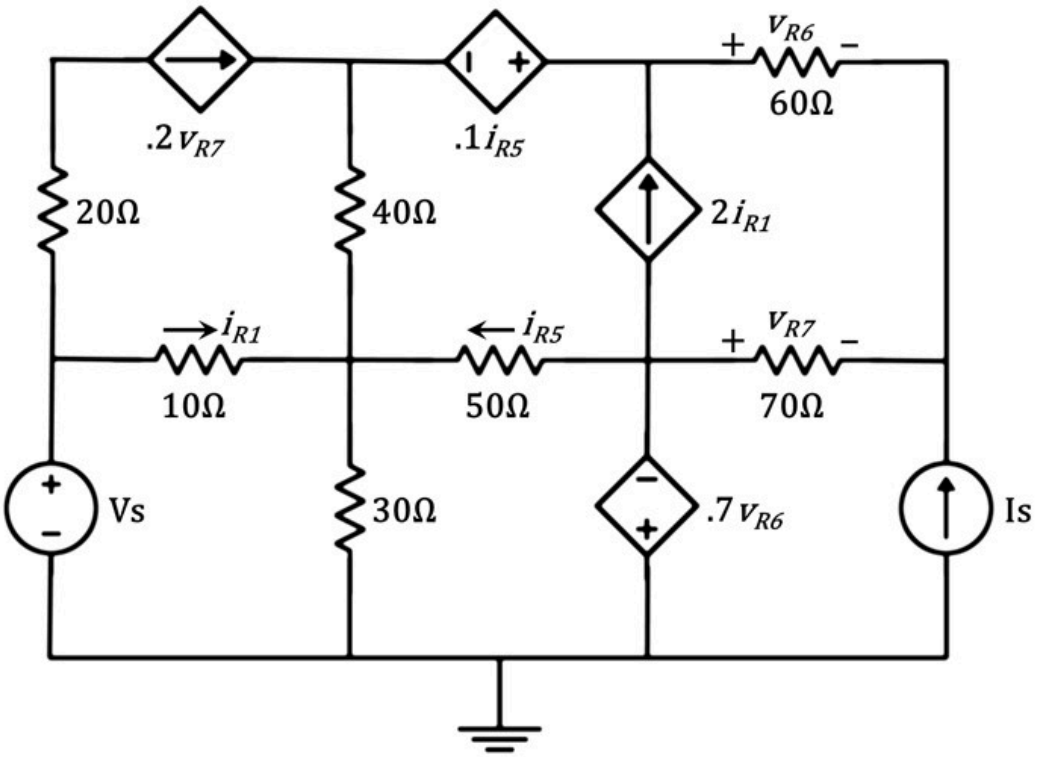
The answer given by the book uses variables with subscripts, which look very elegant. Symbolic answers can be made to read that way too, and the trick is one character: an underscore in a symbolic variable name puts what follows into a subscript. Underscores, and subscripts in your answers (page 298) shows it on this very circuit — optional reading, and it changes no answer, only its look.

The last four problems show Symbulator at its DC best.

The Showing-off Problem (Expert)

In 2013, I invented a problem to showcase how easy it is to describe dependent sources in Symbulator.

Question: Find positive values for V_s and I_s that will result in 80W delivered by the VCCS and 0W dissipated in the CCVS.



Use this circuit description and equations:

Circuit Description

```
es,e,0,vs
js,0,d,is
r1,e,m,10
r2,a,e,20
```

```
r3,m,0,30
r4,b,m,40
r5,n,m,50
r6,c,d,60
r7,n,d,70
jd1,a,b,0.2*v_r7
ed2,c,b,0.1*i_r5
jd3,n,c,2*i_r1
ed4,0,n,0.7*v_r6
```

Symbulator reports the power *consumed* by each element, so delivering 80W is consuming −80W. In **Expert Mode**:

```
Add equation(s)
pjd1 = -80 and ped2 = 0
```

```
Add unknown(s)
vs, is
```

The problem asks for *positive* values, which is what the conditions are for: the two power equations are quadratic, so four solutions fit, and the conditions pick the one meant:

```
Add condition(s)
is > 0 and vs > 0
```

Run it in DC, with **Rounding** at 4 significant digits. Everything comes back numerical: v_s = **17.61** V and i_s = **0.3973** A.

The answers also show *how* the CCVS dissipates nothing: i_{R5} comes back exactly **0** A, and its voltage is 0.1 i_{R5}, so the source itself is zero.

LESSON 4

Shorts, equivalent resistance and Thévenin/Norton

Learn how to describe a *short circuit* using the **s** element. Learn how to find *equivalent resistances*, and *Thévenin and Norton equivalents*.

In this lesson you will learn how to describe a *short circuit* with the **s** element, how to find *equivalent resistances*, and *Thévenin and Norton equivalents*.

4.1 How to describe a short circuit

Shorts are used mostly to find a current in a part of the circuit with no element in it already. Otherwise we would just make it a single node.

> NOTE
>
> **Describing short circuits**
>
> In Symbulator, an ideal short circuit is described with three bits of information, separated by commas: the name of the short, which must start with the letter **s**, then the names of its first and second node.
> For example, an ideal short circuit called **s1**, between nodes **3** and **5**, is described:
> `s1,3,5`

What answers do you get

No power is consumed, and no voltage is dropped, in a short circuit. For each short in a circuit, Symbulator gives only the current through it, flowing from the first node towards the second. For a short called **sx**, that is `isx`.

HK5's Drill Problem 1-13

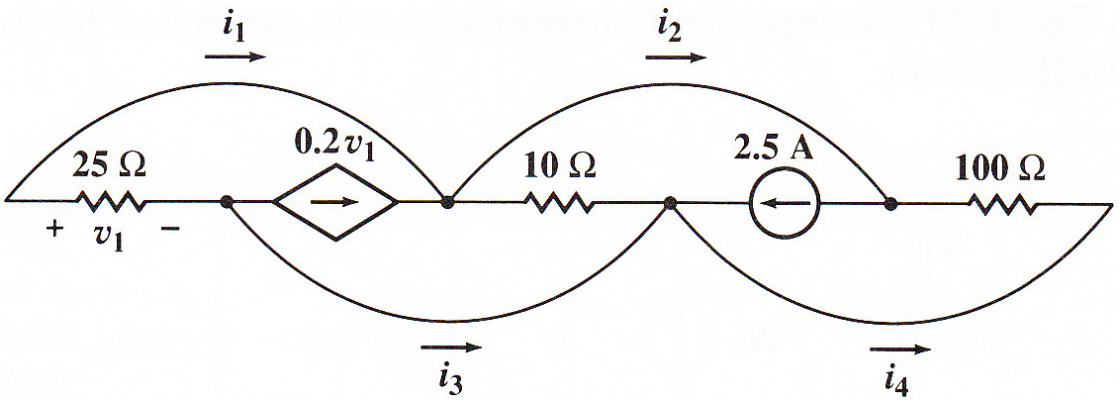
Find i_1, i_2, i_3 and i_4.

SOLUTION

My solution: I define the shorts in the same direction as the arrows in the schematic.

Circuit Description

```
r1,1,0,25
jd,0,2,0.2*v1
r2,2,3,10
```



HK5's Drill Problem 1-13

```
ji,4,3,2.5
r3,4,5,100
s1,1,2
s2,2,4
s3,0,3
s4,3,5
```

Choose *Solve circuit* and *DC*, then run it. The current through each short is given in:

`is1` = **-2** A, `is2` = **3** A, `is3` = **-8** A and `is4` = **-0.5** A.

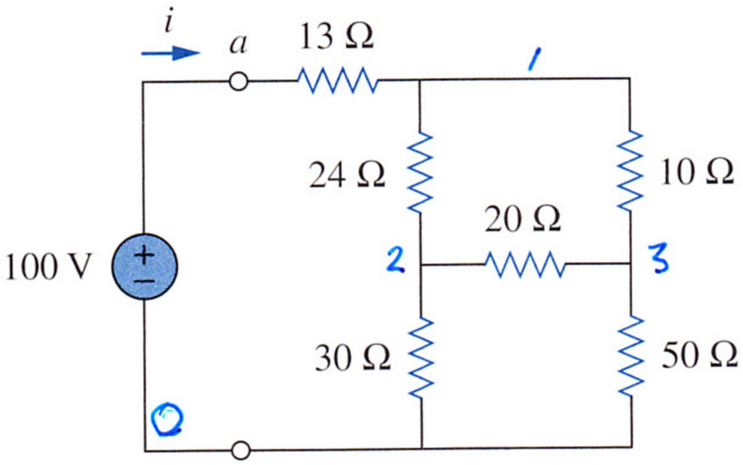
These are correct, and they can only be found using short circuits.

4.2 Finding the equivalent resistance

As we saw, Symbulator gives the equivalent resistance of a circuit as seen from any source. That lets us solve problems like this one.

AS2's Practice Problem 2.15

Find the equivalent resistance as seen by the 100 V source, and the value of current i.



AS2's Practice Problem 2.15

SOLUTION

Circuit Description

```
e,a,0,100
r13,a,1,13
r24,1,2,24
r10,1,3,10
r20,2,3,20
r30,2,0,30
r50,3,0,50
```

Solve it in DC. Because a source sits between the two nodes we care about, the equivalent resistance is one of the answers already: the resistance seen by the source, `re`, is 40 Ω.

The current ***i*** is the current through **r13**: `ir13` = **2.5** A.

Both answers came free, because there was a source connected between the two desired nodes. The rest of this lesson is about what to do when there is not.

What if there is no independent source?

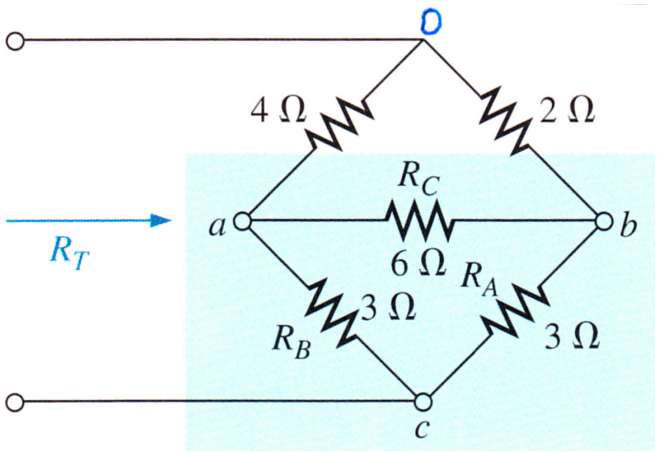
But how do we find the equivalent resistance of a **passive circuit**, one with no *independent* source in it? One way is to connect a 1 A current source between the two nodes and read the voltage drop across it. That is the manual way.

An easier way is to let Symbulator do that for us.

Set **Type of analysis** to *Find equivalent*, and a second menu appears, **Type of equivalent**, whose first entry is *Resistance / impedance*. Two node boxes appear with it, for the terminals you are measuring between. Symbulator does the 1 A trick internally and reports the answer.

B11's Example 8.29

Calculate the equivalent resistance of the circuit shown.



B11's Example 8.29

SOLUTION

My solution: I label the nodes, then describe the circuit.

Circuit Description

```
r4,0,a,4
r2,0,b,2
r6,a,b,6
rb,a,c,3
ra,b,c,3
```

Set **Type of analysis** to *Find equivalent*, **Type of equivalent** to *Resistance / impedance*, and the two node boxes to **0** and **c**. Leave **Analysis** on *DC — direct current* and press **Run Symbulator**.

Results looks different from a normal solve: instead of the node and element listing there is a single answer, headed **equivalent resistance**:

Req = **2.89** Ω

This is correct.

What counts as passive

That example was made of resistors only. *Resistance / impedance* also handles a second kind of passive circuit: one with resistors and dependent sources, but no independent sources. Such a circuit can only be reduced to an equivalent resistance, not to a Thévenin or Norton equivalent, so it is found in exactly the same way, provided the dependent sources are described properly.

4.3 Finding a Thévenin / Norton equivalent

Just as a *passive circuit* can be reduced to an equivalent resistance, an **active circuit** — one with independent sources — can be reduced to a Thévenin or Norton equivalent.

One way is to run a first simulation for the voltage between the two nodes where we want the equivalent (the Thévenin voltage, V_{TH}), then a second with a short between those nodes for the current through it (the Norton current, I_{NO}). R_{EQ} is then V_{TH}/I_{NO}. That is the manual way.

An easier way is to let Symbulator do exactly that automatically: set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*, and give it the two nodes.

RM3's Practice Problem 9-4

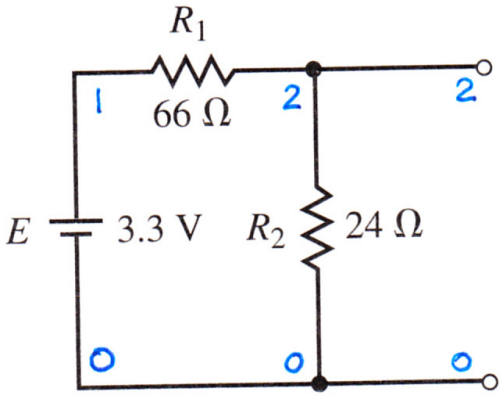
Find the Thévenin and Norton equivalents of the circuit.

SOLUTION

This is my circuit description:

Circuit Description

```
e,1,0,3.3
r1,1,2,66
r2,2,0,24
```



RM3's Practice Problem 9-4

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second — the pair of terminals you are looking into. Leave **Analysis** on *DC — direct current* and press **Run Symbulator**.

Results replaces the node-and-element listing with a single group headed **Thévenin / Norton equivalent**, holding four answers:

- **vth** has the Thévenin voltage
- **ino** has the Norton current
- **req** has the equivalent resistance
- **pmax** has the maximum power that the equivalent can deliver to a hypothetical load

Here they read **vth** = **880** mV, **ino** = **50** mA, **Req** = **17.6** Ω and **pmax** = **11** mW. The first three are what you came for; the fourth is the most this circuit could deliver into a matched load.

Problems with a load

One type of problem books and professors like when teaching the Thévenin / Norton equivalents is what I call an R_L problem. A typical one goes like this: "First, reduce the circuit, as seen by resistor R_L, to its Thévenin or Norton equivalent. Then, find the value of the voltage drop, current and/or power consumed in the load resistor R_L if its value is (whatever) ohms."

Since this is such a typical problem, Symbulator makes provision for it. When the equivalent asked for is *Thévenin / Norton*, a question appears under the two node boxes: **Are you running a problem with a load connected to this equivalent circuit?** Tick it, and **Results** adds three answers under the four, each an expression in terms of the variable ***load***, the value of the load resistor:

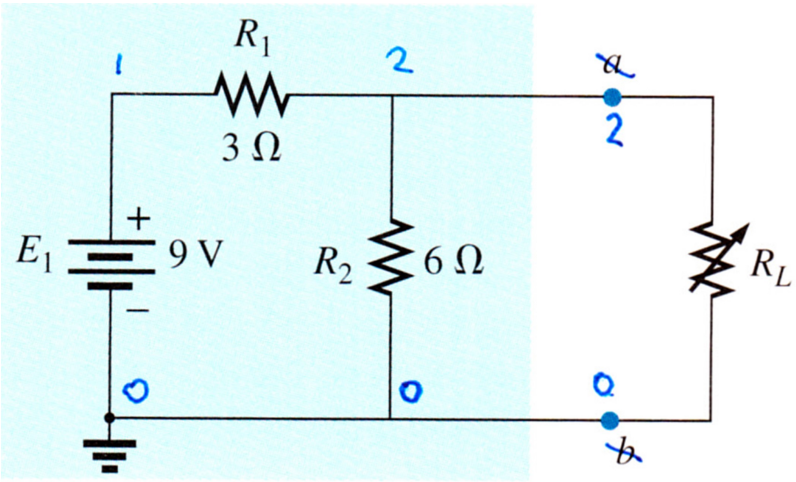
- **irl** has the current in the load
- **vrl** has the voltage drop in the load
- **prl** has the power consumed in the load

To read one of them at a given load, ask **Evaluate** for its name and put the load's value in the **Conditions** box, as in `load = 2`. Untick the question and the four answers are all

you see.

B11's Example 9.6

Find the Thévenin equivalent circuit for the network in the shaded area. Then find the current through R_L for R_L values of 2 Ω, 10 Ω and 100 Ω.



B11's Example 9.6

SOLUTION

First we find the Thévenin equivalent.

Circuit Description

```
e,1,0,9
r1,1,2,3
r2,2,0,6
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second. Tick the question about the load. Choose *DC* and run it.

Results gives:

Thévenin voltage

$$v_{th} = 6 \text{ V}$$

equivalent resistance

$$R_{eq} = 2\ \Omega$$

and, because the load question is ticked:

current in load

$$i_{rl} = \frac{6}{load + 2}$$

Correct. Now we find the current in the load for each value.

Here we use **Rounding** — *approx to n digits*, with **n** = 2, in **Settings**, so the answers read as the book prints them; exact, the third would come back as 1/17. Then ask **Evaluate** for `irl`, with the load's value in the **Conditions** box:

> Evaluate
> `irl`

> Conditions
> `load = 2`

Then `load = 10`, then `load = 100`.

The answers are **1.5** A, **.5** A and **.059** A.

The answers are correct.

Power transfer problems

Another problem often associated with the Thévenin / Norton equivalents is power transfer to a load, particularly the maximum possible. Maximum power is transferred when the load R_L equals the R_{EQ} of the equivalent. Symbulator gives you the maximum power that can be delivered in `pmax`, and the power transferred to the load as a function of its value in `prl`, once the load question is ticked.

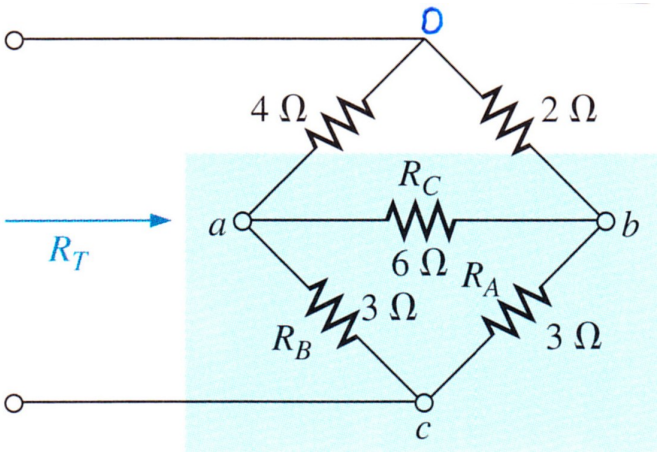
What if it's more than a load?

The formulas for the current, voltage and power in the load apply only when a load is the only thing connected to the equivalent circuit. If the problem you want to solve includes something more complicated, you will have to run your own simulation.

To help you in those cases, Symbulator writes that circuit for you: with the load question ticked, a button under the equivalent's answers drops the Norton equivalent, load and all, into the input fields, ready to be added to. When a load is not the only thing attached (page 300) covers it, with a worked example — optional reading, and nothing else in this lesson depends on it.

4.4 Instructive solved examples

Practice problems for resistive circuits



B11's Example 8.29

Calculate the equivalent resistance of the circuit shown.

You saw this circuit solved step by step in the walkthrough above; here it is again, in the compact form the rest of these problems use.

After I label the nodes, I describe the circuit.

Circuit Description

```
r4,0,a,4
r2,0,b,2
r6,a,b,6
rb,a,c,3
ra,b,c,3
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Resistance / impedance*, with nodes **0** and **c**, in DC. The answer is `req` = **2.889** Ω.

This is correct. Below are many practice examples of this type.

B11's Example 8.30

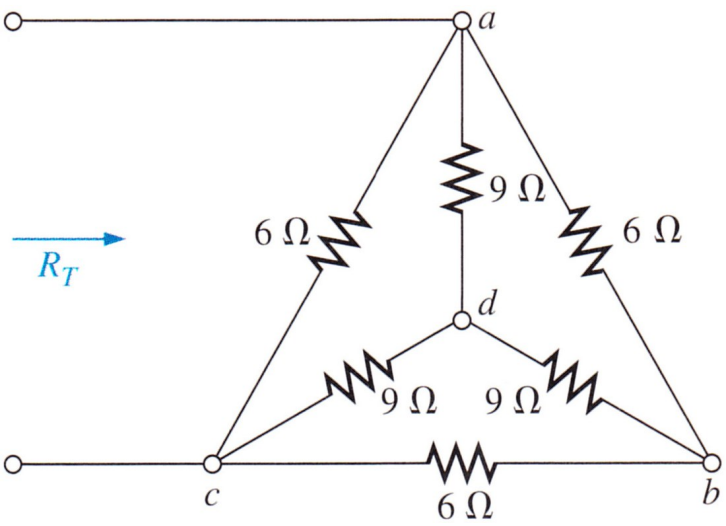
Find the equivalent resistance of the circuit.

Circuit Description

```
rac,a,c,6
rad,a,0,9
rab,a,b,6
rcd,c,0,9
rbc,b,c,6
rbd,b,0,9
```

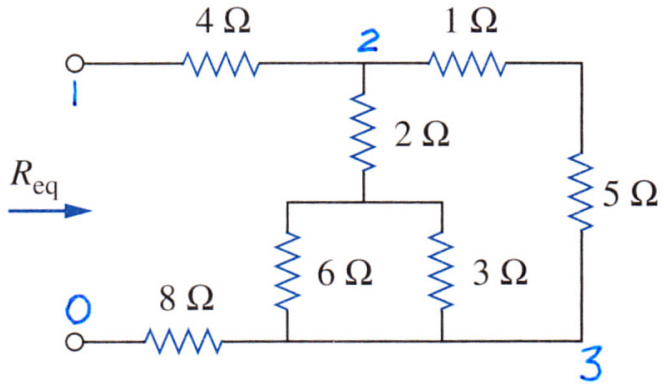
Find equivalent, *Resistance / impedance*, nodes **a** and **c**, in DC: `req` is **3.273** Ω.

The problem names the four nodes **a**, **b**, **c** and **d**. Symbulator needs one of them to be the reference node, so d is called 0 here. Which node you ground makes no difference to the answer in a network with no sources — grounding b gives the same **3.273** Ω.



AS2's Example 2.9

Find the equivalent resistance of the circuit.



We don't need to run a simulation for this. We can reduce it using `pr`.

> Evaluate
> `4+pr(1+5,2+pr(6,3))+8`

Evaluate gives the equivalent resistance: **14.4** Ω.

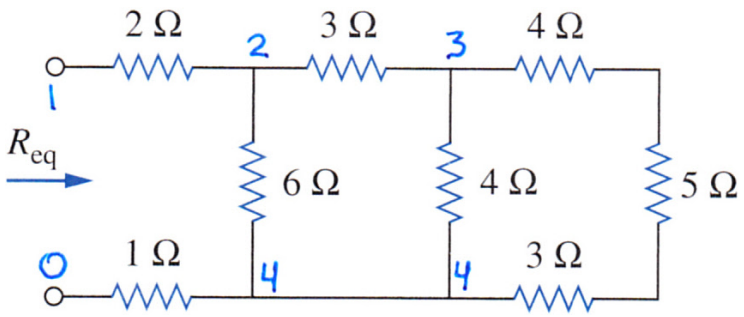
AS2's Practice Problem 2.9

Find the equivalent resistance of the circuit.

We don't need to run a simulation for this. We can reduce it using `pr`.

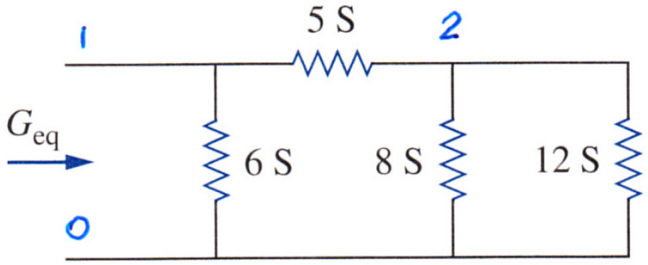
> Evaluate
> `2+pr(6,3+pr(4,4+5+3))+1`

Evaluate gives the equivalent resistance: **6** Ω.



AS2's Example 2.11

Find the equivalent *conductance* of the circuit.



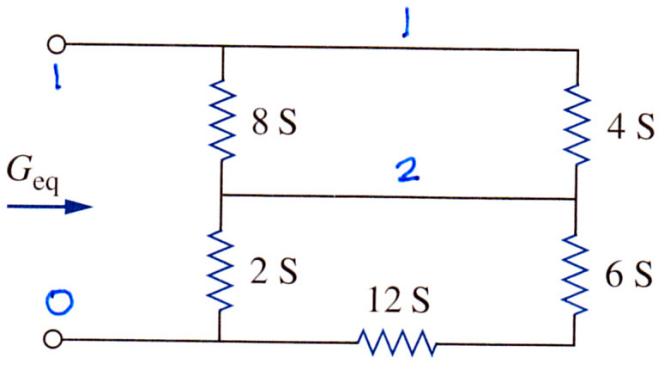
We don't need to run a simulation for this. We can reduce it using `pr`.

```
Evaluate
1/(pr(1/6,1/5+pr(1/8,1/12)))
```

Evaluate gives the equivalent conductance: **10** S.

AS2's Practice Problem 2.11

Find the equivalent *conductance* of the circuit.



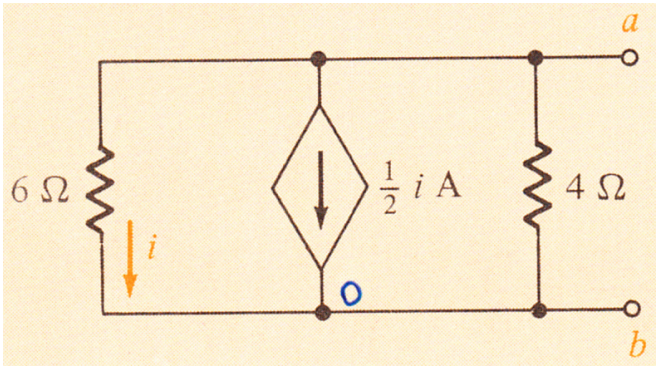
We don't need to run a simulation for this. We can reduce it using `pr`.

```
Evaluate
1/(pr(1/8,1/4)+pr(1/2,1/12+1/6))
```

Evaluate gives the equivalent conductance: **4** S.

Bo2's Drill Problem 3.14

Find the equivalent resistance of the circuit.



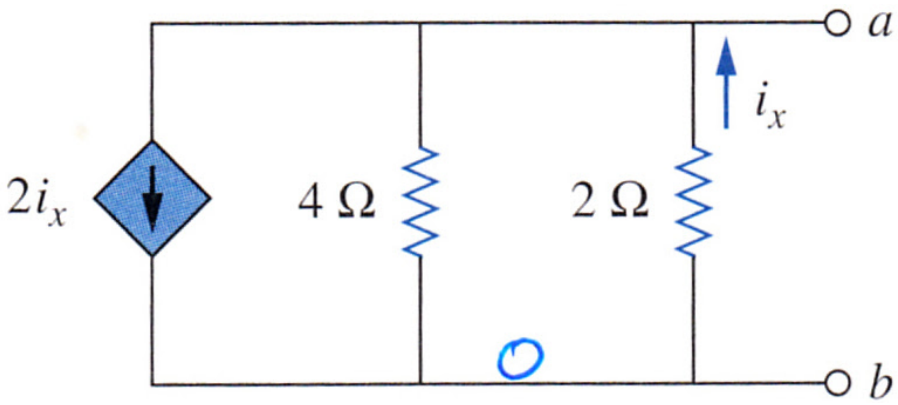
Circuit Description

```
r4,a,0,4
ri,a,0,6
ji,a,0,iri/2
```

Find equivalent, *Resistance / impedance*, nodes **a** and **0**, in DC: `req` is **2** Ω.

AS2's Example 4.10

Find the equivalent resistance of the circuit.



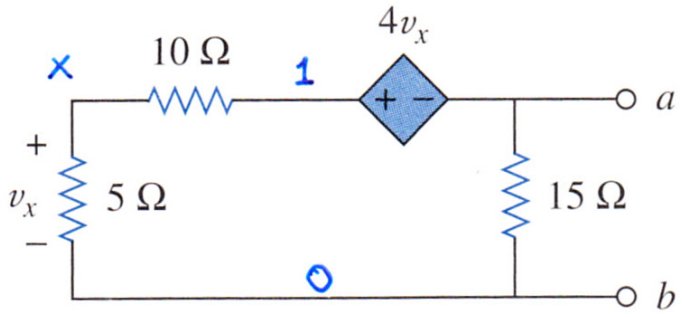
Circuit Description

```
r4,a,0,4
rx,0,a,2
j,a,0,2*irx
```

Find equivalent, *Resistance / impedance*, nodes **a** and **0**, in DC: `req` is **-4** Ω. It may be surprising to have a negative resistance. This is the result of the dependent sources.

AS2's Practice Problem 4.10

Find the equivalent resistance of the circuit.



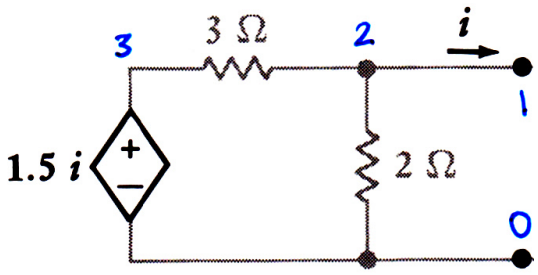
Circuit Description

```
r15,a,0,15
e,1,a,4*vrx
r10,1,x,10
rx,x,0,5
```

Find equivalent, *Resistance / impedance*, nodes **a** and **0**, in DC: `req` is **-7.500** Ω.

HK5's Figure 2-29

Find the equivalent resistance of the circuit.



Circuit Description

```
e,3,0,1.5*is
r3,3,2,3
r2,2,0,2
s,2,1
```

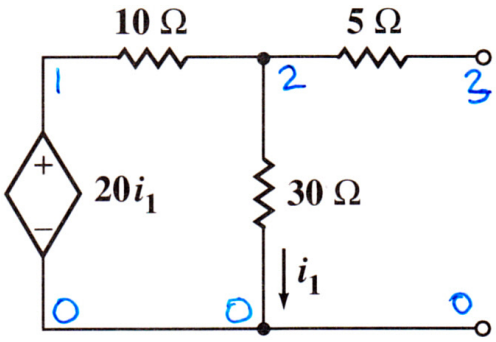
Find equivalent, *Resistance / impedance*, nodes **1** and **0**, in DC: `req` is **0.6000** Ω.

HK5's Drill Problem 2-9d

Find the equivalent resistance of the circuit.

Circuit Description

```
r10,1,2,10
r5,2,3,5
```

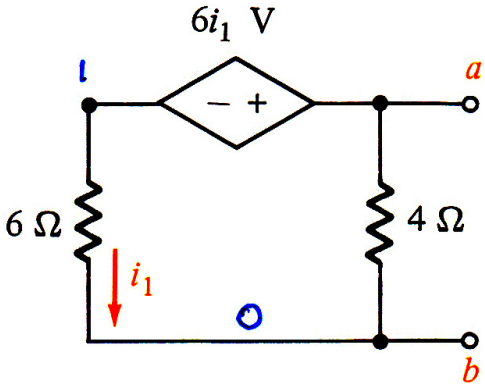



```
r1,2,0,30
e,1,0,20*ir1
```

Find equivalent, *Resistance / impedance*, nodes **3** and **0**, in DC: `req` is **20** Ω.

Bo2's Example 3.12

Find the equivalent resistance of the circuit.



Circuit Description

```
r1,1,0,6
r4,a,0,4
e,a,1,6*ir1
```

Find equivalent, *Resistance / impedance*, nodes **a** and **0**, in DC: `req` is **3** Ω.

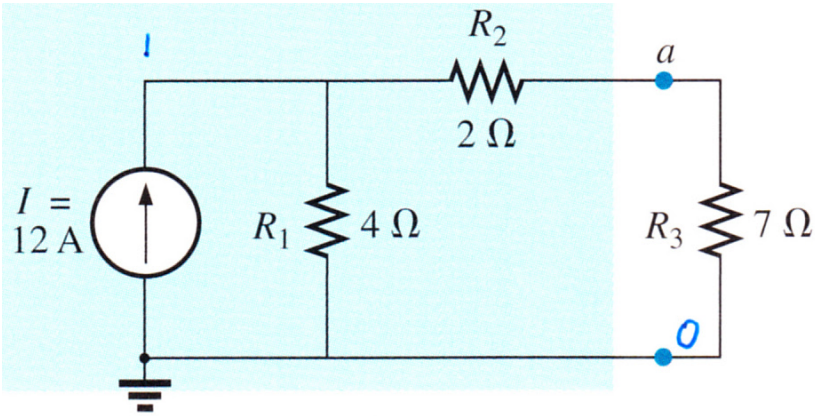
B11's Example 9.7

Find the Thévenin equivalent of the circuit, as seen from the R_3 resistor.

Circuit Description

```
j,0,1,12
r1,1,0,4
r2,1,a,2
```

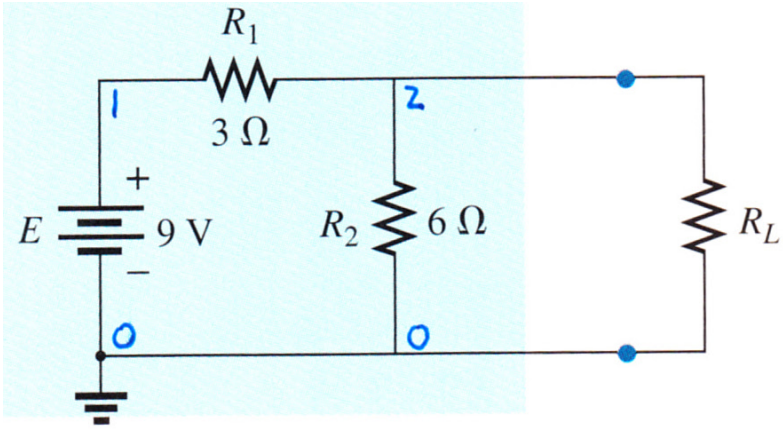
Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **a** in the first and **0** in the second — the pair of terminals you are looking into.



In DC, `vth` gives V_{TH} = **48** V and `req` gives R_{EQ} = **6** Ω.

B11's Example 9.11

Find the Norton equivalent of the circuit, as seen from the R_L resistor.



> Circuit Description
> ```
> e,1,0,9
> r1,1,2,3
> r2,2,0,6
> ```

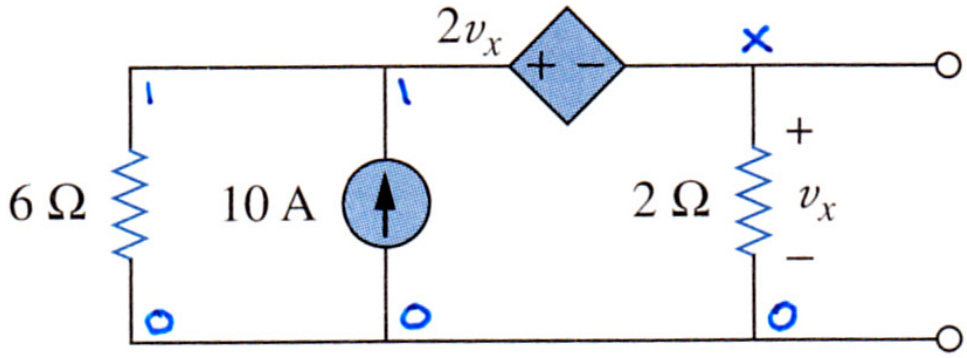
Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `ino` gives I_{NO} = **3** A and `req` gives R_{EQ} = **2** Ω.

AS2's Practice Problem 4.12

Find the Norton equivalent of the circuit.

> Circuit Description
> ```
> r6,1,0,6
> j,0,1,10
> r2,x,0,2
> ```

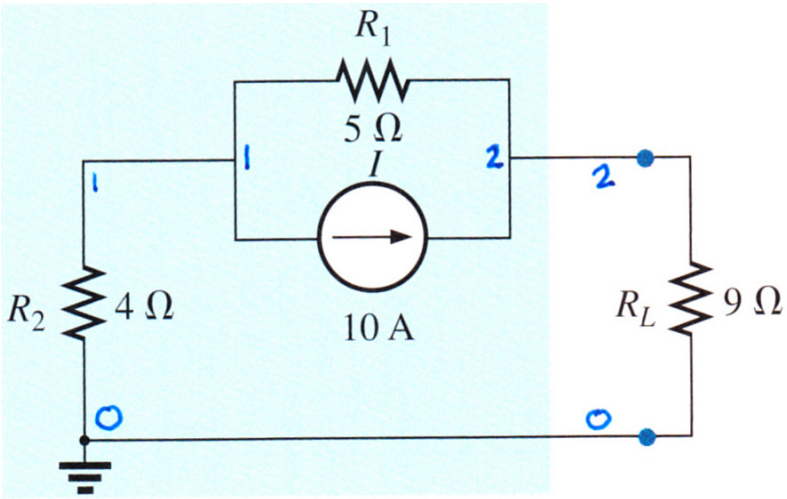



```
e,1,x,2*vx
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **x** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `ino` gives I_{NO} = **10** A and `req` gives R_{EQ} = **1** Ω.

B11's Example 9.12



Find the Norton equivalent of the circuit, as seen from the R_L resistor.

Circuit Description

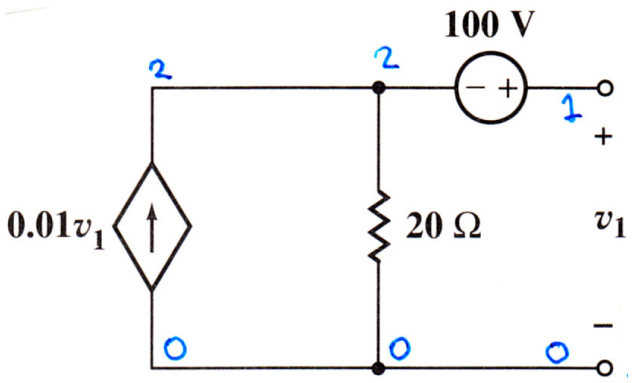
```
r2,1,0,4
r1,1,2,5
j,1,2,10
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `ino` gives I_{NO} = **5.56** A and `req` gives R_{EQ} = **9** Ω.

HK5's Drill Problem 2-8b

Find the Thévenin equivalent of the circuit.



Circuit Description

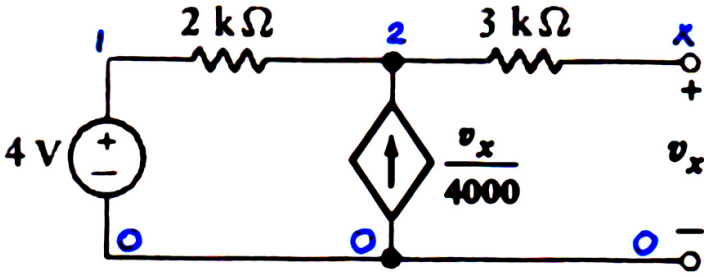
```
j,0,2,0.01*v1
r,0,2,20
e,1,2,100
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **1** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `vth` gives V_{TH} = **125** V and `req` gives R_{EQ} = **25** Ω.

HK5's Figure 2-27

Find the Thévenin equivalent of the circuit.



Circuit Description

```
e,1,0,4
r2,1,2,2'k
r3,2,x,3'k
j,0,2,vx/4000
```

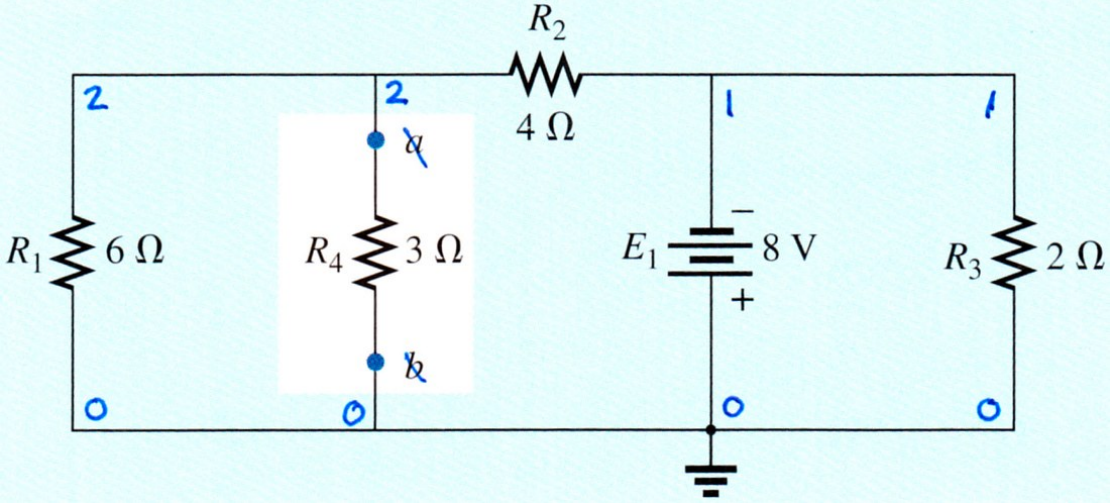
Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **x** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `vth` gives V_{TH} = **8** V and `req` gives R_{EQ} = **10** kΩ.

B11's Example 9.8

Find the Thévenin equivalent of the circuit, as seen from the R_4 resistor.

I ignore the textbook's decision to call the nodes **a** and **b**, since **b** is ground anyway.



Circuit Description

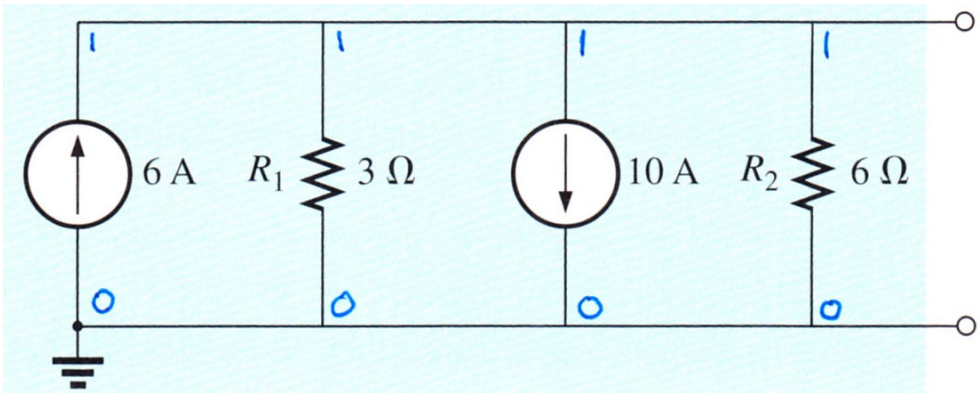
```
r1,2,0,6
r2,2,1,4
r3,1,0,2
e,0,1,8
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `vth` gives V_{TH}= **-4.8** V and `req` gives R_{EQ}= **2.4** Ω.

B11's Example 8.6

Find the Norton equivalent of the circuit.



Circuit Description

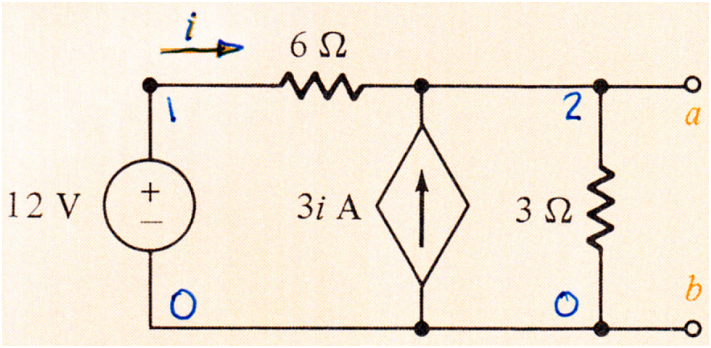
```
j1,0,1,6
r1,1,0,3
j2,1,0,10
r2,1,0,6
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **1** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `ino` gives I_{NO} = **-4** A and `req` gives R_{EQ} = **2** Ω.

Bo2's Drill Exercise 3.12

Find the Norton equivalent of the circuit.



Circuit Description

```
e,1,0,12
r6,1,2,6
j,0,2,3*ir6
r3,2,0,3
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `ino` gives I_{NO} = **8** A and `req` gives R_{EQ} = **1** Ω.

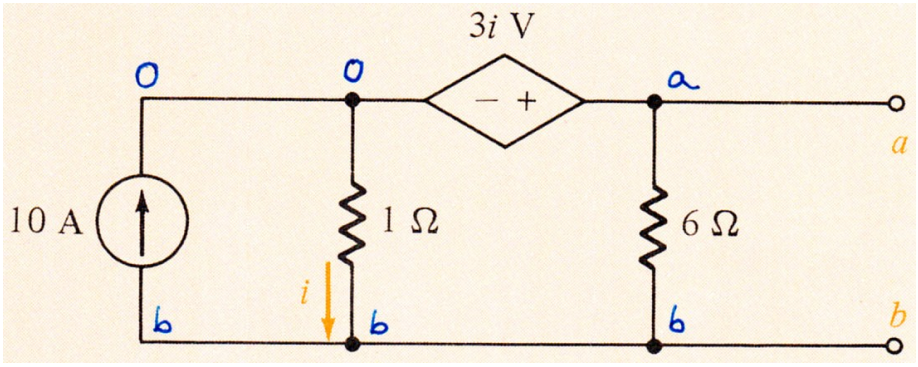
Bo2's Drill Exercise 3.9

Find the Thévenin equivalent of the circuit.

Circuit Description

```
j,b,0,10
r1,0,b,1
e,a,0,3*ir1
r6,a,b,6
```

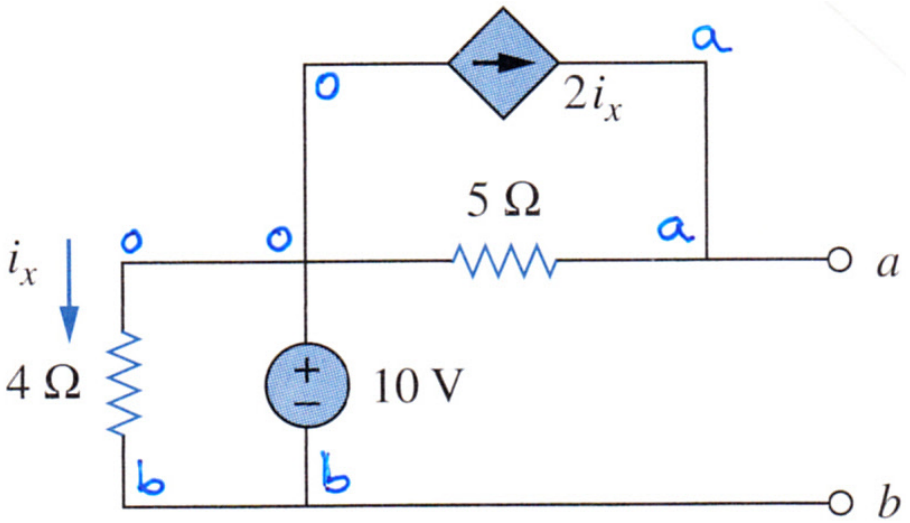
Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **a** in the first and **b** in the second — the pair of terminals you are looking into.



In DC, `vth` gives V_{TH} = **24** V and `req` gives R_{EQ} = **2.4** Ω.

AS2's Example 4.12

Find Norton equivalent of the circuit.



> Circuit Description
>
> ```
> rx,0,b,4
> e,0,b,10
> r5,0,a,5
> j,0,a,2*irx
> ```

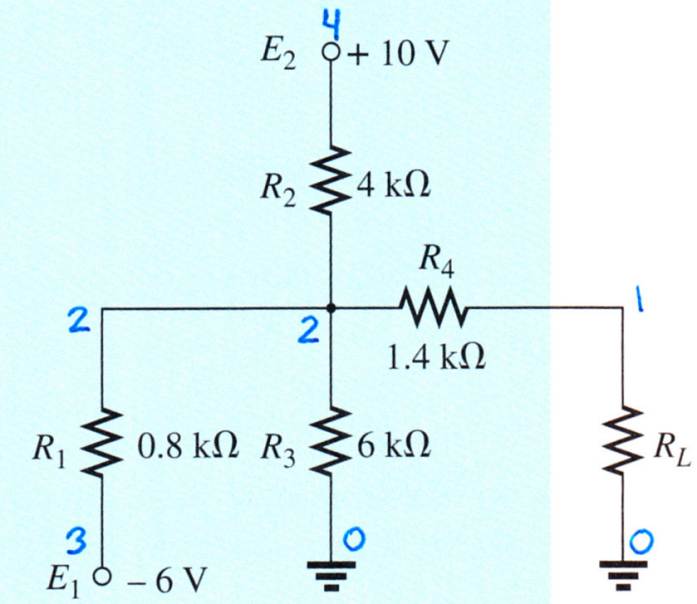
Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **a** in the first and **b** in the second — the pair of terminals you are looking into.

In DC, `ino` gives I_{NO} = **7** A and `req` gives R_{EQ} = **5** Ω.

B11's Example 9.10 (Hidden source)

Find the Thévenin equivalent of the circuit.

My solution is below:



Circuit Description

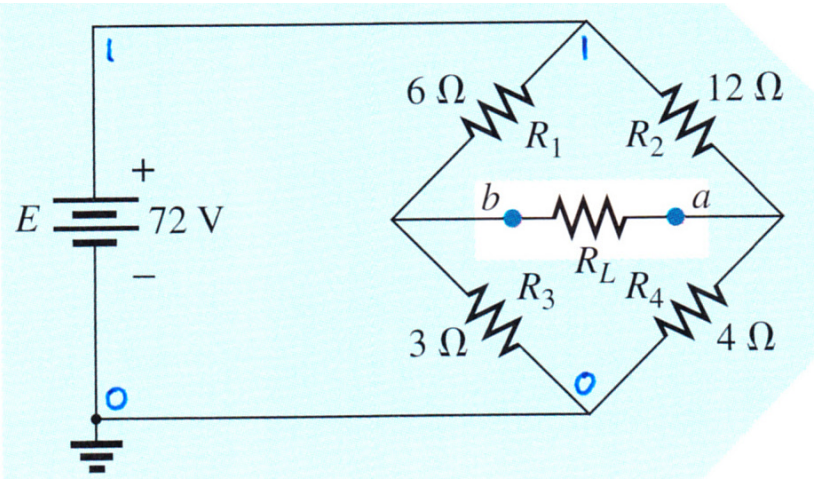
```
e1,3,0,-6
e2,4,0,10
r1,2,3,0.8'k
r2,2,4,4'k
r3,2,0,6'k
r4,2,1,1.4'k
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **1** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `vth` gives V_{TH} = **-3** V and `req` gives R_{EQ} = **2** kΩ.

B11's Example 9.9

Find the Thévenin equivalent of the circuit, as seen from the R_L resistor.



Circuit Description

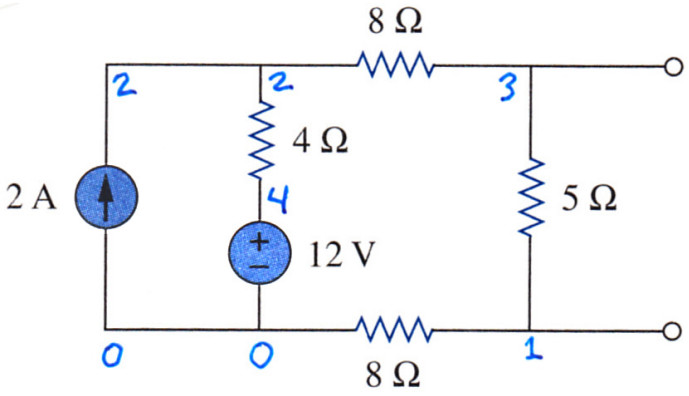
```
e,1,0,72
r1,1,b,6
r2,1,a,12
r3,0,b,3
r4,0,a,4
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **b** in the first and **a** in the second — the pair of terminals you are looking into.

In DC, `vth` gives V_{TH} = **6** V and `req` gives R_{EQ} = **5** Ω.

AS2's Example 4.11

Find the Norton equivalent of the circuit.



Circuit Description

```
j,0,2,2
e,4,0,12
r1,2,4,4
r2,2,3,8
r3,0,1,8
r4,3,1,5
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **3** in the first and **1** in the second — the pair of terminals you are looking into.

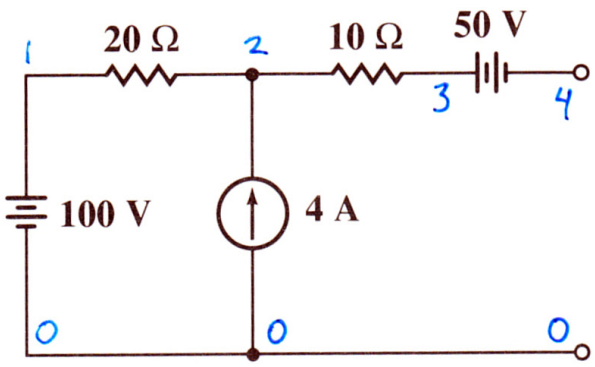
In DC, `ino` gives I_{NO} = **1** A and `req` gives R_{EQ} = **4** Ω.

HK5's Drill Problem 2-8a

Find the Thévenin equivalent of the circuit.

Circuit Description

```
e1,1,0,100
r2,1,2,20
j,0,2,4
```



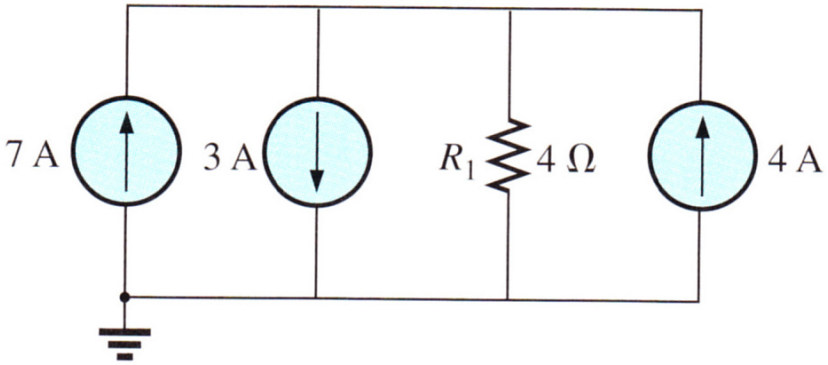

```
r1,2,3,10
e5,3,4,50
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **4** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `vth` gives V_{TH} = **130** V and `req` gives R_{EQ} = **30** Ω.

B11's Example 8.7

Find Norton equivalent of the circuit.



Circuit Description

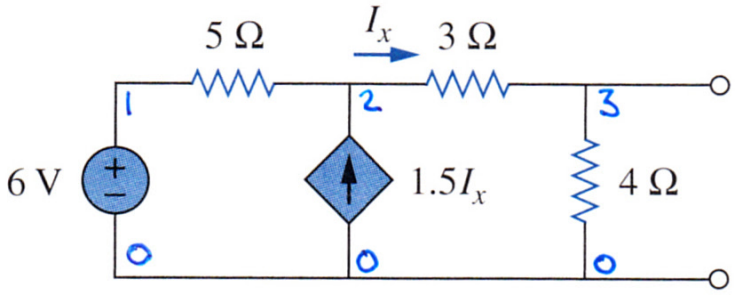
```
j7,0,1,7
j3,1,0,3
r1,1,0,4
j4,0,1,4
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **1** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `ino` gives I_{NO} = **8** A and `req` gives R_{EQ} = **4** Ω.

AS2's Practice Problem 4.9

Find the Thévenin equivalent of the circuit.



Circuit Description

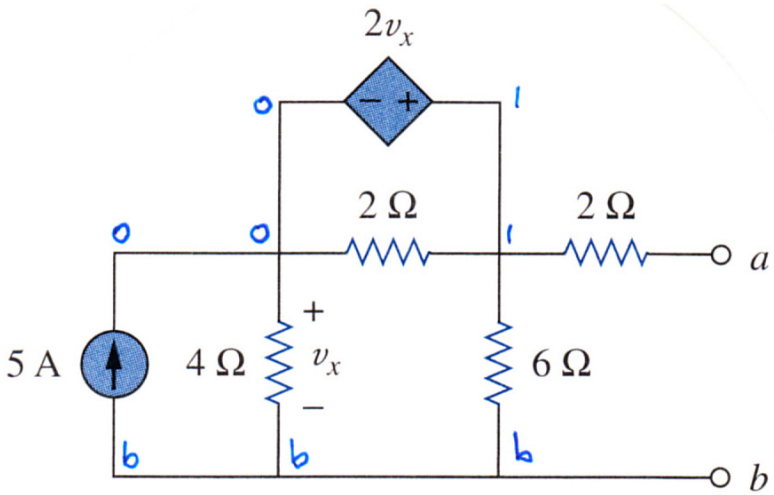
```
e,1,0,6
r5,1,2,5
rx,2,3,3
j,0,2,1.5*irx
r4,3,0,4
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **3** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `vth` gives V_{TH} = **5.33** V and `req` gives R_{EQ} = **0.44** Ω.

AS2's Example 4.9

Find the Thévenin equivalent of the circuit.



Circuit Description

```
j,b,0,5
rx,0,b,4
r1,0,1,2
r2,1,b,6
r3,1,a,2
e,1,0,2*vrx
```

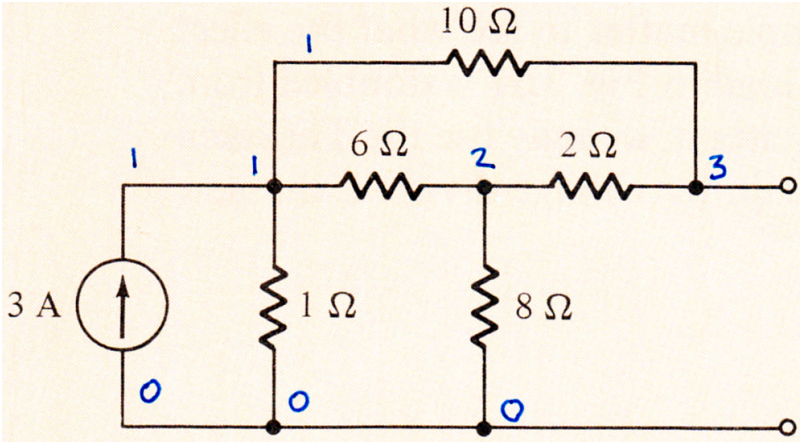
Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **a** in the first and **b** in the second — the pair of terminals you are

looking into.

In DC, `vth` gives V_{TH} = **20** V and `req` gives R_{EQ} = **6** Ω.

Bo2's Drill Exercise 3.8

Find the Thévenin equivalent of the circuit.



Circuit Description

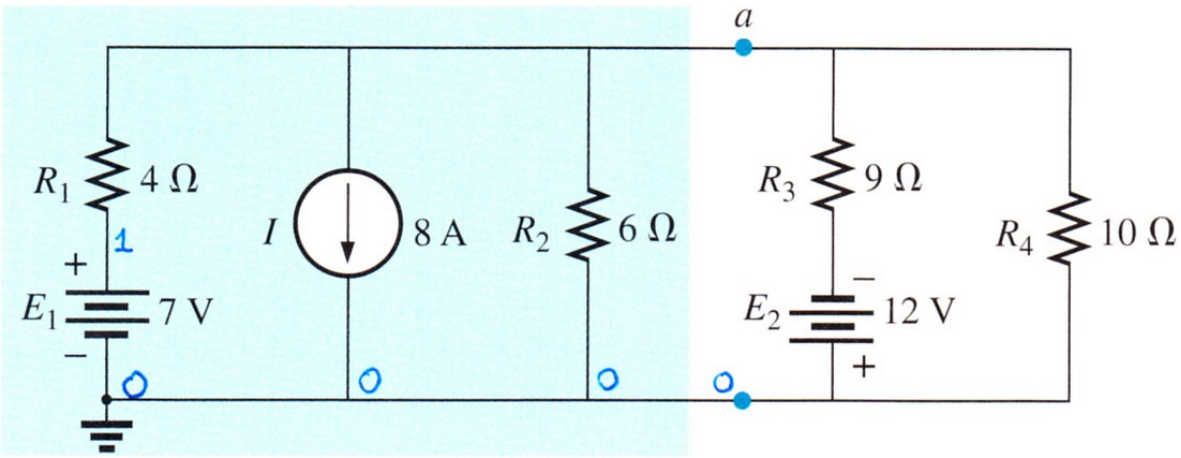
```
j,0,1,3
r1,0,1,1
r6,1,2,6
r10,1,3,10
r8,2,0,8
r2,2,3,2
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **3** in the first and **0** in the second — the pair of terminals you are looking into.

In DC, `vth` gives V_{TH} = **2** V and `req` gives R_{EQ} = **4** Ω.

B11's Example 9.13

Find Norton equivalent of the shaded part of the circuit.



Circuit Description

```
e,1,0,7
r1,1,a,4
j,a,0,8
r2,a,0,6
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **a** in the first and **0** in the second — the pair of terminals you are looking into.

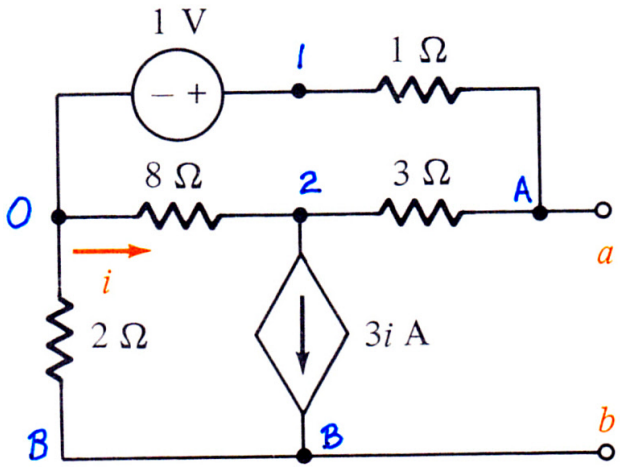
In DC, `ino` gives I_{NO} = **-6.25** A and `req` gives R_{EQ} = **2.4** Ω.

Tricky problems

Some circuit theory books – and some professors – find it entertaining or instructive to surprise unsuspecting students with tricky problems, like the two we solve below.

Bo2's Example 3.11 (Tricky)

Find the Norton equivalent of the circuit.



Since you do not know beforehand that this is a tricky problem, you go for the usual:

Circuit Description

```
r2,0,b,2
r8,0,2,8
r3,2,a,3
r1,1,a,1
e,1,0,1
j,2,b,3*ir8
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **a** in the first and **b** in the second — the pair of terminals you are looking into.

Run it in DC and it is refused: *Could not solve the system of equations.* This often means there is a division by zero somewhere.

Symbulator's own advice is to try again with a symbolic value, which is what we do. We chose to use **x** instead of **3** in the dependent source.

Circuit Description

```
r2,0,b,2
r8,0,2,8
r3,2,a,3
r1,1,a,1
e,1,0,1
j,2,b,x*ir8
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **a** in the first and **b** in the second — the pair of terminals you are looking into.

Now it solves. The expression for `ino` is fine, but the one for `req`,

equivalent resistance

$$R_{eq} = \frac{9x - 35}{4(x - 3)}$$

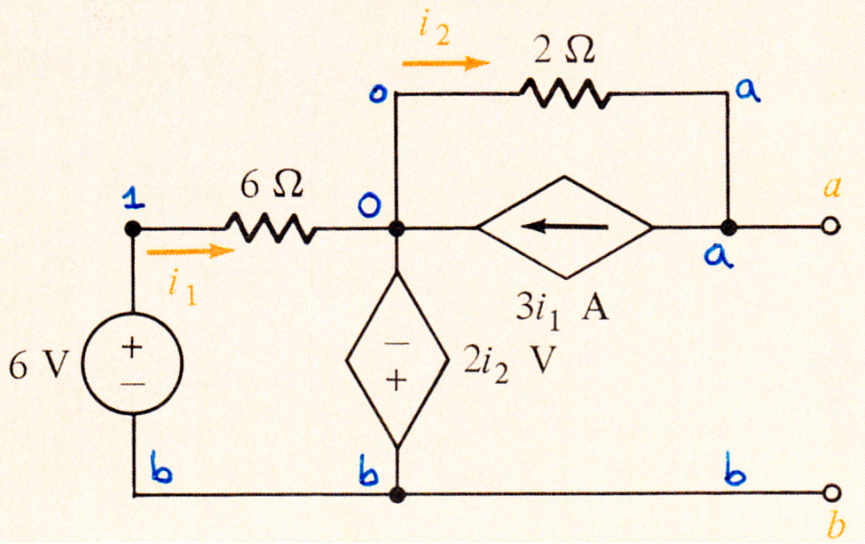
will divide by zero at $x = 3$ — the very value we replaced.

The equivalent is in terms of x, so nothing needs running again. Put `ino` in **Evaluate** with `x = 3` in its **Conditions** box: I_{NO} = **1** A. Ask for `req` the same way: infinite, the denominator being zero.

This means the equivalent resistance is, for practical purposes, infinite. Your idea of fun, right?

Bo2's Drill Exercise 3.13 (Tricky)

Another tricky one. Find the Norton equivalent of the circuit.



As in the previous example, we get an error if we simulate using **2** and **3** in the dependent sources. So, we use **2x** and **3x** instead, where **x** will be later defined as 1.

Circuit Description

```
ei,1,b,6
ed,b,0,2*x*ir2
r1,1,0,6
r2,0,a,2
jd,a,0,3*x*ir1
```

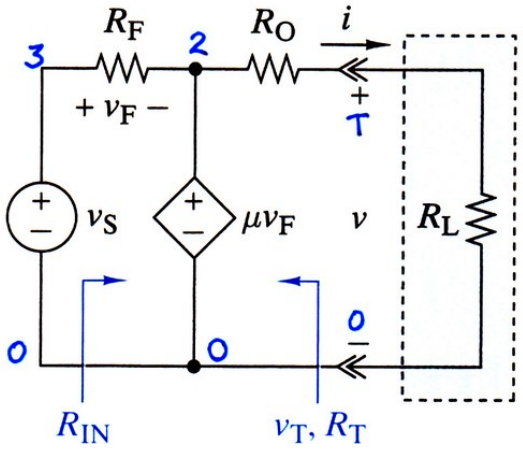
Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **a** in the first and **b** in the second — the pair of terminals you are looking into.

In the answers, the denominator of the expression for `req`, (x-1), divides by zero at x = 1.

Put `ino` in the **Evaluate** card with `x = 1` in its **Conditions** box: I_{NO} = **-3** A. Ask for `req` the same way and the answer is infinite, so R_{EQ} is undefined at that value.

TR5's Exercise 4-6 (Symbolic)

Find the input resistance and output Thévenin equivalent circuit of the circuit



$$R_{IN} = (1 + \mu)R_F$$

$$v_T = \frac{\mu}{\mu + 1} v_S$$

$$R_T = R_O$$

First let's find the input resistance, R_{IN}, i.e. the resistance as seen by the v_S source. We use μ for the constant in the dependent source.

Circuit Description

```
ei,3,0,vs
rf,3,2,rf
ro,2,t,ro
ed,2,0,μ*vrf
```

The answer you want is `rei`, in **Results**.

It reads `rf*(μ+1)`, which is correct. The textbook's answers are shown right of the circuit schematic. Now we find the output Thévenin equivalent circuit as seen by R_L.

Circuit Description

```
ei,3,0,vs
rf,3,2,rf
ro,2,t,ro
ed,2,0,μ*vrf
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **t** in the first and **0** in the second — the pair of terminals you are looking into.

The answers you want are `vth` and `req`, in **Results**.

Results gives

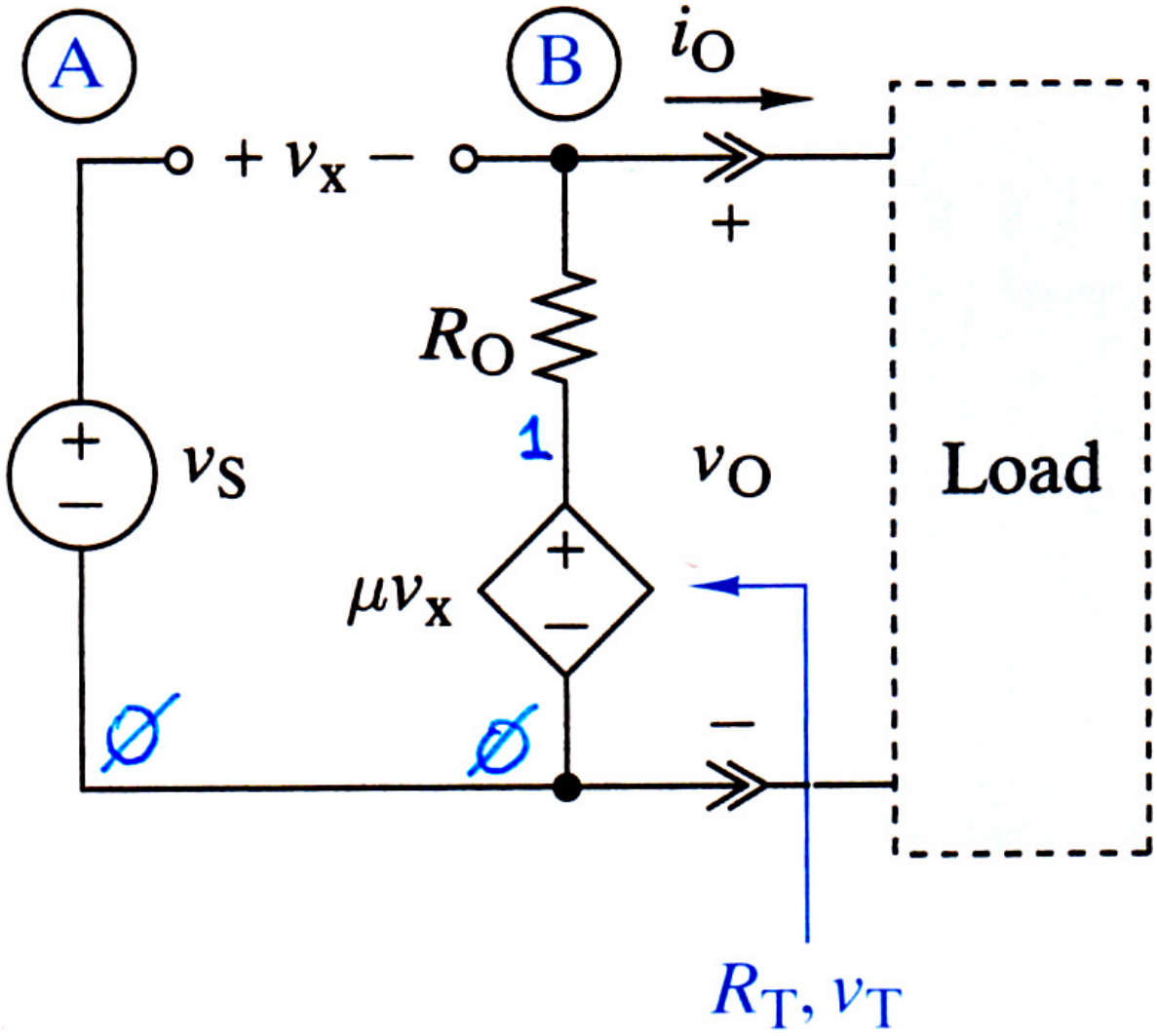
> Thévenin voltage
>
> $$v_{th} = \frac{vs\,\mu}{\mu + 1}$$

> equivalent resistance
>
> $$R_{eq} = ro$$

which is correct, as can be seen in the textbook's answers for v_T and R_T, shown right of the circuit schematic above.

TR5's Example 4-8 (Symbolic)

Find the Thévenin equivalent as seen by the load.



The dependent source's value is the difference between two node voltages, written straight into the value as `va-vb`, as Lesson 3 (page 42) describes.

Now I find the Thévenin equivalent of this circuit description:

Circuit Description

```
ei,a,0,vs
ed,1,0,μ*(va-vb)
ro,b,1,ro
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **b** in the first and **0** in the second — the pair of terminals you are looking into.

The answers you want are `vth` and `req`, in **Results**.

The answers we get in **Results**,

Thévenin voltage

$$v_{th} = \frac{vs\,\mu}{\mu + 1}$$

equivalent resistance

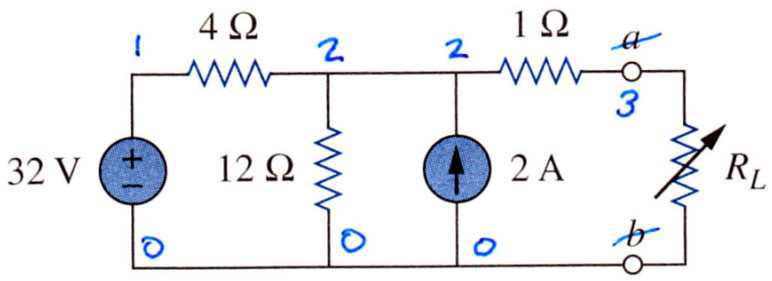
$$R_{eq} = \frac{ro}{\mu + 1}$$

are correct, as can be seen by comparing them to those in the book:

$$v_T = \frac{\mu v_S}{\mu + 1} \quad \text{and} \quad R_T = \frac{R_O}{\mu + 1}$$

AS2's Example 4.8

Find the Thévenin equivalent of the circuit shown to the left of terminals a-b. Then find the current through R_L = 6, 16 and 36Ω.



My solution:

Circuit Description

```
e,1,0,32.
r4,1,2,4
r12,2,0,12
```

```
j,0,2,2
r1,2,3,1
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **3** in the first and **0** in the second — the pair of terminals you are looking into. Tick the question about the load, choose *DC* and run it.

The answers you want are `vth` and `req`, in **Results**, and under them the load answers.

Ask **Evaluate** for `irl`, with the load in the **Conditions** box:

```
Evaluate
irl
```

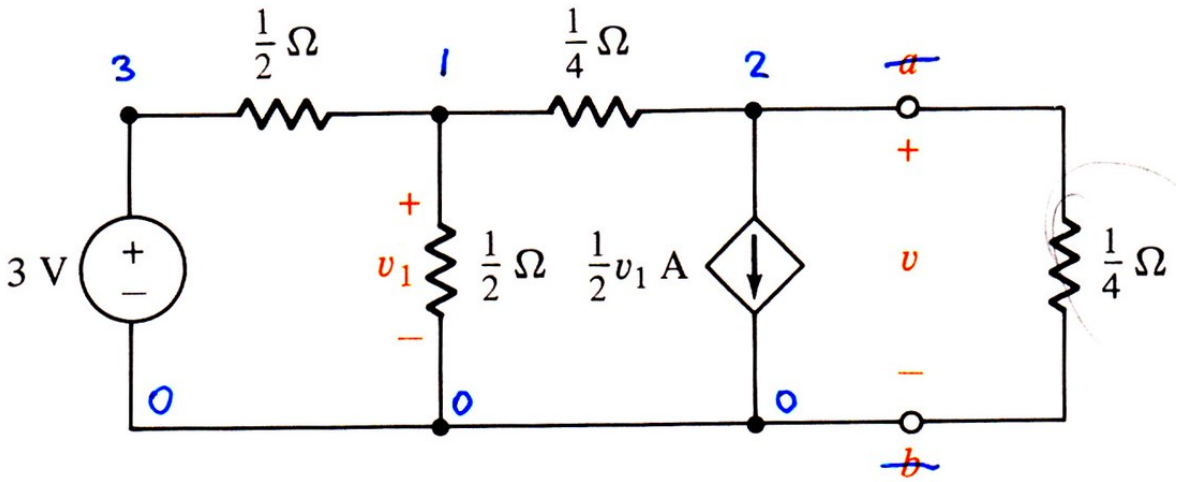
```
Conditions
load = 6
```

Then `load = 16`, then `load = 36`.

The answer is `vth` = **30** V and `req` = **4** Ω, and `irl` is **3** A for 6 Ω, **1.5** A for 16 Ω and **.75** A for 36 Ω. This is correct.

Bo2's Example 3.10

Find the Norton equivalent of the circuit left of the a-b terminals, and then find the voltage drop and the current through the ¼ Ω resistor. My solution:




```
Circuit Description
e,3,0,3
r31,3,1,1/2
r10,1,0,1/2
r12,1,2,1/4
j,2,0,v1/2
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second — the pair of terminals you are looking into. Tick the question about the load, choose *DC* and run it.

The answers you want are `ino` and `req`, in **Results**, and under them the load answers.

Ask **Evaluate** for `vrl`, then for `irl`, with the load in the **Conditions** box each time:

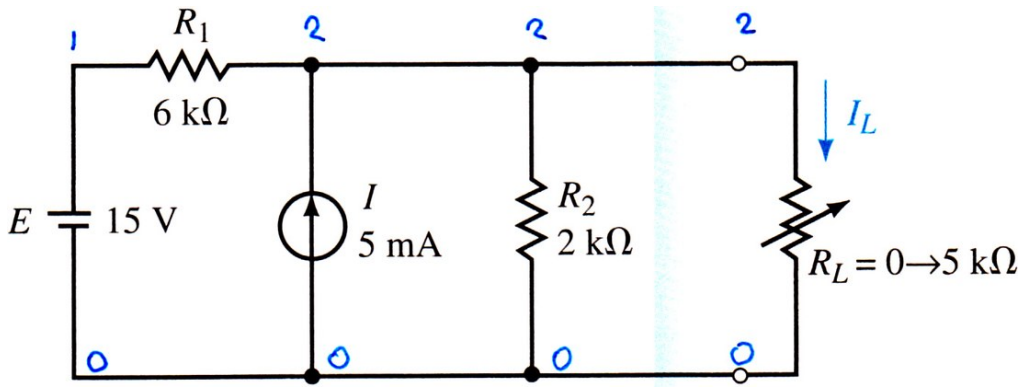
```
Evaluate
vrl
```

```
Conditions
load = 1/4
```

The book gives the answers as fractions. We get it right: `ino` = **21/8** A and `req` = **4/9** Ω, and at that load `vrl` = **21/50** V and `irl` = **42/25** A.

RM3's Example 9-7

Find the Norton equivalent of the circuit external to R_L. Then determine the load current I_L when R_L = 0 Ω, 2 kΩ and 5 kΩ. My solution:




```
Circuit Description
e,1,0,15.
r1,1,2,6'k
j,0,2,5'm
r2,2,0,2'k
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second — the pair of terminals you are looking into. Tick the question about the load, choose *DC* and run it.

The answers you want are `ino` and `req`, in **Results**, and under them the load answers.

Ask **Evaluate** for `irl`, with the load in the **Conditions** box:

```
Evaluate
irl
```

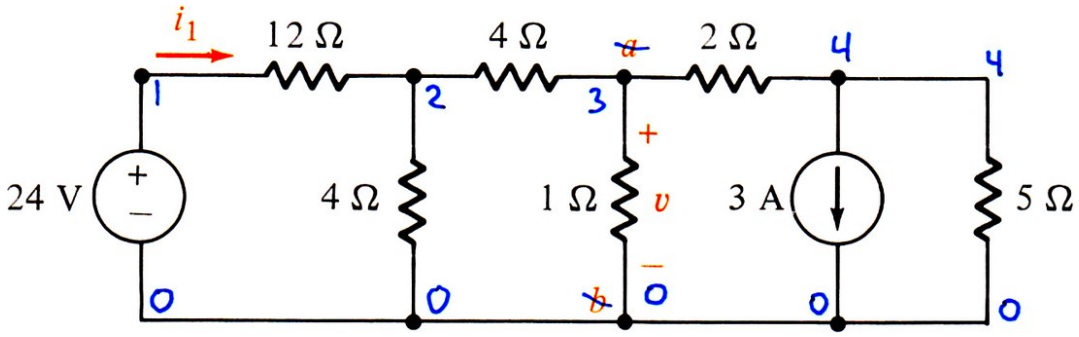
```
Conditions
load = 0
```

Then `load = 2000`, then `load = 5000`.

The answer is `ino` = **.0075** A and `req` = **1500** Ω, and `irl` is **.0075** A for 0 Ω, **.00321** A for 2 kΩ and **.00173** A for 5 kΩ. This is correct.

Bo2's Example 3.5

Find the Norton equivalent of the circuit external to the 1Ω resistor. Then determine the voltage drop across this 1Ω resistor.



My solution:

Circuit Description

```
e,1,0,24
r12,1,2,12
r20,2,0,4
r23,2,3,4
r34,3,4,2
j,4,0,3
r40,4,0,5
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **3** in the first and **0** in the second — the pair of terminals you are looking into. Tick the question about the load, choose *DC* and run it.

The answers you want are `ino` and `req`, in **Results**, and under them the load answers.

Ask **Evaluate** for `vrl`, with the load in the **Conditions** box:

Evaluate

```
vrl
```

Conditions

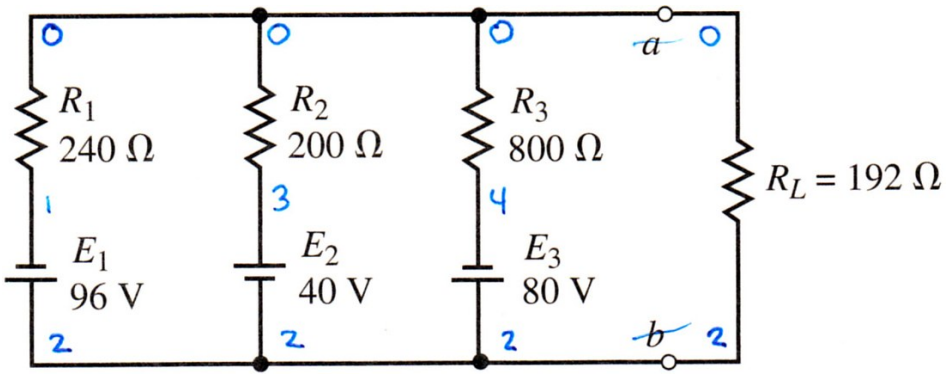
```
load = 1
```

The answer is `ino` = **-9/7** A and `req` = **7/2** Ω, and `vrl` in the 1 Ω load is **-1** V. This is correct.

RM3's Example 9-13

Use Millman's Theorem to simplify the circuit left of a-b so that there is only one voltage and one resistor. Then find the current in the load resistor $\boldsymbol{R_L}$.

I don't know Millman's Theorem, but we can find the Thévenin equivalent with Symbulator.



Circuit Description

```
r1,0,1,240.
e1,2,1,96
r2,0,3,200
e2,3,2,40
r3,0,4,800
e3,2,4,80
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second — the pair of terminals you are looking into. Tick the question about the load, choose *DC* and run it.

The answers you want are `vth` and `req`, in **Results**, and under them the load answers.

Ask **Evaluate** for `irl`, with the load in the **Conditions** box:

Evaluate

```
irl
```

Conditions

```
load = 192
```

The answer is `vth` = **28.8** V and `req` = **96** Ω, and `irl` in the 192 Ω load is **.1** A. This is correct.

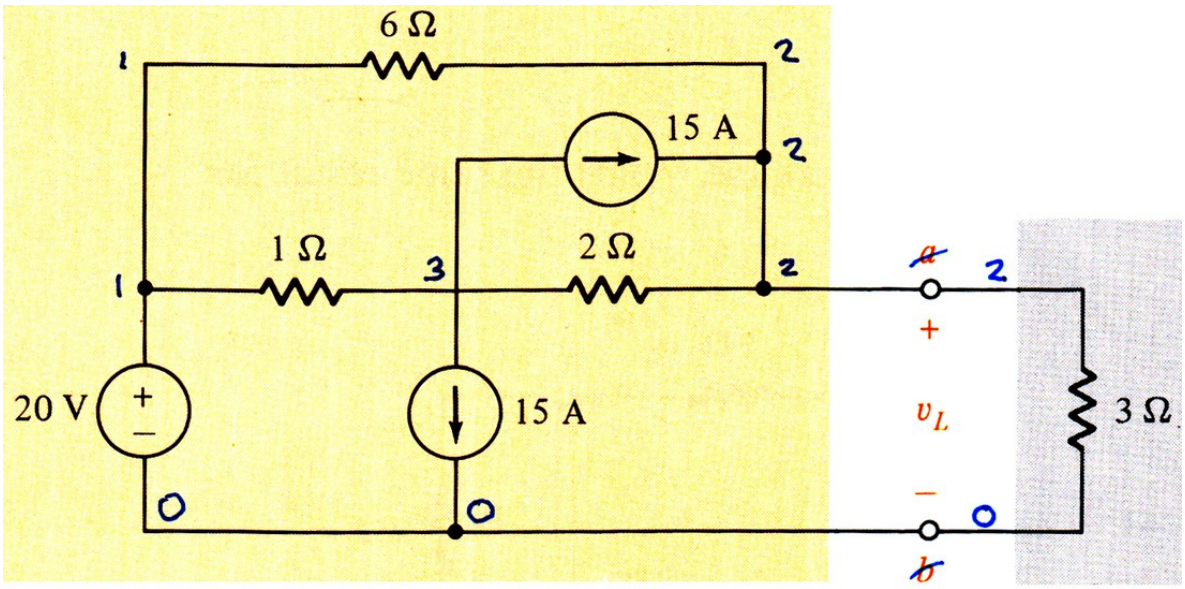
Bo2's Example 3.7

Find the Thévenin equivalent for the circuit left of a-b. Then find the voltage across the 3 Ω resistor, and also if it was 6 Ω. My answer:

Circuit Description

```
e,1,0,20
r6,1,2,6
r1,1,3,1
r2,3,2,2
j32,3,2,15
j30,3,0,15
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second — the pair of terminals you are



looking into. Tick the question about the load, choose *DC* and run it.

The answers you want are `vth` and `req`, in **Results**, and under them the load answers.

Ask **Evaluate** for `vrl`, with the load in the **Conditions** box:

> Evaluate
> `vrl`

> Conditions
> `load = 3`

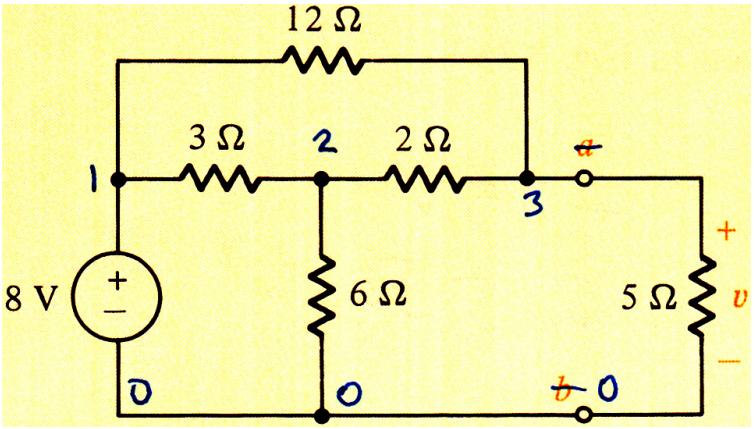
Then `load = 6`..

The answer is `vth` = **30** V and `req` = **2** Ω, and `vrl` is **18** V for 3 Ω and **22.5** V for 6 Ω. This is correct.

Bo2's Drill Exercise 3.7

Find the Thévenin equivalent for the circuit left of a-b. Then find the voltage *v*.

My answer:



Circuit Description

```
e,1,0,8
r3,1,2,3
r12,1,3,12
r6,2,0,6
r2,2,3,2
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **3** in the first and **0** in the second — the pair of terminals you are looking into. Tick the question about the load, choose *DC* and run it.

The answers you want are `vth` and `req`, in **Results**, and under them the load answers.

Ask **Evaluate** for `vrl`, with the load in the **Conditions** box:

Evaluate

```
vrl
```

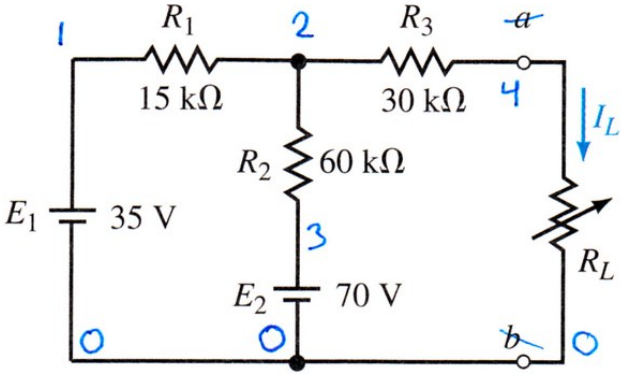
Conditions

```
load = 5.
```

The answer is `vth` = **6** V and `req` = **3** Ω, and `vrl` in the 5 Ω load is **3.75** V. This is correct.

RM3's Practice Problem 9.5

Find the Norton equivalent external to R_L in the circuit. Solve for the current I_L when $R_L = 0$, 10 kΩ, 50 kΩ, and 100 kΩ.



Answers

$R_N = 42$ kΩ, $I_N = 1.00$ mA

For $R_L = 0$: $I_L = 1.00$ mA

For $R_L = 10$ kΩ: $I_L = 0.808$ mA

For $R_L = 50$ kΩ: $I_L = 0.457$ mA

For $R_L = 100$ kΩ: $I_L = 0.296$ mA

My solution:

Circuit Description

```
e1,1,0,35.
r1,1,2,15'k
r2,2,3,60'k
e2,3,0,70
r3,2,4,30'k
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **4** in the first and **0** in the second — the pair of terminals you are looking into. Tick the question about the load, choose *DC* and run it.

The answers you want are `ino` and `req`, in **Results**, and under them the load answers.

Ask **Evaluate** for `irl`, with the load in the **Conditions** box:

> Evaluate
> `irl`

> Conditions
> `load = 0`

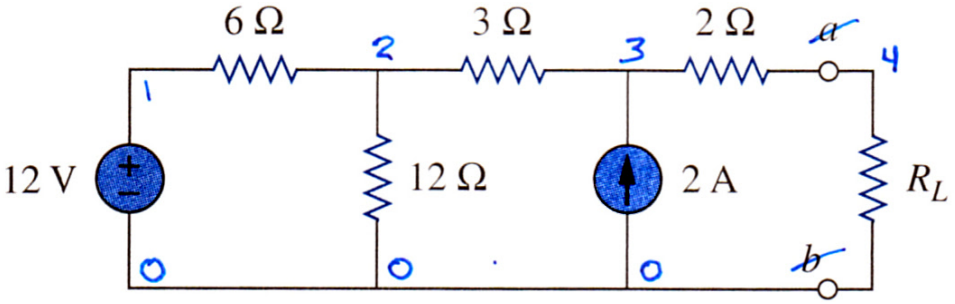
Then `load = 1e4`, then `load = 5e4`, then `load = 1e5`.

The answer is `ino` = **.001** A and `req` = **42000** Ω, and `irl` is **.001** A for 0 Ω, **.000808** A for 10 kΩ, **.000457** A for 50 kΩ and **.000296** A for 100 kΩ. This is correct.

Power transfer problems

AS2's Example 4.13

Find the R_L value for maximum power transfer and the maximum power transferred.



R_L for maximum transfer is `req`. The maximum power is in `pmax`. My solution:

> Circuit Description
> ```
> e,1,0,12
> r6,1,2,6
> r12,2,0,12
> r3,2,3,3
> j,0,3,2
> r2,3,4,2
> ```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **4** in the first and **0** in the second — the pair of terminals you are looking into.

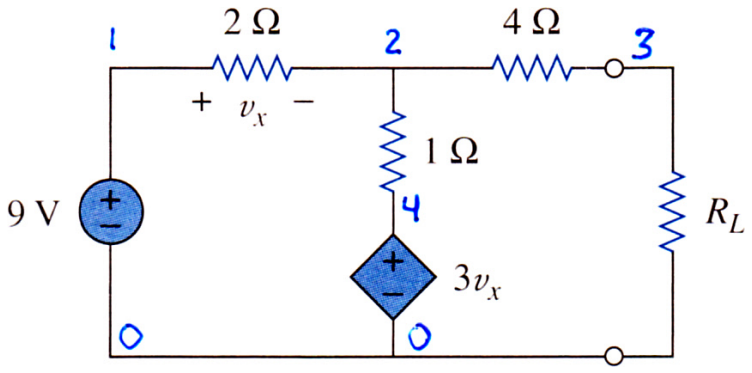
The answers you want are `req` and `pmax`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

Choose DC. The maximum transfer of power occurs when the load is `req` = **9**Ω, and the power transferred is then `pmax` = **13.44**W. Now let's solve another one.

AS2's Practice Problem 4.13

Find the R_L value for maximum power transfer and the maximum power transferred.



Circuit Description

```
ei,1,0,9
rx,1,2,2
r1,2,4,1
ed,4,0,3*vrx
r4,2,3,4
```

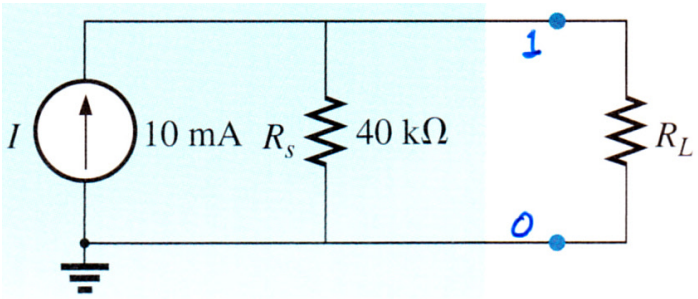
Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **3** in the first and **0** in the second — the pair of terminals you are looking into.

The answers you want are `req` and `pmax`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

Choose DC. The logic of this problem is identical to the previous one. The answer is `req` = **4.22** Ω and `pmax` = **2.901** W.

B11's Example 9.15



My solution to all three questions is presented below.

Circuit Description

```
j,0,1,10'm
rs,1,0,40'k
```

a. Find the load resistance that will result in maximum power transfer to the load, and find the maximum power delivered.
b. If the load were changed to 68 kΩ, would you expect a fairly high level of power transfer to the load based on the results of part (a)? What would the new power level be? Is your initial assumption verified?
c. If the load were changed to 8.2 kΩ, would you expect a fairly high level of power transfer to the load based on the results of part (a)? What would the new power level be? Is your initial assumption verified?

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **1** in the first and **0** in the second — the pair of terminals you are looking into. Tick the question about the load, choose *DC* and run it.

The answers you want are `req` and `pmax`, in **Results**, and under them the load answers.

Ask **Evaluate** for `prl`, with the load in the **Conditions** box:

Evaluate
```
prl
```

Conditions
```
load = 68000.
```

Then `load = 8200.`.

The answers we obtain, `req` = **40000** Ω and `pmax` = **1** W, then `prl` = **.93** W and **.57** W, are correct. Let's deconstruct it.

Part (a) is answered by the first two values: a 40 kΩ resistor as load would receive 1W power. Since this is the maximum – this is the most that any load could receive ever.

Part (b) is answered by the third value. Making use of `prl`, the power delivered by the equivalent to the load as a function of `load`, we find that a load of 68 kΩ receives .93W, which is less than the maximum.

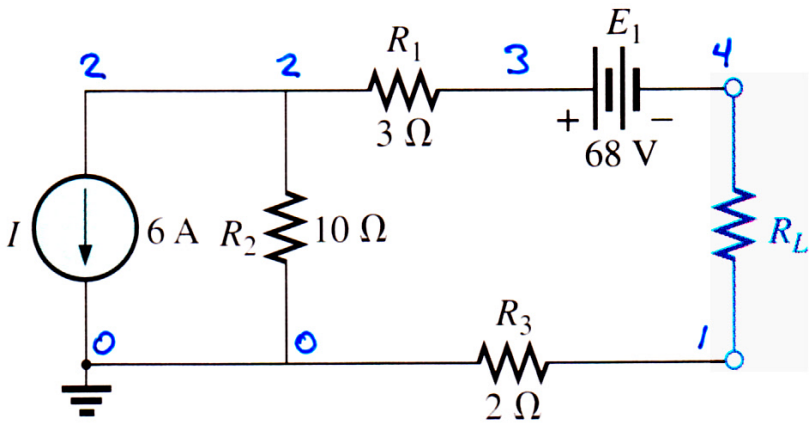
Part (c) is answered in similar manner by the fourth value. A load of 8.2 kΩ receives .57W, which is less than the maximum. Any resistance other than 40 kΩ gets less power.

B11's Example 9.17

Find the R_L value for maximum power transfer and the maximum power transferred.

Circuit Description
```
j,2,0,6
r2,2,0,10
r1,2,3,3
r3,0,1,2
e,3,4,68
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two



node boxes appear: put **4** in the first and **1** in the second — the pair of terminals you are looking into.

The answers you want are `req` and `pmax`, in **Results**.

Here we use **Rounding** — *approx to n digits*, with **n** = 3.

Choose DC. The answer is `req` = **15** Ω and `pmax` = **273.07** W. Let's now see one that is a little different.

LESSON 5

Operational amplifiers

Learn how to describe an ideal *operational amplifier* (Op Amp) using the **o** element. And practice what you learn in a variety of solved op-amp problems, both easy and hard.

In this lesson, you will learn how to describe an ideal *operational amplifier* — also called an Op Amp — using the **o** element. And you will practise what you learn in a variety of solved op-amp problems, both easy and hard.

5.1 How to describe an ideal operational amplifier

There are many types of operational amplifiers. Symbulator can simulate the ideal, linear type.

> ⓘ NOTE
> **Describing op amps**
> In Symbulator, an ideal op amp is described using four bits of information, separated by commas: first, the name of the op amp, which must start with the letter **o**; then, the names of the two input nodes, separated by a comma (the polarity doesn't matter, you can give these input nodes in any order); and finally, the name of the output node.
> For example, an ideal op amp called **o1**, with input nodes **2** and **3** and output node **5**, is described `o1,2,3,5`. And one called **o**, with input nodes **p** and **n** and output node **o**, is `o,p,n,o`

What answers do you get

For each ideal op amp in a circuit, Symbulator gives the following answers:

- The current through the output node, flowing from the output node outwards. For an op amp called **o**, that is `io`.
- The power consumed by the visible part of the op amp. For an op amp called o2, `po2`; the power delivered is the negative of that.
- And, as with every element's nodes, the voltage of each of its nodes with reference to ground. For node **o**, `vo`.

Livin' on the edge

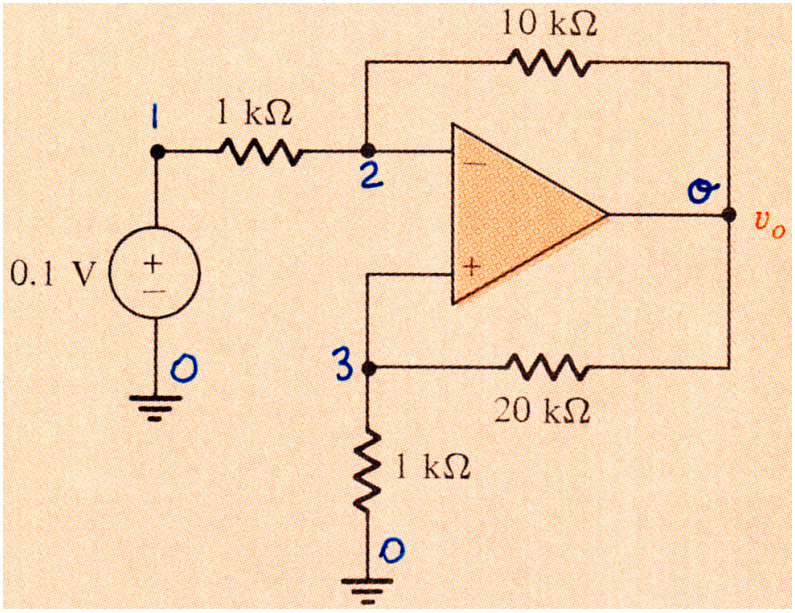
Since I was born to be bad, I like to play with node names to get the answers as close to the book as possible. So do not be surprised if I call one op amp "o" and name its output node "o" as well. That way I can ask for `vo` and `io` and get the output voltage and current. But realise this: `vo` asks for the voltage in node **o**, while in `io` the o stands for the element **o**.

In Symbulator 9 the two cannot collide: **Results** lists node voltages and element quantities separately, so a node called **o** and an op amp called **o** never share a name.

> CAUTION
>
> **Never describe a source as e,#,0,v#**
>
> If you describe a source as `e1,1,0,v1`, you are declaring the voltage of node 1 to be the voltage of node **1**. The resulting equation, v1 = v1, is trivially true and carries no information, which leaves the system one equation short.

Bo2's Drill Exercise 3.2

For the op-amp circuit shown, find vo and the power absorbed by the op amp.



Bo2's Drill Exercise 3.2

SOLUTION

My solution:

Circuit Description

```
e,1,0,.1
r12,1,2,1'k
r2o,2,o,10'k
r30,3,0,1'k
r3o,3,o,20'k
o,3,2,o
```

Read vo and po in **Results**:

- vo = **-2.1** V
- po = **-0.00063** W

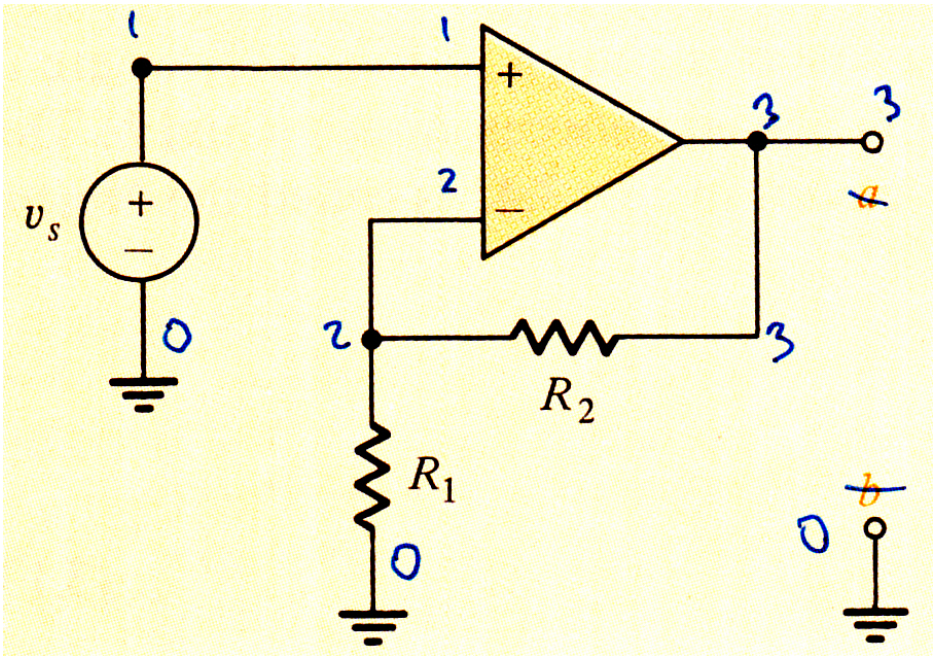
Which are correct. The voltage in node **o** is –2.1 V, and the op amp is absorbing –.63 mW, which is to say it is delivering .63 mW of power to the circuit.

5.2 Instructive Op Amp examples

Solved Op Amp problems

Bo2's Drill Exercise 3.11 (Thévenin)

Find the Thévenin equivalent.



Circuit Description

```
e,1,0,vs
r1,2,0,r1
r2,2,3,r2
o,1,2,3
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **3** in the first and **0** in the second — the pair of terminals you are looking into.

This one catches people out, and Symbulator says so:

Thévenin voltage

$$v_{th} = \frac{vs\,(r_1 + r_2)}{r_1}$$

Norton current

$$i_{no} = \infty$$

equivalent resistance

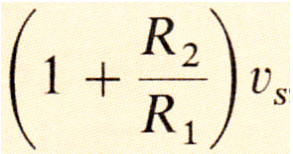
$$R_{eq} = 0$$

maximum deliverable power

$$p_{max} = \infty$$

with a note above the answers: the short-circuit current is unbounded, so the equivalent is a voltage source with nothing in series.

That is not a failure to solve. An ideal op amp holds its output voltage whatever current is drawn, so a short across its output carries an unbounded current, which is exactly a source with no resistance in series: R_{EQ} = **0** Ω. Symbulator finds it by putting a resistance across the terminals and letting it fall to zero. V_{TH} matches the book's answer.



$$\left(1 + \frac{R_2}{R_1}\right) v_s$$

AS2's Example 5.2

Find v_o and i_o.

Circuit Description

```
e,2,0,1.
r5,1,0,5'k
r4,1,o,40'k
r2,o,0,20'k
o,2,1,o
```

The answers you want are `vo` and `io`, in **Results**.

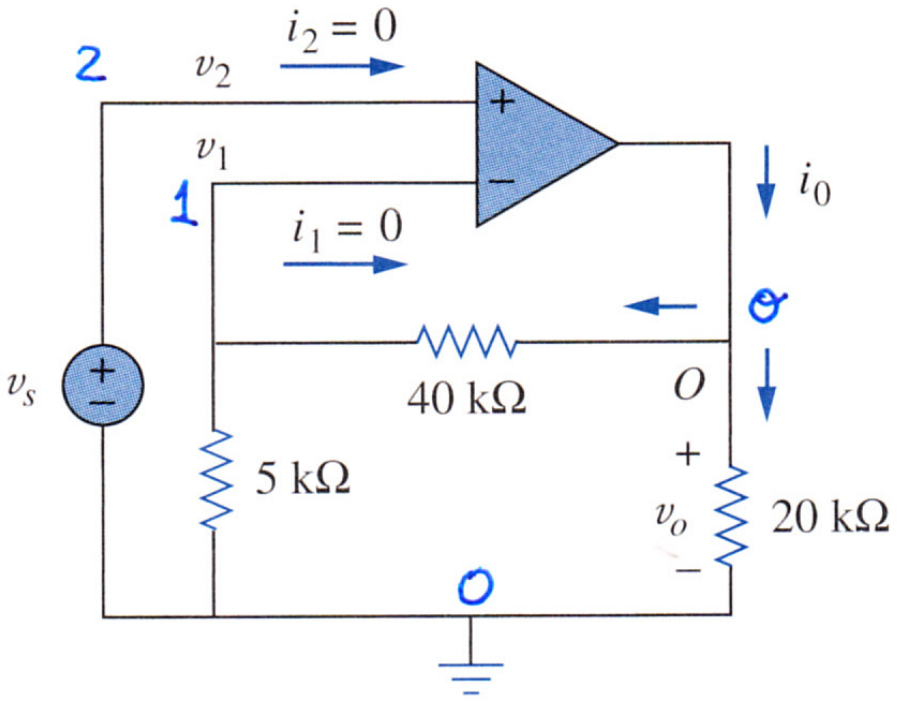
The answer is v_o = **9**V and i_o = **0.65**mA. This is correct.

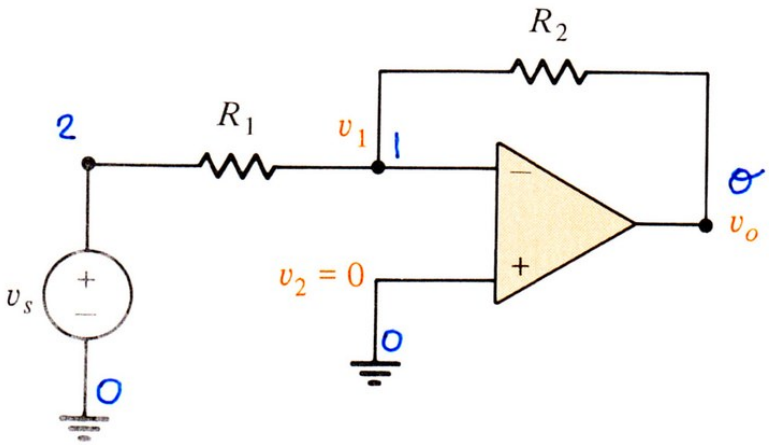
Bo2's Figure 3.3 (Inverting)

Find the gain of the overall circuit, v_o/v_S.

Circuit Description

```
e,2,0,vs
```





$$\frac{v_o}{v_s} = -\frac{R_2}{R_1}$$

```
r1,2,1,r1
r2,1,o,r2
o,1,0,o
```

Ask **Evaluate** for:

```
Evaluate
vo/vs
```

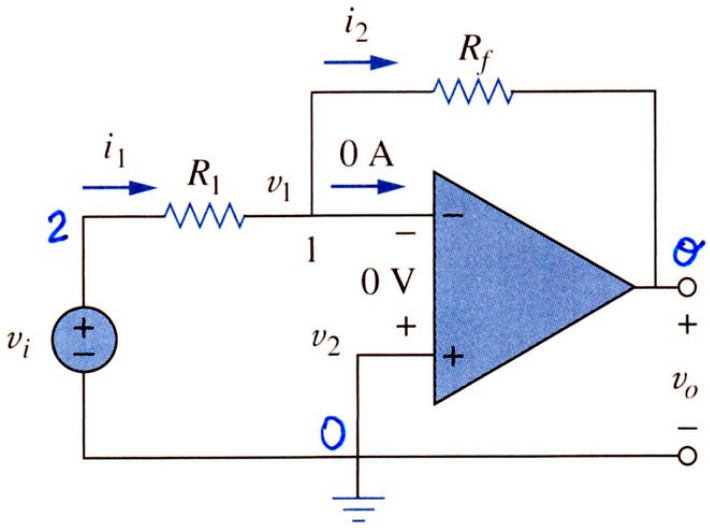
We get

$$\frac{v_o}{vs} = -\frac{r_2}{r_1}$$

which is correct, as can be seen in the book's answer above.

AS2's Figure 5.10 (Inverting)

Find v_o.



$$v_o = -\frac{R_f}{R_1} v_i$$

Circuit Description

```
e,2,0,vi
r1,2,1,r1
rf,1,o,rf
o,0,1,o
```

The answer you want is vo, in **Results**.

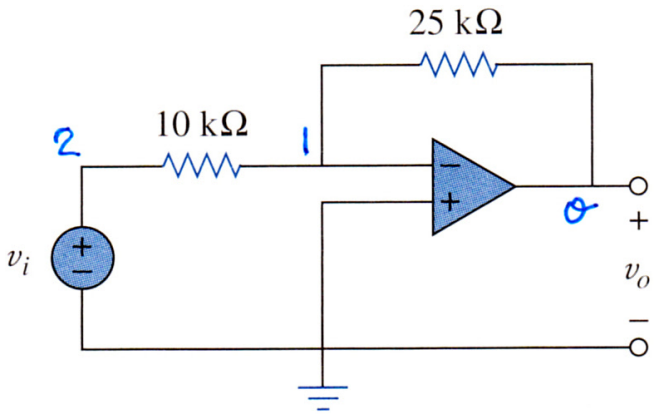
This problem is almost identical to the one above. The answer we get is correct:

voltage of node o

$$v_o = -\frac{rf}{r_1} vi$$

AS2's Example 5.3 (Inverting)

If v_i is 0.5V, calculate the output voltage v_o and the current in the 10 kΩ resistor.



Circuit Description

```
e,2,0,.5
r1,2,1,10'k
rf,1,o,25'k
o,0,1,o
```

The answers you want are in **Results**:

voltage of node o

$$v_o = -1.25 \text{ V}$$

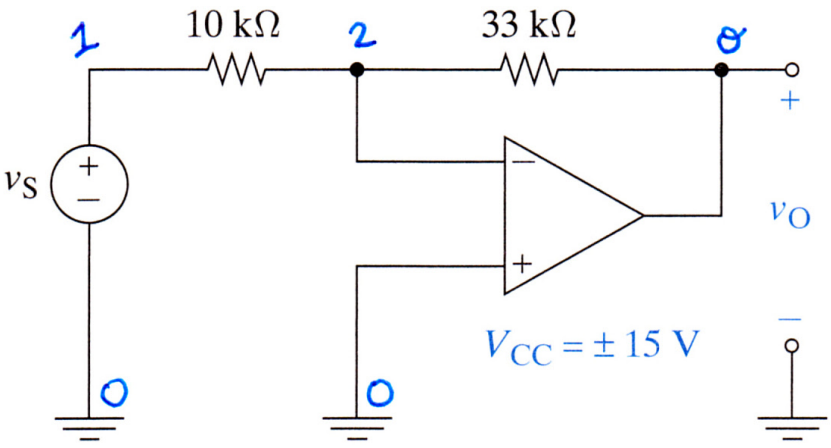
current through r1

$$i_{r1} = 5e - 5 \text{ A}$$

This is correct.

TR5's Exercise 4-11 (Inverting)

Find v_O when v_S is 2V, -4V and 6V. Notice the output of the op amp is limited to ±15V.



This may be the only non-linear problem you will see in this book, because I solved it before I realized it included the ±15V constraint. But anyway, here it goes.

Circuit Description

```
e,1,0,vs
r1,1,2,10'k
r2,2,o,33'k
o,0,2,o
```

Put vo in **Evaluate** and the given value in its **Conditions** box:

Evaluate

```
vo
```

Conditions

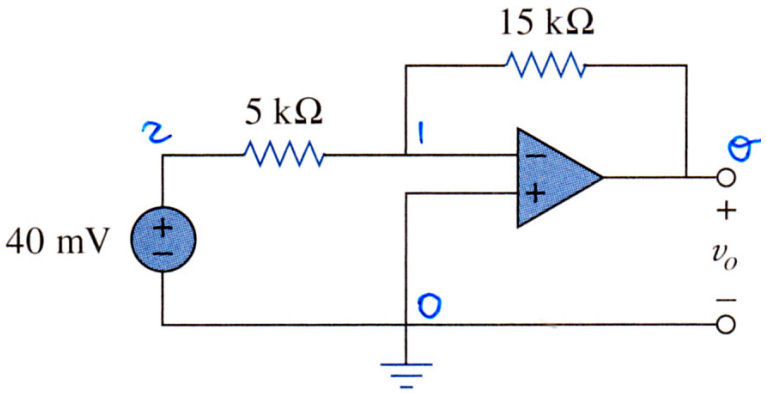
```
vs = 2
```

Then run it again with `vs = -4` and once more with `vs = 6`.

The answers, `vo` = **-6.6** V, **13.2** V and **-19.8** V, are correct within the linear realm, but since the output is constrained to no more than 15V or less than -15V, the answer is v_O=**-15**V for v_S = **6**V.

AS2's Practice Problem 5.3 (Inverting)

Find the output voltage of the op amp (i.e. v_o) and calculate the current through the feedback resistor (i.e. the 15 kΩ resistor).



My answer: Notice we used the m for milli in the value of the voltage source.

Circuit Description

```
e,2,0,40'm
r1,2,1,5'k
rf,1,o,15'k
o,0,1,o
```

The answers are:

voltage of node o

$$v_o = -.12 \text{ V}$$

current through rf

$$i_{rf} = 8e - 6 \text{ A}$$

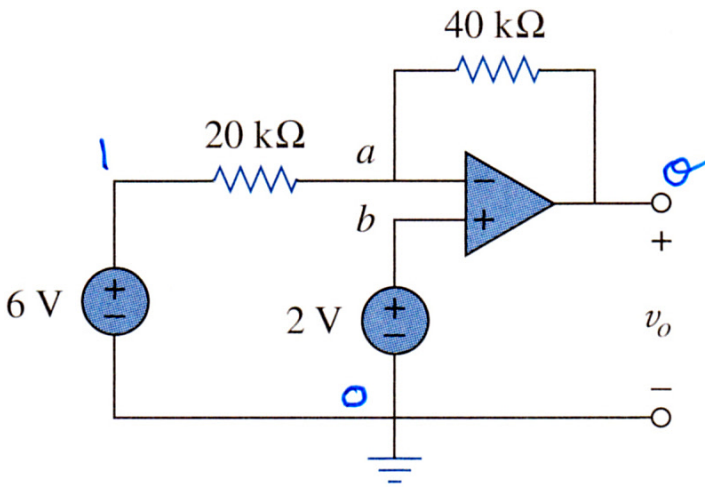
This is correct.

AS2's Example 5.4 (Inverting)

Determine v_o.

Circuit Description

```
e6,1,0,6
r1,1,a,20'k
e2,b,0,2
rf,a,o,40'k
```



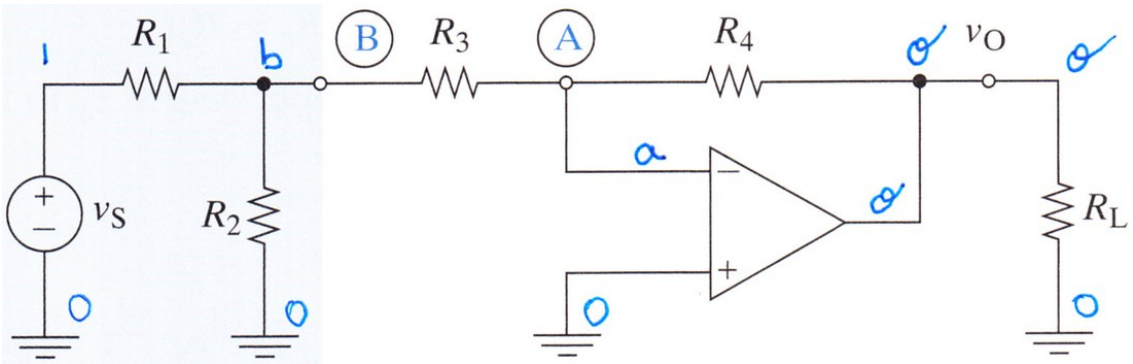

```
o,a,b,o
```

The answer you want is `vo`, in **Results**.

The answer, **-6**, is correct.

TR5's Example 4-14 (Inverting)

Find the input-output relationship of the circuit.



The 'formal' way to *symbulate* this circuit would be as described below.

Circuit Description

```
e,1,0,vs
r1,1,b,r1
r2,b,0,r2
r3,b,a,r3
r4,a,o,r4
o,0,a,o
rl,o,0,rl
```

However, since we are only interested in the ratio of the input to the output, a smarter way to *symbulate* it is this:

Circuit Description

```
e,1,0,1
r1,1,b,r1
r2,b,0,r2
r3,b,a,r3
r4,a,o,r4
o,0,a,o
rl,o,0,1
```

Ask **Evaluate** for `vo/v1` and both approaches give the same answer:

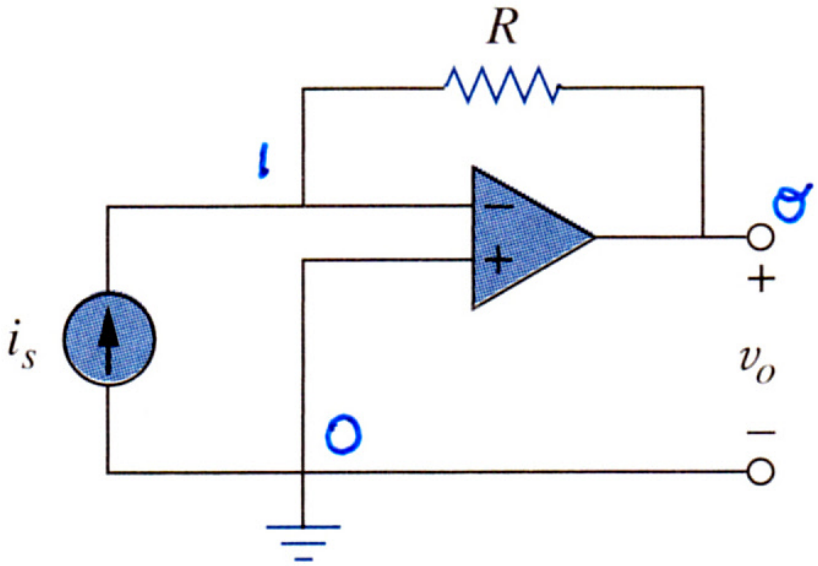
$$\frac{v_o}{v_1} = -\frac{r_2\, r_4}{r_1\, r_2 + r_1\, r_3 + r_2\, r_3}$$

which is correct, as can be seen by comparing it to the book's answer.

$$-\frac{R_2 R_4}{R_1 R_2 + R_1 R_3 + R_2 R_3}$$

AS2's Practice Problem 5.4a (Transresistance)

This is a current-to-voltage converter, also called a *transresistance amplifier*. Find $\boldsymbol{v_o/i_S}$.



Circuit Description

```
j,0,1,is1
r,1,o,r
o,0,1,o
```

Ask **Evaluate** for:

Evaluate

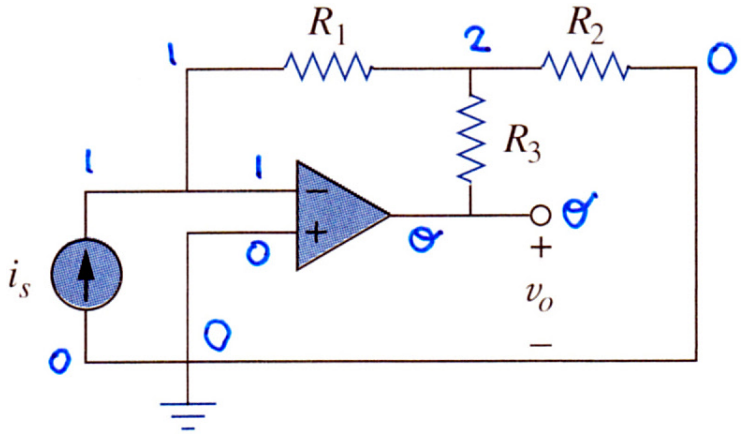
```
vo/is1
```

The answer is correct:

$$\frac{v_o}{is_1} = -r$$

AS2's Practice Problem 5.4b (Transresistance)

This is another *transresistance amplifier*. Again, find v_o/i_S.



Circuit Description

```
j,0,1,is1
r1,1,2,r1
r2,2,0,r2
r3,2,o,r3
o,0,1,o
```

Ask **Evaluate** for `vo/is1`. It gives

$$\frac{v_o}{is_1} = -r_1 - \frac{r_1\, r_3}{r_2} - r_3$$

which is equivalent to the book's answer — the same expression, gathered differently.

$$\frac{v_o}{i_s} = -R_1\left(1 + \frac{R_3}{R_1} + \frac{R_3}{R_2}\right)$$

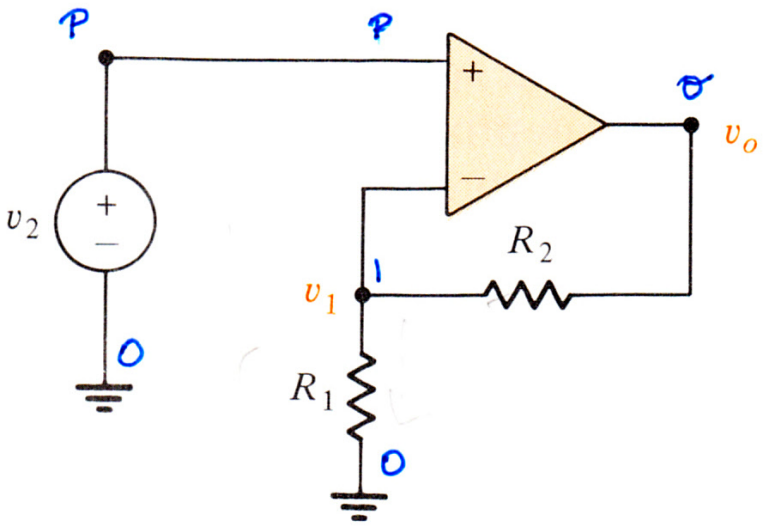
Bo2's Example 3.1 (Non-Inverting Amplifier)

Find v_o/v_1.

Circuit Description

```
e,p,0,v2
r1,1,0,r1
r2,1,o,r2
o,p,1,o
```

Ask **Evaluate** for:



Evaluate

```
vo/v1
```

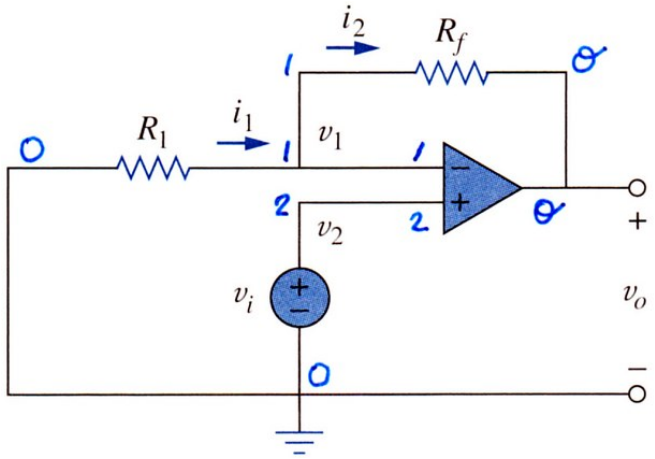
We get

$$\frac{v_o}{v_1} = 1 + \frac{r_2}{r_1}$$

which is correct.

AS2's Figure 5.16 (Non-Inverting Amplifier)

Find v_o.



$$v_o = \left(1 + \frac{R_f}{R_1}\right) v_i$$

Circuit Description

```
e,2,0,vi
r1,0,1,r1
rf,1,o,rf
o,2,1,o
```

The answer you want is `vo`, in **Results**.

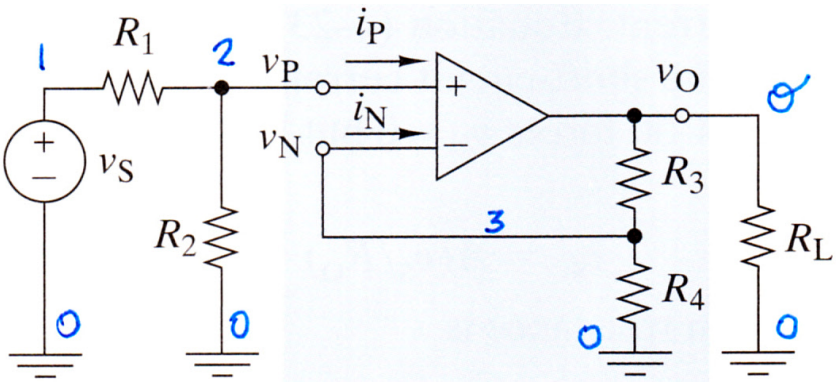
We get the right answer (below), an expression equivalent to the book's answer.

voltage of node o

$$v_o = \left(\frac{r_1 + rf}{r_1}\right) vi$$

TR5's Example 4-13 (Non-Inverting Amplifier)

Find V_o/V_s.



We can solve this problem in one simulation, as shown below.

Circuit Description

```
e,1,0,vs
r1,1,2,r1
r2,2,0,r2
o,2,3,o
r3,o,3,r3
r4,3,0,r4
```

Ask **Evaluate** for:

Evaluate

```
vo/vs
```

We get

$$\frac{v_o}{v_s} = \frac{r_2\,(r_3 + r_4)}{(r_1 + r_2)\,r_4}$$

which is correct, as can be seen by comparing it to the book's answer.

$$\left[\frac{R_2}{R_1 + R_2}\right]\left[\frac{R_3 + R_4}{R_4}\right]$$

We can also solve it in stages, as shown below. First, simulate the left half.

Circuit Description

```
e,1,0,vs
r1,1,2,r1
r2,2,0,r2
```

Ask **Evaluate** for:

Evaluate

```
v2/vs
```

We get

$$\frac{v_2}{v_s} = \frac{r_2}{r_1 + r_2}$$

which is correct for this part. Then, simulate the right half.

Circuit Description

```
e,2,0,1
o,2,3,o
r3,o,3,r3
r4,3,0,r4
```

Ask **Evaluate** for:

Evaluate

```
vo/v2
```

We get

$$\frac{v_o}{v_2} = \frac{r_3 + r_4}{r_4}$$

which is correct for this part. The product of these two partial answers produces the same expression shown above after the big simulation, and is the right answer.

AS2's Figure 5.17 (Voltage Follower)

Find v_o.

Circuit Description

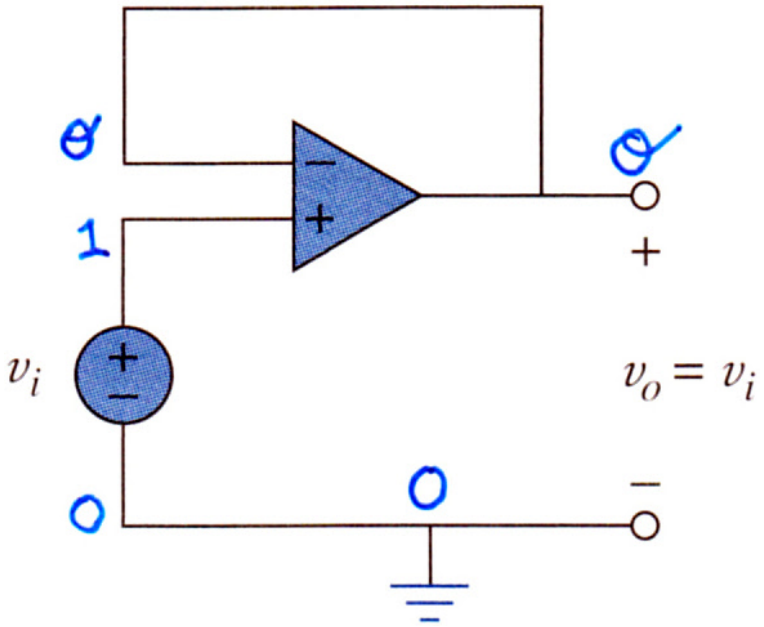
```
e,1,0,vi
o,1,o,o
```

The answer you want is `vo`, in **Results**.

The answer is correct:

voltage of node o

$$v_o = vi$$



TR5's Figure 4-32 (Voltage Follower)

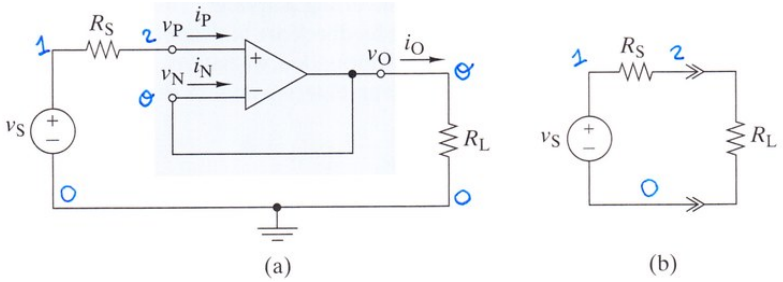


FIGURE 4–32 *(a) Source-load interface with a voltage follower. (b) Interface without the voltage follower.*

The circuits in Figure 4–32 have $v_S = 1.5$ V, $R_S = 2$ kΩ, and $R_L = 1$ kΩ. Compute the maximum power available from the source. Compute the power absorbed by the load resistor in the direct connection in Figure 4–32(b) and in the voltage follower circuit in Figure 4–32(a). Discuss any differences.

Answers: 281 μW; 250 μW; 2250 μW

Alright, so first we simulate the (b) circuit, to find its maximum power and load power:

Circuit Description

```
e,1,0,1.5
rs,1,2,2'k
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Thévenin / Norton*. Two node boxes appear: put **2** in the first and **0** in the second — the pair of terminals you are looking into. Tick the question about the load, choose *DC* and run it.

The answer you want is `pmax`, in **Results**, and under it `prl`, the power in the load as Lesson 4 (page 80) describes. Ask **Evaluate** for it with the load in the **Conditions** box:

Evaluate

```
prl
```

Conditions

```
load = 1000
```

The answers we get, `pmax` = **2.8125e-4** W and `prl` = **2.5e-4** W in the 1 kΩ load, are correct. Now we simulate the (a) circuit.

Circuit Description

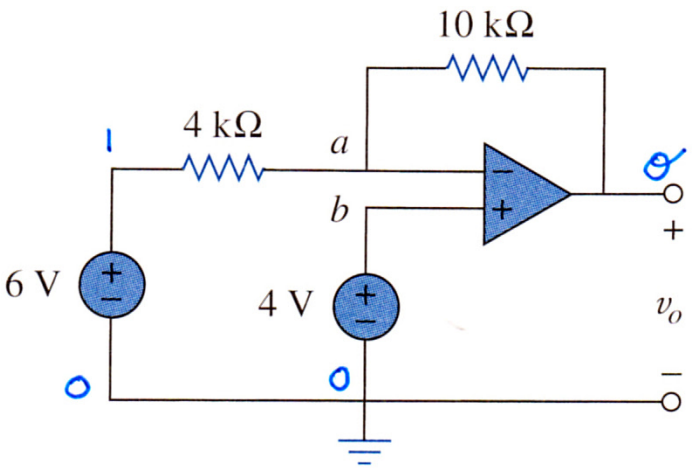
```
e,1,0,1.5
rs,1,2,2'k
o,2,o,o
rl,o,0,1'k
```

The answer you want is `prl`, in **Results** — the power consumed by the load resistor **rl**.

The answer, **.00225**, is correct. The apparent paradox — the load in (a) drawing more power than the source in (b) seems able to provide — evaporates once we remember that the ideal op amp in the schematic is only part of the truth: the real one has its own source of power, which makes up the difference.

AS2's Example 5.5 (Inverting)

Find v_o.



Circuit Description

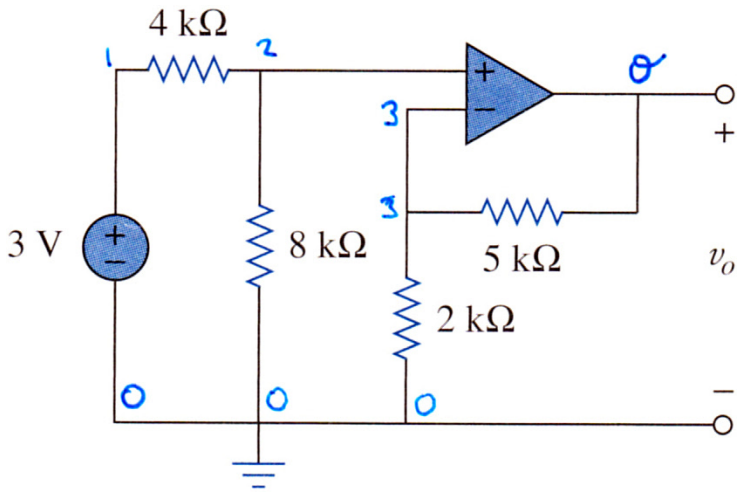
```
e1,1,0,6
r4,1,a,4'k
r10,a,o,10'k
e4,b,0,4
o,b,a,o
```

The answer you want is `vo`, in **Results**.

The answer, **-1**, is correct.

AS2's Practice Problem 5.5 (Non-Inverting)

Calculate v_o.



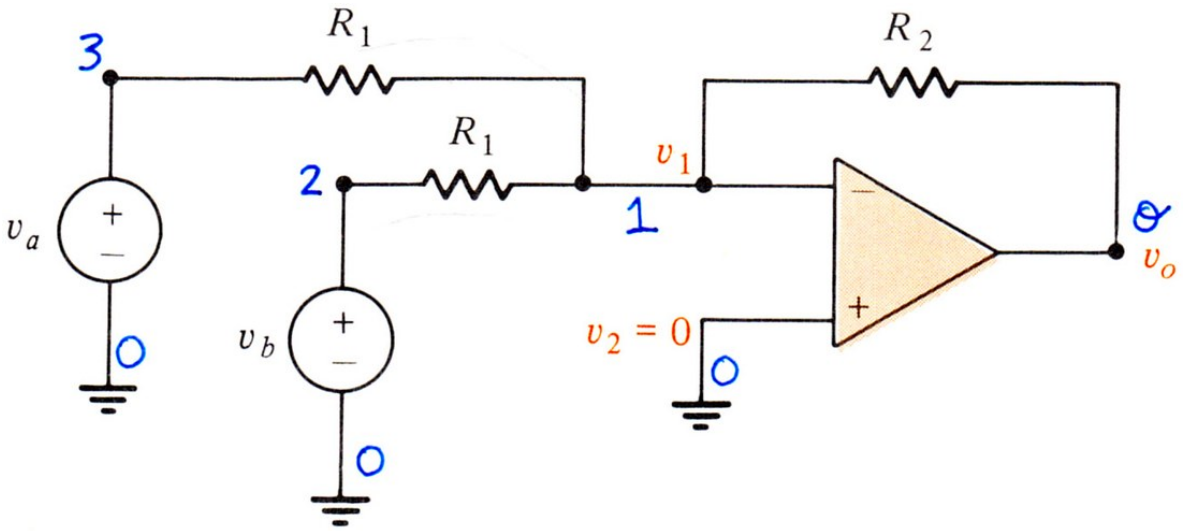
Circuit Description

```
e,1,0,3
r4,1,2,4'k
r8,2,0,8'k
r2,3,0,2'k
r5,3,o,5'k
o,2,3,o
```

The answer for vo, 7, is correct.

Bo2's Example 3.2 (Adder or Summing)

Find v_o.



Circuit Description

```
ea,3,0,va
eb,2,0,vb
r31,3,1,r1
r21,2,1,r1
r1o,1,o,r2
o,0,1,o
```

The answer you want is vo, in **Results**:

voltage of node o

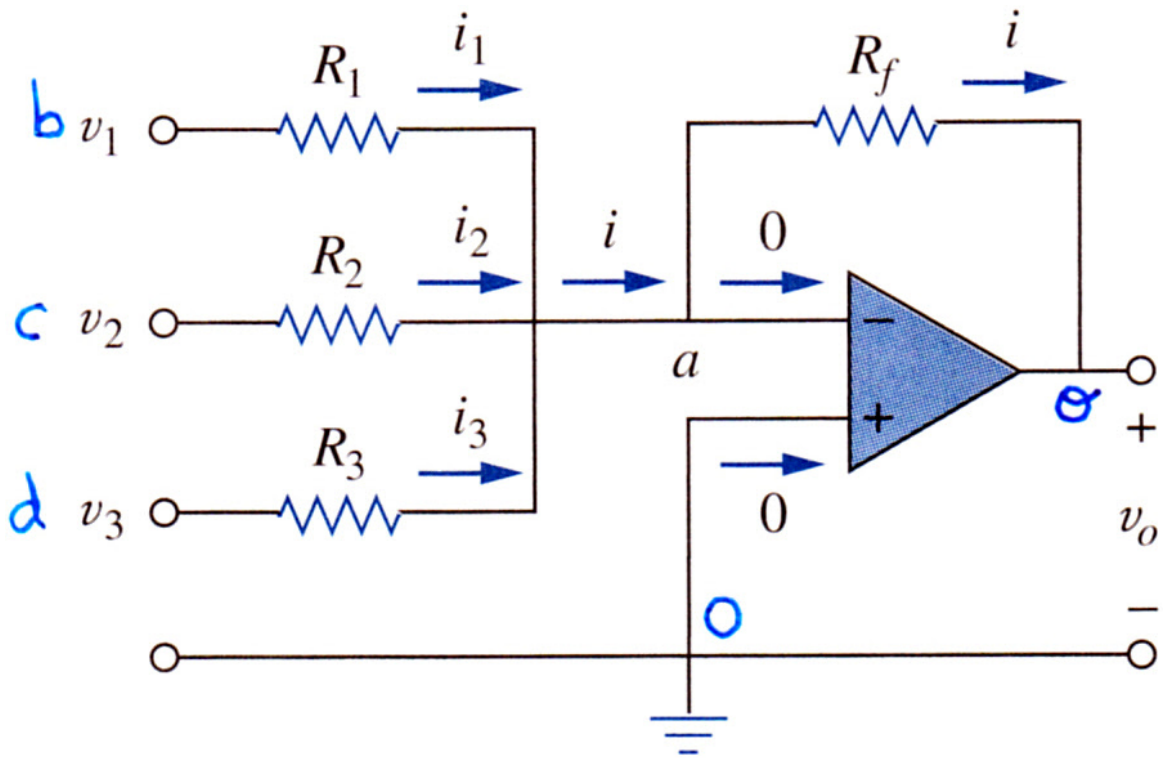
$$v_o = -\frac{r_2}{r_1}(va + vb)$$

which is correct, as can be seen by comparing it to the book's answer.

$$v_o = -\frac{R_2}{R_1}(v_a + v_b)$$

AS2's Figure 5.21 (Adder or Summing)

Find v_o.



Circuit Description

```
e1,b,0,v1
e2,c,0,v2
e3,d,0,v3
```

```
r1,b,a,r1
r2,c,a,r2
r3,d,a,r3
rf,a,o,rf
o,0,a,o
```

Ask **Evaluate** for `expand(vo)`. It gives

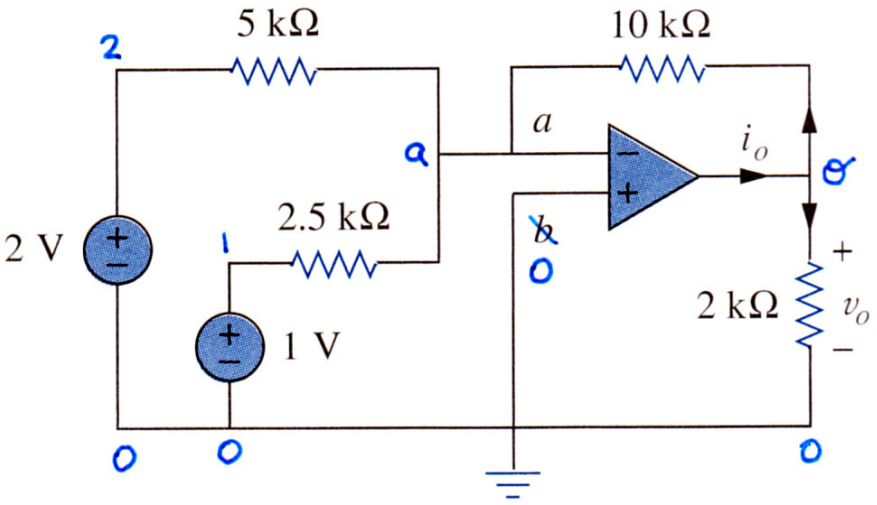
$$v_o = -\frac{rf\, v_1}{r_1} - \frac{rf\, v_2}{r_2} - \frac{rf\, v_3}{r_3}$$

which is correct, as can be seen by comparing it to the book's answer.

$$v_o = -\left(\frac{R_f}{R_1}v_1 + \frac{R_f}{R_2}v_2 + \frac{R_f}{R_3}v_3\right)$$

AS2's Example 5.6 (Adder or Summing)

Find v_o and i_o.




```
Circuit Description
e1,1,0,1
e2,2,0,2
r1,2,a,5'k
r2,1,a,2.5'k
r3,a,o,10'k
r4,o,0,2'k
o,0,a,o
```

The answers you want are in **Results**:

```
voltage of node o
```

$$v_o = -8\ \text{V}$$

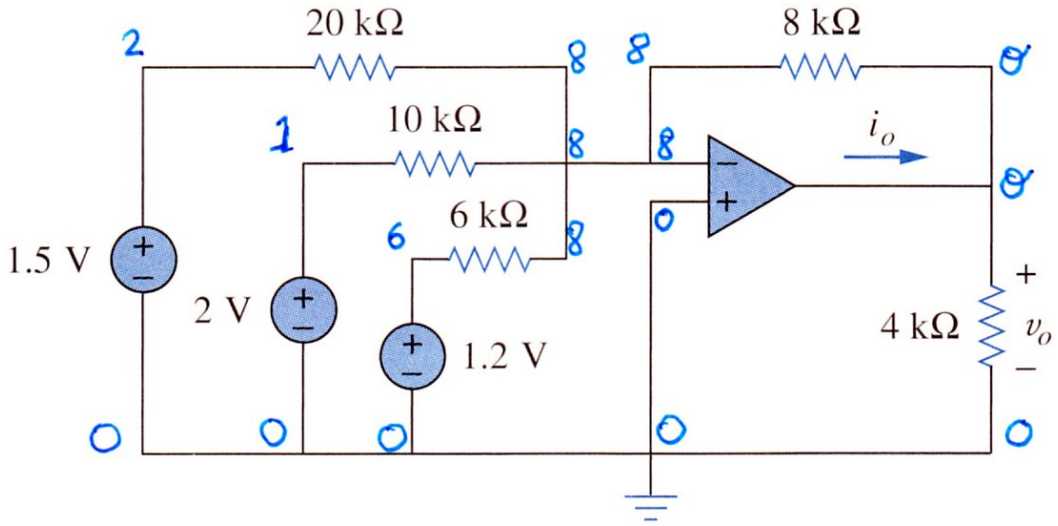
current through o

$$i_o = -.0048 \text{ A}$$

This is correct. Notice a current of 4.8mA is going into the op amp.

AS2's Practice Problem 5.6 (Adder or Summing)

Find v_o and i_o.



Circuit Description

```
e2,2,0,1.5
e1,1,0,2
e6,6,0,1.2
r2,2,8,20'k
r1,1,8,10'k
r6,6,8,6'k
r8,8,o,8'k
r4,o,0,4'k
o,0,8,o
```

The answers you want are in **Results**:

voltage of node o

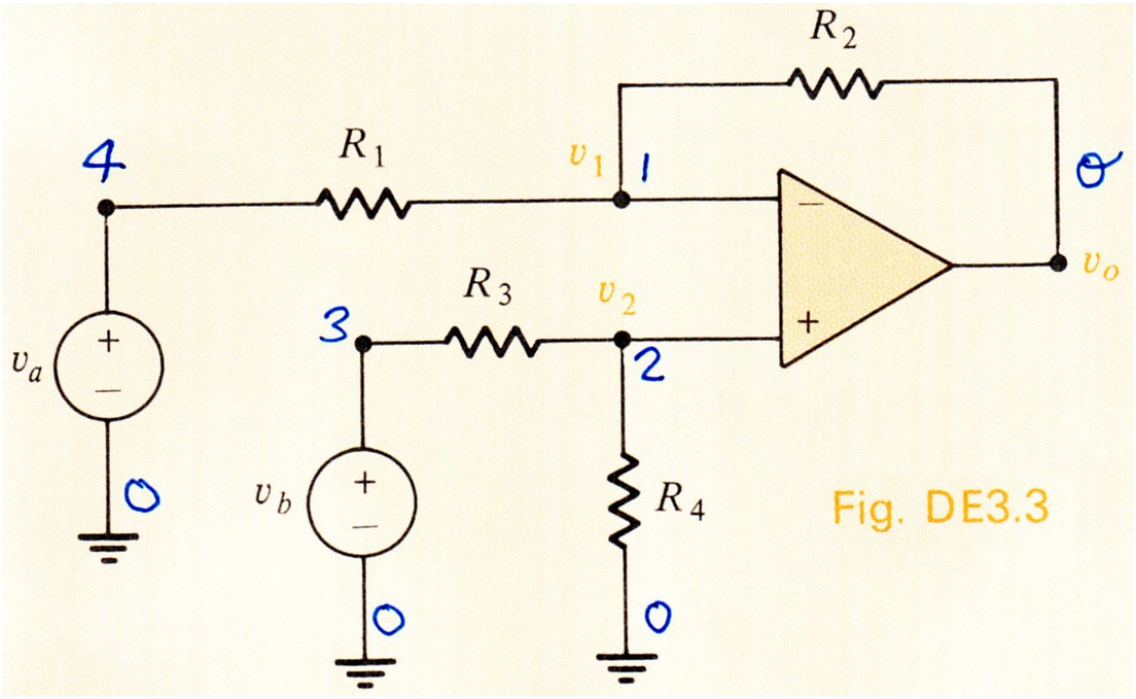
$$v_o = -3.8 \text{ V}$$

current through o

$$i_o = -.001425 \text{ A}$$

This is correct. Again, the current is going into the op amp.

Bo2's Drill Exercise 3.3 (Difference or Differential)



Find v_o for the **difference-amplifier** circuit shown in Fig. DE3.3 when $R_3 = R_1$ and $R_4 = R_2$.

Answer: $\dfrac{R_2}{R_1}(v_b - v_a)$

Circuit Description

```
ea,4,0,va
eb,3,0,vb
r1,4,1,r1
r2,1,o,r2
r3,3,2,r1
r4,2,0,r2
o,2,1,o
```

vo is the correct answer, equivalent to the book's answer above:

voltage of node o

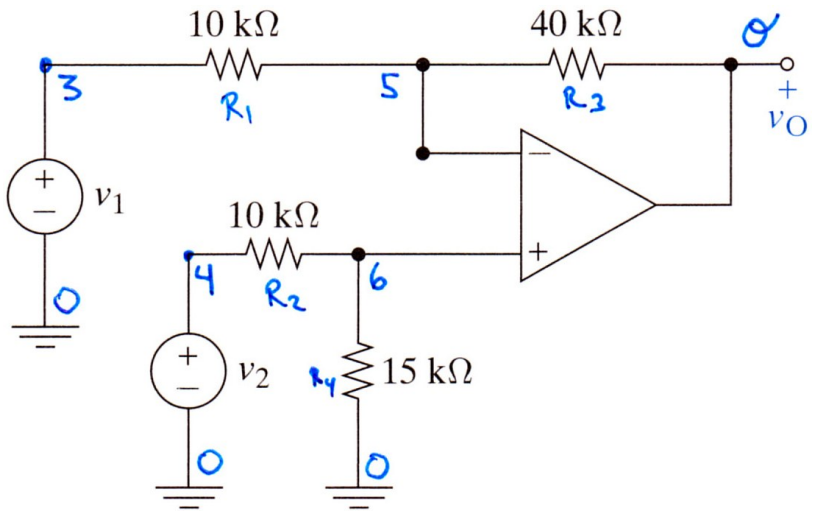
$$v_o = \frac{(vb - va)\, r_2}{r_1}$$

TR5's Exercise 4-13 (Difference or Differential)

Find $\boldsymbol{v_o}$.

Circuit Description

```
e1,3,0,v1
e2,4,0,v2
r1,3,5,10'k
```




```
r2,4,6,10'k
r3,5,o,40'k
r4,6,0,15'k
o,6,5,o
```

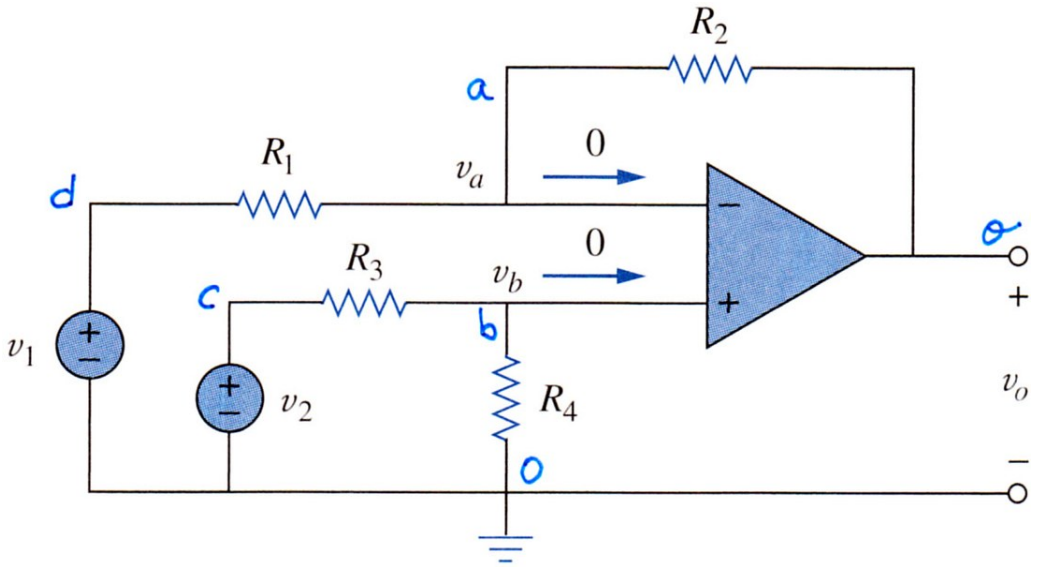
vo is the correct answer:

voltage of node o

$$v_o = 3\,v_2 - 4\,v_1$$

AS2's Figure 5.24 (Difference or Differential)

Find v_o. (And keep the answer on screen: the next three problems reuse it.)



Circuit Description

```
e1,d,0,v1
e2,c,0,v2
```

```
r1,d,a,r1
r3,c,b,r3
r2,a,o,r2
r4,b,0,r4
o,b,a,o
```

The answer you want is in **Results**:

voltage of node o

$$v_o = \frac{r_1\, r_4\, v_2 - r_2\left(r_3\, v_1 + r_4\left(v_1 - v_2\right)\right)}{r_1\left(r_3 + r_4\right)}$$

This expression is equivalent to the book's answer.

$$v_o = \frac{R_2}{R_1}\frac{(1 + R_1/R_2)}{(1 + R_3/R_4)}v_2 - \frac{R_2}{R_1}v_1$$

AS2's Figure 5.24 (Subtractor)

For the same circuit of the previous problem, find v_o when R_1=R_2 and R_3=R_4.

The expression for `vo` is still in **Results** from the previous problem, so we only do this:

Ask **Evaluate** for `expand(vo)` and give it both equalities in the **Conditions** box:

Conditions

```
r2 = r1
r3 = r4
```

The answer we get is correct:

$$v_o = v_2 - v_1$$

AS2's Example 5.7 (Difference or Differential)

Design an op amp circuit with inputs v_1 and v_2 such that v_o = -5v_1 + 3v_2.

My solution follows. The problem statement is a fancy way of saying: for the circuit of the previous problem, find the resistor values that give an output v_o = -5v_1 + 3v_2. Not strictly a Symbulator problem, but it shows how Symbulator fits into design problems.

First we take that part of v_o that is a factor of v_1, and make it equal to -5. Thus:

Conditions again, this time on the inputs rather than the resistors: `v1 = 1` and `v2 = 0`. Then `vo` is

voltage of node o

$$v_o = -\frac{r_2}{r_1}$$

which is equivalent to the book's expression, and it is that which has to equal −5.

Symbulator will not always arrange an expression the way the book does. To rearrange it, **Evaluate** takes `simplify()`, `collect()`, `expand()`, `factor()` and `apart()`.

We get

$$-\frac{r_2}{r_1} = -5$$

Now make that part of v_o that is a factor of v_2 equal to

1. Thus:

With `v1 = 0` and `v2 = 1` instead, `vo` is

voltage of node o

$$v_o = \frac{r_4\,(r_1 + r_2)}{r_1\,(r_3 + r_4)}$$

which is equivalent to the book's expression, and it is that which has to equal 3.

We get

$$\frac{r_4\,(r_1 + r_2)}{r_1\,(r_3 + r_4)} = 3$$

Now, since you have two equations, you can solve for two unknowns. Of the four resistors you get to choose, two can be whatever you want. The book recommends R_1 = 10 kΩ and R_3 = 20 kΩ. Now let's find R_2 and R_4.

Write the two equations out in the **Solve** card, which solves a system that is not a circuit:

Equation(s) to solve in terms of the results

```
-r2/r1 = -5
r4*(r1 + r2)/(r1*(r3 + r4)) = 3
```

with `r2, r4` as the unknowns. The answer is a design rule rather than a pair of numbers:

$$r_2 = 5\,r_1$$

$$r_4 = r_3$$

Put the book's **r1** = 10 kΩ and **r3** = 20 kΩ into it and you get **50** kΩ and **20** kΩ.

Now, if this problem was part of a test, I'd like to verify that the answer is correct. To confirm this, simulate the circuit using the four values given above for the resistors.

Circuit Description

```
e1,d,0,v1
e2,c,0,v2
r1,d,a,10'k
r3,c,b,20'k
```

```
r2,a,o,50'k
r4,b,0,20'k
o,b,a,o
```

vo is the desired output:

> voltage of node o
>
> $$v_o = 3\,v_2 - 5\,v_1$$

The resistor values are correct.

AS2's Practice Problem 5.7 (Difference or Differential)

Design a difference amplifier with gain 4.

My solution follows. Again the statement is a fancy way of saying: for the circuit of the previous problem, find the resistor values that give an output v_o = **4** (v_2-v_1), that is, -4v_1 + 4v_2. Same as before.

Exactly as in the previous problem: solve the circuit twice with conditions on the inputs — `v1 = 1, v2 = 0`, then `v1 = 0, v2 = 1` — and read `vo` each time. The two factors are the same as before, so the two design equations are the same shape with 4 in place of 5 and 3.

This time the book asks that you use R_1 = 10 kΩ and R_3 = 10 kΩ. So we do that.

In the **Solve** card:

> Equation(s) to solve in terms of the results
>
> ```
> -r2/r1 = -4
> r4*(r1 + r2)/(r1*(r3 + r4)) = 4
> ```

with `r2, r4` as the unknowns. It answers

$$r_2 = 4\,r_1$$

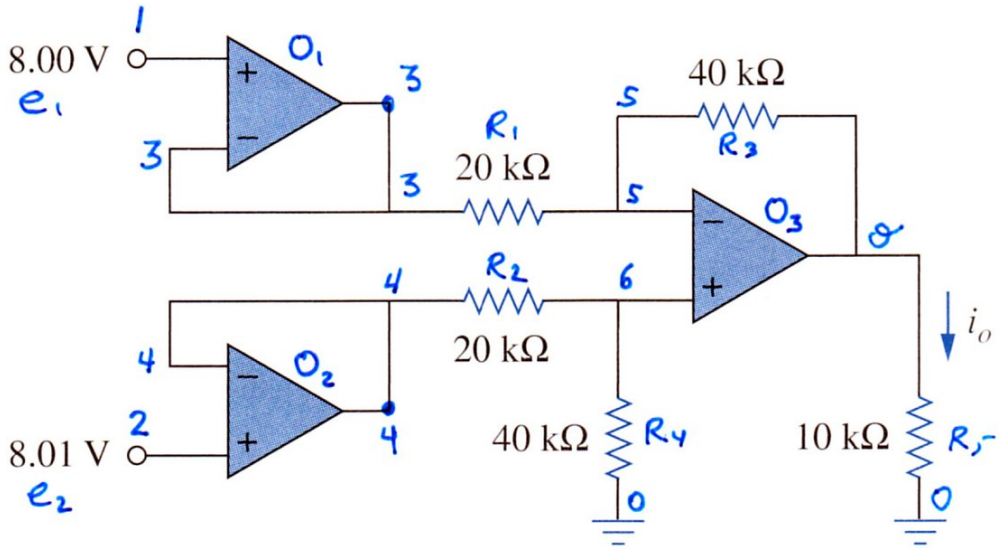
$$r_4 = 4\,r_3$$

which at 10 kΩ each is **40000** Ω and **40000** Ω.

AS2's Practice Problem 5.8 (Instrumentation)

Find i_o.

> Circuit Description
>
> ```
> e1,1,0,8.
> e2,2,0,8.01
> o1,1,3,3
> o2,2,4,4
> r1,3,5,20'k
> ```



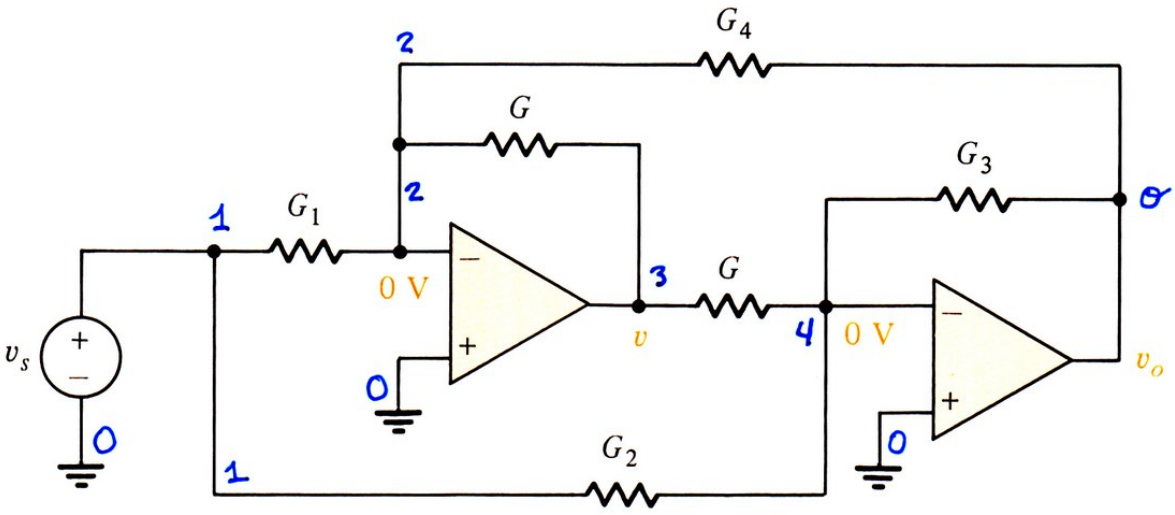

```
r2,4,6,20'k
r3,5,o,40'k
r4,6,0,40'k
o3,6,5,o
r5,o,0,10'k
```

The answer you want is `ir5`, in **Results**.

In the schematic, the current i_o corresponds to `ir5`. The answer, **2e-6**, is correct.

Bo2's Example 3.3 (Cascade)

Find v_o in terms of the conductances and the applied voltage v_S.



Circuit Description

```
e,1,0,vs
r12,1,2,1/g1
r14,1,4,1/g2
r4o,4,o,1/g3
```

```
r2o,2,o,1/g4
r23,2,3,1/g
r34,3,4,1/g
o1,0,2,3
o2,0,4,o
```

vo is:

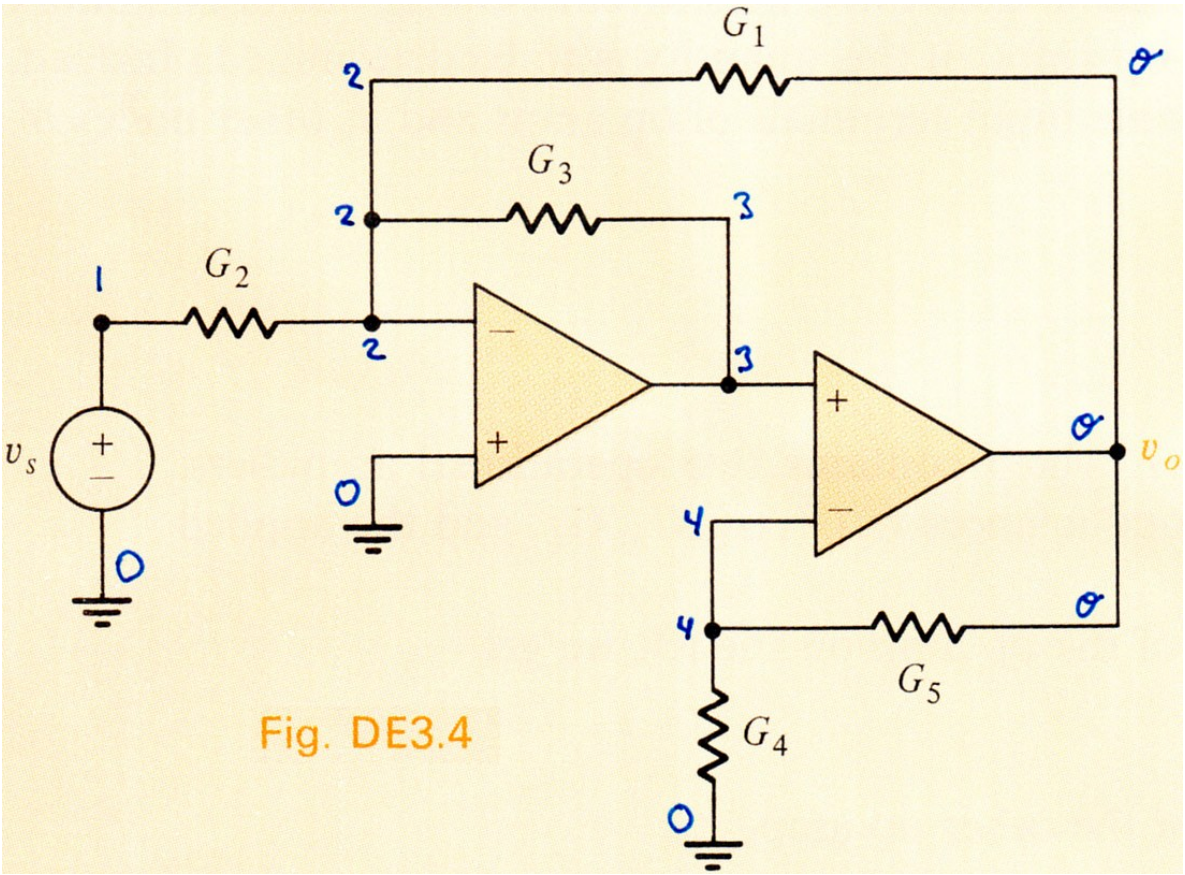
> voltage of node o
>
> $$v_o = \frac{(g_1 - g_2)\, vs}{g_3 - g_4}$$

which is correct, as can be seen by comparing it to the book's answer.

$$v_o = \frac{G_1 - G_2}{G_3 - G_4} v_s$$

Bo2's Drill Exercise 3.4 (Cascade)

For the op-amp circuit shown in Fig. DE3.4, suppose that $G_1 = 1$ S, $G_2 = 2$ S, $G_3 = 3$ S, $G_4 = 4$ S, and $G_5 = 5$ S. Find v_o in terms of v_s. Answer: $-0.75v_s$



Circuit Description

```
e,1,0,vs
r1,2,o,1/1
r2,1,2,1/2
r3,2,3,1/3
r4,4,0,1/4
r5,4,o,1/5
o1,0,2,3
o2,3,4,o
```

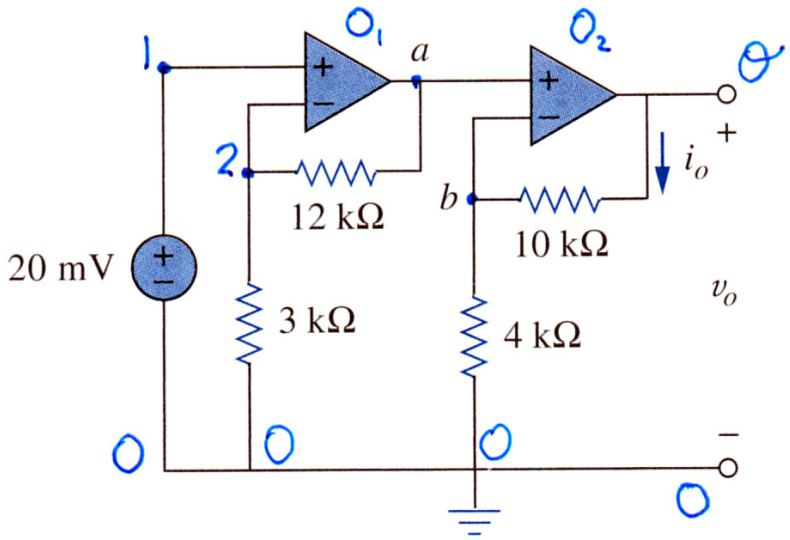
vo is correct:

voltage of node o

$$v_o = -0.75\,vs$$

AS2's Example 5.9 (Cascade)

Find v_o and i_o.



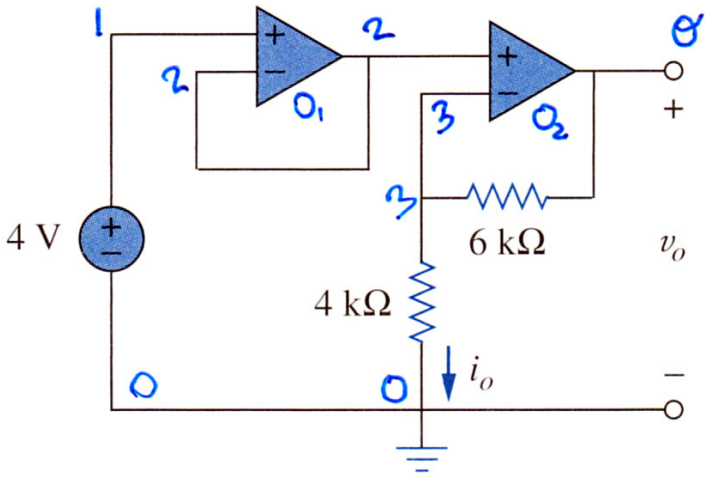
Circuit Description

```
e,1,0,20'm
o1,1,2,a
o2,a,b,o
ro,o,b,10'k
r4,b,0,4'k
r2,a,2,12'k
r3,2,0,3'k
```

vo is **.35** and iro is **2.5e-5**. This is correct.

AS2's Practice Problem 5.9 (Cascade)

Determine v_o and i_o.



Circuit Description

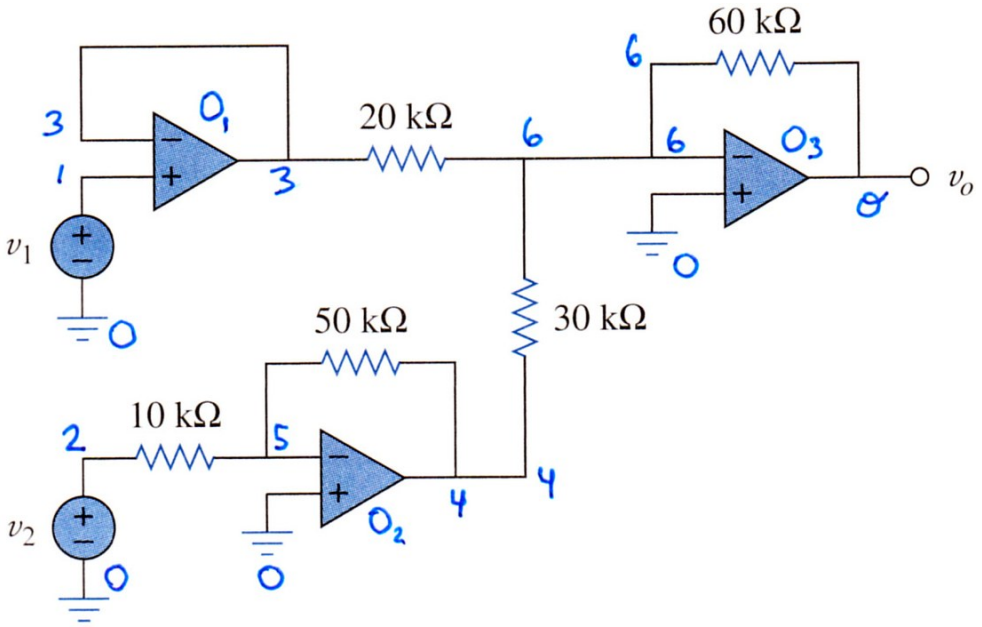
```
e,1,0,4
o1,1,2,2
o2,2,3,o
ro,3,0,4'k
r6,3,o,6'k
```

The answers you want are `vo` and `iro`, in **Results**.

The answer is `vo` = **10** V and `iro` = **1/1000** A, that is 1mA. This is correct.

AS2's Practice Problem 5.10 (Cascade)

If v_1 = 2V and v_2 = 1.5V, find v_o in the circuit.



Circuit Description

```
e1,1,0,2
e2,2,0,1.5
```

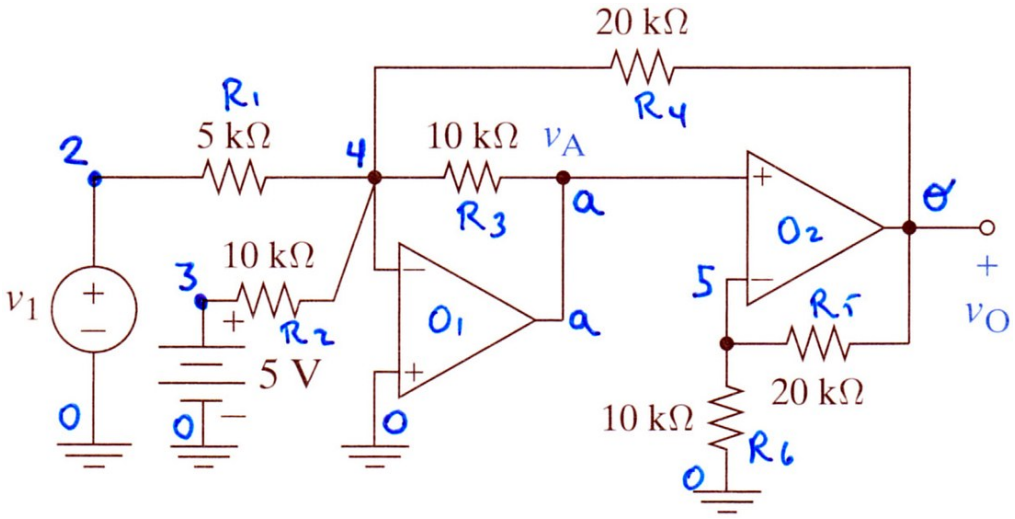
```
o1,1,3,3
r1,2,5,10'k
r2,3,6,20'k
r5,5,4,50'k
o2,0,5,4
r3,6,4,30'k
r6,6,o,60'k
o3,0,6,o
```

The answer you want is `vo`, in **Results**.

The answer, **9**, is correct.

TR5's Example 4-16 (Cascade)

Derive an expression for v_o in terms of the two inputs.



Circuit Description

```
e1,2,0,v1
e5,3,0,5
r1,2,4,5'k
r2,3,4,10'k
o1,0,4,a
r3,4,a,10'k
r4,4,o,20'k
o2,a,5,o
r5,5,o,20'k
r6,5,0,10'k
```

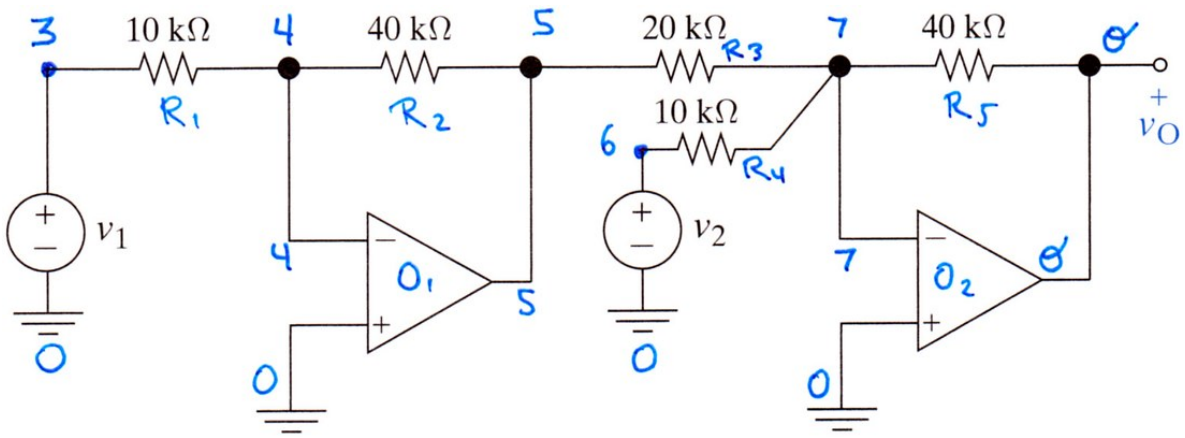
Ask **Evaluate** for `expand(vo)` with **Rounding** set to *approx (full precision)* in **Settings**. The answer is arranged differently from the book's, which is a matter of presentation rather than of arithmetic.

The answer is correct:

$$v_o = -2.4\,v_1 - 6$$

TR5's Exercise 4-14 (Cascade)

Derive an expression for v_o in terms of the inputs v_1 and v_2.



Circuit Description

```
e1,3,0,v1
r1,3,4,10'k
o1,0,4,5
r2,4,5,40'k
r3,5,7,20'k
e2,6,0,v2
r4,6,7,10'k
r5,7,o,40'k
o2,0,7,o
```

Ask **Evaluate** for `expand(vo)`.

The answer is correct:

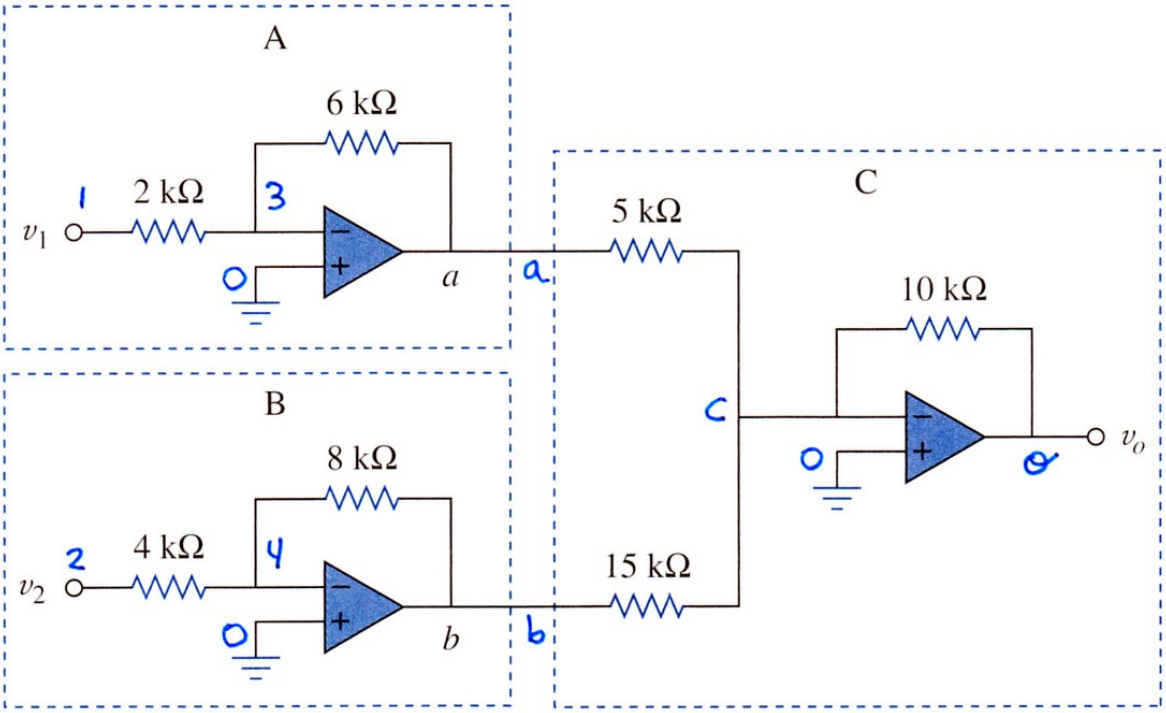
$$v_o = 8\,v_1 - 4\,v_2$$

AS2's Example 5.10 (Cascade)

If v_1 = 1V and v_2 = 2V, find v_o in the circuit.

Circuit Description

```
e1,1,0,1
e2,2,0,2
r2,1,3,2'k
r4,2,4,4'k
r6,3,a,6'k
r8,4,b,8'k
r5,a,c,5'k
r15,b,c,15'k
r10,c,o,10'k
oa,0,3,a
ob,0,4,b
```

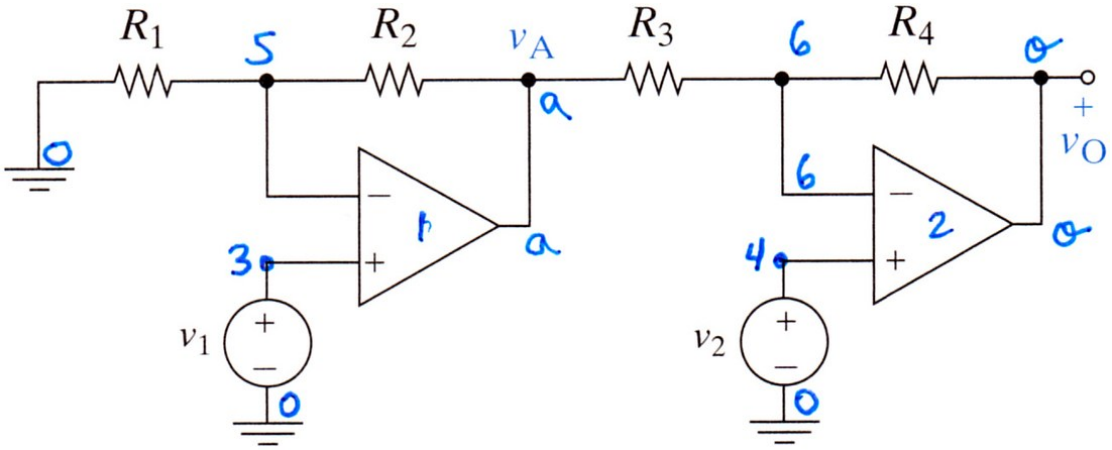



```
oc,0,c,o
```

The answer, **8.667**, is correct.

TR5's Example 4-17 (Cascade)

Derive an expression for v_o in terms of the inputs v_1 and v_2.



Circuit Description

```
r1,5,0,r1
r2,5,a,r2
r3,a,6,r3
r4,6,o,r4
e1,3,0,v1
e2,4,0,v2
o1,3,5,a
```

```
o2,4,6,o
```

The answer you want is `vo`, in **Results**.

We get the right answer. Let's compare it with the book's answer.

$$-\left[\frac{R_4}{R_3}\right]\left[\frac{R_1 + R_2}{R_1}\right]v_1 + \left[\frac{R_3 + R_4}{R_3}\right]v_2$$

Our answer, expanded via `expand(vo)`, is shown below:

voltage of node o

$$v_o = -\frac{r_2\, r_4\, v_1}{r_1\, r_3} - \frac{r_4\, v_1}{r_3} + \frac{r_4\, v_2}{r_3} + v_2$$

Playing with it by hand we get a form that, in my opinion, is *prettier* ;-) than the book's:

voltage of node o

$$v_o = v_2\left(\frac{r_4}{r_3} + 1\right) - v_1\left(\frac{r_4}{r_3}\right)\left(\frac{r_2}{r_1} + 1\right)$$

TR5's Example 4-18 (Multiple)

Derive an expression for v_o in terms of the inputs v_1 and v_2.

Circuit Description

```
e2,a,0,v2
e1,f,0,v1
o2,a,c,b
o1,f,d,e
r1,b,c,r1
r2,c,d,r2
r3,d,e,r3
```

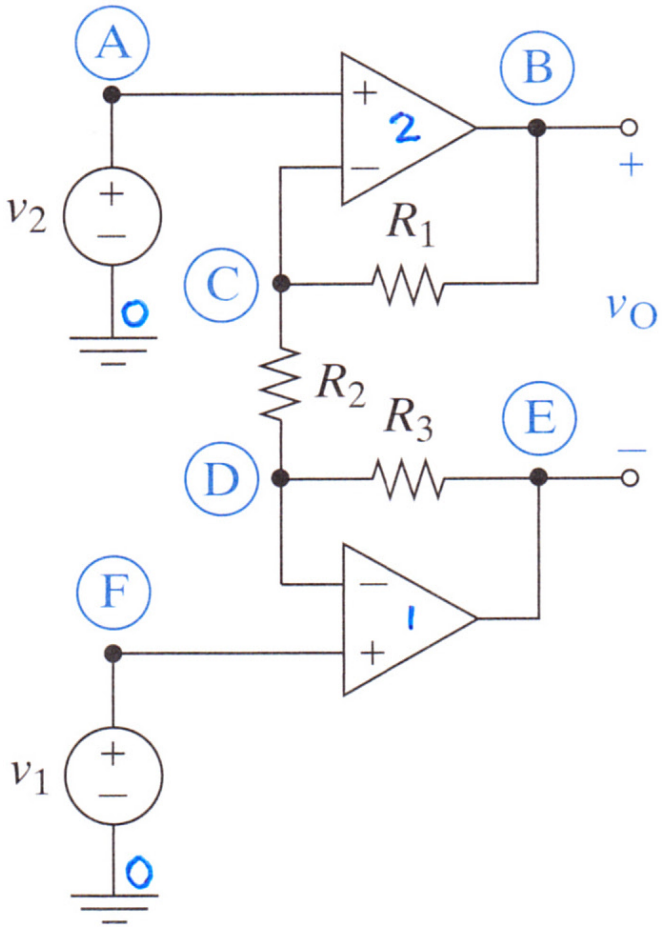
To find vo, we ask **Evaluate** for `vb-ve`. We get an expression that can easily be rearranged to look like this:

voltage of node o

$$v_o = -\left(\frac{r_1 + r_2 + r_3}{r_2}\right)(v_1 - v_2)$$

This is exactly the answer from the book:

If you are ever in doubt whether two expressions are the same, subtract one from the other in **Evaluate**: if the answer is `0`, they are the same.



$$v_O = v_B - v_E = \left[\frac{R_1 + R_2 + R_3}{R_2}\right](v_2 - v_1)$$

LESSON 6

Transient analysis

Learn to run a *transient* time-domain analysis. Learn to describe *capacitors* with **c** and *inductors* with **l**. Limit the results, and plot answers in time.

Symbulator can simulate two energy-storing elements, capacitors and inductors, in three kinds of analysis: transient time-domain (TR), alternating current (AC), and complex frequency-domain (FD). This lesson teaches the transient analysis of a circuit, and how to describe capacitors and inductors for it. AC and FD come in later lessons.

6.1 How to describe a capacitor

> ⓘ NOTE
> **Describing capacitors**
> When describing an ideal capacitor for the purpose of a transient analysis, we need five pieces of information, separated by commas: a unique name to identify the capacitor (must start with the letter **c**), the names of its first and second nodes, its value in farads (F), and its initial condition in volts, understood as the voltage in the first node minus the voltage in the second.
> For example, an ideal capacitor called **ca**, connected between two nodes called **1** and **2**, with a capacitance of **3** F and an initial condition of 0.1 V, would be described as follows: `ca,1,2,3,0.1`

Values can use SI prefixes

The values of the capacitance and the initial condition of the capacitor can use prefixes of the International System.

What answers do you get

For each capacitor, a transient analysis will provide two answers: the voltage drop in the capacitor, in volts, and the current through it, flowing from the first node towards the second, in amperes.

> ⓘ NOTE
> **Note on capacitors in DC**
> In a DC simulation, the current through a capacitor is defined as 0. It basically becomes an open circuit.

6.2 How to describe an inductor

> ⓘ NOTE
>
> **Describing inductors**
>
> When describing an ideal inductor for the purpose of a transient analysis, we need five pieces of information, separated by commas: a unique name to identify the inductor (must start with the letter **l**), the names of its first and second nodes, its value in henries (H), and its initial condition in amperes, understood as the current flowing through it from the first node towards the second.
> For example, an ideal inductor called **l1**, connected between two nodes called **1** and **2**, with an inductance of **0.1** H and an initial condition of 0.2 A, would be described as follows: `l1,1,2,0.1,0.2`

For each inductor, a transient analysis will provide the voltage drop across it and the current through it, with the same conventions as for capacitors.

> ⓘ NOTE
>
> **Note on inductors in DC**
>
> In a DC simulation, the voltage drop in an inductor is defined as 0. It basically becomes a short circuit.

6.3 Running a transient analysis

You will learn the subtleties of transient simulations through a series of examples. For now, to run one set **Analysis** to *TR — transient / time domain*.

Bo2's Example 4.15

EXAMPLE 4.15

For the circuit shown in Fig. 4.37(a), let us determine v given that i is described by Fig. 4.37(b). In this case, let us consider three intervals of time.

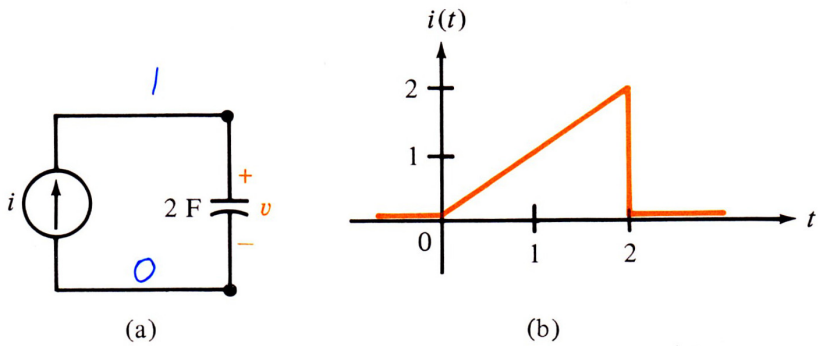


Fig. 4.37 (a) Circuit with a capacitor. (b) Applied current.

Bo2's Example 4.15

SOLUTION

For t ≤ 0 s, since the current source is inactive, there is nothing going on. Without doing

a simulation, we know the voltage v is 0 volts.

For 0 s < t $\leq$ 2 s, the source is active. From the graph, its value in this interval is t: the current in amperes equals the time in seconds. We describe the circuit and run the transient analysis:

Circuit Description

```
j,0,1,t
c,1,0,2,0
```

The source has four terms: name (starting with j), first node, second node, and value — here t. The capacitor has five: name, first node, second node, value in farads, and initial condition in volts — here 0 volts.

Run it in TR and read the capacitor's voltage, `vc`, in **Results**:

voltage drop in c

$$v_c = \frac{t^2}{4}$$

For 2 s < t, the voltage is equal to whatever value it had at time 2 s:

Put `vc` in **Evaluate** and the time in its **Conditions** box:

Evaluate

```
vc
```

Conditions

```
t = 2
```

It gives 1 V, which is correct.

Describing source values

For the purpose of a transient analysis, Symbulator accepts many types of source values. The only thing you must do is describe these values properly.

Step values: symbolic. A source with an unknown step value starting at t=0 is described with a variable and the step function u(t). For example, a voltage source **e1** between nodes **1** and **0**, of V volts starting at t=0, is `e1,1,0,V*u(t)`.

Step values: numerical. A step source with a known numerical value can be described the same way, `e1,1,0,12*u(t)`. To save typing, you can skip the u(t) when the value is numerical, and Symbulator assumes a step: `e1,1,0,12`. The results are the same.

Impulse values. A source with an impulse value at t=0 is described with its value, symbolic or numerical, and the delta function δ(t). For example, a current source **j1** between nodes **0** and **1**, with an impulse of i amperes at t=0, is `j1,0,1,i*δ(t)`.

Values as functions of time. A source whose value is a function of time, such as a ramp, a sinusoid or an exponential, is described by writing the value as an expression in terms of t, as in `j,0,1,t`.

Dependent values. Sources with dependent values are described as we have seen before, for example `j,0,1,3*vr1`.

A word on intervals

In a transient analysis, every switch opening or closing ends one time interval and begins another, and a simulation covers one interval only. When we run a TR simulation and get answers in terms of t, this variable must be understood as the time elapsed since the start of *that* interval in particular.

This is a distinction without a difference as long as the interval we simulate starts at t=0, which is most often the case. However, sometimes problems have switch changes at other times — **Bo2's Example 5.6 (page 169)**, where a switch closes at t=1 s, and **Bo2's Drill Exercise 5.6 (page 171)**, where one closes at t=2 s, are examples in the solved problems.

6.4 Two useful time-savers

Limiting the results

Unless you say otherwise, a Symbulator simulation gives you the whole set of answers: voltages in all nodes, and voltage drops, currents and power consumed in all elements. When the expressions are complicated, finding them all takes time.

To get only a chosen few, tick **Do you want to limit the results to save time?** in **Settings** once TR is chosen, and list them in the box that opens, separated by commas.

Limiting the results to save time (page 296) covers it, with the answer names you may use and when the tick is worth reaching for. It is optional reading — every problem in this lesson solves without it.

Plotting an answer

Symbulator draws plots in the **Plotting Tools** card below the results. Solve the circuit first, since the plot is of an answer, then open the card and fill in four things:

- **Plot type**: *Plot a function of time (TR)*
- **Variable to plot**: the answer you want, such as `v2`
- **Start time (s)** and **End time (s)**: the window, for example 0 and 0.005
- **Points**: how finely to sample it; 300 is the default and is usually plenty

Press **Run** and the curve appears under the card.

The variable must be an answer's name, not an expression, and one this circuit has, as **Results** spells it: `v2` or `ir1`.

> ⚠ WARNING
> **A curve the inverse Laplace could not find**

When the inverse Laplace transform cannot produce an expression, Symbulator samples the s-domain answer numerically, so the plot still appears even when **Results** shows nothing for that variable. A blank plot usually means a misspelled variable.

6.5 Instructive solved examples

Transient analysis of RC circuit

Bo2's Example 5.1

EXAMPLE 5.1

Let us begin this example by considering the circuit in Fig. 5.4, in which there is a switch that is closed for time $t < 0$, is opened at time $t = 0$, and stays open for all time $t > 0$. For this circuit, let us determine $v_C(t)$ for all values of t.

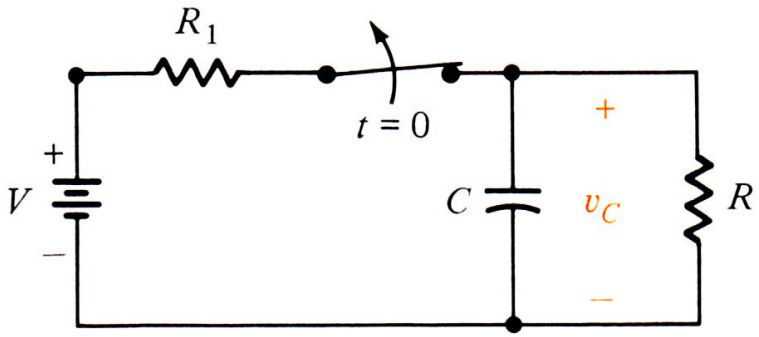


Fig. 5.4 *RC* circuit with a switch.

Solution. Here we have two intervals. In the first, **t < 0s**, we assume things have been steady for a long time and any transient has passed, so we run a DC simulation to find the initial conditions for the second.

Circuit Description

```
e,1,0,V
r1,1,2,r1
c,2,0,c
r,2,0,r
```

Run it in DC and read `vc`, the voltage in the capacitor **c**. It is the expression shown below, which is correct.

For the second interval, when **0s** $\leq$ **t**, we do a transient simulation. We give the capacitor as initial condition its voltage from the previous interval.

Circuit Description

```
c,2,0,c,r*v/(r+r1)
```

```
r,2,0,r
```

Run it in TR and read `vc` again. It is the expression below, which is correct.

voltage drop in c

$$v_c = e^{-t/cr}(rv)/(r + r_1)$$

Bo2's Drill Exercise 5.1

DRILL EXERCISE 5.1

For the circuit shown in Fig. DE5.1, find a single expression for all t for (a) $v_C(t)$ and (b) $i_C(t)$.
Answer: (a) $6 - 6(1 - e^{-4t})u(t)$ V; (b) $-2e^{-4t}$ A

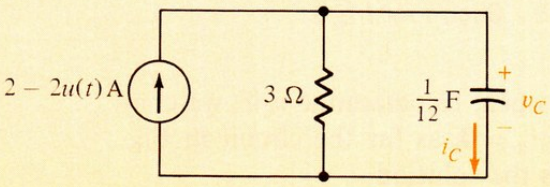


Fig. DE5.1

Solution. Again, we have two intervals: the first for DC, the second for TR.

For the first interval, when **t < 0s**, the current source is 2A. Simulate in DC.

Circuit Description

```
j,0,1,2
r,1,0,3
c,1,0,1/12
```

The answers you want are in **Results**:

voltage drop in c

$$v_c = 6 \text{ V}$$

current through c

$$i_c = 0 \text{ A}$$

So the voltage in the capacitor is 6V. We give this value the name $\mathbf{vc}_0$ in **Define**, to use it as the initial condition of the capacitor for the second interval.

For the second interval, when $\mathbf{0s} \leq \mathbf{t}$, we do a transient simulation. The current source is 0A, and the capacitor has an initial condition: voltage $\mathbf{vc}_0$.

Circuit Description

```
j,0,1,0
r,1,0,3
```

```
c,1,0,1/12,vc0
```

Define
```
vc0 = 6
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are `vc` and `ic`, in **Results**.

Run it, and the two answers are the expressions below. They are right.

voltage drop in c

$$v_c = 6e^{-4t}$$

current through c

$$i_c = -2e^{-4t}$$

Bo2's p224 5.2

EXAMPLE 5.2

For the circuit shown in Fig. 5.7, let us determine $v_C(t)$, $i_C(t)$, and $v(t)$ for all time.

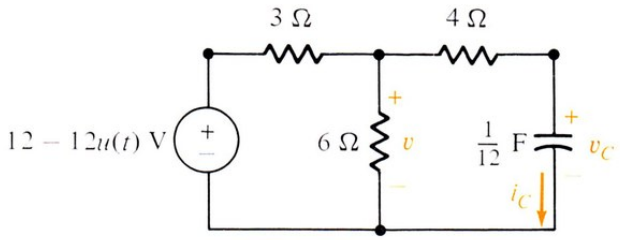


Fig. 5.7 Circuit for Example 5.2.

Solution. Again, we have two intervals: the first for DC, the second for TR.

For the interval when **t < 0s**, the voltage source is 12 V. Simulate in DC.

Circuit Description
```
e,1,0,12
r3,1,2,3
r6,2,0,6
r4,2,3,4
c,3,0,1/12
```

The answers you want are in **Results**:

voltage drop in c

$$v_c = 8\ \text{V}$$

current through c

$$i_c = 0\,\text{A}$$

voltage of node 2

$$v_2 = 8\,\text{V}$$

For the interval when **0s** $\leq$ **t**, the voltage source is 0V. Simulate in TR, giving the capacitor as initial condition its voltage from the previous interval.

Circuit Description

```
e,1,0,0
r3,1,2,3
r6,2,0,6
r4,2,3,4
c,3,0,1/12,8
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

voltage drop in c

$$v_c = 8e^{-2t}$$

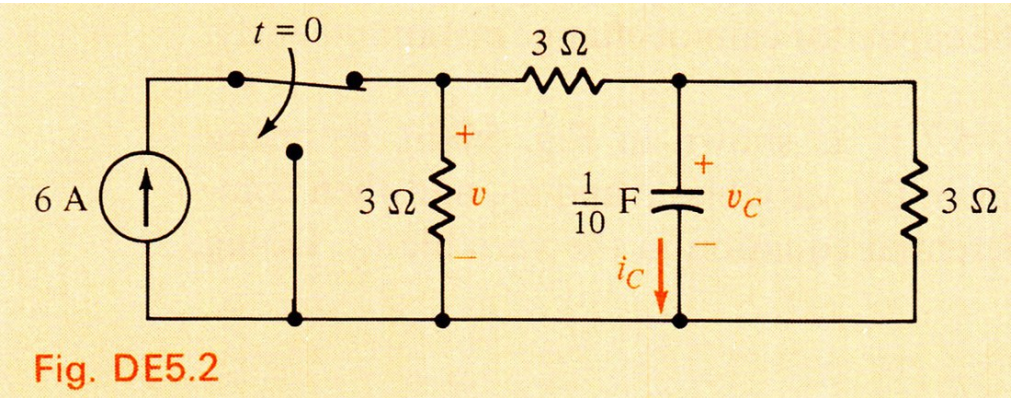
current through c

$$i_c = (-4/3)e^{-2t}$$

voltage of node 2

$$v_2 = (8/3)e^{-2t}$$

Bo2's Drill Exercise 5.2



Solution. For **t < 0s**, the 6A source is part of the circuit. Simulate in DC.

DRILL EXERCISE 5.2

For the circuit shown in Fig. DE5.2, find an expression for all t for (a) $v_C(t)$, (b) $i_C(t)$, and (c) $v(t).$
Answer: (a) $6 - 6(1 - e^{-5t})u(t)$ V; (b) $-3e^{-5t}u(t)$ A; (c) $12 - 12u(t) + 3e^{-5t}u(t)$ V

Circuit Description

```
j,0,1,6
r10,1,0,3
r12,1,2,3
r20,2,0,3
c,2,0,1/10
```

The answers you want are in **Results**:

voltage drop in c

$$v_c = 6 \text{ V}$$

current through c

$$i_c = 0 \text{ A}$$

voltage of node 1

$$v_1 = 12 \text{ V}$$

For **0s** $\leq$ **t**, the source is no longer relevant and can be excluded from the circuit. Simulate in TR, giving the capacitor its proper initial condition.

Circuit Description

```
r10,1,0,3
r12,1,2,3
r20,2,0,3
c,2,0,1/10,6
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

voltage drop in c

$$v_c = 6e^{-5t}$$

current through c

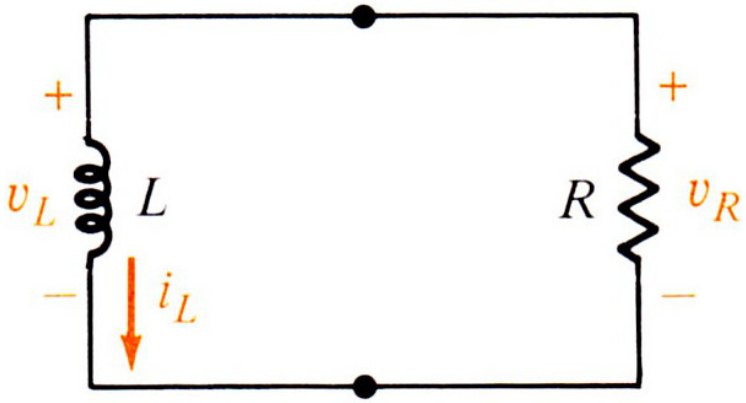
$$i_c = -3e^{-5t}$$

voltage of node 1

$$v_1 = 3e^{-5t}$$

Bo2's Figure 5.10a

Look at the simple resistor-inductor (RL) circuit shown, where at time t=0 the inductor current is $i_L(0)$. Determine $v_L(t)$, $i_L(t)$ and $v_R(t)$ for t≥0.



Since we are already given the initial condition, we only run the transient simulation for t≥0. In our circuit description below, the initial condition is **il0**.

Circuit Description

```
l,1,0,l,il0
r,1,0,r
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are `vl`, `il` and `vr`, in **Results**.

We get the right answers:

voltage drop in l

$$v_l = -il_0 re^{(-rt)/l}$$

current through l

$$i_l = il_0 e^{(-rt)/l}$$

voltage drop in r

$$v_r = -il_0 re^{(-rt)/l}$$

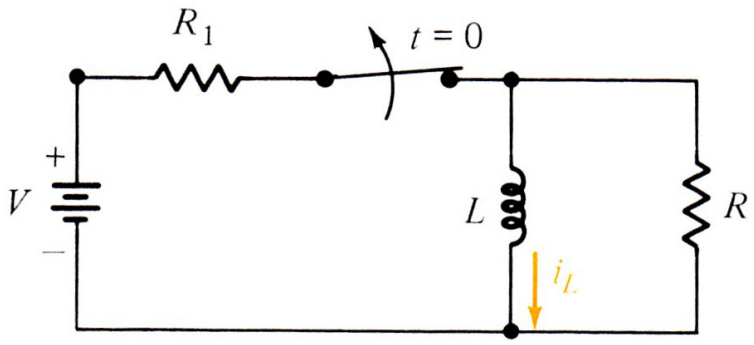
Bo2's Example 5.3

Determine $i_L(t)$ for all t.

Solution. For **t < 0s**, simulate in DC.

Circuit Description

```
e,1,0,v
r1,1,2,r1
```




```
l,2,0,l
r2,2,0,r
```

The answer you want is `il`, in **Results**.

For **t ≥ 0s**, simulate in TR, giving the inductor its initial condition.

Circuit Description

```
l,2,0,l,v/r1
r2,2,0,r
```

Set **Analysis** to *TR — transient / time domain*.

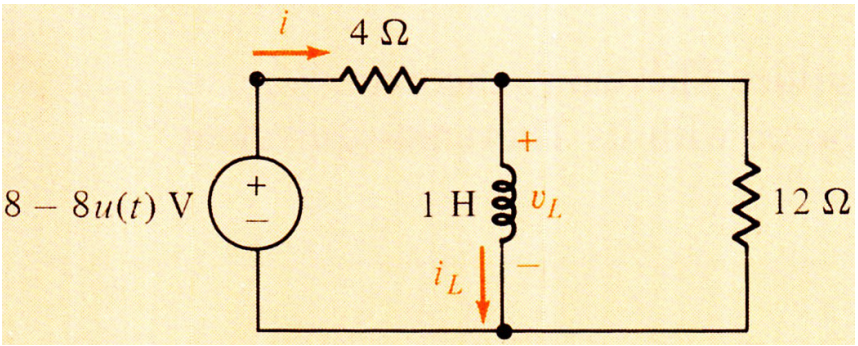
The answer you want is in **Results**:

current through l

$$i_l = e^{(-rt)/l} v/r_1$$

Bo2's Drill Exercise 5.3

Find $\boldsymbol{i_L}(t)$, $\boldsymbol{v_L}(t)$ and $\boldsymbol{i}(t)$ for all t.



Solution. For **t < 0s**, simulate in DC.

Circuit Description

```
e,1,0,8
r4,1,2,4
l,2,0,1
r12,2,0,12
```

The answers you want are in **Results**:

> current through l
>
> $$i_l = 2\ \text{A}$$

> voltage drop in l
>
> $$v_l = 0\ \text{V}$$

> current through r4
>
> $$i_{r4} = 2\ \text{A}$$

For **t ≥ 0s**, simulate in TR, giving the inductor its initial condition of 2A.

> Circuit Description
> ```
> e,1,0,0
> r4,1,2,4
> l,2,0,1,2
> r12,2,0,12
> ```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

> current through l
>
> $$i_l = 2e^{-3t}$$

> voltage drop in l
>
> $$v_l = -6e^{-3t}$$

> current through r4
>
> $$i_{r4} = (3/2)e^{-3t}$$

Bo2's p230 (Dependent source)

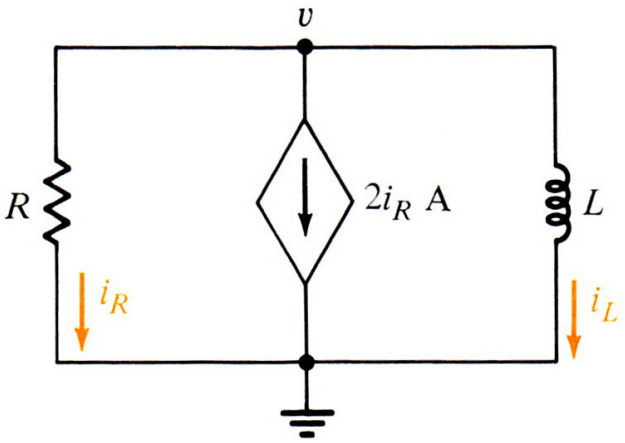
Find $\boldsymbol{i_L}(t)$ for t≥0, given that $\boldsymbol{i_L}(0)$ = 5A.

We only run the transient simulation for t≥0, with initial condition 5A.

> Circuit Description
> ```
> r,v,0,r
> j,v,0,2*ir
> l,v,0,l,5
> ```

Set **Analysis** to *TR — transient / time domain*.

The answer you want is in **Results**:

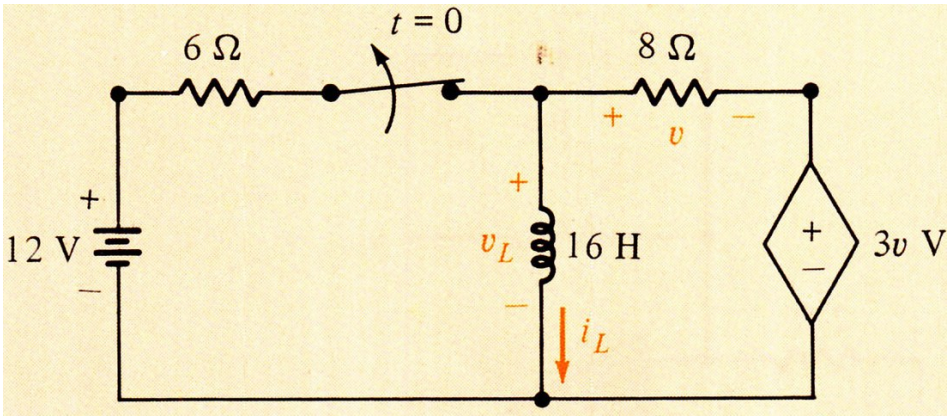


current through l

$$i_l = 5e^{((-Rt)/3L)}$$

Bo2's Drill Exercise 5.4 (Dependent source)

Find $\boldsymbol{i_L}(t)$ and $\boldsymbol{v_L}(t)$ for all t.



Solution. For **t < 0s**, simulate in DC. This gives us the initial condition.

Circuit Description

```
ei,1,0,12
r6,1,2,6
l,2,0,16
r8,2,3,8
ed,3,0,3*vr8
```

The answers you want are in **Results**:

current through l

$$i_l = 2\ \text{A}$$

voltage drop in l

$$v_l = 0\ \text{V}$$

For **t ≥ 0s**, simulate in TR, giving the inductor its initial condition.

Circuit Description

```
l,2,0,16,2
r8,2,3,8
ed,3,0,3*vr8
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

current through l

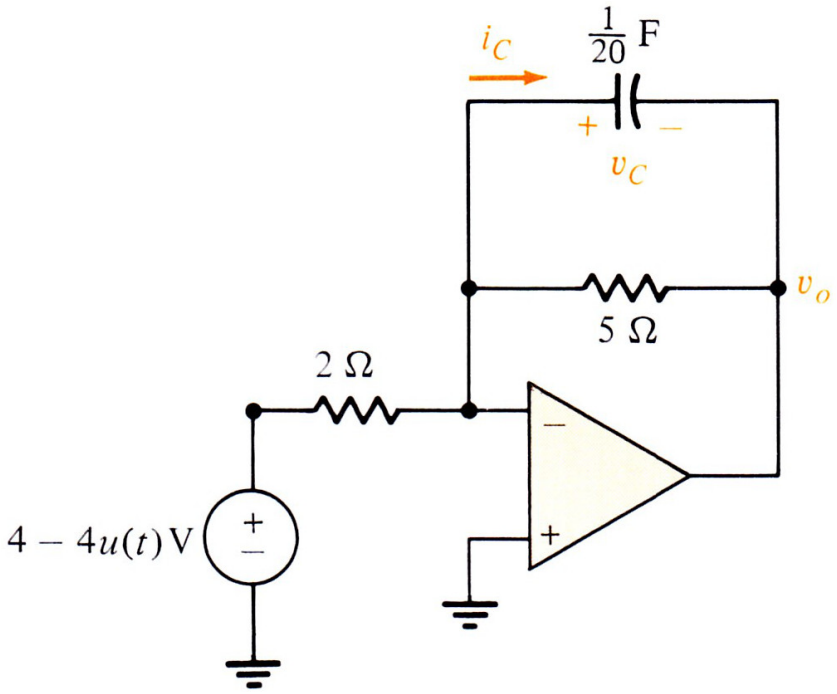
$$i_l = 2e^{-2t}$$

voltage drop in l

$$v_l = -64e^{-2t}$$

Bo2's Example 5.5 (Op Amp)

Determine $\boldsymbol{v_C}$(t), $\boldsymbol{i_C}$(t) and $\boldsymbol{v_o}$(t) for all t.



Solution. For **t < 0s**, simulate in DC. This gives us the initial condition.

Circuit Description

```
e,3,0,4
r2,3,1,2
r5,1,o,5
```

```
c,1,o,1/20
o,0,1,o
```

The answers you want are in **Results**:

voltage drop in c

$$v_c = 10\ \text{V}$$

current through c

$$i_c = 0\ \text{A}$$

voltage of node o

$$v_o = -10\ \text{V}$$

For **t ≥ 0s**, simulate in TR, giving the capacitor its initial condition.

Circuit Description

```
r2,0,1,2
r5,1,o,5
c,1,o,1/20,10
o,0,1,o
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

voltage drop in c

$$v_c = 10e^{-4t}$$

current through c

$$i_c = -2e^{-4t}$$

voltage of node o

$$v_o = -10e^{-4t}$$

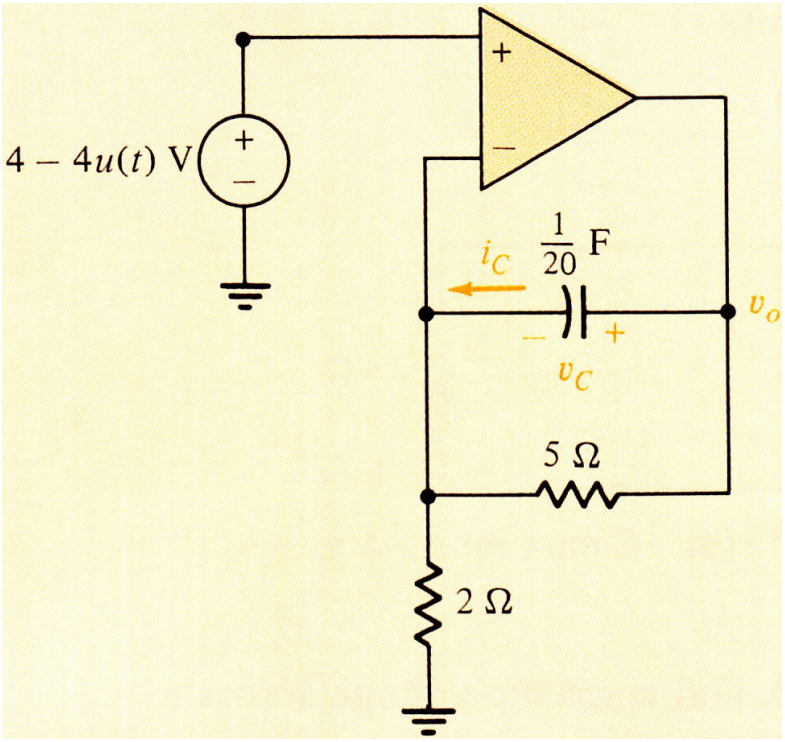
Bo2's Drill Exercise 5.5 (Op Amp)

Determine $v_C(t)$, $i_C(t)$ and $v_o(t)$ for all t.

Solution. For **t < 0s**, simulate in DC. This gives us the initial condition.

Circuit Description

```
e,2,0,4
o,2,1,o
c,o,1,1/20
r5,o,1,5
```




```
r2,1,0,2
```

The answers you want are in **Results**:

voltage drop in c

$$v_c = 10\ \text{V}$$

current through c

$$i_c = 0\ \text{A}$$

voltage of node o

$$v_o = 14\ \text{V}$$

For **t ≥ 0s**, simulate in TR, giving the capacitor its initial condition.

Circuit Description

```
o,0,1,o
c,o,1,1/20,10
r5,o,1,5
r2,1,0,2
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

voltage drop in c

$$v_c = 10e^{-4t}$$

current through c

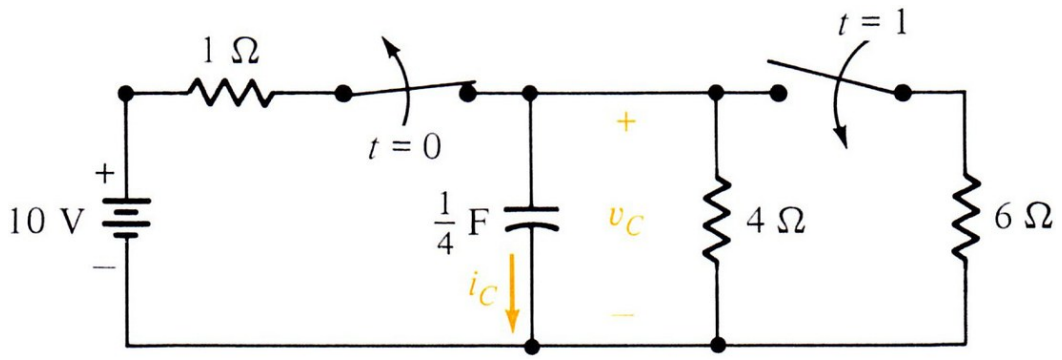
$$i_c = -2e^{-4t}$$

voltage of node o

$$v_o = 10e^{-4t}$$

Bo2's Example 5.6

In the circuit shown, there are two switches: one that opens at time t=0 and one that closes at time t=1 second. Determine $\boldsymbol{v_C}$(t) and $\boldsymbol{i_C}$(t) for all t.



As we mentioned before, every time a switch moves marks the end of one interval and the beginning of another interval for Symbulator.

For the first interval, which corresponds to t<0, we simulate in DC.

Circuit Description

```
e,2,0,10
r1,2,1,1
c,1,0,1/4
r4,1,0,4
```

The answers you want are in **Results**:

voltage drop in c

$$v_c = 8\ \mathrm{V}$$

current through c

$$i_c = 0\ \mathrm{A}$$

The voltage in the capacitor at the end of this interval will serve as the initial condition of the capacitor for the next interval. The second interval runs between 0 and 1 second, i.e. $0 < t \leq 1$ second. We simulate in TR.

Circuit Description

```
c,1,0,1/4,8
r4,1,0,4
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

voltage drop in c

$$v_c = 8e^{-t}$$

current through c

$$i_c = -2e^{-t}$$

The capacitor's voltage at the end of this second interval is the initial condition for the third. We can use its exact value, `8*exp(-1)`, but the textbook prefers using its approximate value, i.e. `2.943`.

The third interval corresponds to t > 1 second. We simulate in TR. To make things easier for Symbulator, we replace the resistors by their equivalent.

Circuit Description

```
c,1,0,1/4,2.943
re,1,0,[4,6]
```

The expressions we get are equivalent to:

voltage drop in c

$$v_c = 2.943e^{(-5(t-1))/3}$$

current through c

$$i_c = -1.226e^{(-5(t-1))/3}$$

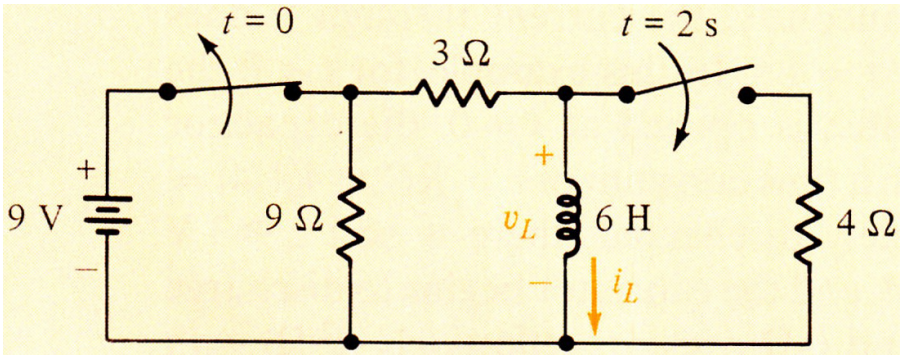
These are the right answers, as can be seen by checking the book's answers.

$$v_C(t) = \begin{cases} 8\ \text{V} & \text{for} \quad t < 0 \\ 8e^{-t}\ \text{V} & \text{for} \quad 0 \le t < 1\ \text{s} \\ 2.943e^{-5(t-1)/3}\ \text{V} & \text{for} \quad t \ge 1\ \text{s} \end{cases}$$

$$i_C(t) = \begin{cases} 0\ \text{A} & \text{for} \quad t < 0 \\ -2e^{-t}\ \text{A} & \text{for} \quad 0 \le t < 1\ \text{s} \\ -1.226e^{-5(t-1)/3}\ \text{A} & \text{for} \quad t \ge 1\ \text{s} \end{cases}$$

Bo2's Drill Exercise 5.6

Let's see another example where a TR simulation is done for an interval that starts at a time other than t=0. Determine $\boldsymbol{i_L}(t)$ and $\boldsymbol{v_L}(t)$ for all t.



For the first interval, which corresponds to t<0, we simulate in DC.

Circuit Description

```
e,1,0,9
r9,1,0,9
r3,1,2,3
l,2,0,6
```

The answers you want are in **Results**:

current through l

$$i_l = 3\ \text{A}$$

voltage drop in l

$$v_l = 0\ \text{V}$$

The 3A current in the inductor is its initial condition for the second interval, which goes from 0 to 2 seconds, i.e. $0 < t \leq 2$ seconds. We simulate in TR.

Circuit Description

```
r,1,0,9+3
l,1,0,6,3
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

current through l

$$i_l = 3e^{-2t}$$

voltage drop in l

$$v_l = -36e^{-2t}$$

The inductor's current at the end of this second interval is the initial condition for the third. Find its approximate value thus:

Put `il` in **Evaluate** with `t = 2` in its **Conditions** box. It reads

.055

The third interval corresponds to t > 2 second. We simulate in TR. To make things easier for Symbulator, we replace the resistors by their equivalent.

Circuit Description

```
r,1,0,[9+3,4]
l,1,0,6,.055
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

current through l

$$i_l = .055e^{(-(t-2)/2)}$$

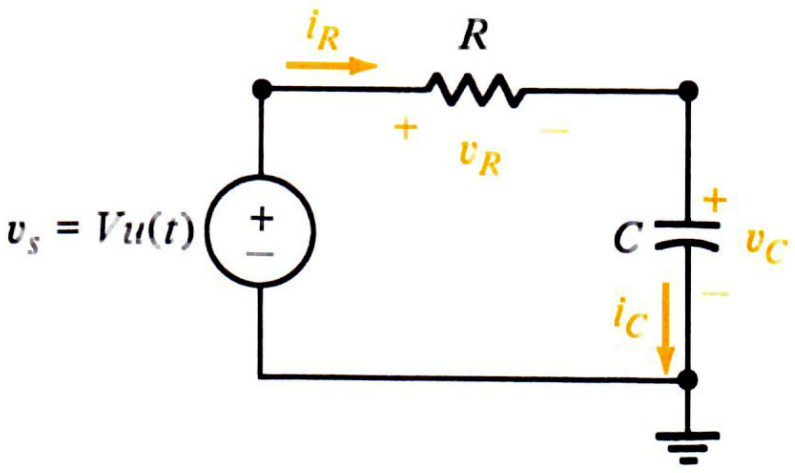
voltage drop in l

$$v_l = -.165e^{(-(t-2)/2)}$$

These are the kind of expressions your book or professor are looking for.

Bo2's Figure 5.19

In the following circuit, assume that all the initial conditions are zero. The source is a step function **u(t)** with value **V** volts. Find i_R, v_R, i_C and v_C.



As discussed before, a step source with symbolic value requires that we use the `u(t)` nomenclature to describe it. Since the value is V volts, we describe the source as `V*u(t)`.

Circuit Description

```
e,1,0,V*u(t)
r,1,2,r
```

```
c,2,0,c,0
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are `vc`, `vr` and `ic`, in **Results**.

There is no need to look for i_R, since i_R=i_C. We get the following expressions:

voltage drop in c

$$v_c = v - ve^{(-t/(cr))}$$

voltage drop in r

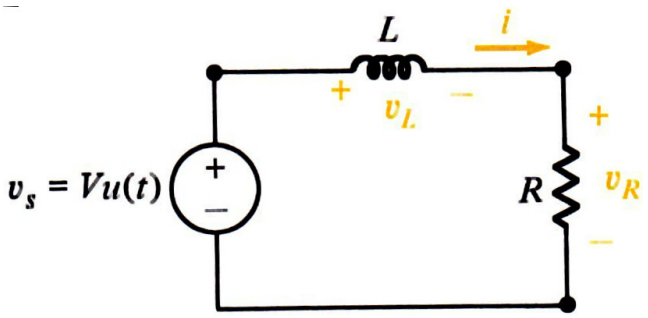
$$v_r = ve^{(-t/(cr))}$$

current through c

$$i_c = (v/r)e^{(-t/(cr))}$$

Bo2's Figure 5.24

In the following circuit, assume that all the initial conditions are zero. The source is a step function **u(t)** with value **V** volts. Find i_R, v_R, i_L and v_L.



Circuit Description

```
e,1,0,V*u(t)
l,1,2,l,0
r,2,0,r
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are `vl`, `vr` and `il`, in **Results**.

There is no need to look for i_R, since in a series circuit it will be identical to i_L.

voltage drop in l

$$v_l = ve^{(-rt)/l}$$

voltage drop in r

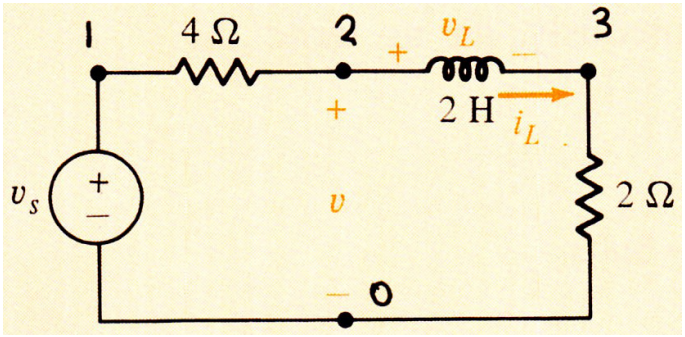
$$v_r = v(1 - e^{(-rt)/l})$$

current through l

$$i_l = (v/r)(1 - e^{(-rt)/l})$$

Bo2's Drill Exercise 5.7

For the circuit, with a step source of 12 volts, find i_L, v_L, i_L and v.



As discussed before, a step source with numerical value can be described without the `u(t)` nomenclature. Thus, we describe the source as `12`.

Circuit Description

```
e,1,0,12
r4,1,2,4
l,2,3,2,0
r2,3,0,2
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

current through l

$$i_l = 2 - 2e^{-3t}$$

voltage drop in l

$$v_l = 12e^{-3t}$$

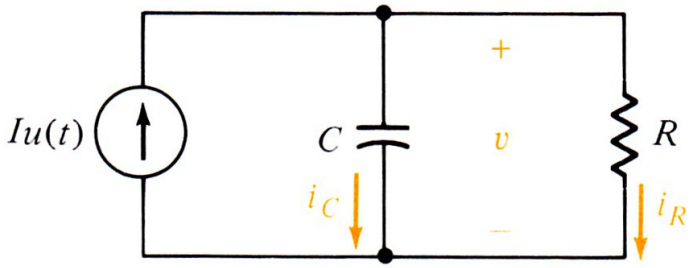
voltage of node 2

$$v_2 = 8e^{-3t} + 4$$

Bo2's Figure 5.26

In the following circuit, assume that all the initial conditions are zero. The source is a step function **u(t)** with value **I** amperes. Find i_C, i_R, and v.

As discussed before, a step source with symbolic value requires that we use the `u(t)` nomenclature to describe it. Since the value is I amperes, we describe the source as `I*u(t)`. Thus:



Circuit Description

```
j,0,1,i*u(t)
c,1,0,c,0
r,1,0,r
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

voltage of node 1

$$v_1 = (i - ie^{-t/(cr)}) \cdot r$$

current through r

$$i_r = i - ie^{-t/(cr)}$$

current through c

$$i_c = ie^{-t/(cr)}$$

Bo2's Example 5.7 (Op Amp)

Suppose that $v_s(t)=u(t)$. Find v_C, i_C, and v_o.

This source is a step source with a value of 1 volt. Since its value is numerical, the `u(t)` nomenclature is not needed. We describe the source plainly as `1`.

Circuit Description

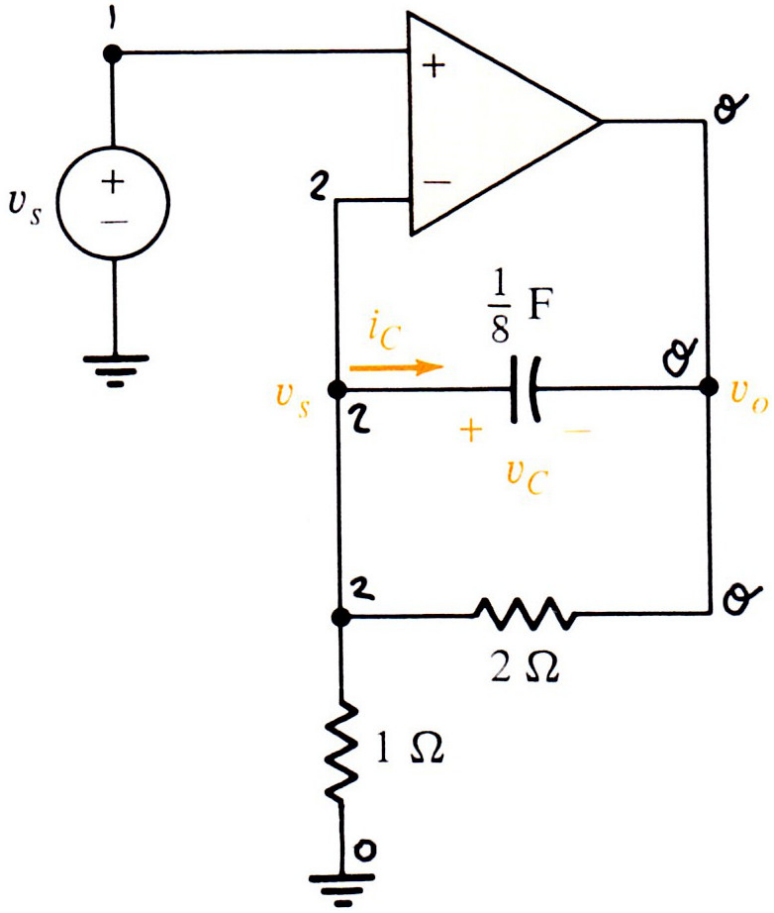
```
e,1,0,u(t)
o,1,2,o
c,2,o,1/8,0
r2,2,o,2
r1,2,0,1
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

voltage drop in c

$$v_c = 2e^{-4t} - 2$$



current through c

$$i_c = -e^{-4t}$$

voltage of node o

$$v_o = 3 - 2e^{-4t}$$

Bo2's Drill Exercise 5.8 (Op Amp)

Suppose that $v_s(t)=u(t)$. Find v_C, i_C, and v_o.

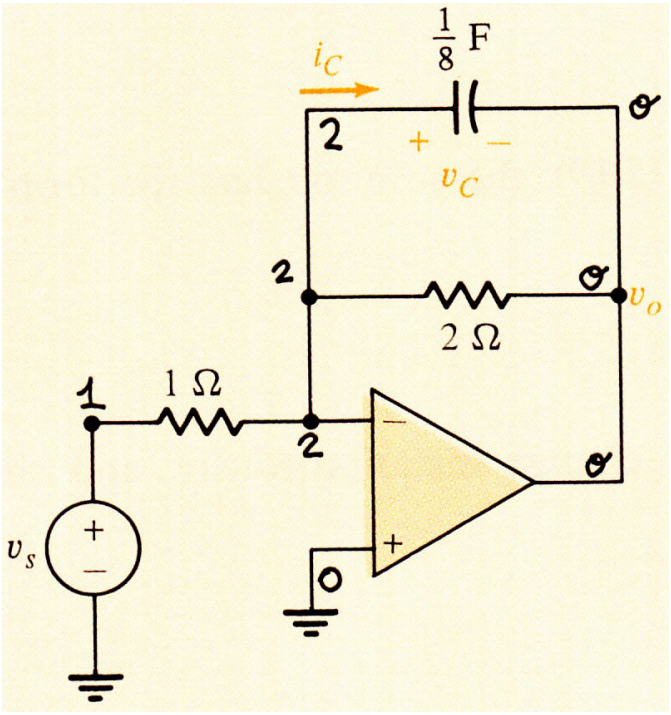
Again, following the same logic as above, we describe the source plainly as `1`.

Circuit Description

```
e,1,0,u(t)
r1,1,2,1
r2,2,o,2
c,2,o,1/8,0
o,0,2,o
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:



voltage drop in c

$$v_c = 2 - 2e^{-4t}$$

current through c

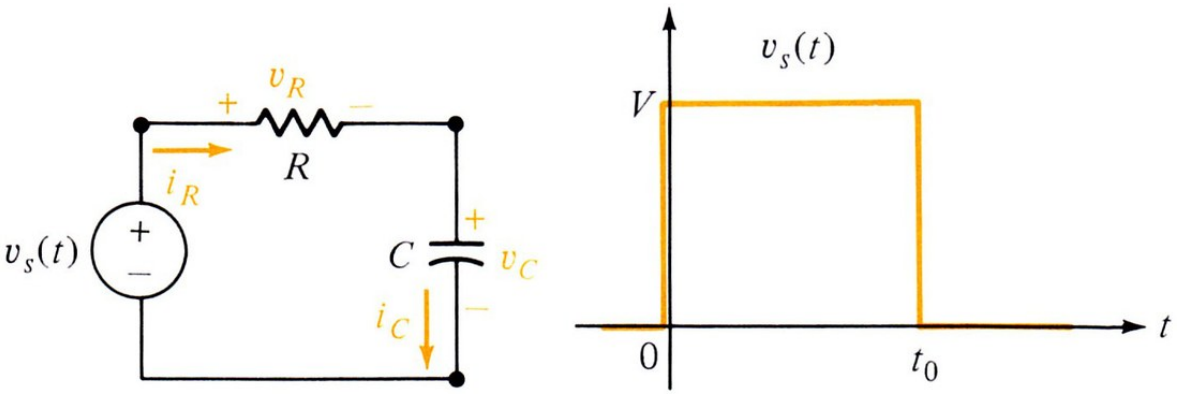
$$i_c = e^{-4t}$$

voltage of node o

$$v_o = 2e^{-4t} - 2$$

Bo2's p249 F5.28

For the circuit left, the source is $v_s(t) = V\,u(t) - V\,u(t\text{-}t_0)$. This is shown in the plot to the right. Given this, find the voltage drop in the capacitor for all t.



Because the source has two steps, this problem has to be analyzed in two intervals. For

the first interval, the initial conditions are zero and the source value is V*u(t). Our circuit description for the first interval is given below.

Circuit Description

```
e,1,0,V*u(t)
r,1,2,r
c,2,0,c,0
```

Set **Analysis** to *TR — transient / time domain*.

The answer you want is `vc`, in **Results**.

The expression for the capacitor's voltage drop in the first interval is:

voltage drop in c

$$v_c = v - ve^{(-t/(cr))}$$

The second interval starts at t_0. In it, the initial condition of the capacitor is given by the value of the expression above when t=t_0. Here is how we find it:

Put `vc` in the **Evaluate** card and write `t = to` in its **Conditions** box, to evaluate this expression under this condition. The answer comes back $V - Ve^{-\frac{to}{cr}}$.

The circuit description for the second interval uses that expression as the capacitor's initial condition. The source is now 0 volts — another way of saying it becomes a short — so it need not appear in the description.

voltage drop in c

$$v_c = (v - ve^{(-to/(cr))})e^{((-t+to)/(cr))}$$

Circuit Description

```
r,0,2,r
c,2,0,c,v-e^(-to/(c*r))*v
```

Set **Analysis** to *TR — transient / time domain*.

The answer you want is `vc`, in **Results**.

This is the expression for the capacitor's voltage drop in the second interval.

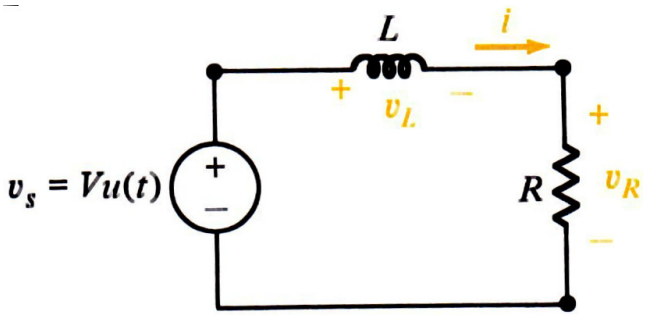
Bo2's Drill Exercise 5.9

Find the current through the inductor if the source is v_s(t) = V u(t) – V u(t-t_0).

Because the source has two steps, this problem has to be analyzed in two intervals. For the first interval, the initial conditions are zero and the source value is V*u(t). Our circuit description for the first interval is given below.

Circuit Description

```
e,1,0,V*u(t)
l,1,2,l,0
```




```
r,2,0,r
```

Set **Analysis** to *TR — transient / time domain*.

The answer you want is `il`, in **Results**.

The expression for the inductor's current in the first interval is:

current through l

$$i_l = (v/r)(1 - e^{((-rt)/l)})$$

The second interval starts at t_0. In it, the initial condition of the inductor is given by the value of the expression above when t=t_0. Here is how we find it:

Put `il` in the **Evaluate** card and write `t = to` in its **Conditions** box. The answer comes back $\frac{V}{r} - \frac{Ve^{-\frac{rto}{l}}}{r}$, which is the same expression gathered differently.

The circuit description for the second interval uses that expression as the inductor's initial condition. The source becomes a short, so it need not appear in the description.

current through l

$$i_l = (v/r)(1 - e^{((-tor)/l)})e^{((-rt)/l)}$$

Circuit Description

```
l,0,2,l,v/r-e^(-to*r/l)*v/r
r,2,0,r
```

Set **Analysis** to *TR — transient / time domain*.

The answer you want is `il`, in **Results**.

To put this in terms of the same t as the first interval, we replace **t** with **t-t**$_o$:

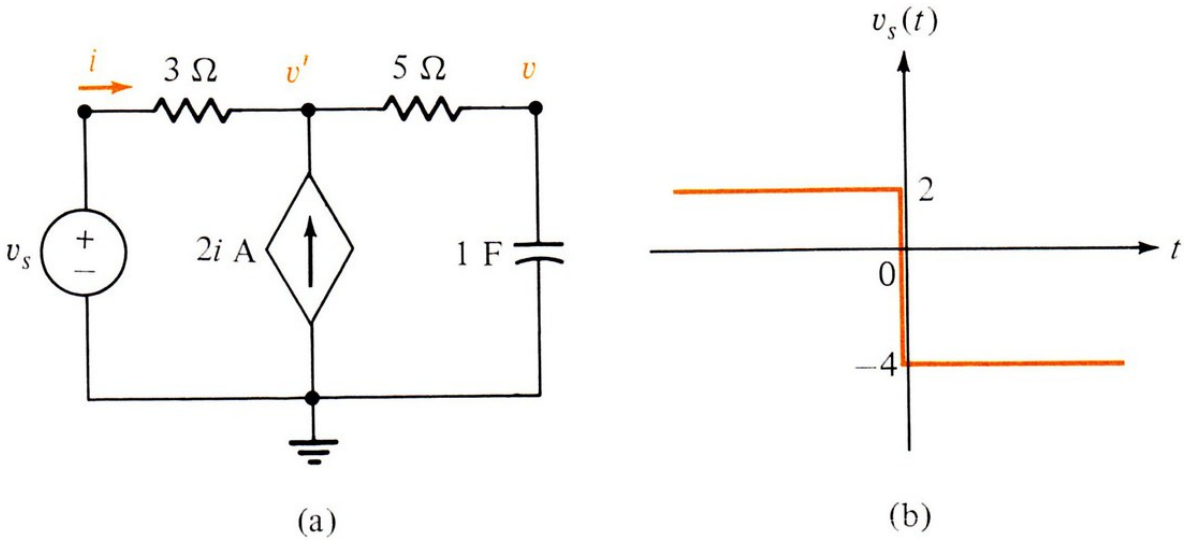
current through l

$$i_l = (v/r)(1 - e^{((-tor)/l)})e^{((-r(t-to))/l)}$$

This is the expression for the inductor's current in the second interval.

Bo2's Figure 5.36

For the circuit shown in (a), find the voltage drop in the capacitor, given that the voltage source is as shown in (b).



This problem requires two intervals. In the first interval, a DC analysis of the circuit with a source of 2 V gives us the voltage in the capacitor.

Circuit Description

```
e,1,0,2
r3,1,2,3
r5,2,3,5
c,3,0,1
j,0,2,2*ir3
```

The answer you want is `vc`, in **Results**. It reads

2

This is the initial condition for the capacitor in the second interval, which we analyze using TR and a source value of -4 V.

Circuit Description

```
e,1,0,-4*u(t)
r3,1,2,3
r5,2,3,5
c,3,0,1,2
j,0,2,2*ir3
```

Set **Analysis** to *TR — transient / time domain*.

The answer you want is in **Results**:

voltage drop in c

$$v_c = 6e^{(-t/6)} - 4$$

This is the expression for the voltage drop in the capacitor after t=0.

Limiting the results

Bo2's Drill Exercise 5.11

For the circuit of **Bo2's Example 5.7 (page 175)**, change the value of the capacitor to ¼ F. Find $v_C(t)$ for the case that the source is $v_S(t)$=1 V for t<0 and 3V for t≥0.

We can tell Symbulator that we are only interested in one answer, `vc`.

Optional in Symbulator 9, which solves quickly enough: if you want to, tick **Do you want to limit the results to save time?** in **Settings** and list `vc` in the field it reveals.

Then we run the simulation for the first interval, just as we did before, but with the new capacitor and source values.

Circuit Description

```
e,1,0,1
o,1,2,o
c,2,o,1/4
r2,2,o,2
r1,2,0,1
```

The answer you want is `vc`, in **Results**. It reads

-2

This is the initial condition for the next interval. Again, we can limit the results to the one answer, `vc`.

Optional in Symbulator 9, which solves quickly enough: if you want to, tick **Do you want to limit the results to save time?** in **Settings** and list `vc` in the field it reveals.

Then we run the simulation for the second interval, just as we did before, but with the new capacitor and source values.

Circuit Description

```
e,1,0,3*u(t)
o,1,2,o
c,2,o,1/4,-2
r2,2,o,2
r1,2,0,1
```

Set **Analysis** to *TR — transient / time domain*.

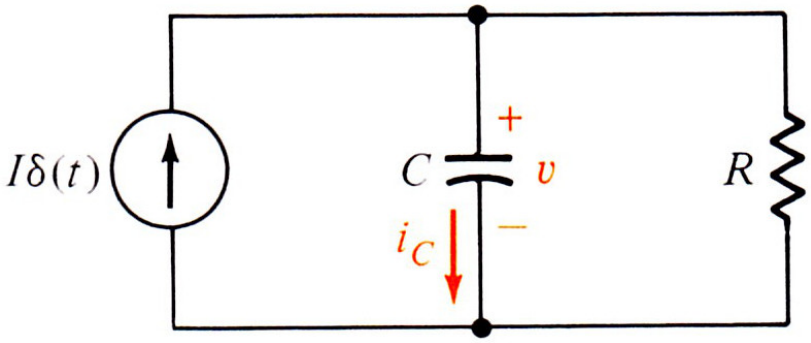
The answer you want is in **Results**:

voltage drop in c

$$v_c = 4e^{(-2t)} - 6$$

Bo2's Figure 5.38 (Impulse)

Find the voltage drop and current in the capacitor, given a source of I δ(t) A, where δ(t) is the impulse function and I is a constant.



As we mentioned before, impulse sources must be described using the δ(t) nomenclature. Typing `delta(t)` spells it without the Greek.

Circuit Description

```
j,0,1,i*δ(t)
c,1,0,c,0
r,1,0,r
```

Once it has run, the answers we want are:

voltage drop in c

$$v_c = (i/c)e^{(-t/(cr))}$$

current through c

$$i_c = (-i/(cr))e^{(-t/(cr))}$$

They are already in **Results**: `vc` is $\frac{ie^{-\frac{t}{cr}}}{c}$ and `ic` is $i\delta\left(t\right) - \frac{ie^{-\frac{t}{cr}}}{cr}$.

The impulse survives in the capacitor's current, which is right — the charge arrives all at once, and the delta is that instant.

Bo2's Drill Exercise 5.13 (Impulse)

For the circuit of **Bo2's Example 5.7 (page 175)**, find v_C(t), i_C(t) and v_o(t) due to an impulse voltage source of v_S(t)= δ(t) (that is to say, a 1V impulse in t=0.)

Nothing new here. Again we use the δ(t) nomenclature for the source.

Circuit Description

```
e,1,0,δ(t)
o,1,2,o
c,2,o,1/8,0
```

```
r2,2,o,2
r1,2,0,1
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

voltage drop in c

$$v_c = -8e^{-4t}$$

current through c

$$i_c = 4e^{-4t}$$

voltage of node o

$$v_o = 8e^{-4t}$$

Bo2's Figure 5.39 (Ramp)

Find i_L(t) and v_L(t) for a circuit that has in parallel a resistor R, an inductor L and a current source with a ramp value I r(t). Initial conditions are zero.

A ramp value I r(t) means a value of I t. We describe this source simply as `i*t`.

Circuit Description

```
j,0,1,i*t
r,1,0,r
l,1,0,l,0
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

current through l

$$i_l = ((il)/r)(e^{((-rt)/l)} - 1) + it$$

voltage drop in l

$$v_l = il(1 - e^{((-rt)/l)})$$

Bo2's Drill Exercise 5.14 (Ramp)

For the circuit of **Bo2's Example 5.7 (page 175)**, find v_C(t) and i_C(t) due to a ramp input voltage of v_S(t)= r(t).

Nothing new here. The description below should be clear to you by now.

Circuit Description

```
e,1,0,t
o,1,2,o
c,2,o,1/8,0
r2,2,o,2
r1,2,0,1
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

voltage drop in c

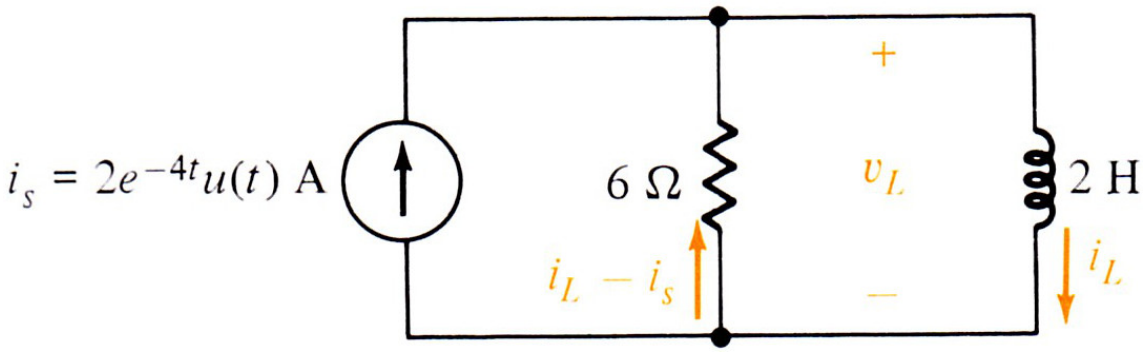
$$v_c = (1/2)(1 - e^{-4t}) - 2t$$

current through c

$$i_c = (1/4)(e^{-4t} - 1)$$

Bo2's Example 5.12 (Exponential)

Assume initial conditions zero. Find $\boldsymbol{i_L}(t)$ and $\boldsymbol{v_L}(t)$, given that $\boldsymbol{i_S}(t) = 2\,e^{-4t}$ A.



Describe the source as given to you by the problem, using an exponential.

Circuit Description

```
j,0,1,2e^(-4t)
r,0,1,6
l,1,0,2,0
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

current through l

$$i_l = 6e^{-3t} - 6e^{-4t}$$

voltage drop in l

$$v_l = 48e^{-4t} - 36e^{-3t}$$

Bo2's p265 (Exponential)

Repeat **Bo2's Example 5.12 (page 184)** if the current source is now $i_S(t) = 2\,e^{-3t}$ A.

All we do is change one digit in the circuit description: replace 4 with 3.

Circuit Description

```
j,0,1,2e^(-3t)
r,0,1,6
l,1,0,2,0
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

current through l

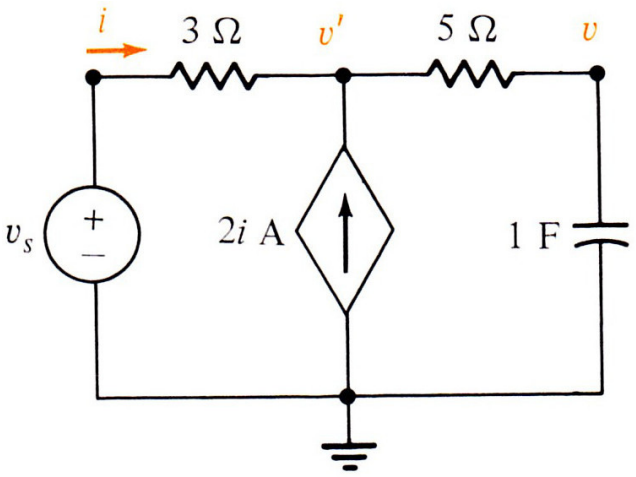
$$i_l = 6te^{-3t}$$

voltage drop in l

$$v_l = (12 - 36t)e^{-3t}$$

Bo2's Example 5.13 (Exponential)

Find the voltage drop in the capacitor and the current in the 3 Ω resistor, if the source value is $v_S(t)=18e^{-t/2}$.



Circuit Description

```
e,1,0,18e^(-t/2)
r3,1,2,3
r5,2,3,5
c,3,0,1,0
j,0,2,2*ir3
```

Set **Analysis** to *TR — transient / time domain*.

The answers you want are in **Results**:

> voltage drop in c
>
> $$v_c = 9e^{(-t/6)} - 9e^{(-t/2)}$$

> current through r3
>
> $$i_{r3} = (3/2)e^{(-t/2)} - (1/2)e^{(-t/6)}$$

Bo2's Drill Exercise 5.16 (Exponential)

For the circuit of **Bo2's Example 5.7 (page 175)**, find $v_C(t)$ and $i_C(t)$ for the case when the source's value is $v_S(t) = 2\,e^{-4t}$. As before, assume initial conditions are zero.

> Circuit Description
>
> ```
> e,1,0,2e^(-4t)
> o,1,2,o
> c,2,o,1/8,0
> r2,2,o,2
> r1,2,0,1
> ```

Set **Analysis** to *TR — transient / time domain*.

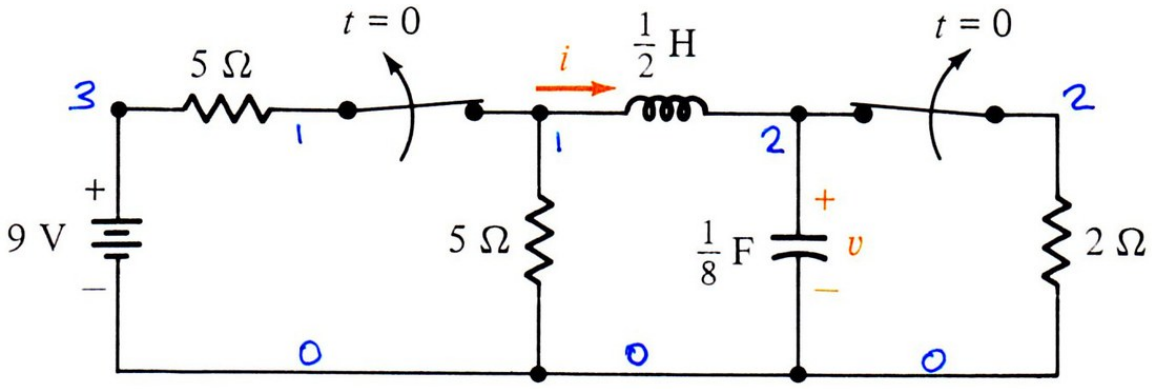
The answers you want are in **Results**:

> voltage drop in c
>
> $$v_c = -16te^{-4t}$$

> current through c
>
> $$i_c = (8t - 2)e^{-4t}$$

Bo2's Example 6.1

Find the current in the inductor $i(t)$ and the voltage drop in the capacitor $v(t)$.



Since we have switches changing in time t=0, this is a two interval problem. The first interval is for t<0. We can analyze the first interval using DC.

Circuit Description

```
e,3,0,9
r2,3,1,5
r1,1,0,5
l,1,2,1/2
c,2,0,1/8
r3,2,0,2
```

The answers you want are in **Results**:

voltage drop in c

$$v_c = 2\ \text{V}$$

current through l

$$i_l = 1\ \text{A}$$

These are the initial conditions for the second interval. The second is for t≥0. We can analyze the second interval using TR. We only care about `vc` and `il`.

Circuit Description

```
r1,1,0,5
l,1,2,1/2,1
c,2,0,1/8,2
```

voltage drop in c

$$v_c = 4e^{-2t} - 2e^{-8t}$$

current through l

$$i_l = 2e^{-8t} - e^{-2t}$$

Plotting an answer

Imagine, for example, that in **Bo2's Example 6.1 (page 186)** we are asked to plot ***i***(t) and ***v***(t) for times between 0 s and 1.5 s.

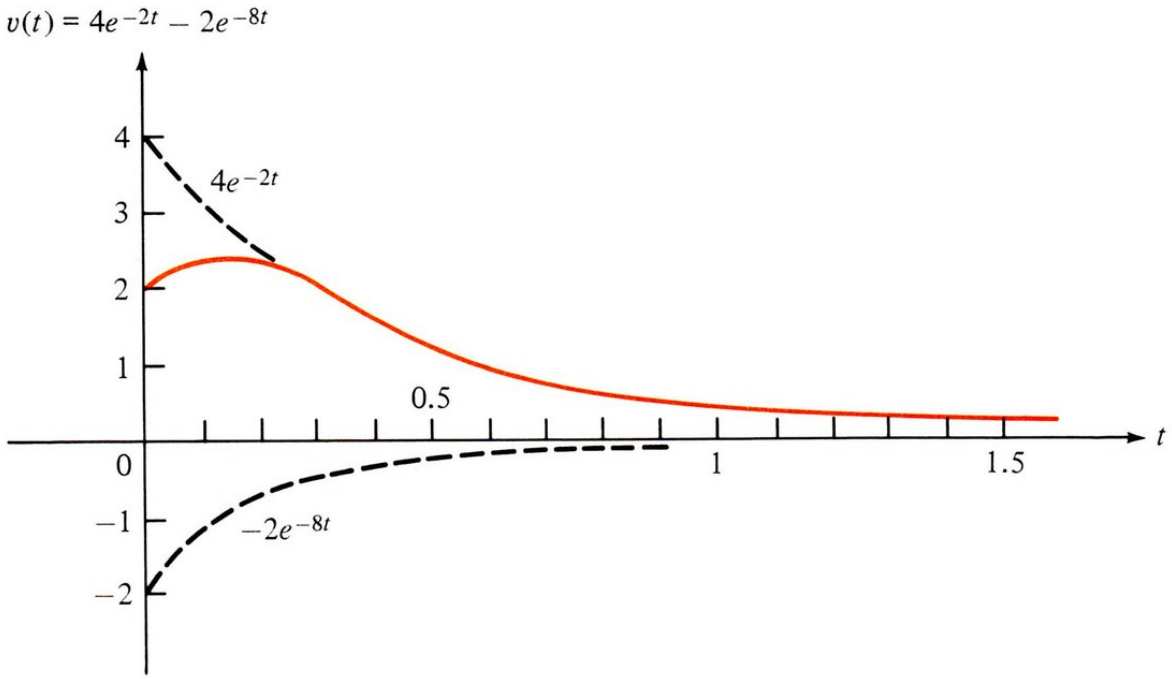
Open the **Plotting Tools** card, leave **Plot type** on *Plot a function of time (TR)*, and give it `vc` as the variable, `0` as the start and `1.5` as the end. The graph appears beneath.

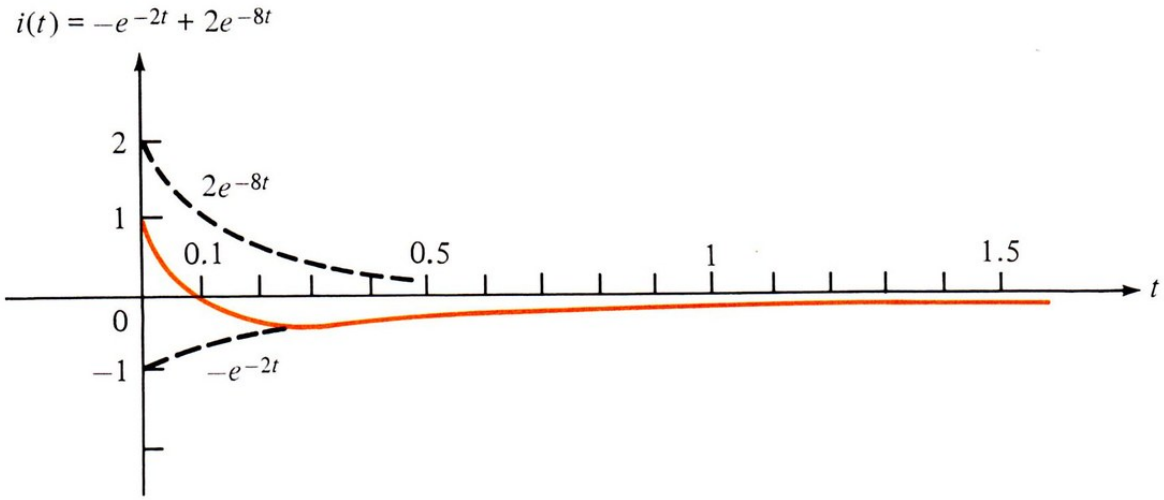
Compare it to the graph given by the textbook. Repeat this procedure giving `il` as the function. Compare the resulting graph to that given by the textbook.

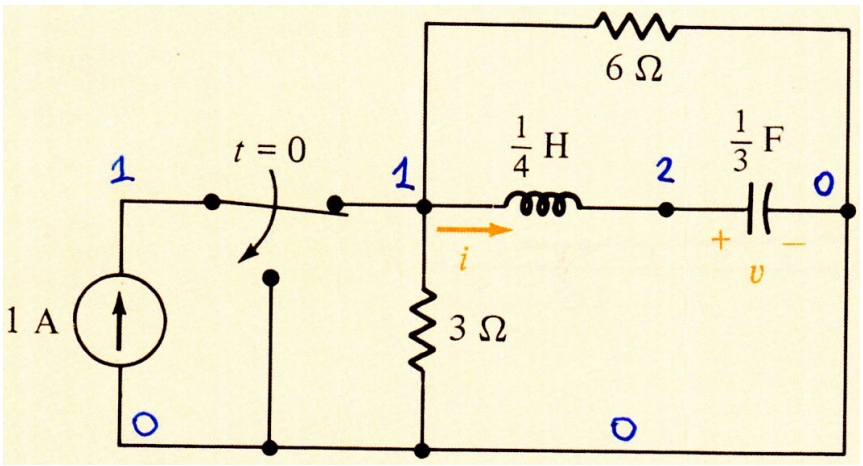
Bo2's Drill Exercise 6.1

For the circuit shown, find ***v***(t) and ***i***(t) for t≥0.

Nothing new here. This is a two interval problem, as others seen before. The first interval, for t<0, is analyzed in DC to find the initial conditions.


$v(t) = 4e^{-2t} - 2e^{-8t}$
4
3
2
1
0
−1
−2
$4e^{-2t}$
$-2e^{-8t}$
0.5
1
1.5
t


$i(t) = -e^{-2t} + 2e^{-8t}$
2
1
0
−1
$2e^{-8t}$
$-e^{-2t}$
0.1
0.5
1
1.5
t


6 Ω
$\frac{1}{4}$ H
$\frac{1}{3}$ F
t = 0
1
1
2
0
i
+
v
−
1 A
3 Ω
0
0

Circuit Description

```
j,0,1,1
r3,1,0,3
r6,1,0,6
l,1,2,1/4
c,2,0,1/3
```

The answers you want are in **Results**:

voltage drop in c

$$v_c = 2\ \text{V}$$

current through l

$$i_l = 0\ \text{A}$$

These conditions are used in the TR analysis of the second interval, for t≥0.

The answers are `vc` = $3e^{-2t} - e^{-6t}$ and `il` = $-2e^{-2t} + 2e^{-6t}$, both in **Results**.

Circuit Description

```
r3,1,0,3
r6,1,0,6
l,1,2,1/4,0
c,2,0,1/3,2
```

voltage drop in c

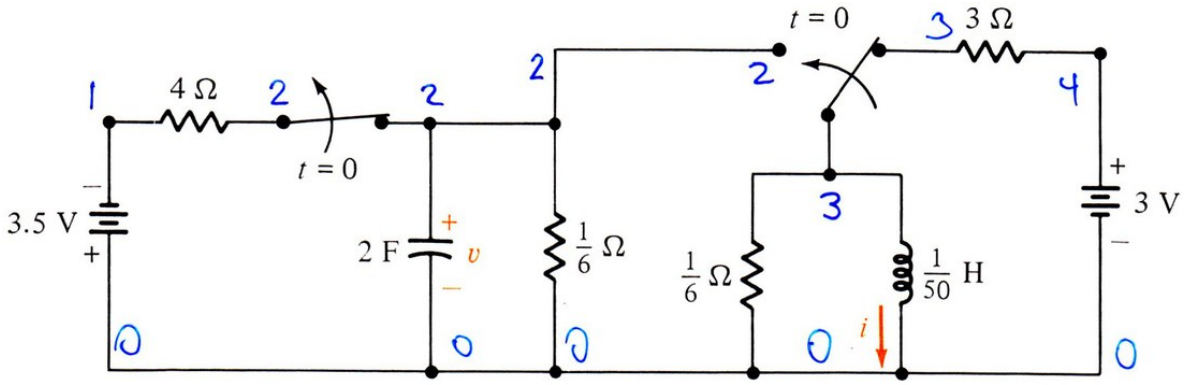
$$v_c = 3e^{-2t} - e^{-6t}$$

current through l

$$i_l = 2e^{-6t} - 2e^{-2t}$$

Bo2's Example 6.2

Find ***i***(t) and ***v***(t) for the circuit shown.



I decided to simulate this using only fractions, not decimals.

Write the value as the fraction it is, `7/2` rather than `3.5`: a fraction stays exact, a decimal point makes the arithmetic approximate.

7/2

Because of the switches, this is a two interval problem. The first interval is to be analyzed using DC. Notice that for t<0 the circuit is such that we basically have two separate circuits. We simulate each one of them separately:

Circuit Description

```
e,0,1,7/2
r1,1,2,4
c,2,0,2
r2,2,0,1/6
```

The answer you want is `vc`, in **Results**.

-7/50

Circuit Description

```
r3,3,0,1/6
l,3,0,1/50
r4,3,4,3
e2,4,0,3
```

The answer you want is `il`, in **Results**.

1

The second interval, for t≥0, is analyzed using TR. I could have described the circuit again from scratch, but out of laziness I copied the descriptions from the DC simulations. To avoid renaming nodes, I simulated the right switch as a short between nodes **2** and **3**.

`vc` comes back $\frac{(-\sin(4t) - 7\cos(4t))\, e^{-3t}}{50}$ and `il` $(-\sin(4t) + \cos(4t))\, e^{-3t}$.

Circuit Description

```
c,1,0,2,-7/50
r,1,0,[1/6,1/6]
l,1,0,1/50,1
```

voltage drop in c

$$v_c = (-7/50)e^{-3t}\cos(4t) - (1/50)e^{-3t}\sin(4t)$$

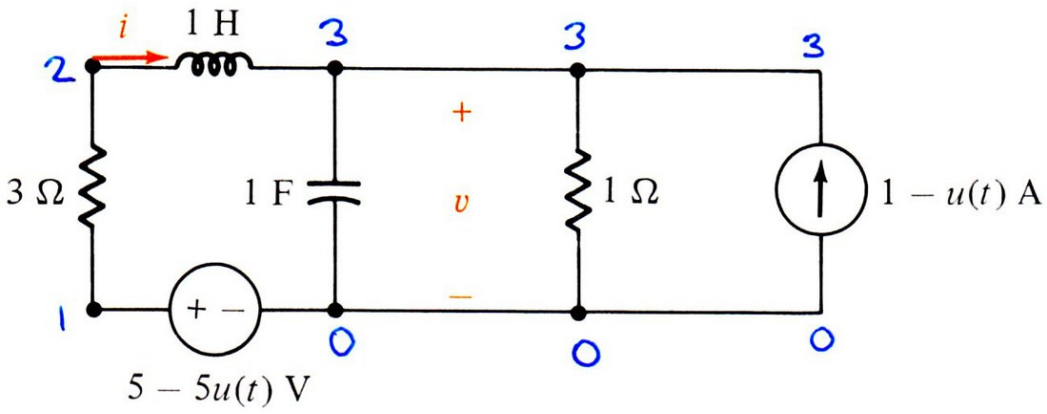
current through l

$$i_l = e^{-3t}\cos(4t) - e^{-3t}\sin(4t)$$

These are the right answers.

Bo2's Example 6.3

Find ***i***(t) and ***v***(t) for the circuit shown.



Since the sources change value, this is a two interval problem. For t<0, in DC:

Circuit Description

```
e,1,0,5
r3,1,2,3
l,2,3,1
c,3,0,1
r1,3,0,1
j,0,3,1
```

The answers you want are in **Results**:

voltage drop in c

$$v_c = 2\ \text{V}$$

current through l

$$i_l = 1\ \text{A}$$

Using these as initial conditions, now we use TR for t≥0:

Circuit Description

```
r3,0,2,3
l,2,3,1,1
c,3,0,1,2
r1,3,0,1
```

voltage drop in c

$$v_c = (3t + 2)e^{-2t}$$

current through l

$$i_l = (1 - 3t)e^{-2t}$$

Is the textbook wrong?

Anybody who has written a textbook, or read one for that matter, can attest that errors in the problems or the answers come with the territory. (I am sure this book has many.) We are all only human. But errors in an engineering textbook can be *very* frustrating to a student: confusion and hair-pulling ensue. If you think you have found one in your circuit analysis textbook, Symbulator can help you confirm whether your guess is right.

Below is an example of using Symbulator to confirm an error in a textbook.

Bo2's Drill Exercise 6.4

DRILL EXERCISE 6.4

For the *RLC* circuit shown in Fig. DE6.4, suppose that $v_s(t) = 2 - 2u(t)$ V. Find $i(t)$ and $v(t)$ for $t \geq 0$.
Answer: $8te^{-4t} + 2e^{-4t}$ A; $-16te^{-4t}$ V

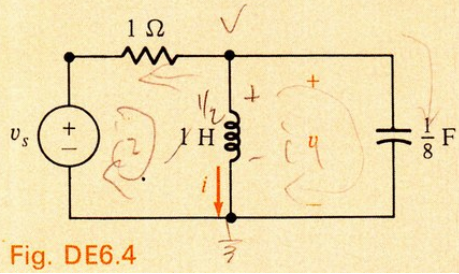


Fig. DE6.4

My copy of Bo2 is second-hand, and this problem had scribbles on it. The previous owner evidently struggled with it and concluded that the inductor had to be ½ H, not the 1 H the schematic says. Symbulator can show he is right and the book wrong. First, solve the circuit as printed, with a 1 H inductor.

Circuit Description

```
e,2,0,2
r,1,2,1
l,1,0,1
c,1,0,1/8
```

The answers you want are in **Results**:

current through l

$$i_l = 2 \text{ A}$$

voltage drop in c

$$v_c = 0 \text{ V}$$

Circuit Description

```
r,1,0,1
l,1,0,1,2
c,1,0,1/8,0
```

The answer we get doesn't match the answer provided by the book. Now let's repeat the TR simulation using the ½ H value for the inductor:

Circuit Description

```
r,1,0,1
l,1,0,1/2,2
c,1,0,1/8,0
```

Now the answer matches the book's. An expert user can reach for FD analysis and the `t2s` tool, which we have not discussed yet, to find the inductor's value from the book's answer. This is how I found the ½ H myself, working backwards:

Evaluate

```
t2s(-16*t*e^(-4*t))
```

which gives $-\frac{16}{(s+4)^2}$.

-16/(s+4)^2

2*(s+8)/(s+4)^2

Circuit Description

```
r,1,0,1
l,1,0,l,2
c,1,0,1/8,0
```

Set **Analysis** to *FD — complex frequency domain*.

The answer you want is `vc`, in **Results**.

-16*l/(l*s^2+8*l*s+8)

That is what the **Solve** card is for — a system that is not a circuit:

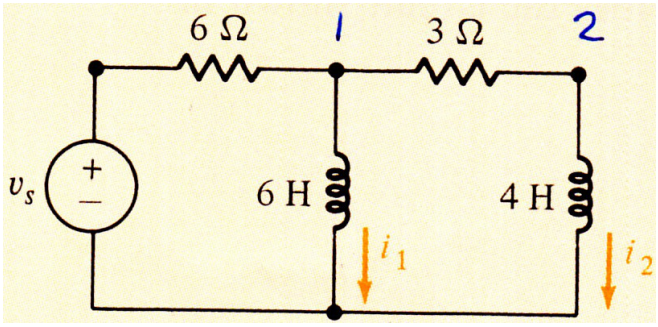
Equation(s) to solve in terms of the results

```
-16/(s+4)^2 = -16*l/(l*s^2+8*l*s+8)
```

with `l` as the unknown. It answers `l` = **1/2** H.

l=1/2

FD analysis and the `t2s` tool are used for frequency domain analysis, and will be discussed separately in a subsequent part of Symbulator's documentation.



Bo2's Drill Exercise 6.5

In the circuit, suppose that $v_S(t) = 12 - 12\ u(t)$ V. Find $i_2(t)$ for $t \geq 0$.

Circuit Description

```
e,3,0,12
r6,3,1,6
l1,1,0,6
r3,1,2,3
l2,2,0,4
```

The answers you want are in **Results**:

current through l1

$$i_{l1} = 2\ \text{A}$$

current through l2

$$i_{l2} = 0\ \text{A}$$

Circuit Description

```
r6,0,1,6
l1,1,0,6,2
r3,1,2,3
l2,2,0,4,0
```

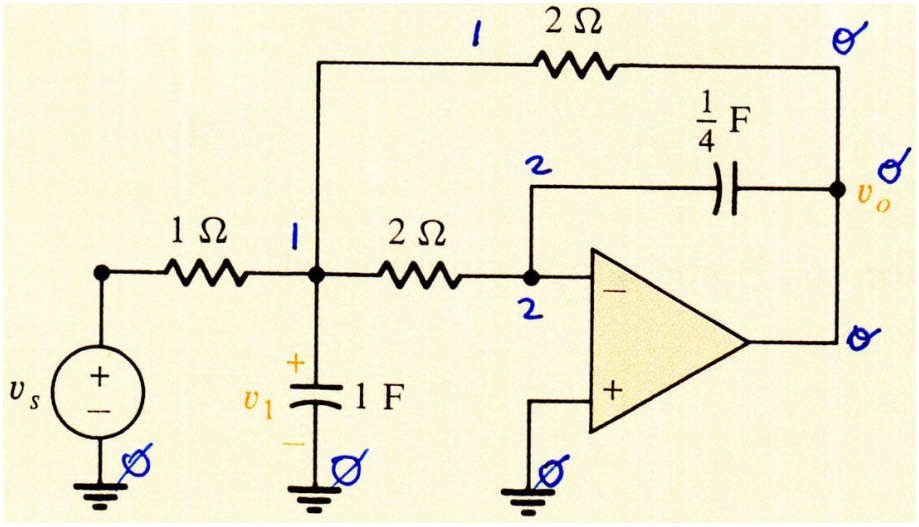
current through l2

$$i_{l2} = (12/11)(e^{-3t} - e^{-t/4})$$

Bo2's Drill Exercise 6.6 (Op Amp)

In the circuit, suppose that $v_S(t) = 2 - 2\ u(t)$ V. Find $v_o(t)$ for $t \geq 0$.

Simulate the first interval in DC to find the initial conditions of the capacitors.



Circuit Description

```
e,3,0,2
r1,3,1,1
r2,1,2,2
r3,1,o,2
ca,1,0,1
cb,2,o,1/4
o,0,2,o
```

The answers you want are in **Results**:

voltage drop in ca

$$v_{ca} = 0 \text{ V}$$

voltage drop in cb

$$v_{cb} = 4 \text{ V}$$

Then simulate the second interval in TR, to find the voltage in node *o*.

voltage of node o

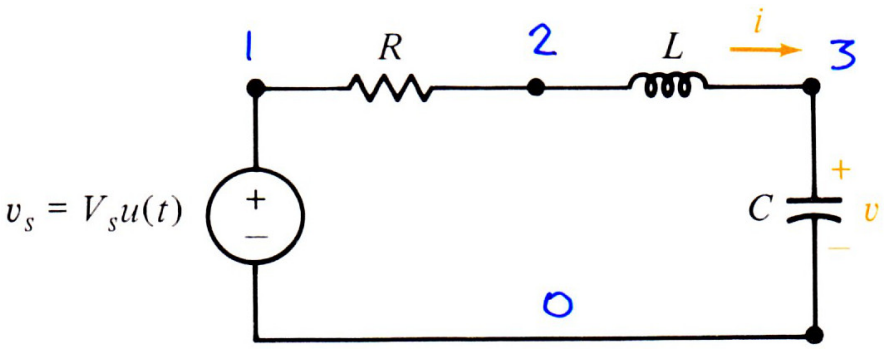
$$v_o = (-4t - 4)e^{-t}$$

Circuit Description

```
r1,0,1,1
r2,1,2,2
r3,1,o,2
ca,1,0,1,0
cb,2,o,1/4,4
o,0,2,o
```

Bo2's Example 6.5 (Plot)

In the circuit, use v_S=2/5V, R=12Ω, L=2H and C=1/50F. Find v(t) and i(t), and plot them for time 0<t<1.5 seconds.



Circuit Description

```
e,1,0,2/5*u(t)
r,1,2,12
l,2,3,2,0
c,3,0,1/50,0
```

voltage drop in c

$$v_c = -\tfrac{2}{5}e^{-3t}\cos(4t) - \tfrac{3}{10}e^{-3t}\sin(4t) + \tfrac{2}{5}$$

current through l

$$i_l = (1/20)e^{-3t}\sin(4t)$$

`vc` is the first of those and `il` the second.

These are the right answers. To plot them, open **Plotting Tools**, enter `vc` as the variable, `0` as the start and `1.5` as the end. Compare to the graph from the book. Repeat the plot for `il`, and compare that one to the book's graph too.

Bo2's Drill Exercise 6.7

For the circuit in **Bo2's Example 6.5 (page 196)**, use v_S=3V, R=5Ω, L=1/2H and C=1/8F. Find v(t) and i(t).

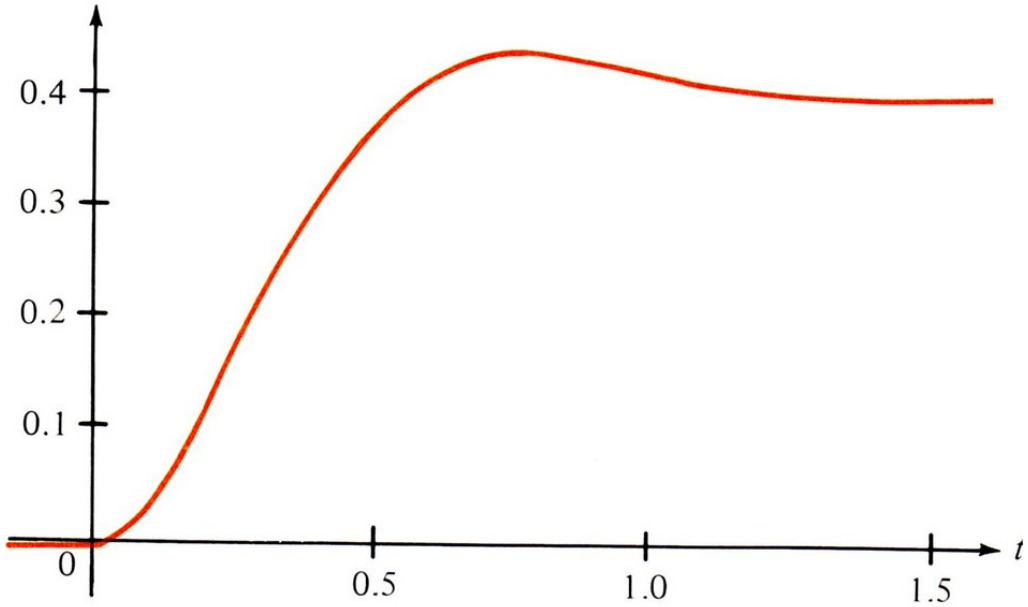
Circuit Description

```
e,1,0,3*u(t)
r,1,2,5
l,2,3,1/2,0
c,3,0,1/8,0
```

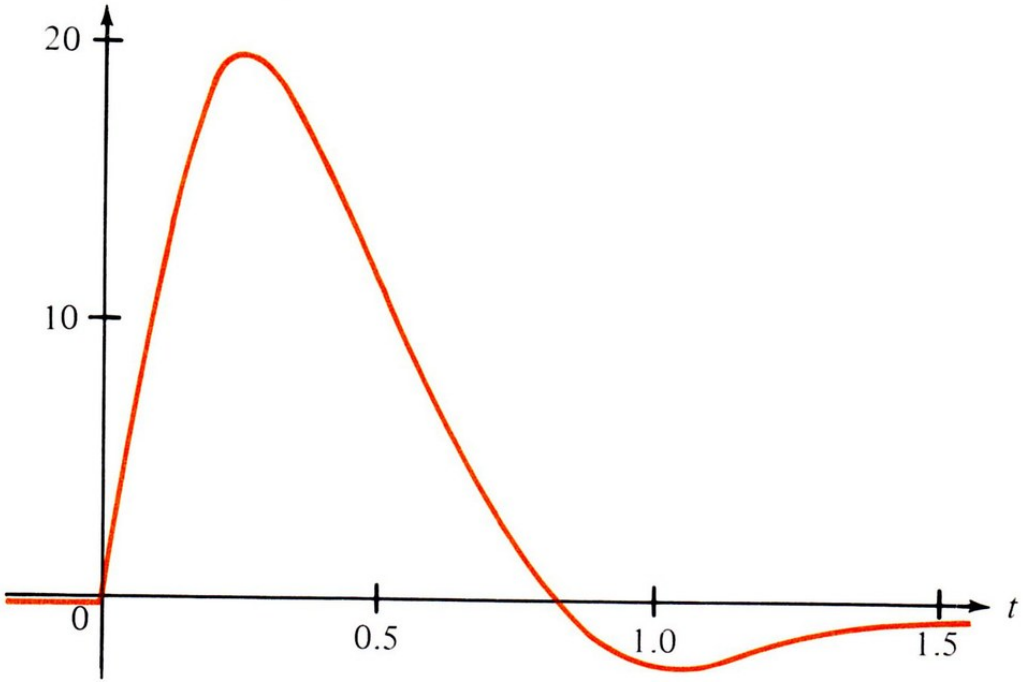
voltage drop in c

$$v_c = -4e^{-2t} + e^{-8t} + 3$$

$v(t) = [0.4 - 0.5\,e^{-3t}\cos(4t - 0.64)]\,u(t)$



$i(t) = [0.05\,e^{-3t}\sin 4t]\,u(t)$



> current through l
>
> $$i_l = e^{-2t} - e^{-8t}$$

Bo2's Drill Exercise 6.8 (Ramp)

For the circuit in **Bo2's Drill Exercise 6.7 (page 196)**, find the voltage drop in the capacitor v(t) if the source has a value v_S(t) = 3 r(t).

Circuit Description

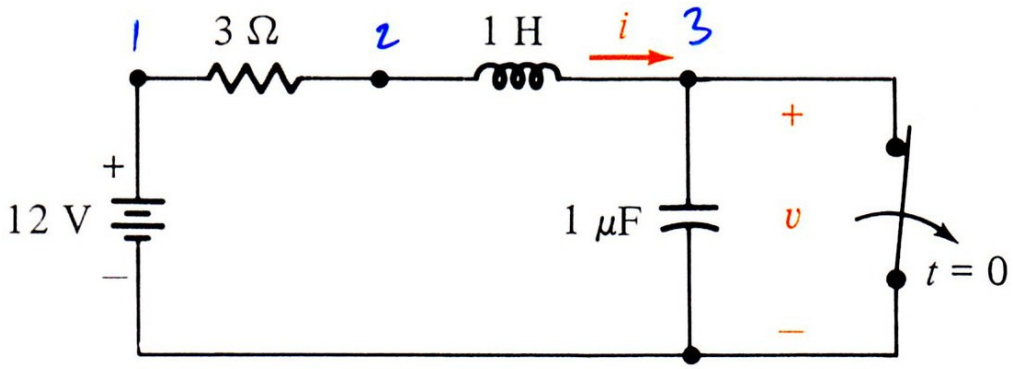
```
e,1,0,3t
r,1,2,5
l,2,3,1/2,0
c,3,0,1/8,0
```

voltage drop in c

$$v_c = 2e^{-2t} - (e^{-8t})/8 + 3t - 15/8$$

Bo2's Example 6.6 (Plot)

For the circuit, plot the voltage drop in the inductor for 0 < t < 5ms.



First, find the initial conditions for t<0 by running a DC simulation.

Circuit Description

```
e,1,0,12
r,1,2,3
l,2,3,1
c,3,0,1'μ
s1,3,0
```

The answers you want are in **Results**:

current through l

$$i_l = 4\ \text{A}$$

voltage drop in c

$$v_c = 0\ \text{V}$$

Then, find the `v1` for t≥0 by running a TR simulation.

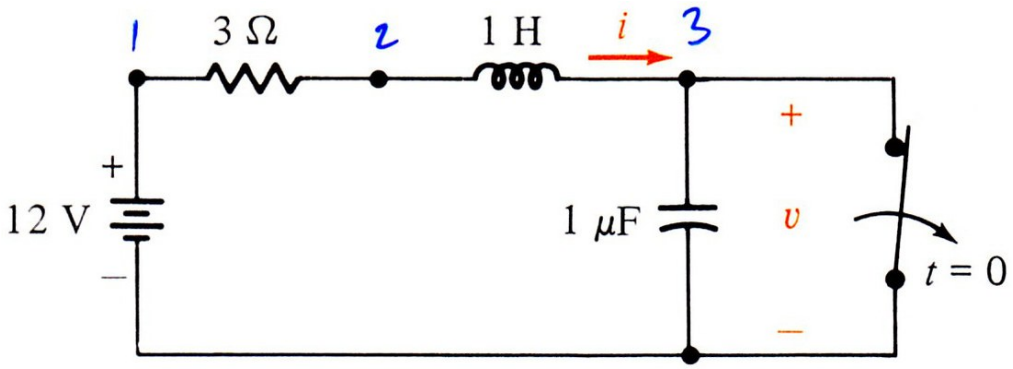
Circuit Description

```
e,1,0,12*u(t)
r,1,2,3
l,2,3,1,4
c,3,0,1'μ,0
```

Finally, plot `v1` in **Plotting Tools** from `0` to `0.005` s. In the resulting plot you will see something outstanding: around t = 1.57 ms, the voltage drop across the inductor is 3991 volts!

Bo2's Drill Exercise 6.9

For the circuit, R=1Ω, L=2H, C=1/2F and i_S(t)=u(t). Find i(t) and v(t).



Circuit Description

```
j,0,1,u(t)
r,1,0,1
l,1,0,2,0
c,1,0,1/2,0
```

current through l

$$i_l = (-t-1)e^{-t} + 1$$

voltage drop in c

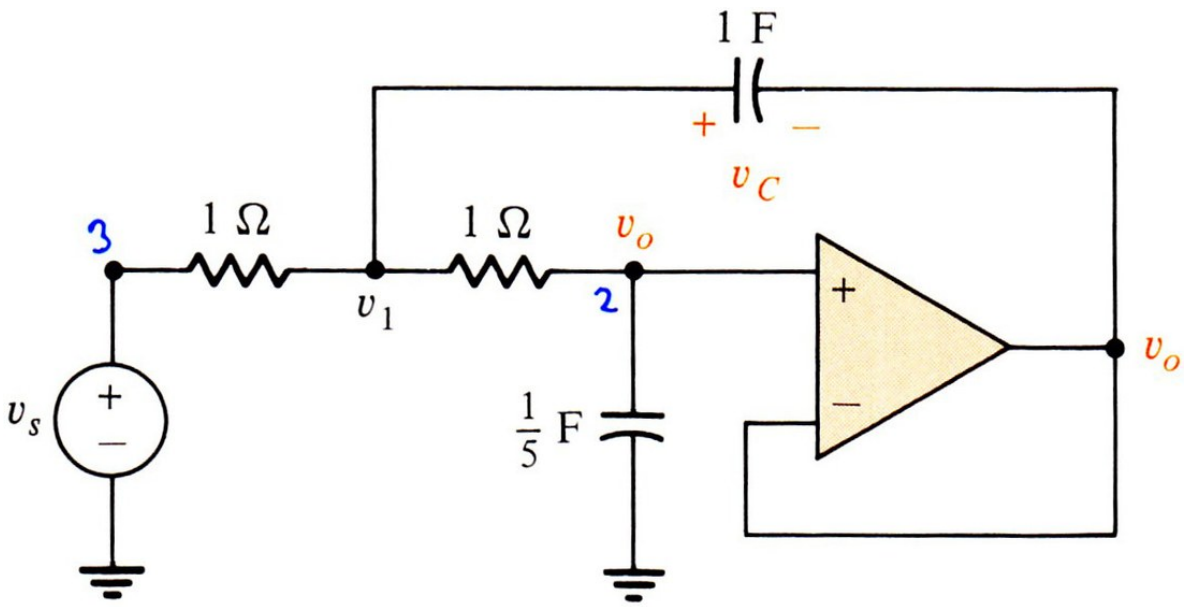
$$v_c = 2te^{-t}$$

Bo2's Figure 6.23 (Op Amp)

For the circuit, find v_o(t) if v_S(t) = u(t).

Circuit Description

```
e,3,0,u(t)
r1,3,1,1
r2,1,2,1
ca,2,0,1/5,0
cb,1,o,1,0
o,2,o,o
```



voltage of node o

$$v_o = 1 - e^{-t}\cos(2t) - (1/2)e^{-t}\sin(2t)$$

Bo2's Drill Exercise 6.10 (Op Amp)

In the previous circuit, change the capacitor's value from 1/5 F to 25/16 F. Find $\boldsymbol{v_o}(t)$ if $\boldsymbol{v_S}(t) = 3\ u(t)$.

Circuit Description

```
e,3,0,3*u(t)
r1,3,1,1
r2,1,2,1
ca,2,0,25/16,0
cb,1,o,1,0
o,2,o,o
```

$-4e^{-2t/5}+e^{-8t/5}+3$

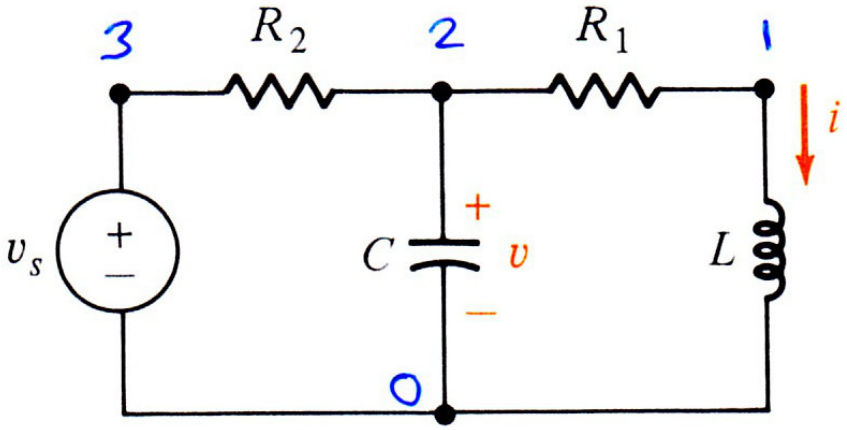
Bo2's Drill Exercise 6.11 (p307)

For the circuit, suppose that $\boldsymbol{R_1} = \boldsymbol{R_2} = 1\Omega$, $\boldsymbol{L} = 1\text{H}$, $\boldsymbol{C} = 1\text{F}$ and $\boldsymbol{v_S}(t) = 2e^{-2t}\ u(t)$. Find $\boldsymbol{i}(t)$ if all initial conditions are zero.

Circuit Description

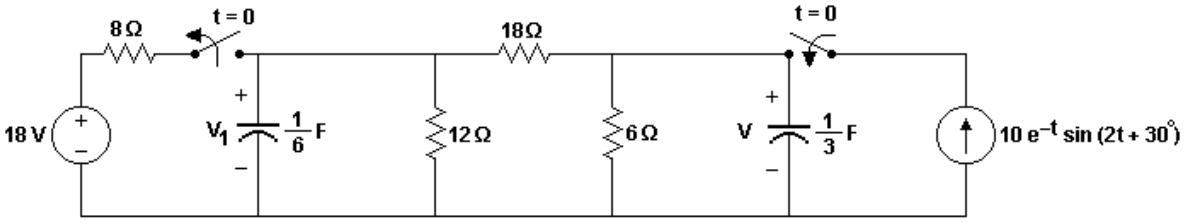
```
e,3,0,2e^(-2t)
r2,3,2,1
r1,2,1,1
c,2,0,1,0
l,1,0,1,0
```

$-(e^t\cos(t)-e^t\sin(t)-1)\ e^{-2t}$



A more complex problem

The circuit comes from the circuits analysis class of Professor Eliane Boulet de Cabrera, from Universidad Tecnologica de Panama. Find v_1 and v.



Because of the switches, this is a two interval problem. The first is run in DC.

Circuit Description

```
e,3,0,18
r8,3,1,8
ca,1,0,1/6
r12,1,0,12
r18,1,2,18
r6,2,0,6
cb,2,0,1/3
```

The answers you want are in **Results**:

voltage drop in ca

$$v_{ca} = 9 \text{ V}$$

voltage drop in cb

$$v_{cb} = 9/4 \text{ V}$$

These serve as initial conditions for the second interval, which is run in TR. Since we only want v1 and v2, it pays off to let Symbulator know this:

Circuit Description

```
ca,1,0,1/6,9
r12,1,0,12
r18,1,2,18
r6,2,0,6
cb,2,0,1/3,9/4
j,0,2,10e^(-t)*sin(2t+30*pi/180)
```

Set **Analysis** to *TR — transient / time domain*.

voltage of node 1

$$\begin{aligned} v_1 = {} & (5\sqrt{3}/17 - 20/17)e^{-t}\cos(2t) \\ & + (-20\sqrt{3}/17 - 5/17)e^{-t}\sin(2t) \\ & + (80\sqrt{3}/17 + 193/34)e^{-t/2} + (9/2 - 5\sqrt{3})e^{-t} \end{aligned}$$

voltage of node 2

$$\begin{aligned} v_2 = {} & (-245\sqrt{3}/34 - 20/17)e^{-t}\cos(2t) \\ & + (245/34 - 20\sqrt{3}/17)e^{-t}\sin(2t) \\ & + (80\sqrt{3}/17 + 193/34)e^{-t/2} + (5\sqrt{3}/2 - 9/4)e^{-t} \end{aligned}$$

Symbulator 9 answers this one in about a second; `v1` and `v2` are in **Results**.

These are the correct answers, which explains why Professor Boulet – who made the problem up - is a living legend among circuit students at UTP.

Advanced use of Expert Mode in TR

Expert Mode can be very useful in transient analysis. Using it, however, requires some knowledge. Here's two things you need to know in order to use **Expert Mode** like a boss:

First, when Symbulator solves a problem using TR, it follows these general steps:

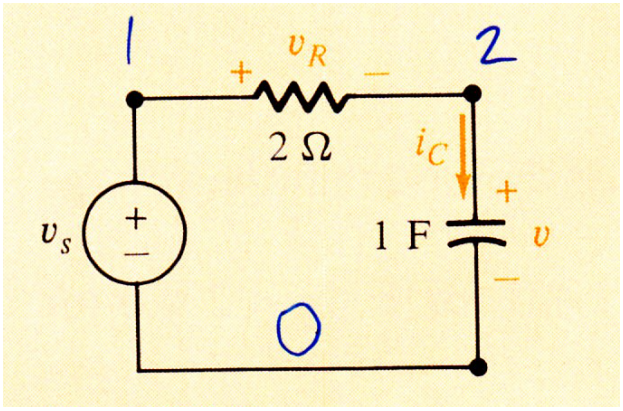
- Move every independent source into the frequency domain, and generate a set of equations and unknowns for the system there.
- Solve these frequency domain equations.
- Convert the answers back to the time domain.

Second, when Symbulator solves a problem in **Expert Mode**, it freezes this process halfway between steps 1 and 2, so that you can tinker with the equations and unknowns before they are solved.

In version 9, what you *type* is read in the **time domain** and converted on the way in, though the equations are still solved in the frequency domain. Write the answer you know the way you would on paper. That covers everything you add: equations, conditions and the expressions in them. A relation between plain parameters, such as `x = 3`, is left alone, having nothing to transform.

Bo2's Drill Exercise 4.5 (Expert)

For this circuit, we know that the capacitor's initial condition is zero, that v_s is an unknown step source (i.e. of the form $A\ u(t)$, where A is a constant value in volts) and that the voltage drop in the capacitor for t>0 is found to be 1-e^(-t/2). Find the voltage drop in the resistor, the current through the capacitor and the value of the source.



This problem is a match made in heaven for the **expert** tool, because we have one unknown value in the circuit (i.e. the value of the step source) and we have one known answer (i.e. the voltage drop in the capacitor.) So, the game plan here is to run this circuit through Symbulator's **Expert Mode**, add one new equation and one new unknown, and then solve. First, let's generate the new equation.

The new equation is simply `1-e^(-t/2) = vc`: **Expert Mode** reads it in the time domain, like the answers on screen.

If you would rather convert by hand, **t2s** is still there and an equation already written in s is left alone rather than transformed twice:

Evaluate
```
t2s(1-e^(-t/2))
```

which gives $\frac{1}{s\,(2s+1)}$ — the same statement, one domain over.

Now let's run the Expert simulation of the circuit. Let's define the value of the source as `a*u(t)`, since we know it's a step source; the variable `a` will serve as the unknown value, for which we will solve in **Expert Mode**.

Circuit Description
```
e,1,0,a*u(t)
r,1,2,2
c,2,0,1,0
```

Choose TR, then open **Expert Mode**. **Add equation(s)** takes one per line:

Add equation(s)
```
1-e^(-t/2) = vc
```

Add unknown(s) takes the name:

> Add unknown(s)
> `a`

Let Symbulator solve the equations now. Once it is done, we want to do a quick sanity check. If the system solved correctly, we should see that Symbulator has as voltage drop in the capacitor the expression we already know from the problem statement:

`vc` in **Results** reads $1 - e^{-\frac{t}{2}}$, which it does.

This is correct, and the answers we want are on screen with it: `ic` is $\frac{e^{-\frac{t}{2}}}{2}$, and the voltage drop `vr` is $e^{-\frac{t}{2}}$.

These are the correct answers. The value of the step source is 1 volt.

Bo2's Drill Exercise 4.13 (Expert)

For the circuit shown in Fig. DE4.13, the voltage across the 1-Ω resistor is $v_R(t) = e^{-t}$ V for $t \geq 0$. Given that $v_C(0) = 0$ V, find the following functions for $t \geq 0$: (a) $i_R(t)$; (b) $v_C(t)$; and (c) $v_s(t)$.
Answer: (a) e^{-t} A; (b) $1 - e^{-t}$ V; (c) 1 V

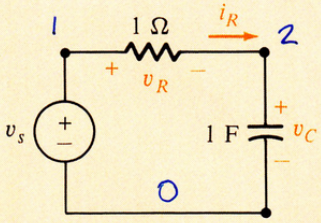


Fig. DE4.13

This problem is solved just like the one we already saw. First, find the new equation:

No transform this time either — write what you know:

> Add equation(s)
> `vr = e^(-t)`

Then run the Expert simulation. I decided to use `vs` as the step value of the source:

> Circuit Description
> ```
> e,1,0,vs*u(t)
> r,1,2,1
> c,2,0,1,0
> ```

Choose TR, and put `vs` in **Add unknown(s)** beside the equation above.

Run it. The sanity check is already on screen: `vr` reads e^{-t}, which is what we told it.

And the answers with it: `ir` is e^{-t} and the source's value `vs` is **1** V.

Ain't that a beauty? Yes, ma'am! Atta boy, Symbulator! Atta boy!

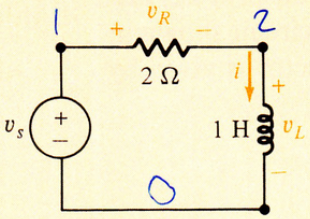
Bo2's 164 (Expert)

For the circuit shown in Fig. DE4.2, the current through the inductor is

$$i(t) = \begin{cases} 0 & \text{for } -\infty < t < 0 \\ 1 - e^{-2t} & \text{for } 0 \le t < \infty \end{cases}$$

Find, for $t \ge 0$, (a) $v_L(t)$, (b) $v_R(t)$, (c) $v_s(t)$, and (d) the power absorbed by the inductor.

Answer: (a) $2e^{-2t}$ V; (b) $2(1 - e^{-2t})$ V; (c) 2 V; (d) $2(e^{-2t} - e^{-4t})$ W



This is the same thing as the previous two, so there is no need for commentary.

Circuit Description

```
e,1,0,vs*u(t)
r,1,2,2
l,2,0,1,0
```

Choose TR, put `vs` in **Add unknown(s)**, and give **Add equation(s)** the answer you know — in time, as always here:

Add equation(s)

```
1-e^(-2*t) = il
```

Run it, then put the sanity check into **Evaluate**:

It does add up — `il` reads $1 - e^{-2t}$ — and the answers are on screen with it: `vl` is $2e^{-2t}$, `vr` is $2 - 2e^{-2t}$, and the source's value `vs` is **2** V.

This is the end of the TR section.

LESSON 7

Alternating current analysis

Learn how to run an *alternating current* analysis (via phasor analysis). Learn how to describe capacitors, inductors, *impedances and admittances*, and sources for AC analysis.

In this lesson you will learn how to use the **AC** analysis to solve AC circuits in Symbulator. You will also learn how to describe the elements you already know so they suit AC analysis, and how to find equivalents in AC.

7.1 AC analysis in Symbulator

The main thing to know is that in AC mode Symbulator understands phasors — complex numbers. It uses them in the input, in the analysis and in the output.

> CAUTION
> **Symbulator does not understand sinusoids**
> You should not feed Symbulator any sinusoids, or expect any sinusoid answers from it. Any conversion to and from sinusoids has to be done by you.

The AC analysis

Set **Analysis** to *AC — alternating current*. Unlike DC and TR, it needs one more input: the frequency in radians per second, typed into the ω **— angular frequency** box beside the menu.

AC mode in the other tools

The equivalence tools work in AC the same way: choose *Find equivalent* as usual, set **Analysis** to *AC — alternating current*, and give the ω box a frequency where the circuit needs one.

The Mini-Tools card

One more thing before the first AC problem, because AC is where it earns its place. Under the results sits the **Mini-Tools** card: four small calculators that work on values you type rather than on the circuit, and are ready before and after any solve. Choose one in its **Tool** menu, fill in the boxes it shows, and press **Run**:

- **aa — amplitude and angle** turns a complex value into polar form. It takes one value:

an answer's name, such as `ir1`, or any expression, such as `v1 - v2`. This lesson uses it throughout.

- **pf — power factor** takes one value: a complex power, such as `se`, read as given, or the name of an element, such as `e`, read on the power a source delivers or an impedance consumes. It answers with the power factor and whether it leads or lags, and for a name says which power it read. Lesson 8 (page 222) introduces it.
- **gain — voltage, current and power gain** takes the voltage and current at a two-port's input and at its output. Lesson 13 (page 277) introduces it.
- **pz — poles and zeros** takes one value: a transfer function, such as `v2/vg` after an FD run with the source left as a symbol, or any ratio of polynomials in *s*. It answers with the roots of the denominator, the poles, and the roots of the numerator, the zeros, after cancelling any factor the two share.

They are mini-tools, not analyses: none of them solves the circuit, and none needs the circuit to have been solved, though an answer's name only means something once there are answers to read.

Describing elements for AC

When describing a circuit for AC analysis, element values have to be written the way this analysis expects. Valueless elements, like the short circuit and the op amp, are unchanged. Here's how to describe the ones with values:

- Current sources **j** and voltage sources **e** accept complex values. You can declare their value in rectangular form, e.g. `10-3i`, or in angular form, e.g. `(100∠120°)`.
- The element **r** no longer represents just a resistor: it now represents an **impedance**. If you are wondering why we didn't use the letter z, wait until we get to two-ports. In AC analysis, r accepts complex numbers as its value, such as `10-3i`. We use r to describe resistors (real values) and conductances (real values, inverted), but also impedances (complex values) and admittances (the inverse of their complex value).
- The elements **c** and **l** are used to describe capacitors and inductors *only* when their values are given in farads and henries, respectively. For AC analysis their description no longer requires the fifth field for the initial condition, since alternating current analysis focuses on the steady state after transient effects are long gone. So we only use the name, the two nodes, and the value.

> ⚠ WARNING
>
> **If it's in Ω, it's an r**
>
> When the problem gives us the values of capacitors and inductors in ohms, we **must** describe them as impedances. If a capacitor has its value in farads, describe it with **c**. If an inductor has its value in henries, describe it with **l**. But anything that has its value in ohms — resistor, impedance, capacitor or inductor — must be described using **r**.

The values of all of these elements can use SI prefixes.

What answers do you get

You get the same answers you did in DC, except that now they are phasors. The one difference is that now, besides the real power consumed, you also get the complex power consumed. We will discuss this in Lesson 8 (page 222).

Rectangular or polar

A phasor can be written two ways, and circuit problems use both: in *rectangular* form as `3 + 4j`, or in *polar* form as an amplitude and an angle, $5\angle 53.13°$. They are the same number. Which one you want depends on the question — rectangular adds and subtracts easily, polar multiplies and divides easily, and textbook answers are usually quoted in polar.

Symbulator answers in rectangular form unless you ask otherwise. In **Settings**, under *Display*, tick **Show AC answers as polar phasors** and every answer is shown as an amplitude and an angle. The circuit is re-solved as soon as you tick it.

Three things it leaves alone:

- **Average power stays a plain number.** It is real, so an angle would be meaningless. Complex power *is* converted: its polar form is the apparent power and the power-factor angle. See Lesson 8 (page 222).
- **Symbolic answers are left as they are.** There is no angle to take of $v_{in}r_b/(r_a + r_b)$.
- **It applies to AC only.** DC answers are real, and TR and FD answers are functions of t or s. The checkbox greys out there.

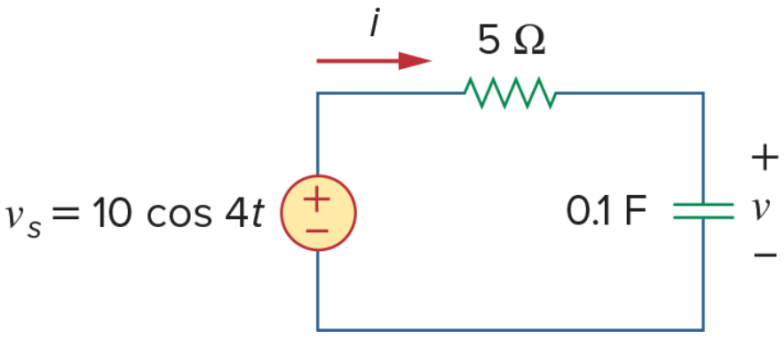
An answer that happens to be real still gets its angle, 0° or 180°, as the *aa* mini-tool does.

> TIP
>
> **It combines with SI prefixes**
>
> With SI prefixes on too, the amplitude carries the prefix and the angle stays in degrees: `632.5 m∠108.4°`.
> A polar answer is always a decimal, even under *exact*: an angle in degrees has no closed form to preserve.

The *aa* mini-tool is still the better choice for converting one value, or an expression that is not an answer by itself, such as a difference between two node voltages. Lesson 9 (page 230) uses it that way throughout.

7.2 Solved numerical examples

With values in F and H



AS7's Example 9.9

AS7's Example 9.9

Find ***v***(t) and ***i***(t).

SOLUTION

Since this is the first AC problem we will solve, I will explain every step, including the manual ones, in full detail.

Step 1: sinusoids to phasors. We have to express the value of the source in terms Symbulator can understand, so we convert it from a sinusoid to a phasor: $10\angle 0°$. We can pass this as `(10∠0°)` or, since the angle is zero, simply as `10`.

Step 2: note the frequency. The circuit has a capacitor with a value given in farads. Because of this, Symbulator will need the frequency, given by the problem as 4 rad/s, to convert that value into an impedance in ohms.

Step 3: describe the circuit.

> Circuit Description
> ```
> e,1,0,10
> r1,1,2,5
> c,2,0,.1
> ```

Choose *AC — alternating current*, and put **4** in the ω **— angular frequency** box that appears beside it.

Notice that, in our circuit description, the order we give to the nodes of the resistor and the capacitor is chosen to be convenient for the answers we will read.

Step 4: read the answers. Once Symbulator is done, you should take a look at what it found:

- the usual voltages in the nodes, voltage drops in the elements, and currents through the elements, under the names that should be familiar by now
- the *average power consumed* in the source and the resistor, in `ape` and `apr1`. None is given for the capacitor, since capacitors and inductors do not consume real power.
- the *complex power consumed* in all elements, in `sc`, `se` and `sr1`
- the *equivalent impedance* of the rest of the circuit as seen by the source, in `ze`

The answers this problem wants are `ir1` and `vc`. It is likely that Symbulator will give you the answers in rectangular form, with a real part and an imaginary part.

If we want to see them as an amplitude and angle, open the **Mini-Tools** card, leave the mini-tool set to *aa — amplitude and angle*, and give it the answer's name:

> Value
> `ir1`

It reads **1.789∠26.57°** A.

To see *every* answer in polar form, use the setting described in *Rectangular or polar*, above.

This is correct. You manually convert it to a sinusoid, by putting it back in the same terms the input was given in:

$$i(t) = 1.789\cos(4t + 26.57°)\ \text{A}$$

Asking for the capacitor voltage the same way gives 4.472∠-63.43°, that is,

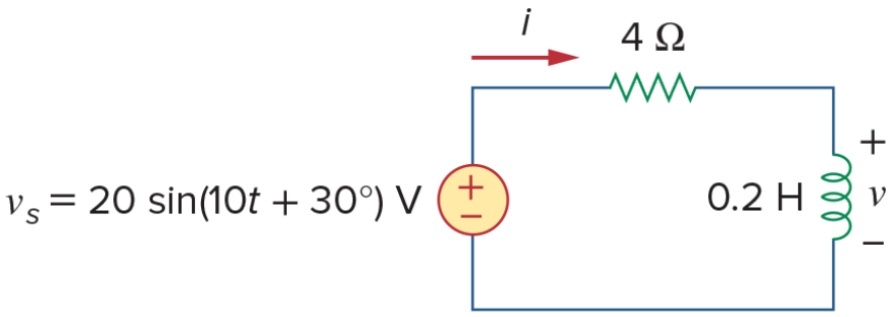
$$v(t) = 4.472\cos(4t - 63.43°)\ \text{V}$$

which is correct.

Future solved problems will not include this level of detail in the solution, only the circuit description and the settings we choose.

AS7's Practice Problem 9.9

Determine ***v***(t) and ***i***(t).



AS7's Practice Problem 9.9

SOLUTION

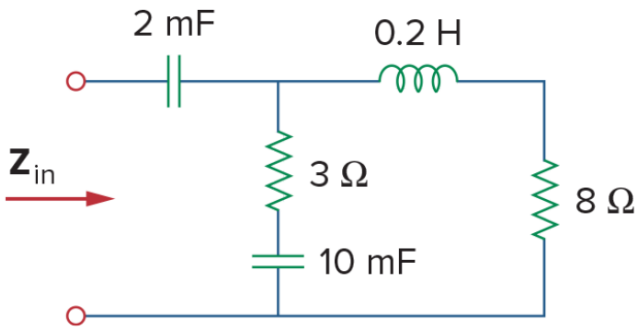
> Circuit Description
> `e,1,0,(20∠30°)`
> `r1,1,2,4`
> `l,2,0,.2`

The source is given in polar form and typed as written: magnitude, the angle sign, then degrees.

Then AC, with **10** for ω. **Mini-Tools** with *aa* reads `ir1` as **4.472∠3.43°** A and `vl` as **8.944∠93.43°** V. Both are correct.

AS7's Example 9.10

Find the input impedance of the circuit. Assume that the circuit operates at $\omega = 50$ rad/s.



AS7's Example 9.10

SOLUTION

Since this is a passive circuit, to reduce it we find its equivalent impedance.

Circuit Description

```
ca,1,2,2'm
r1,2,3,3
cb,3,0,10'm
l1,2,4,.2
r2,4,0,8
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Resistance / impedance*, with nodes **1** and **0**, in AC at ω **50**. The answer is called `zeq`.

equivalent impedance

$$Z_{eq} = 3.22 - 11.07\,\mathrm{j}\ \Omega$$

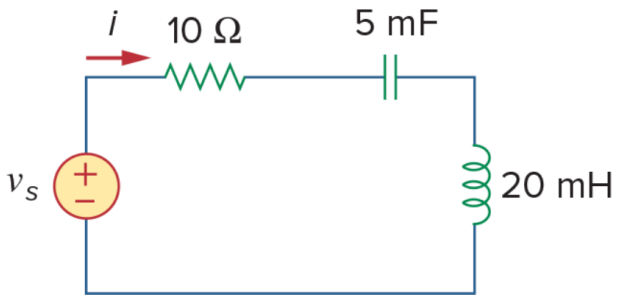
This is correct.

> TIP
>
> **The pr tool only reduces impedances**
>
> In case you were tempted to reduce this circuit by hand instead of simulating it, remember that parallel reduction works on resistors and impedances only. You would need to convert the capacitors and the inductor to impedances in ohms before you do that.

AS7's Problem 9.35

Find the steady-state current $\boldsymbol{i}$ in the circuit when $\boldsymbol{v_s}(t) = 50 \cos 200t$ V.

SOLUTION



AS7's Problem 9.35

The capacitor is in farads and the inductor in henries, so Symbulator needs the frequency, and the source is entered as its phasor: 50 cos 200t is `50` at an angular frequency of 200.

Circuit Description

```
e,1,0,50
r,1,2,10
c,2,3,5'm
l,3,0,20'm
```

AC, with ω — **angular frequency** set to **200**.

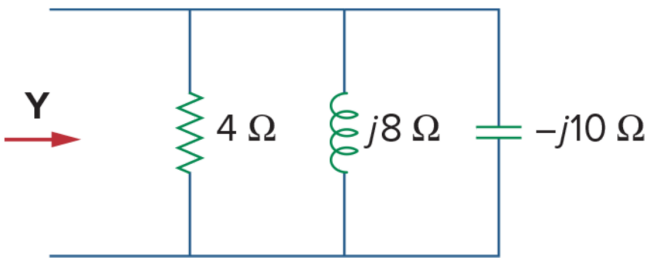
Read it with **Mini-Tools** set to *aa*: `aa(ir)` gives **4.789∠-16.70°** A. Correct.

With values in Ω only

When the values of all the capacitors and inductors in the circuit are given in imaginary ohms, they must be entered in the circuit description as impedances, **r**. In these cases, Symbulator will not need a frequency.

AS7's Problem 9.37

Determine the admittance Y for the circuit.



AS7's Problem 9.37

SOLUTION

Since the values are in ohms, these are all impedances. This is a passive circuit, so we could find its equivalent. However, the structure of this circuit is so simple that we can reduce it by hand. The problem asks for the equivalent admittance, so our answer will be the inverse of the equivalent impedance:

Evaluate

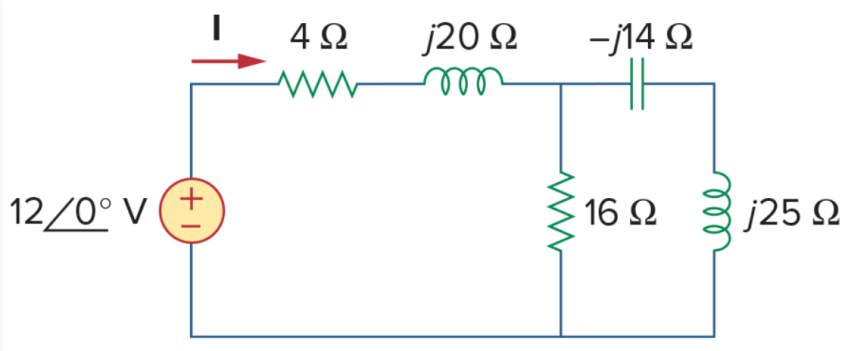
```
1/pr(4,8j,-10j)
```

$$0.25 - 0.025\,\mathrm{j}\ \mathrm{S}$$

This is correct.

AS7's Problem 9.39

For the circuit shown, find the equivalent impedance, and use that to find the current I. Let omega = 10 rad/s.



AS7's Problem 9.39

SOLUTION

The frequency the question gives is superfluous: every value is already in ohms. Nothing here needs a full simulation either; the equivalent impedance is one expression, using the parallel shorthand:

Evaluate

```
4+20j+pr(16,-14j+25j)
```

$$9.135 + 27.47\,\mathrm{j}\ \Omega$$

That is 9.135 + j27.47 Ω. The current is the source voltage divided by it:

Put the whole thing in **Mini-Tools** with *aa*: `aa(12/(4+20j+pr(16,-14j+25j)))` gives **0.4145∠-71.60°** A. Correct.

TIP

The same answer from a simulation

You can also let Symbulator do the reduction, using the square-bracket shorthand for parallel elements inside a value:

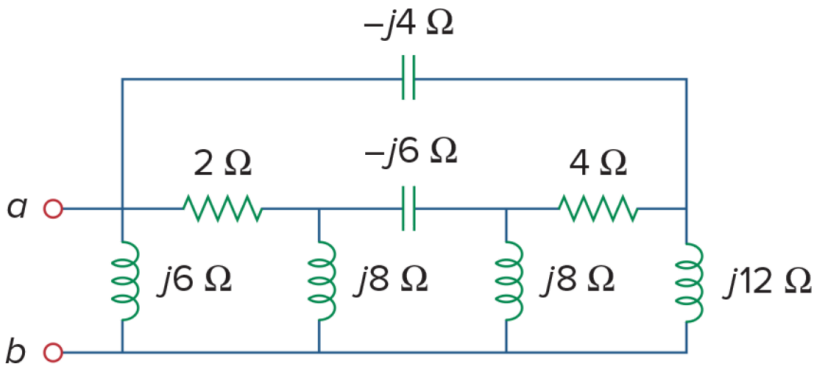
Circuit Description

```
e,1,0,12
r1,1,0,4+20j+[16,-14j+25j]
```

AC asks for a frequency anyway; put in anything, since every value is already an impedance. Then `aa(ir1)` gives the same **0.4145∠-71.60°** A.

AS7's Problem 9.73

Determine the equivalent impedance for the circuit.



AS7's Problem 9.73

SOLUTION

Eight impedances and no source: a job for *Find equivalent* rather than a solve. The names carry the two nodes each element bridges, for convenience.

Circuit Description

```
r10,1,0,6j
r20,2,0,8j
r30,3,0,8j
r40,4,0,12j
r12,1,2,2
r23,2,3,-6j
r34,3,4,4
r14,1,4,-4j
```

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Resistance / impedance*, with nodes **1** and **0**, in AC. The answer is called `zeq`.

equivalent impedance

$$Z_{eq} = 0.3794 + 1.46\,\mathrm{j}\ \Omega$$

That is 0.3794 + j1.46 Ω, which is correct.

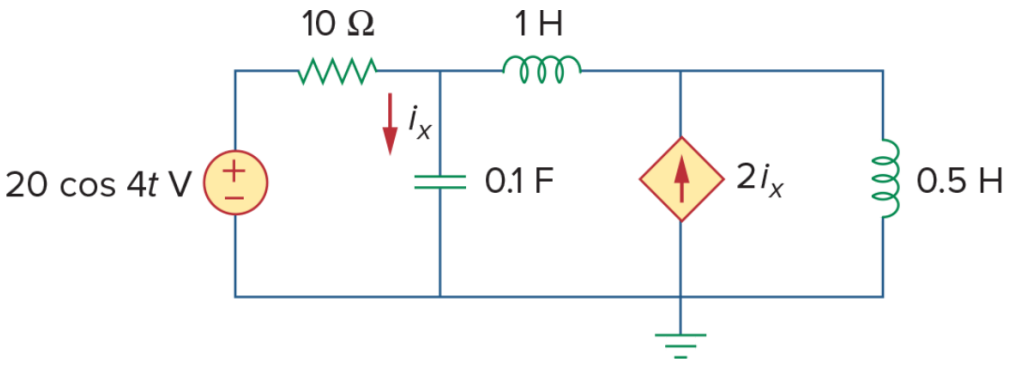
With dependent sources

A dependent source is described exactly like an independent one — the value is just an expression naming another element's answer instead of a number.

The name is the one the answers use: the current through **cx** is `icx`, the voltage across **rx** is `vrx`.

AS7's Example 10.1

Find i_x in the circuit.



AS7's Example 10.1

SOLUTION

The current source is controlled by the current through the capacitor **cx**, which is what the last field of **j1** says.

Circuit Description

```
e,1,0,20
r1,1,2,10
cx,2,0,.1
l1,2,3,1
j1,0,3,2*icx
l2,3,0,.5
```

AC, with ω — **angular frequency** set to **4**.

`aa(icx)` gives **7.589∠108.4°** A. Correct.

AS7's Practice Problem 10.1

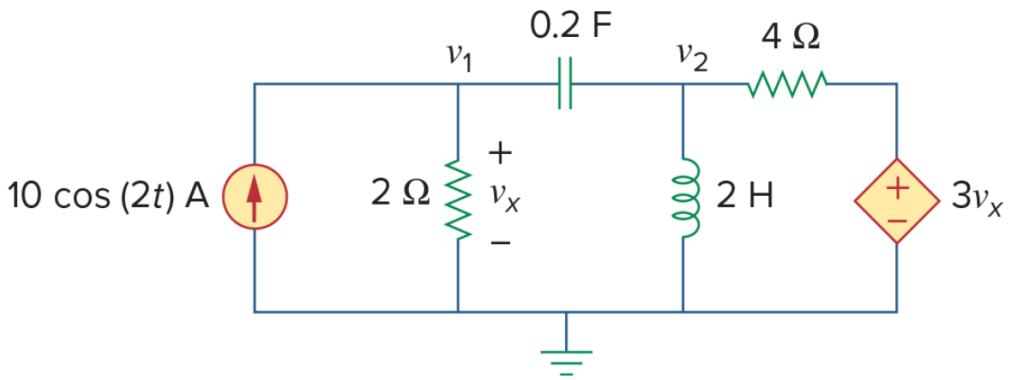
Find v_1 and v_2 in the circuit.

SOLUTION

This one is controlled by a *voltage* rather than a current — three times the voltage across **rx**.

Circuit Description

```
j,0,1,10
rx,1,0,2
c,1,2,.2
l,2,0,2
r,2,3,4
e,3,0,3*vrx
```



AS7's Practice Problem 10.1

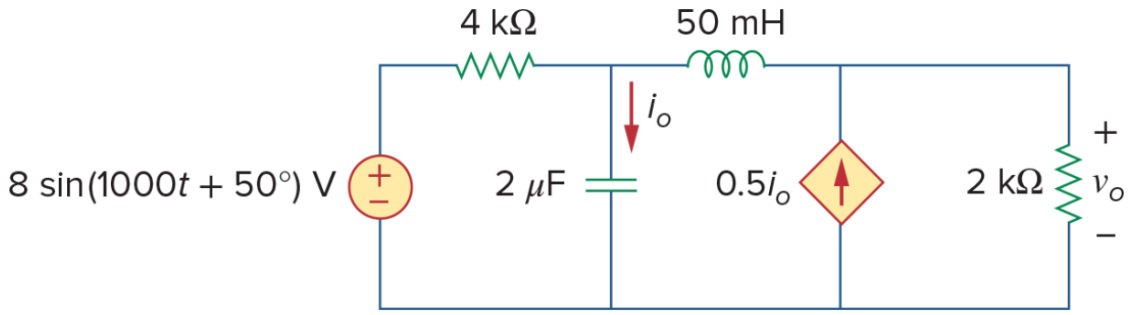
AC, with ω — **angular frequency** set to **2**.

`aa(v1)` gives **11.33∠60.02°** V and `aa(v2)` gives **33.02∠57.13°** V.

Both are correct.

AS7's Example 10.13

Obtain $\boldsymbol{v_o}$ and $\boldsymbol{i_o}$ in the circuit.



AS7's Example 10.13

SOLUTION

The book wants its answers in terms of cosine rather than sine, so the source is taken as 8 cos(1000t – 40°) — as a phasor, `(8∠-40°)`.

Note the decimal points on the values. They make the arithmetic approximate, which here is what you want: an exact solve of this circuit carries unresolved square roots through every step for no benefit.

Circuit Description

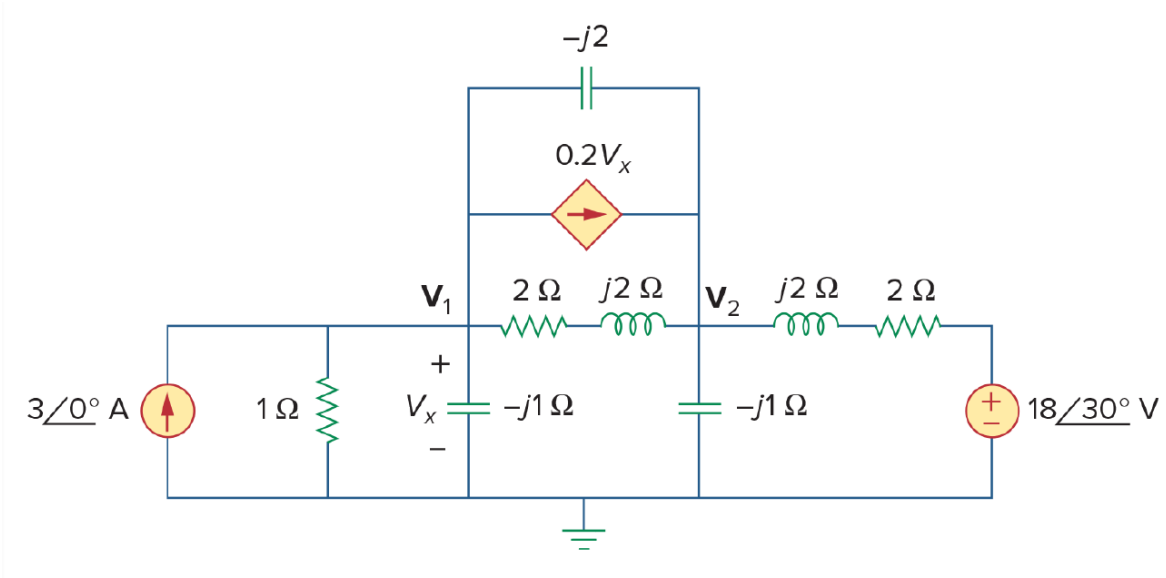
```
e,1,0,(8∠-40°)
r1,1,2,4'k
co,2,0,2'μ
l1,2,3,50'm
j1,0,3,.5*ico
ro,3,0,2'k
```

AC, with ω — **angular frequency** set to **1000**.

`aa(vro)` gives **1.550∠-95.18°** V and `aa(ico)` gives **0.003264∠-3.743°** A. Both correct.

AS7's Example 10.14

Find V_1 and V_2 in the circuit.



AS7's Example 10.14

SOLUTION

Every value here is already in ohms, so no frequency matters. Again the decimal points keep the arithmetic approximate.

Circuit Description

```
j1,0,1,3
r1,1,0,1
rx,1,0,-1j
r3,1,2,-2j
j2,1,2,.2*vrx
r4,1,2,2+2j
r5,2,0,-1j
r6,2,3,2+2j
e,3,0,(18∠30°)
```

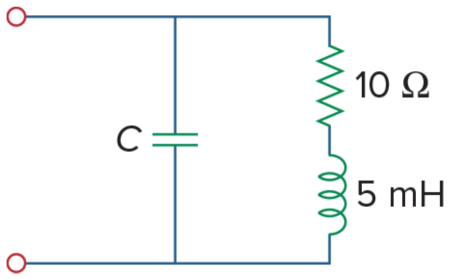
AC. The frequency is asked for but never used, so anything will do.

`aa(v1)` gives **2.708∠-56.73°** V and `aa(v2)` gives **6.914∠-80.70°** V. Both correct.

7.3 Solved numerical-from-symbolic examples

AS7's Problem 9.89

Calculate the value of C so that the net impedance is purely resistive at 2 kHz.



AS7's Problem 9.89

SOLUTION

The answer given by the book is 25 μF. But I think this is one of those unfortunately common instances where there is a mistake in the book. I believe they meant to ask "at 2k rad/s", because that is the frequency at which the answer they give is right. Let me tell you how I solved this problem and I will let you be the judge.

Circuit Description

```
c,1,0,c
r1,1,2,10
l,2,0,5'm
```

Find equivalent, *Resistance / impedance*, nodes **1** and **0**, AC. The problem gives 2000 Hz and the ω box wants radians per second, so type `2*pi*2e3` straight into it — it takes an expression, not only a number.

Once the simulation finishes, solve for the value of c that makes the imaginary part of the equivalent impedance zero:

Now open the **Solve** card and ask for the capacitance that leaves no imaginary part:

Equation(s) to solve in terms of the results

```
im(zeq) = 0
```

Unknown(s) to solve for

```
c
```

Tick **real solutions only** before running it. Without it the answer comes back carrying an `im(c)` term, because nothing has told Symbulator that a capacitance is a real number.

The answer is **c** = **0.000001235**, or 1.235 μF. This is not the answer the book gives.

I first solved this over the complex numbers, which also works, but my friend Qifan Wang — who verified my answers — pointed out, correctly, that it is not necessary: focusing on the imaginary part removes every reference to the complex operator, so we are solving an equation in real terms only.

Let's now repeat the process, using what I suspect is the frequency they meant: 2000 rad/s. I also want to show you something cool. Now that we know the answer is in the range of μF, we can declare the value of the capacitor as `c'`μ, so that the value of c we get will be in that scale:

Circuit Description

```
c,1,0,c'μ
r1,1,2,10
l,2,0,5'm
```

Find equivalent, *Resistance / impedance*, nodes **1** and **0**, in AC at ω **2000**. That gives `zeq` = $\frac{500i}{-c+25+25i}$, and the condition goes to the **Solve** card:

Equation(s) to solve in terms of the results

```
im(zeq) = 0
```

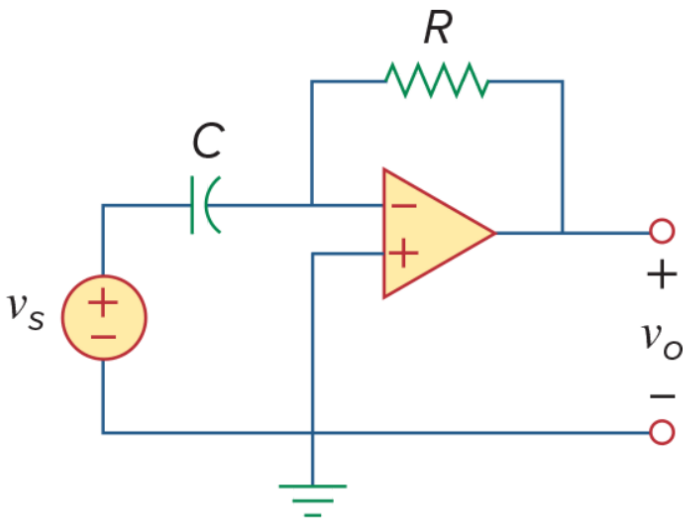
with `c` as the unknown and **real solutions only** ticked — the same equation as before, now against the new `zeq` — which answers `c` = **25**.

The answer is **c** = **25**, that is to say 25 μF. This is the right answer in the book. So they asked the wrong question — they asked it using the wrong frequency unit.

7.4 Solved symbolic examples

AS7's Problem 10.69

Find V_o/V_s.



AS7's Problem 10.69

SOLUTION

Circuit Description

```
e,1,0,vs
c,1,2,c
r1,2,o,r
o,0,2,o
```

The ω box takes a name as readily as a number: type `omega` into it and the answers come back as functions of it.

Evaluate gives `vo/vs` as `-c·ω·r·i`, which is correct.

NOTE

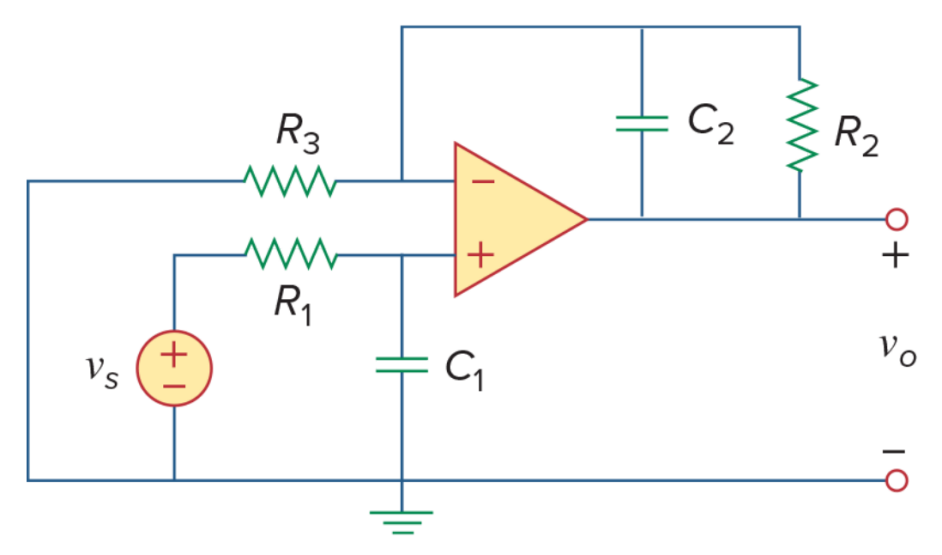
Know Symbulator's limits

Symbulator is a symbolic solver, and symbolic effort grows quickly with the number of nodes. A circuit that would fill a page with algebra is one you should be solving numerically anyway.

What matters is that you know the limits, so that they don't take you by surprise in an exam. Think of Symbulator as a wonderful Swiss knife of circuit simulators: no matter how good it is, it is still a pocket knife. You would not try to chop a tree down with it. SPICE and its children — LTspice, PSpice, MultiSim — are the chainsaws of circuit simulation. They won't give you symbolic answers, but when you need to chop a tree down, they are the way to go.

AS7's Problem 10.77

Compute the closed-loop gain V_o/V_s for the op-amp circuit.



AS7's Problem 10.77

SOLUTION

Here is an example that takes Symbulator to the limit of what it can solve symbolically — everything in it is a symbol, including the frequency.

Circuit Description

```
e,1,0,vs
r1,1,2,r1
ca,2,0,ca
r3,0,3,r3
o,2,3,o
cb,3,o,cb
r2,3,o,r2
```

Then ask **Evaluate** for the ratio:

```
Evaluate
vo/vs
```

$$\frac{V_o}{V_s} = \frac{j\,c_b r_2 r_3 \omega + r_2 + r_3}{r_3\left(-c_a c_b r_1 r_2 \omega^2 + j\,c_a r_1 \omega + j\,c_b r_2 \omega + 1\right)}$$

It looks different from the answer in the book, but evaluating both shows they are the same expression arranged differently.
This circuit takes Symbulator 9 about ten seconds, with every value symbolic — and about twice that on a phone.

LESSON 8

Power in AC circuits

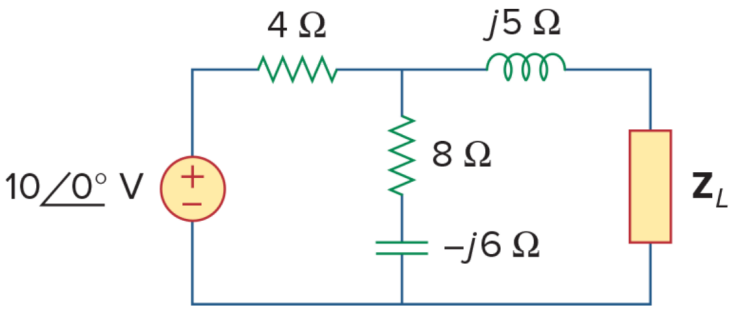
Learn to solve *average power* and *complex power* problems in AC. Learn how you can use RMS values in Symbulator with the **RMS phasors** setting. Solve power factor problems using the **pf** mini-tool.

In this lesson you will learn how to run more advanced *alternating current* simulations. You will learn how to solve *average power* problems, when and how to use the **RMS phasors** setting, and how to solve *complex power* problems.

8.1 Average power

AS7's Example 11.5

Determine the load impedance ZL that maximises the average power drawn from the circuit. What is the maximum average power?



AS7's Example 11.5

SOLUTION

Thévenin / Norton helps us find all the answers we need. Notice this circuit description does not include the load, because it is not needed.

Circuit Description

```
e,1,0,10
r1,1,2,4
r2,2,0,8-6j
r3,2,3,5j
```

Find equivalent, *Thévenin / Norton*, nodes **3** and **0**, in AC. In AC the equivalent impedance is called `zeq` rather than `req`.

Thévenin / Norton reports the equivalent impedance, `zeq`, as 2.933 + j4.467. The load that

will deliver the maximum power is its conjugate, 2.933 – j4.467.

The average maximum power delivered by the circuit is `pmax` = 2.3674 W.

In AC, `pmax` is the *average* maximum power, computed from the real part of the equivalent impedance: **Results** shows `pmax` = **2.367** W, the power delivered when the load is the conjugate of `zeq`.

8.2 RMS and PF

In the previous lesson we mentioned that after an AC simulation you also get the complex power consumed. How to read these values depends on the **RMS phasors** setting.

The RMS setting

You will learn in your circuits course that phasor analysis can be done in RMS values. There are advantages to it, and you will have to solve AC circuits both in normal values (also called magnitude values) and in RMS. Working in RMS simply means every current and voltage in the analysis is taken to be RMS. That's it.

The switch is in **Settings**, under **AC power convention**: tick **RMS phasors**.

With the tick **off**, Symbulator takes every phasor you type as a *peak* value — the amplitude the sinusoid rises to — and the average power in an element is computed with the factor of one half that peak values require:

$$P = \tfrac{1}{2} V_{\text{peak}} I_{\text{peak}} \cos\theta$$

With the tick **on**, it takes every phasor you type as an *RMS* value, and the half disappears:

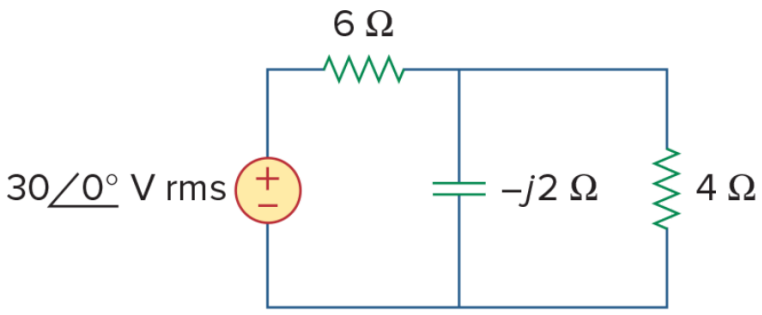
$$P = V_{\text{rms}} I_{\text{rms}} \cos\theta$$

That is the whole difference. The currents and voltages you get back are in whatever convention you typed the sources in, unchanged; only the power answers — `p`, `q` and `s` — depend on the tick, and they keep their names whichever way it is set: `pe` is the average (real) power, `qe` the reactive power, `se` the complex power (the calculator named the average `ap` with peak phasors and `p` with RMS ones; `ape` still answers, as the same number as `pe`). With the tick on, the cards call every current and voltage *effective*, the RMS values being the effective values. So for the same numbers typed in, RMS on reports twice the power that RMS off does, and neither is wrong: they answer different questions. Match the book you are working from. The setting is always in view, so it is easier for you to notice whether it is on or off.

AS7's Example 11.10

Determine the power factor of the entire circuit as seen by the source. Calculate the average power delivered by the source.

SOLUTION



AS7's Example 11.10

The source value in this problem is given in RMS values, so we must say so.

Circuit Description

```
e,1,0,30
r1,1,2,6
r2,2,0,-2j
r3,2,0,4
```

AC, with `omega` typed into the ω box, and **RMS phasors** ticked in **Settings**.

Or, if you want to simplify the impedances, the resistors can be collapsed into a single parallel combination.

The average power *consumed* by the source is `pe`; the power *delivered* is its opposite, which the source's card reads as `-pe`, 125.4 W.

The pf mini-tool

pf is the second of the mini-tools Lesson 7 (page 206) introduced, and it works as the calculator's did. In the **Mini-Tools** card choose *pf — power factor*; it asks for one value, and there are two things you can give it.

A complex power. Give it the name of a complex power answer, such as `se`, or any complex expression — a number in rectangular or polar form, an impedance, an expression with symbols in it:

Value

```
se
```

It answers **0.97342 lagging**: the absolute value of the real part divided by the magnitude of the whole thing, and the word for the value as given — nothing more, because by now you know what `se` is. It is the power the source *consumes*, and the word belongs to that power.

The name of an element. Give it the name of an element of the circuit you have just solved in AC:

Value

```
e
```

It answers **0.97342 leading**: the value and the word together, and under them *this is the power factor for the power delivered by source e*. This form works only when the element's voltage and current came out as numbers.

The same number, the opposite word. That is not a contradiction: `se` is the power the source consumes and `e` reads the power it delivers, and the two powers are opposite, so their words are. The question *what is the power factor of the circuit as seen by the source?* is answered by the name, `e`.

In a nutshell: given a complex power, or any complex expression, pf returns the power factor of that value as given, with its word and no explanation. Given the name of an element, right after an AC solve, it reads the right power for that element's kind and says in words which power that was. With symbols in the value there is no word, only the expression.

> NOTE
>
> **Which power the reading is taken on**
>
> The two forms do not read the same power, and the difference is what makes the word come out right.
>
> Given a **variable** such as `se`, `sj` or `sr1`, the calculation is done on the value as it stands — the complex power *consumed*, which is what Symbulator stores for every element, source or load alike — and the word is that value's own. For an impedance that is its own reading. For a source it is the reading of the power it *consumes*, which is the opposite of the power it delivers, so the word comes out opposite to the source's own. The tool cannot know better from a number alone: the same complex power is consumed by one side of a branch and delivered by the other.
>
> Given a **name**, the tool reads the sign convention off the element's kind, the way the calculator did. For an impedance — `r`, and in version 9 `l` and `c` as well — the angle is taken between the element's voltage and the current it *consumes*, so the reading is the element's own: an inductive load reads lagging. For a source — `e` or `j` — the current is negated first, so the angle is taken on the power the source *delivers*, and the reading is that of the whole circuit the source sees. A source feeding an inductive load reads lagging, the same word as the load.
>
> The value is the same either way; only the word depends on it. A source read on its consumed power, `se`, says the opposite word — which is why the question *as seen by the source* wants the name, and the tool does the negating itself.

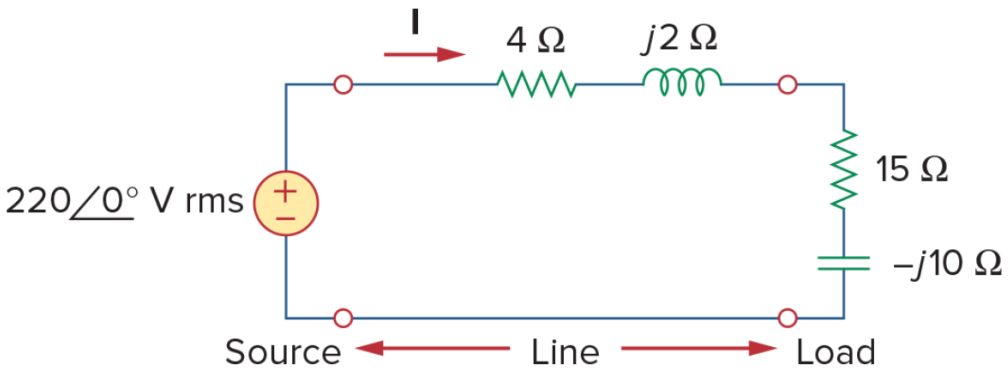
8.3 Complex power

AS7's Example 11.13

The figure shows a load being fed by a voltage source through a transmission line. The impedance of the line is represented by the (4+j2) Ω impedance and a return path. Find the real power and reactive power delivered by the source, and absorbed by the line and load.

SOLUTION

The value of this circuit's source is given in RMS. Since the problem asks for answers



AS7's Example 11.13

regarding the line and the load, we should have separate impedances, one for each.

Circuit Description

```
e,1,0,220
r1,1,2,4+2j
r2,2,0,15-10j
```

Again AC with `omega` and **RMS phasors** ticked.

The complex power *delivered* by the source is the opposite of the power it consumes, `-se`; the complex power *absorbed* in the line and in the load are `sr1` and `sr2`:

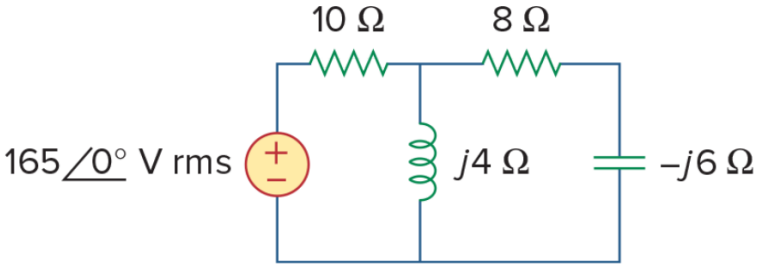
Results shows:

- `-se` = **2163.8** − **911.1i** VA
- `sr1` = **455.5** + **227.8i** VA
- `sr2` = **1708.2** − **1138.8i** VA

These are correct.

AS7's Practice Problem 11.10

Calculate the power factor of the entire circuit as seen by the source. What is the average power supplied by the source?



AS7's Practice Problem 11.10

SOLUTION

The values are RMS, so tick **RMS phasors** in **Settings**.

Circuit Description

```
e,1,0,165
r1,1,2,10
r2,2,0,4j
r3,2,3,8
r4,3,0,-6j
```

Written out as four elements rather than one, which is clearer and costs nothing here. The average power supplied is the opposite of the power the source consumes: `pe` reads **-2007.1** W, so the source supplies **2007.1** W, which its card reads as `-pe`.

For the power factor, use **Mini-Tools** with *pf*, giving it the source's name, so that the reading is of the circuit the source sees:

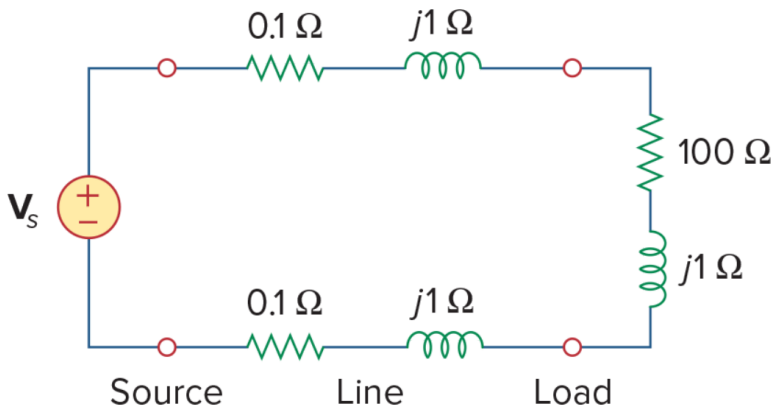
Value

```
e
```

It reads **0.93595** **lagging**.

AS7's Problem 11.97

A power transmission system is modelled as shown. If V_s = 240 V rms, find the average power absorbed by the load.



AS7's Problem 11.97

SOLUTION

Three impedances in series: the outgoing line, the load, and the return path.

Circuit Description

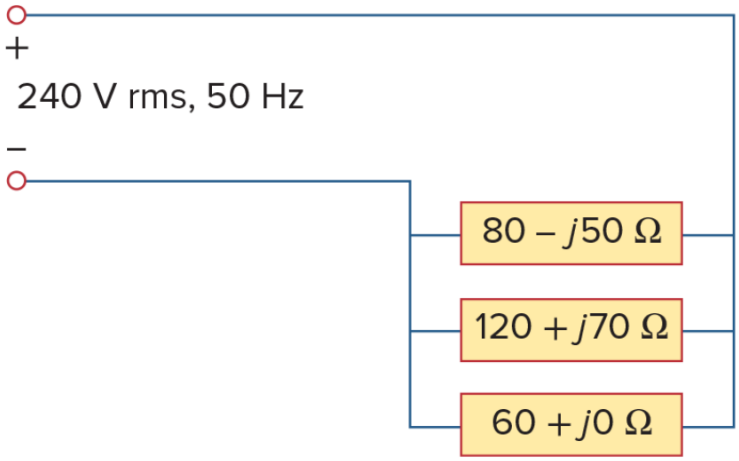
```
evs,1,0,240
rl1,1,2,.1+1j
rl,2,3,100+1j
rl2,3,0,.1+1j
```

AC with **RMS phasors** ticked. The power consumed `prl` reads **573.2** W.

Correct.

AS7's Problem 11.75

Consider the power system shown. Calculate the total complex power, the power factor, and the parallel capacitance needed for a unity power factor.



AS7's Problem 11.75

SOLUTION

Three loads in parallel across a 240 V rms source. The first two parts need no frequency at all, so leave omega as anything.

Circuit Description

```
e,1,0,240
r1,1,0,80-50j
r2,1,0,120+70j
r3,1,0,60
```

(a) The complex power delivered is the opposite of the power the source consumes: the opposite of `se` gives **1835.9** – **114.7**j VA.

(b) The power factor, from *pf* in **Mini-Tools** given the source's name, `e`, is **0.99805** leading.

(c) This one needs a frequency, because it needs a capacitor. Add one in parallel with a symbolic value (let's say x), and run at the stated 50 Hz — $(2\pi)(50)$ in rad/s.

Circuit Description

```
e,1,0,240
r1,1,0,80-50j
r2,1,0,120+70j
r3,1,0,60
c,1,0,x
```

Put `2*pi*50` in the ω **— angular frequency** box — it takes an expression. Give *pf* the complex power `se` now and it answers with an expression in `x`: the power factor as a function of the capacitance, which is what the calculator solved for 1. In version 9, ask

the **Solve** card instead for the value of x that leaves no reactive power, which is the same condition:

> Equation(s) to solve in terms of the results
> `im(se) = 0`

> Unknown(s) to solve for
> `x`

Tick **real solutions only**.

It answers **-6.34** μF.

A negative capacitance, and no positive value satisfies the equation — which is probably why the textbook gives no number for this part. It is not a failure of the method: the load is *already* leading, as part (b) said, so no capacitor can bring it to unity. What it needs is the opposite.

Try an inductor instead:

> Circuit Description
> ```
> e,1,0,240
> r1,1,0,80-50j
> r2,1,0,120+70j
> r3,1,0,60
> l,1,0,x
> ```

Solving the same way gives **1.5987** H, which is positive and therefore the real answer.

Check it by putting that number back in as the inductor's value: the power factor comes out **1**. An inductor of about 1.6 H in parallel is what brings this system to unity.

LESSON 9

Three-phase circuits

Learn to solve simple *three-phase* circuits in Y-Y, Y-Δ, Δ-Δ and Δ-Y configurations, both balanced and unbalanced, to find line and phase currents, voltages and complex power.

In this lesson you will learn to use Symbulator to solve simple **three-phase** systems, in their four basic configurations of wye-wye, wye-delta, delta-delta and delta-wye. We will see both balanced and unbalanced examples, where you are asked to find currents, voltages and complex power in the source, line and load.

9.1 About solving three-phase circuits

Three-phase circuits can be tricky to solve in Symbulator, for several reasons. It is not always clear which node to use as ground. There are three times as many nodes as in an equivalent single-phase circuit. And sources in a delta array bring mathematical trouble: more unknowns than equations.

Symbulator does a great job on the simple three-phase circuits of a basic Circuits I / II textbook, and here we will see examples and one or two tricks. It would not be my tool of choice for anything larger: as circuits grow, the number of nodes quickly passes what your device can solve.

9.2 Wye-Wye

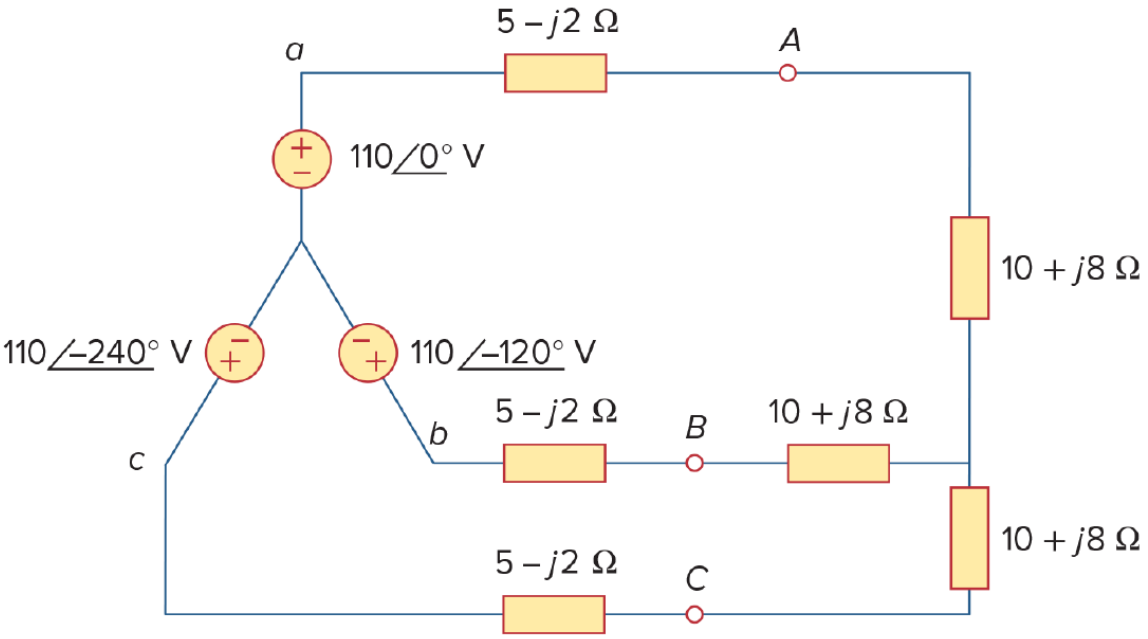
Balanced wye-wye system

AS7's Examples 12.2 & 12.6

Calculate the line currents in the three-wire Y-Y system. Determine the total average power, reactive power, and complex power absorbed at the source and at the load. Assume that the values given for the source are RMS.

SOLUTION

Although not specified in the textbook, to get the answers they give we need to assume the values are RMS. One must describe the circuit carefully in the case of three-phase circuits, because it is very easy to make a mistake. Below is how I describe this one:



AS7's Examples 12.2 and 12.6

Circuit Description

```
ea0,ag,0,(110∠0°)
eb0,bg,0,(110∠-120°)
ec0,cg,0,(110∠120°)
rat,ag,ad,5-2j
rbt,bg,bd,5-2j
rct,cg,cd,5-2j
ra0,ad,0,10+8j
rb0,bd,0,10+8j
rc0,cd,0,10+8j
```

AC, with **RMS phasors** ticked. Every value is already an impedance, so the frequency is asked for but never used.

Some observations about my description:

- Notice that I have specified the node at the centre of both Y's as node **0**. This is something you can do only in the case of balanced Y-Y systems, not for other configurations and not for unbalanced systems.
- Since Symbulator does not differentiate between lower and upper case in names, nodes called **a** and **A** would be considered the same node. Instead, we use the names **ag** and **ad**, where the g reminds us a node is on the generation side and the d reminds us it is on the demand side.

Once the simulation is completed, we look in the results for the currents on the transmission lines:

Tick **Show AC answers as polar phasors** in **Settings** and each line impedance's *current through* reads straight off:

- `irat` = **6.809∠-21.8°** A

- `irbt` = **6.809∠-141.8°** A
- `irct` = **6.809∠98.2°** A

The complex power at the source is `sea0+seb0+sec0`, which gives −2086.2 − 834.5**i** VA. The real part is the average power absorbed by the source; since it is negative, the source is delivering an average power of 2086 W. The imaginary part is the reactive power: 834 var.

The complex power at the load is `sra0+srb0+src0`, which gives 1390.8 + 1112.6**i** VA. The load is consuming an average power of 1391 W and a reactive power of 1113 var.

Before we move on, try simulating this circuit with a different node as the centre of the load's Y array — call it d0, for the demand side. The voltage in that node evaluates to 0 V, because the system is balanced. That is why we could use 0 for both. In an unbalanced system we cannot.

Further wye-wye problems

AS7's Practice Problem 12.2 & 12.6

Calculate the line voltages and the line currents. Also calculate the complex power at the source and at the load. Assume the source voltage is given as RMS.

> A Y-connected balanced three-phase generator with an impedance of $0.4 + j0.3\ \Omega$ per phase is connected to a Y-connected balanced load with an impedance of $24 + j19\ \Omega$ per phase. The line joining the generator and the load has an impedance of $0.6 + j0.7\ \Omega$ per phase. Assuming a positive sequence for the source voltages and that $\mathbf{V}_{an} = 120 \underline{/30^\circ}$ V, find: (a) the line voltages, (b) the line currents.

AS7's Practice Problem 12.2 and 12.6

SOLUTION

The textbook does not say so, but you only get its answers if you take the values as RMS.

There are **three** impedances in each line here: one inside the generator, one for the transmission line, and one for the load. The node names below say which is which — **ag** in the generator, **as** on the supply side, **ad** on the demand side.

Circuit Description

```
ea0,ag,0,(120∠30°)
eb0,bg,0,(120∠-90°)
ec0,cg,0,(120∠150°)
rag,ag,as,.4+.3j
rbg,bg,bs,.4+.3j
rcg,cg,cs,.4+.3j
rat,as,ad,.6+.7j
rbt,bs,bd,.6+.7j
rct,cs,cd,.6+.7j
```

```
ra0,ad,0,24+19j
rb0,bd,0,24+19j
rc0,cd,0,24+19j
```

AC, with **RMS phasors** ticked. Every value is already an impedance, so the frequency is asked for but never used.

The question asks for the line voltages — the differences between the lines. For the answers the textbook wants, take these differences:

In **Mini-Tools** with *aa*:

- `aa(vag-vbg)` = **207.8∠60.00°** V
- `aa(vbg-vcg)` = **207.8∠-60.00°** V
- `aa(vcg-vag)` = **207.8∠180.0°** V

> TIP
>
> **Which line voltage did you want?**
>
> If you ask me, that is not really the voltage between the lines, because the transmission line starts *after* the generator. At the start of the transmission line the answer is **204.6∠60.03°** V, and at the end of it **198.7∠59.71°** V — from `aa(vas-vbs)` and `aa(vad-vbd)` respectively. All three are different, and which one you want depends on where you are standing.

The line currents are the currents through the transmission-line impedances:

With polar phasors on, each transmission-line impedance's *current through* reads:

- `irat` = **3.748∠-8.660°** A
- `irbt` = **3.748∠-128.66°** A
- `irct` = **3.748∠111.34°** A

Finally the complex power. At the source:

Evaluate `sea0+seb0+sec0`, which gives **-1054** − **842.9i** VA.

Note this does *not* include the power lost in the source's own internal impedances, which you could add if you wanted them. At the load:

Evaluate `sra0+srb0+src0`, which gives **1012** + **800.8i** VA.

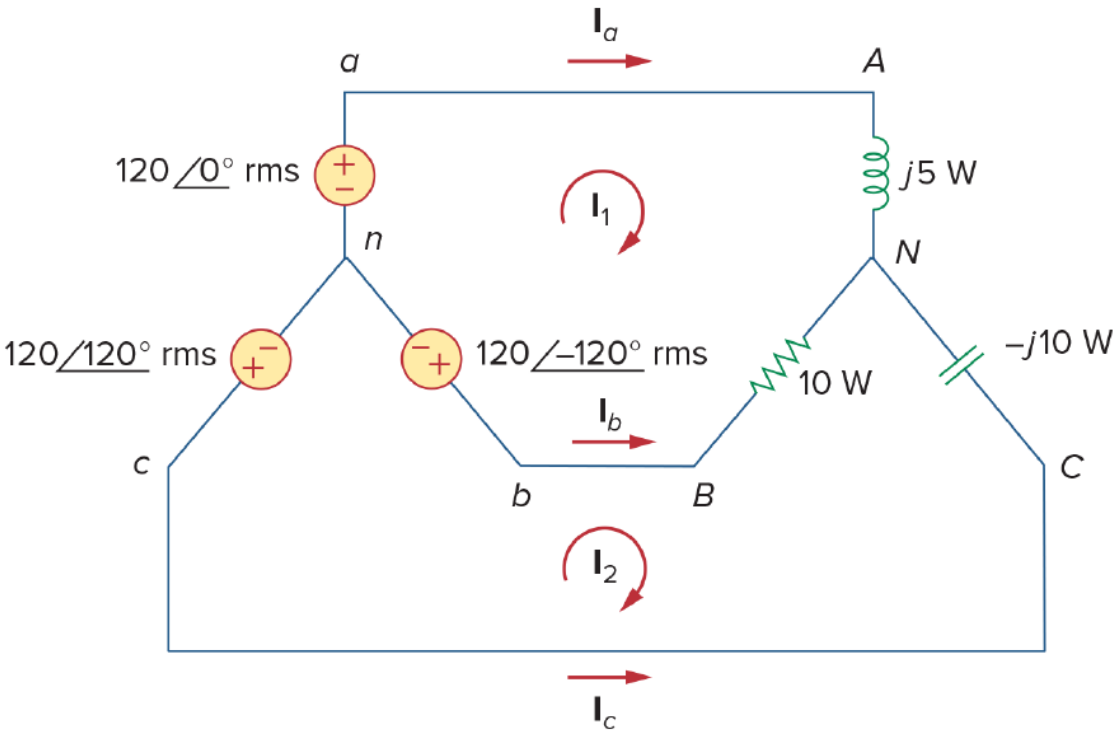
All correct.

Unbalanced wye-wye system

The balanced case let you use node **0** for the centre of *both* wyes, because the voltage at the centre of the load is zero. Unbalanced, it is not — so the load needs a centre node of its own.

AS7's Example 12.10

For the unbalanced circuit, find (a) the line currents, (b) the total complex power absorbed by the load, and (c) the total complex power absorbed by the source.



AS7's Example 12.10

SOLUTION

The centre of the generator's wye is ground, and the centre of the load's wye is a node of its own, called `n`.

Circuit Description

```
ea,a,0,(120∠0°)
eb,b,0,(120∠-120°)
ec,c,0,(120∠120°)
ra,a,n,5j
rb,b,n,10
rc,c,n,-10j
```

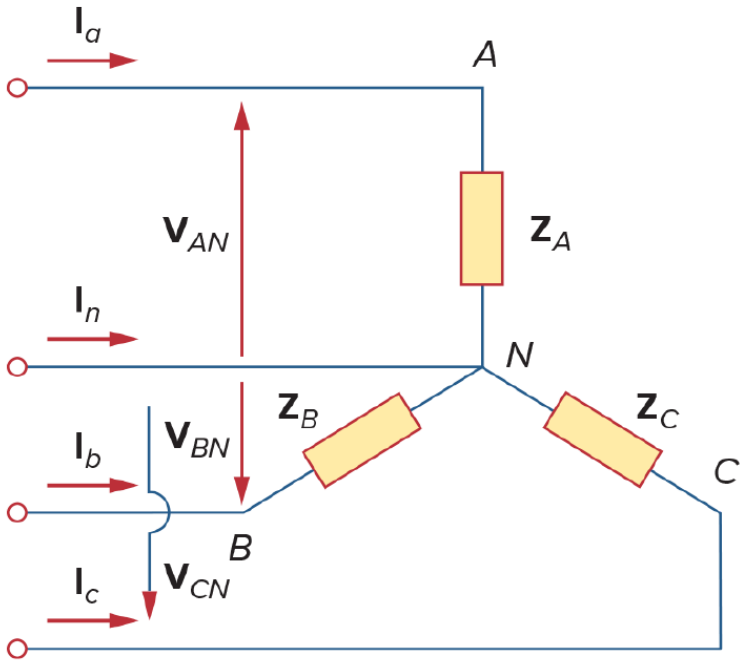
AC, with **RMS phasors** ticked.

The three line currents read **56.78∠0°**, **25.46∠135.0°** and **42.76∠-155.1°** A. The complex powers in the load are **16120i**, **6480** and **-18280i** VA, and in the three sources **-6814**, **790.6** – **2951i** and **-456.5** + **5111i** VA.

All correct. Two things are worth checking for yourself. Complex power is conserved, so `sea+seb+sec+sra+srb+src` evaluates to zero. And the voltage at the centre of the load is *not* zero, as it was in the balanced case — `vn` is **120.0** – **283.9i** V.

AS7's Example 12.9

The unbalanced Y-load has balanced voltages of 100 V in the *acb* sequence. Calculate the line currents and the neutral current. Take $\boldsymbol{Z_A}$ = 15 Ω, $\boldsymbol{Z_B}$ = 10 + j5 Ω and $\boldsymbol{Z_C}$ = 6 − j8 Ω.



AS7's Example 12.9

SOLUTION

Two things differ from the last one. The source values are *not* RMS this time. And there is a neutral line joining the centre of the load to the centre of the source, which you describe as a short circuit — an **s** element, which takes just a name and two nodes.

Note also the angles: this is the *acb* sequence, so they run the other way round.

Circuit Description

```
ea,a,0,(100∠0°)
eb,b,0,(100∠120°)
ec,c,0,(100∠-120°)
sn,0,n
ra,a,n,15
rb,b,n,10+5j
rc,c,n,6-8j
```

AC, with **RMS phasors** left unticked this time.

The three line currents read **6.667∠0°**, **8.944∠93.43°** and **10.00∠-66.87°** A, and the neutral current **10.06∠178.5°** A.

All four are correct. The neutral carries a current precisely because the load is unbalanced; in the balanced case it would be zero, which is why a balanced three-wire system needs no neutral at all.

9.3 Wye-Delta

A wye source feeding a delta load. The source still has a centre to use as ground; the load does not, which is the only new thing here.

Balanced

AS7's Example 12.3

For the balanced Y-Δ circuit, find the phase currents and the line currents.

> A balanced *abc*-sequence Y-connected source with $V_{an} = 100 \angle 10°$ V is connected to a Δ-connected balanced load $(8 + j4)\Omega$ per phase. Calculate the phase and line currents.

AS7's Example 12.3

SOLUTION

Nothing says these values are RMS, so leave the convention alone — the question asks for no powers, so it makes no difference either way. The three sources get the angles of the abc sequence.

Circuit Description

```
ea0,a,0,(100∠10°)
eb0,b,0,(100∠-110°)
ec0,c,0,(100∠130°)
rab,a,b,8+4j
rca,c,a,8+4j
rbc,b,c,8+4j
```

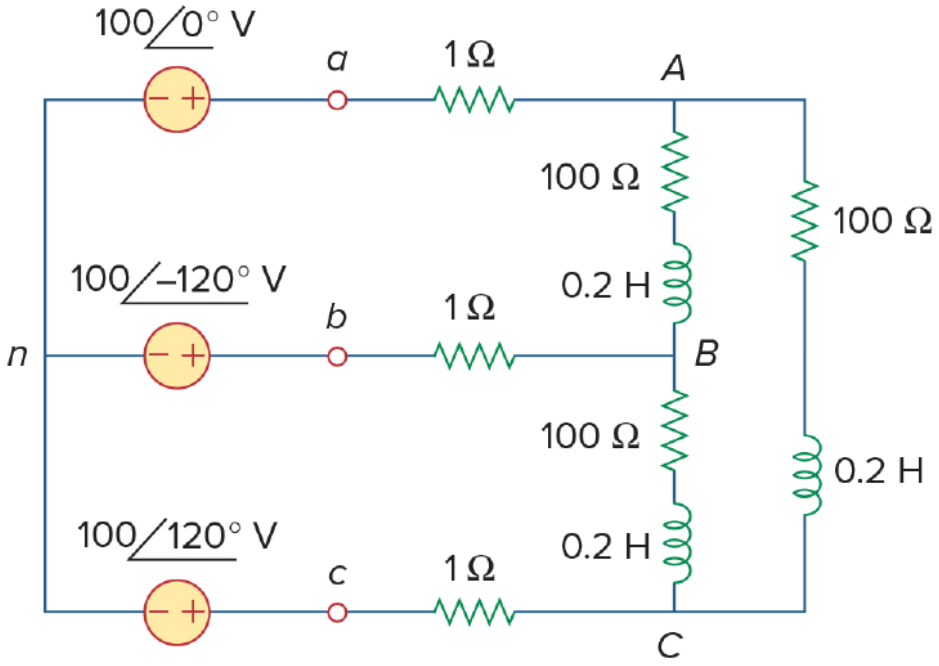
The phase currents are the currents in the three load impedances:

`aa(irab)`, `aa(irbc)` and `aa(irca)` read **19.36∠13.43°**, **19.36∠-106.57°** and **19.36∠133.43°** A.

The line currents are the currents the sources deliver, which is the opposite of the current through each source element:

`aa(-iea0)`, `aa(-ieb0)` and `aa(-iec0)` read **33.54∠-16.57°**, **33.54∠-136.57°** and **33.54∠103.43°** A.

Both sets are correct, and they show the relationship you would expect of a delta load: the line current is √3 times the phase current, and lags it by 30°.



AS7's Example 12.11

AS7's Example 12.11

For the balanced Y-Δ circuit, find the line current I_{aA}, the phase voltage V_{AB}, and the phase current I_{AC}. The source frequency is 60 Hz.

SOLUTION

This one has line impedances, so each source reaches the load through a resistor, and the load's three nodes are separate from the source's three.

Circuit Description

```
ea1,na1,0,100
eb1,nb1,0,(100∠-120°)
ec1,nc1,0,(100∠120°)
raa,na1,na2,1
rbb,nb1,nb2,1
rcc,nc1,nc2,1
rac,na2,nc2,100+24*pi*j
rcb,nc2,nb2,100+24*pi*j
rba,nb2,na2,100+24*pi*j
```

The line current is the current in one of the line resistors, and the phase voltage is the difference between two load nodes:

`aa(iraa)` reads **2.35∠-36.2°** A and `aa(vna2-vnb2)` reads **169.94∠30.81°** V. Both are correct.

For the phase current, `aa(irac)` reads **1.357∠-66.2°** A.

Unbalanced

If you find a simple unbalanced wye-delta problem, let me know.

9.4 Delta-Delta

Until now, choosing a ground node was easy: the centre of the wye of sources. A delta has no centre, which is the first problem. It is solved by picking one of the delta's own nodes on the generator side and calling it **0**.

The second problem is subtler, and it is not Symbulator's alone — SPICE-like simulators dislike it too. **A triangle of three voltage sources cannot be solved.** The third source adds no information, because the first two already fix the voltages at all three nodes, but it does add an unknown: the current through it. Drop the redundant equation and the system has one unknown too many.

> ⚠ WARNING
>
> **Describe a delta source with two sources, not three**
>
> The trick is to leave one source out — the one opposite the node you chose as ground. Two sources fix all three node voltages, so everything outside the delta behaves exactly as it should.
>
> The cost is that you learn nothing about the currents *inside* the sources: the currents you get for the two you kept are not the currents they would carry in a real three-source arrangement. Everything beyond the delta is right; the inside of the generator is not.

Balanced

AS7's Example 12.4

A balanced Δ-connected load of 20 − j15 Ω is fed by a Δ-connected, positive-sequence generator with V_{ab} = 330∠0° V. Find the phase currents of the load and the line currents.

> A balanced Δ-connected load having an impedance $20 - j15\ \Omega$ is connected to a Δ-connected, positive-sequence generator having $\mathbf{V}_{ab} = 330\underline{/0^\circ}$ V. Calculate the phase currents of the load and the line currents.

AS7's Example 12.4

SOLUTION

Node **c** becomes ground, so the source left out is the one opposite it. To read the line currents we add three shorts to act as the lines.

Circuit Description

```
e0a,0,ag,(330∠120°)
eb0,bg,0,(330∠-120°)
```

```
sat,ag,ad
sbt,bg,bd
sct,0,cd
rab,ad,bd,20-15j
rbc,bd,cd,20-15j
rca,cd,ad,20-15j
```

The line currents are the currents through the shorts:

`aa(isat)`, `aa(isbt)` and `aa(isct)` read **22.86∠6.87°**, **22.86∠-113.13°** and **22.86∠126.87°** A.

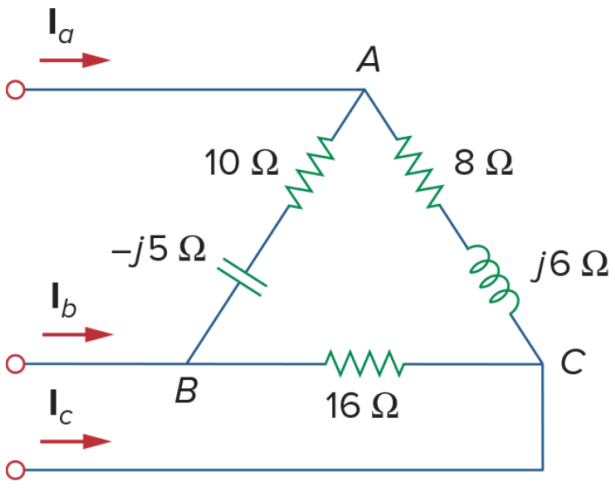
Correct.

Unbalanced

Nothing changes in the method. The load impedances simply differ, and the answers stop being three copies of one another.

AS7's Practice Problem 12.9

The unbalanced Δ-load is supplied by balanced line-to-line voltages of 440 V in positive sequence. Find the line currents, taking $\boldsymbol{V_{ab}}$ as the reference.



AS7's Practice Problem 12.9

SOLUTION

Circuit Description

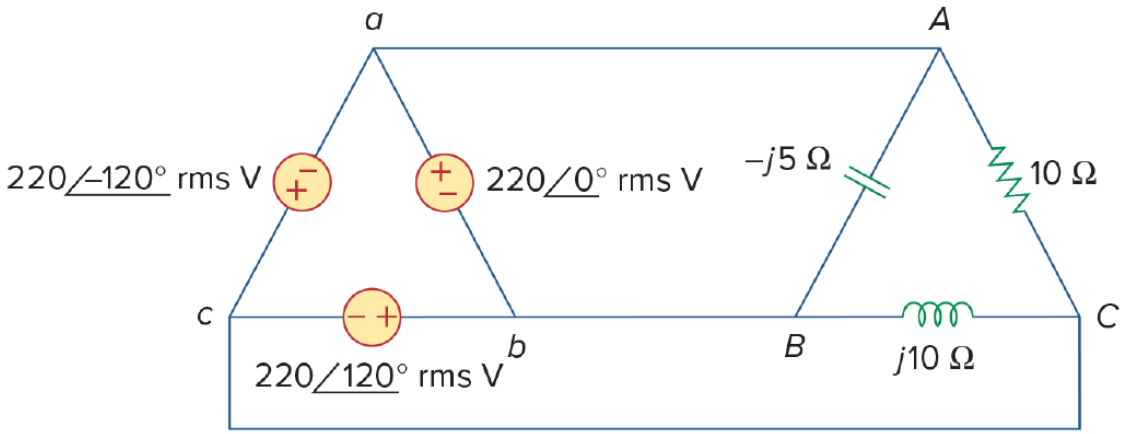
```
e0a,0,ag,(440∠120°)
eb0,bg,0,(440∠-120°)
sla,ag,ad
slb,bg,bd
slc,0,cd
rab,ad,bd,10-5j
rbc,bd,cd,16
rca,cd,ad,8+6j
```

`aa(isla)`, `aa(islb)` and `aa(islc)` read **39.71∠-41.07°**, **64.12∠-139.77°** and **70.13∠74.27°** A.

Correct — and unlike the balanced case, all three differ in magnitude as well as angle.

AS7's Practice Problem 12.10

Find the line currents in the unbalanced three-phase circuit, and the real power absorbed by the load.



AS7's Practice Problem 12.10

SOLUTION

These values are RMS, so tick **RMS phasors** in **Settings** — this one does ask for power.

Circuit Description

```
e0a,0,ag,(220∠-120°)
eb0,bg,0,(220∠120°)
sla,ag,ad
slb,bg,bd
slc,0,cd
rab,ad,bd,-5j
rbc,bd,cd,10j
rca,cd,ad,10
```

The three line currents read **64.00∠80.1°**, **38.11∠-60°** and **42.50∠-135°** A.

For the power, add the three loads' consumption in **Evaluate**:

Evaluate

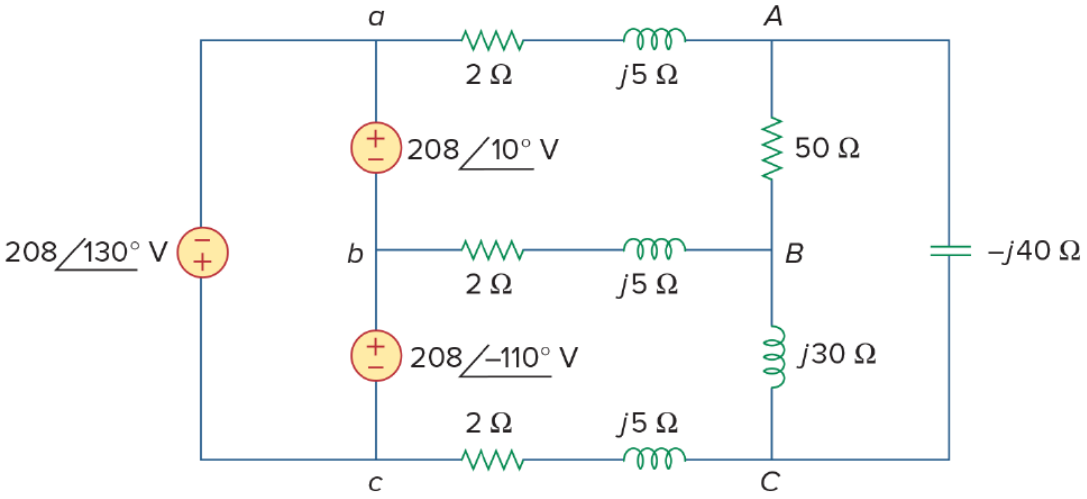
```
prca+prab+prbc
```

It gives **4840** W.

Correct.

AS7's Example 12.12

For the unbalanced Δ-Δ circuit, find the generator current I_{ab}, the line current I_{bB} and the phase current I_{BC}.



AS7's Example 12.12

SOLUTION

Two of the three are ordinary element currents. The first — the current *inside* the generator — is the awkward one, and it is dealt with after them.

Circuit Description

```
e0a,0,ag,(208∠130°)
eb0,bg,0,(208∠-110°)
rla,ag,ad,2+5j
rlb,bg,bd,2+5j
rlc,0,cd,2+5j
rab,ad,bd,50
rbc,bd,cd,30j
rca,cd,ad,-40j
```

Find equivalent is not needed here — a plain AC solve gives both. `aa(irlb)` reads **9.106∠168.48°** A and `aa(irbc)` reads **5.500∠172.47°** A, matching the printed answers.

The remaining answer, the generator current, is trickier. They ask for the current in the source we did not simulate — and even for one of the two we did have, we could not trust it, since the real circuit uses three.

Now, I tried something, and I think I got lucky, because I got the answer the book gives. This is what I tried:

- `(ie0a+ieb0)/3` in **Mini-Tools** with *aa*, which takes an expression as readily as a name and answers with the magnitude and the angle.

What I thought was: the current coming out of the two sources in my simulation would, in reality, come out of three. So adding those two currents and dividing by three may approximate the current out of one source. I am confident that holds in a balanced circuit.

This one is not balanced, but it was the best I had. So I tried it.

5.959∠-177.18° A

And it worked. That is the answer in the book.

> ⚠ WARNING
>
> **Treat that as a guess that was checked, not as a method**
>
> The argument holds in a balanced circuit. This one is not balanced; it was used because there was nothing better to hand, and it happened to land on the book's figure. Check it against something before trusting it elsewhere.

9.5 Delta-Wye

The Δ-Y is the platypus of three-phase systems. The sensible route is usually to convert the Δ source into an equivalent Y and solve it as a Y-Y — and if you are doing this by hand, do that.

Inside Symbulator there is another way: make the **centre of the wye load** the ground node, and describe the delta source with two sources as before.

Balanced

AS7's Example 12.5

For the balanced Δ-Y circuit, find the line currents.

> A balanced Y-connected load with a phase impedance of 40 + j25 Ω is supplied by a balanced, positive sequence Δ-connected source with a line voltage of 210 V. Calculate the phase currents. Use V_{ab} as a reference.

AS7's Example 12.5

SOLUTION

Circuit Description

```
eca,c,a,(210∠120°)
ebc,b,c,(210∠-120°)
ra,a,0,40+25j
rb,b,0,40+25j
rc,c,0,40+25j
```

Node **0** here is the centre of the load's wye, not a node of the source at all.

`aa(ira)`, `aa(irb)` and `aa(irc)` read **2.570∠-62.01°**, **2.570∠177.99°** and **2.570∠57.99°** A.

Correct — balanced, so three equal magnitudes 120° apart.

Unbalanced

If you find a simple unbalanced delta-wye problem, let me know.

LESSON 10

Mutual inductance and ideal transformers

Learn how to describe a *mutual inductance* between two inductors or two impedances using the **m** element. Learn how to describe an ideal *transformer* using the **t** element.

Two coils that share a magnetic field are not two separate elements, and a circuit description has to say so. This lesson covers the two ways of saying it: **m** for a mutual inductance between a pair of elements, and **t** for an ideal transformer.

10.1 How to describe mutual inductance

Textbooks give a coupled pair one of two ways, and Symbulator takes both.

- Two inductors in **henries**, coupled by a mutual inductance also in henries.
- Two impedances in **j**Ω, coupled by a mutual impedance also in jΩ. Written this way the coils are **r** elements with imaginary values.

Either way the coupling itself is an **m** element with four fields: a name starting with **m**, the names of the two coupled elements, and the value of the coupling. For example: `m1, l1,l2,1.5`

That is a mutual inductance of 1.5 H between inductors **l1** and **l2**. In jΩ it looks the same, naming two **r** elements instead: `m1,r1,r2,2j`

A book that gives the coupling as a coefficient k rather than as an inductance is written as it states it, `m1,l1,l2,k=0.5`, and Symbulator works out $M = k\sqrt{(L_1 L_2)}$ itself. The same spelling works for coils in jΩ. NR9's Assessment Problem 9.14, the last of the problems below, is written that way.

> ⚠ WARNING
>
> **Don't mix the two**
>
> `m1,l1,r1,2` couples an inductor given in henries to one given in jΩ. Symbulator refuses it, and refuses a coil or a coupling whose value is not of the right kind – real and positive in henries, positive imaginary in jΩ – or a coupling stronger than $\sqrt{(L_1 L_2)}$. Keep all three values in the same units: all henries, or all jΩ.

The dots are the node order

Every coupled-inductor problem in a textbook has a dot on each coil, marking the ends that share polarity. Symbulator has no way to draw a dot, so **the first node you name in each coupled element is its dotted end**.

That is the whole convention, and it is the thing most likely to give you a correct-looking wrong answer. If you list the nodes of one coil the other way round, the sign of the coupling flips and every current comes back wrong.

> ⚠ WARNING
>
> **Keep the dots consistent**
>
> Read the schematic, decide which end of each coil is dotted, and declare that node first — in both coupled elements. Look at the solved examples below and notice that the dotted ends are always the first node given.

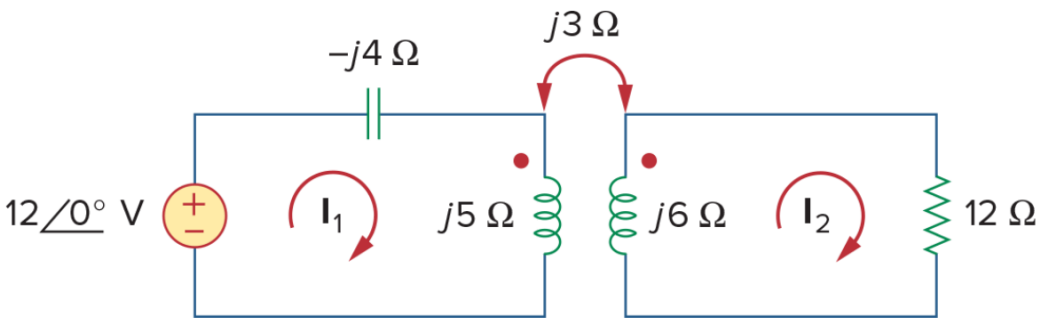
The value of a mutual inductance can use SI prefixes.

What answers do you get? None. An **m** element is a statement about two other elements, not a component with its own voltage and current — its effect shows up in the currents of the elements it couples.

10.2 Instructive m problems

AS7's Example 13.1

Calculate the phasor currents I_1 and I_2 in the circuit.



AS7's Example 13.1

SOLUTION

Three things to notice in the description. The bottom node of both halves is ground. The coils are given in jΩ, so they are **r** elements with imaginary values, and the mutual is imaginary too, to match. And the dotted node of each coupled element is named first.

Circuit Description

```
e,1,0,12
```

```
r1,1,2,-4j
r2,2,0,5j
m,r2,r3,3j
r3,3,0,6j
r4,3,0,12
```

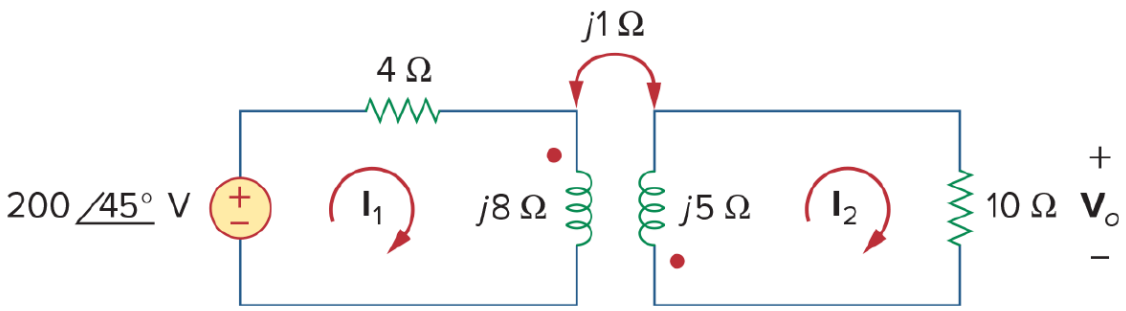
AC, and leave **RMS phasors** unticked. Then read the two currents with **Mini-Tools** set to *aa*:

`aa(ir2)` reads **13.02∠-49.4°** A and `aa(ir4)` reads **2.910∠14.04°** A.

Both are correct.

AS7's Practice Problem 13.1

Determine the voltage V_o in the circuit.



AS7's Practice Problem 13.1

SOLUTION

Again the dotted node of the coupled inductor comes first.

Circuit Description

```
e,1,0,(200∠45°)
r1,1,2,4
r2,2,0,8j
m,r2,r3,1j
r3,0,o,5j
r4,o,0,10
```

`aa(vo)` reads **20.00∠-134.43°** V.

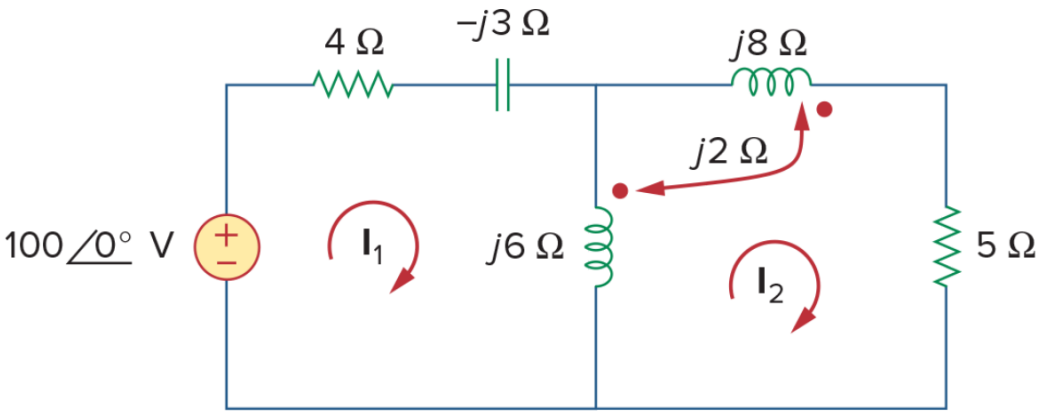
Correct.

AS7's Example 13.2

Calculate the mesh currents in the circuit.

SOLUTION

Notice the node order again, matching the dots.



AS7's Example 13.2

Circuit Description

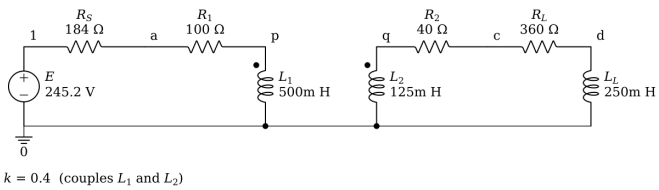
```
e,1,0,100
r1,1,2,4-3j
r2,2,0,6j
m,r2,r3,2j
r3,3,2,8j
r4,3,0,5
```

`aa(ir1)` reads **20.30∠3.5°** A and `aa(ir4)` reads **8.693∠19.03°** A.

Correct.

NR9's Assessment Problem 9.14

A linear transformer couples a load of a 360 Ω resistor in series with a 0.25 H inductor to a sinusoidal source of 245.20 V maximum at 800 rad/s, whose internal impedance is 184 Ω. The transformer has $\boldsymbol{R_1}$ = 100 Ω, $\boldsymbol{L_1}$ = 0.5 H, $\boldsymbol{R_2}$ = 40 Ω, $\boldsymbol{L_2}$ = 0.125 H and a coefficient of coupling k = 0.4, with the dots at the top of both coils. Calculate (a) the reflected impedance, (b) the primary current and (c) the secondary current.



NR9's Assessment Problem 9.14, drawn by Symbulator

SOLUTION

The book gives the coupling as a coefficient, so the **m** line takes it as it is, `m,l1,l2,k=0.4`, and there is no mutual inductance to work out first. Both dots are at the top, so both coils are written top node first. We call the source's internal impedance `rs`, and the load's resistor and inductor `rl` and `ll`. The book draws the two sides on separate bottom rails. We join them at ground, which changes no current, because nothing conducts from one winding to the other.

Circuit Description

```
e,1,0,245.2
rs,1,a,184
r1,a,p,100
l1,p,0,0.5
m,l1,l2,k=0.4
l2,q,0,0.125
r2,q,c,40
rl,c,d,360
ll,d,0,0.25
```

Set **Analysis** to *AC — alternating current* and put **800** in the **ω — angular frequency** box. In **Settings**, set **Rounding** to *approx to n digits* with **n** = 4 and tick **Show AC answers as polar phasors**.

Parts (b) and (c) are read straight off the cards. The primary current is the current through `r1`, and the secondary current the current through `r2`, each in the direction of the book's arrow:

current through r1

$$i_{r1} = 0.5000\angle -53.13^\circ$$

current through r2

$$i_{r2} = 0.08000\angle 0^\circ$$

So **(b)** the primary current is 0.5 cos(800t − 53.13°) A, and **(c)** the secondary current is 0.08 cos 800t A.

(a) The reflected impedance is what the secondary adds to the impedance the source sees. The source's card reports that impedance as `ze`. Without the coupling it would be the source's own 184 Ω, the primary winding's 100 Ω and the primary coil's $j\omega L_1 = j800 \times 0.5 = j400$ Ω, so we type the difference into **Evaluate**:

Evaluate

```
ze - (184 + 100 + 400j)
```

It gives **10.24 - 7.68j** Ω. All three agree with the answers the book prints. The coupling written as the inductance it stands for, `m,l1,l2,0.1`, gives the same, since $M = 0.4 \times \sqrt{(0.5 \times 0.125)} = 0.1$ H.

10.3 How to describe ideal transformers

An ideal transformer has four terminals — two a side. Symbulator simulates the case where **one terminal on each side is grounded**, which covers the textbook problems; you name only the two live nodes.

A **t** element takes five fields: a name starting with `t`, the live node on the first side, the live node on the second side, and the turns on each side. For example: `t1,1,2,80,800`

That is 80 turns on the side connected to node **1** and 800 on the side connected to node

2, wound the same way. Only the ratio matters, so `1,10` would describe the same transformer.

Polarity is a minus sign. If the dots are on opposite ends, make one of the two turn counts negative: `t,a,b,1,-5`

It does not matter which of the two you make negative, as long as one of them is.

The turns may be written as a pair in brackets, `t1,1,2,[80,800]`, which means exactly the same as the five-field form.

All four terminals. When neither bottom terminal is ground, write each side as a bracketed pair, top node then bottom node, and the turns as a pair:

> Circuit Description
> `t1,[1,3],[2,4],[80,800]`

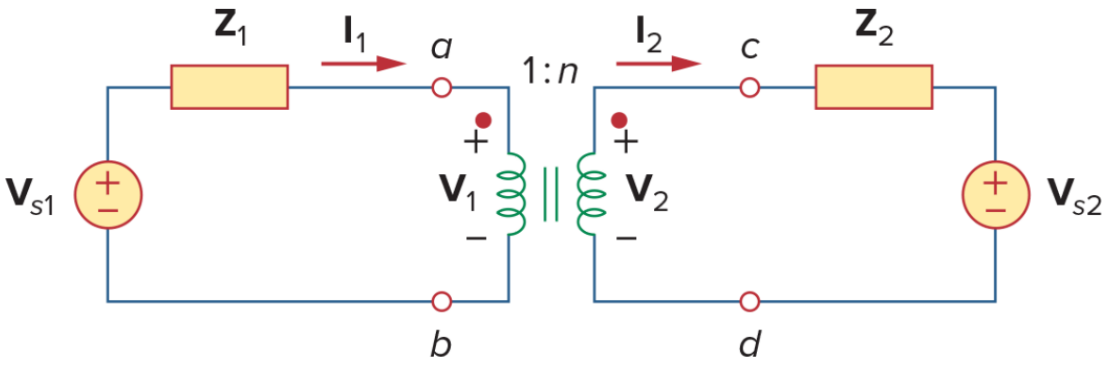
That is a primary between nodes **1** and **3**, a secondary between nodes **2** and **4**, 80 and 800 turns. The dots sit at the two top nodes, as before. An ideal transformer conducts nothing from one side to the other, so a side with no path of its own to ground is an island: Symbulator takes one of its nodes — the winding's lower terminal — as that side's own reference, holds it at 0, and says so in a note above the results. Every current and every voltage difference on that side is the same whichever node is chosen; only its absolute potentials depend on it, and they have no meaning anyway. Either terminal of a pair may be `0`, so `t1,[1,0],[2,0],[80,800]` is the two-node form written out.

What answers do you get? The voltages at the live nodes, as for any node in the circuit, and **the current entering the transformer at each of its live nodes**, named with the transformer and the node — for a transformer `t` on nodes **1** and **2**, that is `it1` and `it2`. With all four terminals named, there are four such currents, one per node.

10.4 Instructive t problems

AS7's Figure 13.33

Obtain V_{Th} and Z_{Eq} for the part of the circuit to the right of nodes **a** and **b**.



AS7's Figure 13.33

SOLUTION

Everything left of a and b is ignored. Note that this one is entirely symbolic — the turns ratio, the impedance and the source are all names.

Circuit Description

```
t,2,3,1,n
r2,3,4,z2
e2,4,0,vs2
```

Find equivalent, *Thévenin / Norton*, nodes **2** and **0**, in DC.

Results gives

Thévenin voltage

$$v_{th} = \frac{vs_2}{n}$$

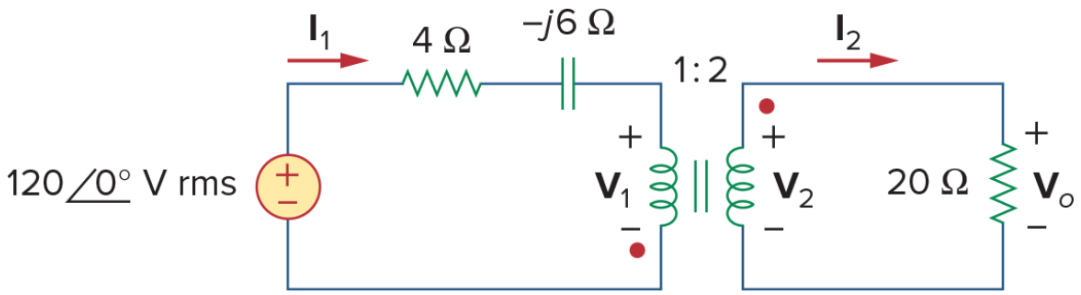
equivalent resistance

$$R_{eq} = \frac{z_2}{n^2}$$

Correct — the transformer refers the source by the turns ratio and the impedance by its square, which is exactly what an ideal transformer does.

AS7's Example 13.8

For the ideal transformer circuit, find the source current I_1, the output voltage V_o, and the complex power supplied by the source.



AS7's Example 13.8

SOLUTION

One of the turn counts is negative here, because the dots are of opposite polarity.

Circuit Description

```
e,1,0,120
r1,1,2,4-6j
t,2,3,1,-2
ro,3,0,20
```

AC, with **RMS phasors** ticked this time — the question asks for complex power, and that is the convention it wants.

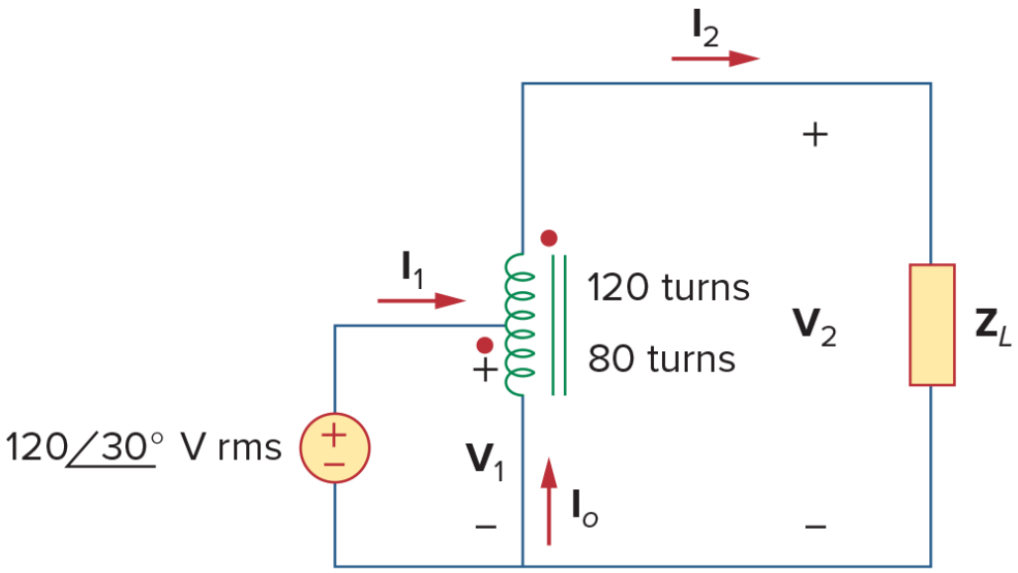
Three readings from **Mini-Tools** with *aa*: `ir1` is **11.09∠33.69°** A, `vro` is **110.9∠-146.31°** V, and `-se` — the complex power *supplied*, so the opposite of the power consumed by the source — is **1331∠-33.69°** VA.

Three conversions in a row is the point at which the setting is less work than the tool: tick **Show AC answers as polar phasors** in **Settings** and the first two are already in that form when the circuit solves. The third still wants *aa*, because `-se` is an expression rather than an answer.

All three are correct.

AS7's Example 13.11

Refer to the autotransformer circuit. Calculate I_1, I_2 and I_o if Z_L = 8 + j6 Ω, and the complex power supplied to the load.



AS7's Example 13.11

SOLUTION

Symbulator has no dedicated element for an autotransformer. The same **t** element does it: the input is node **1**, the output is node **2**, and the turns are the 80 of the lower winding and the 80 + 120 of the whole.

Circuit Description

```
e,1,0,(120∠30°)
t,1,2,80,80+120
rl,2,0,8+6j
```

AC, with **RMS phasors** ticked.

This is how the three currents come out of the two the transformer reports:

- I_1 is the current entering the transformer at node **1**, `it1`.

- I_2 is the opposite of the current entering it at node **2**, `it2`.
- I_o is the opposite of the sum of the two, since what enters the transformer at its two live nodes leaves it through the common one.

Results gives, under **t**,

> current into port at node 1
>
> $$i_{t1} = 75\angle -6.87^\circ$$

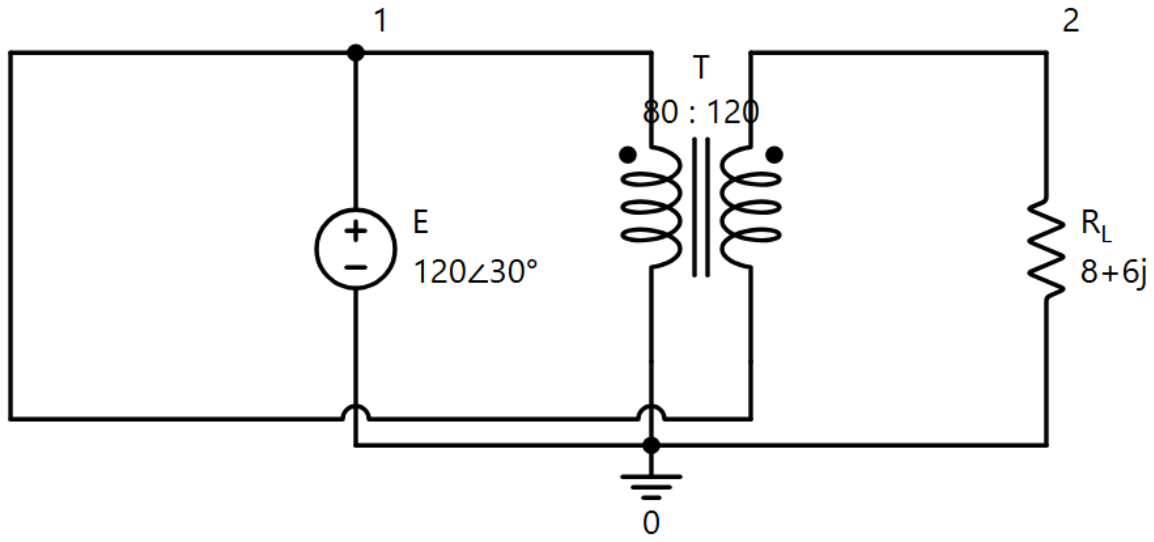
> current into port at node 2
>
> $$i_{t2} = 30\angle 173.13^\circ$$

so I_1 is **75∠-6.87°** A, I_2 is the opposite of `it2`, **30∠-6.87°** A, and I_o, read with *aa* on `-it1-it2`, is **45∠173.13°** A.

All three are correct.

The autotransformer as one tapped winding

The same autotransformer, described as what it physically is: one winding with a tap. The lower section, 80 turns, runs from the tap to ground; the upper section, 120 turns, runs from the top of the winding down to the tap. Find I_1 and I_2 again.



The autotransformer as one tapped winding, drawn by Symbulator

SOLUTION

The second winding's bottom terminal is the first winding's top, node **1**, so this needs the four-terminal form: the first winding is the pair `[1,0]`, the second the pair `[2,1]`.

> Circuit Description
>
> ```
> e,1,0,(120∠30°)
> t,[1,0],[2,1],[80,120]
> rl,2,0,8+6j
> ```

AC, with **RMS phasors** ticked. **Results** gives, under **t**,

current into port at node 1

$$i_{t1} = 75\angle -6.87^\circ$$

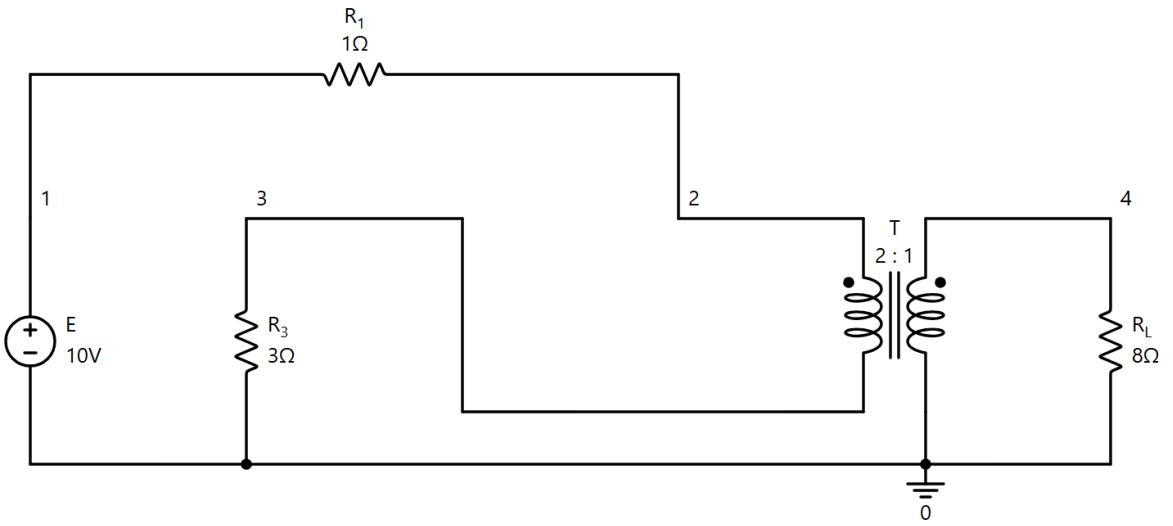
current into port at node 2

$$i_{t2} = 30\angle 173.13^\circ$$

Node **1** is a terminal of both windings, so `it1` is the total entering the transformer there, which is the source current I_1, **75∠-6.87°** A; and I_2 is the opposite of `it2`, **30∠-6.87°** A. The same answers as with the two-node description above, from a description that looks like the circuit.

A transformer whose primary is not grounded

A 10 V source feeds a 1 Ω resistor and then the primary of a 2:1 ideal transformer; the far end of the primary returns to ground through a 3 Ω resistor. The secondary is grounded and drives an 8 Ω load. Find the load current and the current into each terminal of the transformer.



A transformer whose primary is not grounded, drawn by Symbulator

SOLUTION

The primary sits between nodes **2** and **3**, neither of them ground, so it is written as the pair `[2,3]`; the secondary's bottom is ground, and the pair `[4,0]` says so.

Circuit Description

```
e,1,0,10
r1,1,2,1
t,[2,3],[4,0],[2,1]
r3,3,0,3
rl,4,0,8
```

DC. **Results** gives, under **t**,

current into port at node 2

$$i_{t2} = \frac{5}{18}$$

current into port at node 3

$$i_{t3} = -\frac{5}{18}$$

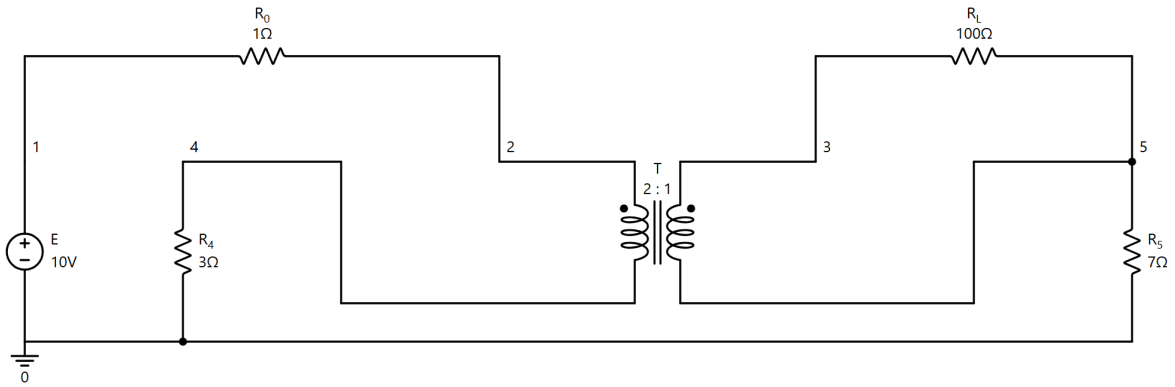
current into port at node 4

$$i_{t4} = -\frac{5}{9}$$

and `irl` is **5/9** A. Check: the 8 Ω load reflects to the primary as 8 × (2/1)² = 32 Ω, in series with 1 + 3 Ω, so the primary current is 10/36 = 5/18 A, entering at node **2** and leaving at node **3**; the 2:1 ratio doubles it on the secondary, where `it4` is negative because that current is leaving the transformer into the load.

Both windings between live nodes, and why a side needs a ground

The same 2:1 transformer, now with its secondary between two live nodes as well: the 100 Ω load sits across the secondary, and node **5** is tied to ground through a 7 Ω resistor. Find the currents.



Both windings between live nodes, drawn by Symbulator

SOLUTION

Circuit Description

```
e,1,0,10
r0,1,2,1
t,[2,4],[3,5],[2,1]
r4,4,0,3
rl,3,5,100
r5,5,0,7
```

DC. **Results** gives, under **t**,

current into port at node 2

$$i_{t2} = \frac{5}{202}$$

current into port at node 4

$$i_{t4} = -\frac{5}{202}$$

current into port at node 3

$$i_{t3} = -\frac{5}{101}$$

current into port at node 5

$$i_{t5} = \frac{5}{101}$$

and `ir5` is **0**: the secondary's current circulates through the load alone, and the resistor to ground carries nothing. What it does is give the secondary side a reference. An ideal transformer conducts nothing from one winding to the other, so without `r5` that side would have no path to node **0** at all — delete it and Symbulator still solves, taking node **5** as the side's own reference and saying so in a note: *Node(s) 5, 3 have no path to node 0 (they lie behind a port or a coupling), so their voltages are measured against 5, taken as 0.* The currents and `v3 - v5` come out the same either way; only the two absolute potentials differ, and on a real bench they would be whatever the wiring made them.

10.5 Switching with coupled coils

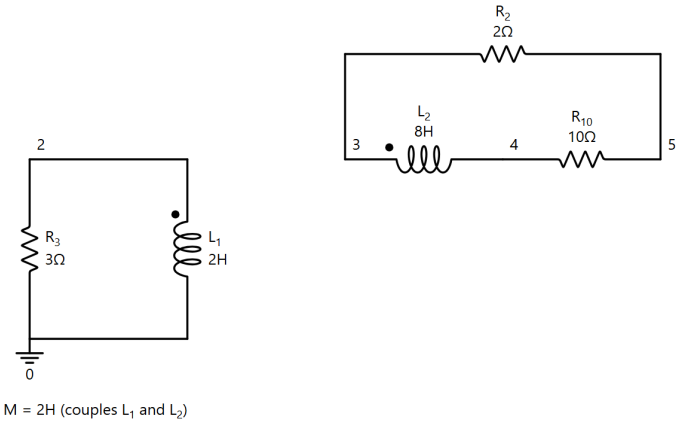
Seven problems from Nilsson and Riedel's *Electric Circuits*: four from the 11th edition, each with a switch that moves at t = 0 and a pair of coupled coils on the far side of it, and three from the 12th, in which a switch closes onto a source with nothing stored. The method is the one Lesson 6 uses for every switch: a DC solve before the switch gives the coil currents, and those become the initial conditions of a transient solve after it — or simply zeros, when the book says no energy is stored. In the first four the coupled side has no ground of its own, and none is needed: a coupling conducts nothing across, so Symbulator takes one node of that side as its reference and says so. The variable asked for is the one the book marks in blue.

NR11's 60 V source and coupled 2 H and 8 H coils

A 60 V source feeds a 9 Ω resistor and, through a switch at **a**, a 3 Ω resistor in series with a 2 H coil. The coil is coupled, M = 2 H, to an 8 H coil closed on 2 Ω and 10 Ω. The switch has been at **a** a long time and moves to **b** at t = 0, shorting the 3 Ω and the 2 H together. Find i_1 in the 2 H coil and i_2 in the 8 H coil for t > 0. Both dots are at the coils' upper ends.

SOLUTION

Before the switch, in DC, with the source in and the switch at **a**:



The circuit after the switch, drawn by Symbulator

Circuit Description

```
e,1,0,60
r9,1,a,9
r3,a,2,3
l1,2,0,2
l2,3,4,8
r2,3,5,2
r10,5,4,10
m,l1,l2,2
```

Both coils are shorts in DC, so `il1` = **5** A (60 V over 9 + 3 Ω) and `il2` = **0**. The secondary has no path to node **0**, and the results say so in a note: its voltages are measured against node **4**.

After the switch the source is gone and the 3 Ω and the 2 H are shorted through **b**; the 5 A is the coil's initial condition. In TR:

Circuit Description

```
r3,0,2,3
l1,2,0,2,5
l2,3,4,8,0
r2,3,5,2
r10,5,4,10
m,l1,l2,2
```

current through l1

$$i_{l1} = \frac{5}{2}\left(e^{-t} + e^{-3t}\right)\ \text{A}$$

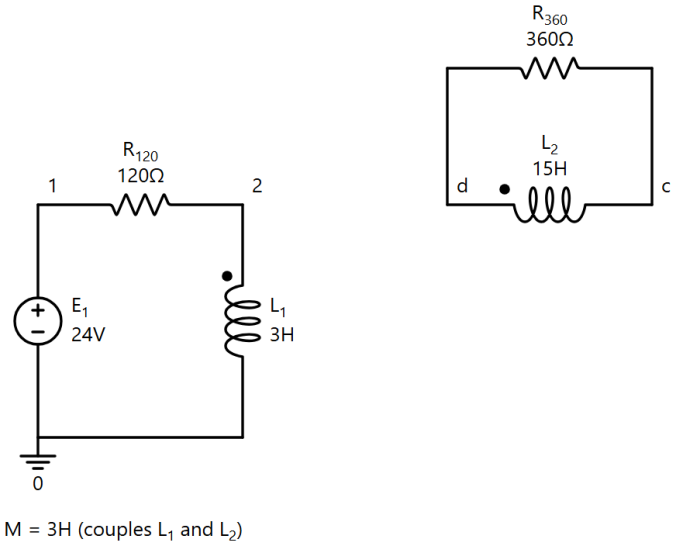
current through l2

$$i_{l2} = \frac{5}{4}\left(e^{-t} - e^{-3t}\right)\ \text{A}$$

Two modes, at 1 and 3 per second: i_1 starts from 5 A and i_2 from 0, and both vanish. Both satisfy the pair of coil equations exactly, which is how they were checked.

NR11's two switches and coupled 3 H and 15 H coils

On the left, a 24 V source and 120 Ω feed a 3 H coil through switch **1**, which has been open a long time and closes at t = 0. On the right, a 15 H coil, coupled to the first by M = 3 H, is connected through switch **2** to either **a** — a 10 Ω resistor and a 20 V source — or **b**, a 360 Ω resistor. Switch 2 has been at **a** a long time and moves to **b** at t = 0. Find i_1 for t > 0. The dot of the 3 H coil is at its upper end, the dot of the 15 H coil at its lower end.



The circuit after the switches, drawn by Symbulator

SOLUTION

Before the switches there is no primary current at all, and the 20 V drives 2 A through the 10 Ω and down through the 15 H coil. In DC:

Circuit Description

```
e,a,0,20
r10,a,c,10
l2,d,c,15
```

The dot of the 15 H coil is at its bottom, so its first node is **d**, the bottom, and the current from **d** to **c** reads `il2` = **-2** A.

After the switches the 24 V drives the primary from rest and the secondary, starting at 2 A, decays through the 360 Ω. In TR, ticking **Do you want to limit the results to save time?** and asking for the one current:

What results are you after? List the variables here

```
il1
```

Circuit Description

```
e1,1,0,24
r120,1,2,120
l1,2,0,3,0
l2,d,c,15,-2
```

```
r360,c,d,360
m,l1,l2,3
```

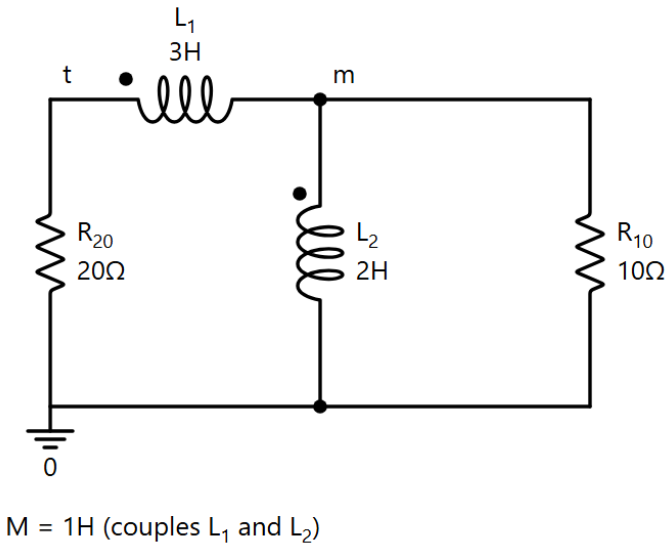
current through l1

$$i_{l1} = \frac{1}{5} - \frac{31}{20}\,e^{-20t} + \frac{27}{20}\,e^{-60t}\ \text{A}$$

It starts from 0, settles at 24/120 = 0.2 A, and its two modes are at 20 and 60 per second. The dots are at opposite ends, which is why the secondary is written bottom node first with a positive M; writing it top node first with M = −3 gives the same answer.

NR11's 90 V source and coupled 3 H and 2 H coils

A 90 V source and 5 Ω feed, through a switch at **a**, a 3 H coil whose far end joins a 2 H coil and a 10 Ω resistor to ground; the coils are coupled, M = 1 H, dots at both upper ends. The switch has been at **a** a long time and moves to **b** at t = 0, replacing the source by a 20 Ω resistor. Find i_o, the current into the 3 H coil, for t > 0.



The circuit after the switch, drawn by Symbulator

SOLUTION

Before the switch both coils are shorts in DC, and the 2 H shorts the 10 Ω:

Circuit Description
```
e,1,0,90
r5,1,t,5
l1,t,m,3
l2,m,0,2
r10,m,0,10
m,l1,l2,1
```

`il1` = **18** A, `il2` = **18** A and `ir10` = **0**: all of it goes through the coils.

After the switch, in TR, asking only for `il1` in the same way:

Circuit Description

```
r20,t,0,20
l1,t,m,3,18
l2,m,0,2,18
r10,m,0,10
m,l1,l2,1
```

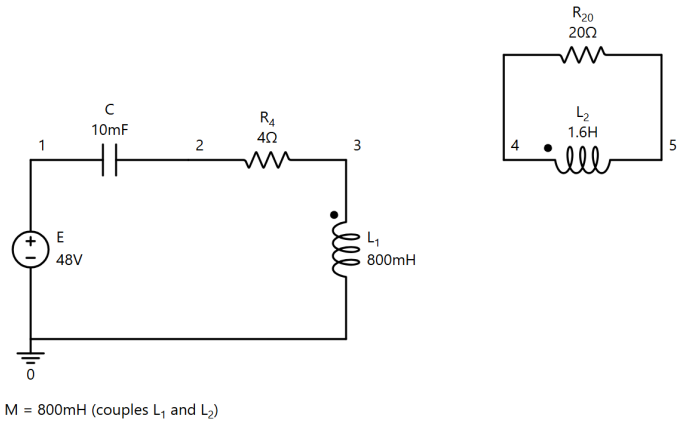
current through l1

$$i_{l1} = 12\,e^{-2t} + 6\,e^{-20t}\ \text{A}$$

From 18 A to zero, with modes at 2 and 20 per second.

NR11's capacitor shorted by a switch, coupled 0.8 H and 1.6 H coils

A 48 V source feeds a 10 mF capacitor, a 4 Ω resistor and a 0.8 H coil in series. A switch across the capacitor has been closed a long time and opens at t = 0. The coil is coupled, M = 0.8 H, to a 1.6 H coil across a 20 Ω resistor; both dots are at the upper ends. Find $\boldsymbol{v_o}$, the voltage across the 20 Ω, for t > 0.



The circuit after the switch opens, drawn by Symbulator

SOLUTION

Before the switch the capacitor is shorted, so it is left out of the DC picture:

Circuit Description

```
e,1,0,48
r4,1,3,4
l1,3,0,0.8
l2,4,5,1.6
r20,4,5,20
m,l1,l2,0.8
```

`il1` = **12** A and `il2` = **0**.

After the switch opens the capacitor enters the loop, uncharged, with the coil carrying its 12 A. In TR, and then in **Evaluate**:

Evaluate

```
v4-v5
```

Circuit Description

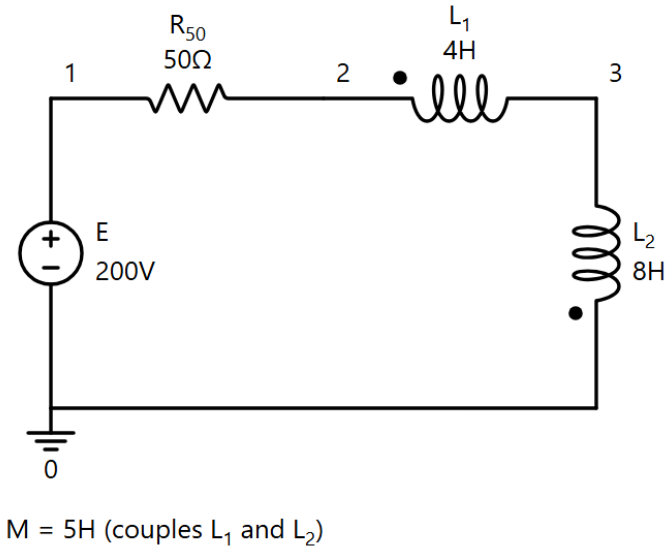
```
e,1,0,48
c,1,2,10'm,0
r4,2,3,4
l1,3,0,0.8,12
l2,4,5,1.6,0
r20,4,5,20
m,l1,l2,0.8
```

$$v_o = e^{-5t}\,(60\cos 10t - 120\sin 10t) - 60\,e^{-25t}\ \text{V}$$

A damped oscillation at 10 rad/s, the primary's series RLC, plus the secondary's own mode at 25 per second; it starts from 0 and returns to 0. Since the secondary's reference is node **5**, v4 alone reads the same thing.

NR12's Problem 7.68

There is no energy stored in the circuit when the switch closes at t = 0. A 200 V source and 50 Ω feed a 4 H coil in series with an 8 H coil, coupled by M = 5 H; the 4 H coil's dot is at its left end, the 8 H coil's at its bottom. Find $\boldsymbol{i}$(t), $\boldsymbol{v_1}$(t) across the 4 H and $\boldsymbol{v_2}$(t) across the 8 H, and say whether the answers make sense.



NR12's Problem 7.68, drawn by Symbulator

SOLUTION

The current enters the 4 H at its dot and the 8 H at its undotted end, so the 8 H is written bottom node first and the pair opposes:

Circuit Description

```
e,1,0,200
```

```
r50,1,2,50
l1,2,3,4,0
l2,0,3,8,0
m,l1,l2,5
```

In TR, with the fifth field of each coil, the initial current, at 0:

current through l1

$$i_{l1} = 4 - 4\,e^{-25t}\ \text{A}$$

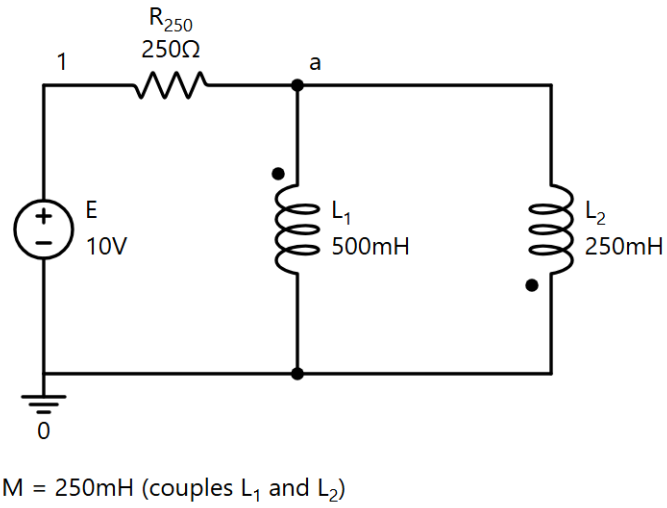
voltage at node 3

$$v_3 = 300\,e^{-25t}\ \text{V}$$

and **Evaluate** with `v2-v3` gives $\boldsymbol{v_1} = -100\,e^{-25t}$ V. The pair reduces to 4 + 8 − 2·5 = 2 H, whence the 25 per second (2 H over 50 Ω), and $\boldsymbol{v_1}$ is *negative* because the 5 H coupling outweighs the coil's own 4 H; the two voltages still add to $200\,e^{-25t}$, which is 200 − 50 $\boldsymbol{i}$.

NR12's Problem 7.70

There is no energy stored when the switch closes. A 10 V source and 250 Ω feed a 0.5 H coil and a 0.25 H coil in parallel, coupled by M = 0.25 H; the 0.5 H coil's dot is at its top, the 0.25 H coil's at its bottom. Find $\boldsymbol{i_o}$, $\boldsymbol{v_o}$, $\boldsymbol{i_1}$ and $\boldsymbol{i_2}$, both branch currents taken downward.



NR12's Problem 7.70, drawn by Symbulator

SOLUTION

The 0.25 H coil is written bottom node first, since that is its dotted end:

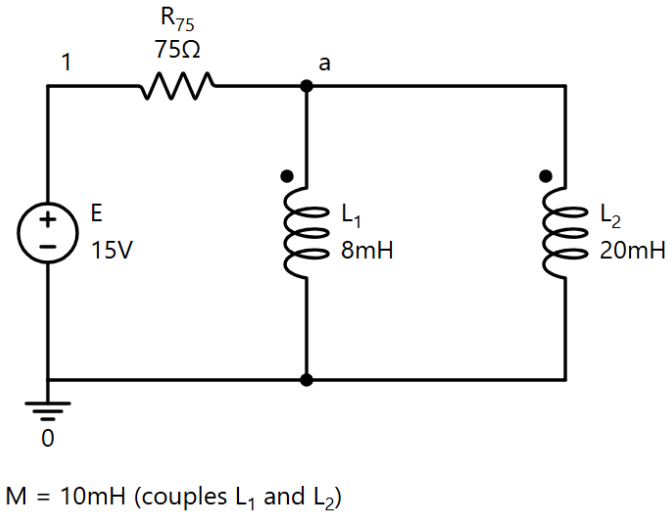
Circuit Description

```
e,1,0,10
r250,1,a,250
l1,a,0,0.5,0
l2,0,a,0.25,0
m,l1,l2,0.25
```

TR. **Results** gives `ir250` = $i_o = 0.04(1 - e^{-5000t})$ A, `va` = $v_o = 10\,e^{-5000t}$ V and `il1` = $i_1 = 0.016(1 - e^{-5000t})$ A; i_2 is downward, the opposite of `il2`, so **Evaluate** with `-il2` reads $0.024(1 - e^{-5000t})$ A. The parallel pair opposes and reduces to (0.5·0.25 − 0.25²)/(0.5 + 0.25 + 2·0.25) = 0.05 H, whence the 5000 per second with 250 Ω.

NR12's Problem 7.71

There is no energy stored when the switch closes. A 15 V source and 75 Ω feed an 8 mH coil and a 20 mH coil in parallel, coupled by M = 10 mH, both dots at the top. Find i_o, v_o, i_1 and i_2.



NR12's Problem 7.71, drawn by Symbulator

SOLUTION

Both coils are written top node first:

Circuit Description

```
e,1,0,15
r75,1,a,75
l1,a,0,8'm,0
l2,a,0,20'm,0
m,l1,l2,10'm
```

TR. **Results** gives `ir75` = $i_o = 0.2(1 - e^{-10000t})$ A, `va` = $v_o = 15\,e^{-10000t}$ V, `il1` = $i_1 = 0.25(1 - e^{-10000t})$ A and `il2` = $i_2 = -0.05(1 - e^{-10000t})$ A. The pair aids and reduces to (160 − 100)/(28 − 20) mH = 7.5 mH, whence the 10 000 per second with 75 Ω. The backwards i_2 is the part worth a second look: with both coils shorts at the end, the split between the branches is set by their flux linkages rather than by resistance, and 0.25 − 0.05 is the 0.2 A the 75 Ω allows.

LESSON 11

Bode plots and resonance

Learn to do gain and phase *Bode plots* for transfer functions of ω and s in the **Plotting Tools** card. Find the *resonant frequency* of an answer by solving for the frequency at which the reactance vanishes.

Two things that only make sense across a range of frequencies rather than at one: the Bode plot, which draws a transfer function's gain and phase as frequency sweeps, and resonance, which is the one frequency where a circuit's reactance vanishes.

11.1 Do Bode plots

A Bode plot shows the gain and the phase of a transfer function against frequency, on a logarithmic axis.

Symbulator draws it in the **Plotting Tools** card below the results. Solve the circuit first, then set **Plot type** to *Bode plot of a variable (FD)*; the two time boxes become frequencies.

Circuit Description

```
e,1,0,1
r1,1,2,1000
c1,2,0,1e-6
```

- **Variable to plot**: `v2`
- **Start frequency (Hz)** and **End frequency (Hz)**: 10 and 100000
- **Points**: 300

Both curves appear together, magnitude in decibels above and phase in degrees below. For this low-pass RC that is the rolloff you would expect: flat at low frequency, falling 20 dB per decade after the corner, with the phase heading for −90°.

⚠ WARNING

The variable, not an expression

The box takes an answer's *name*, as **Results** spells it, not a formula. A formula belongs in the transfer-function plot type, next.
Solving in AC and plotting against frequency would not work: an AC result is one phasor at one ω. The plot re-solves the circuit at every frequency.

When you have H(s) itself

The textbook often hands you the transfer function and no circuit. Symbulator takes it directly: set **Plot type** to *Bode plot of a transfer function H(s) (FD)*, and the variable box becomes **Transfer function H(s)**. Type the function there; the Circuit Description is ignored for this plot type, so it works on an empty page.

- Write the function **in terms of s**. A textbook function of jω becomes one of s by writing `s` wherever jω appears — a Bode plot sweeps the imaginary axis, where s *is* jω.
- The expression takes the same shorthand a circuit value does — `^` for powers, implied multiplication, `1'k` — and it must be numeric apart from `s`. A leftover symbol is refused by name.
- The frequency boxes are in **hertz**. The examples in this lesson give their ranges in rad/s; divide by 2π for the same sweep in Hz (0.1 to 300 rad/s is about 0.016 to 48 Hz).

AS7's Example 14.3

Construct the Bode plots for the given transfer function.

$$H(\omega) = \frac{200\, j\omega}{(j\omega + 2)(j\omega + 10)}$$

Symbulator takes the expression directly: set **Plot type** to *Bode plot of a transfer function H(s) (FD)* and type the problem's function into the box, writing `s` wherever the textbook writes jω — no imaginary operator involved. The frequency boxes are in hertz, so the sweep of 0.1 to 300 rad/s is about 0.016 to 48 Hz.

> Transfer function H(s)
> `200*s/((s+2)*(s+10))`

- **Start frequency (Hz)** and **End frequency (Hz)**: 0.016 and 48

Reading the answers off the curve: the maximum of **24.4** dB at 4.47 rad/s sits near 0.71 Hz, and the zero crossing at 199.8 rad/s near 31.8 Hz.

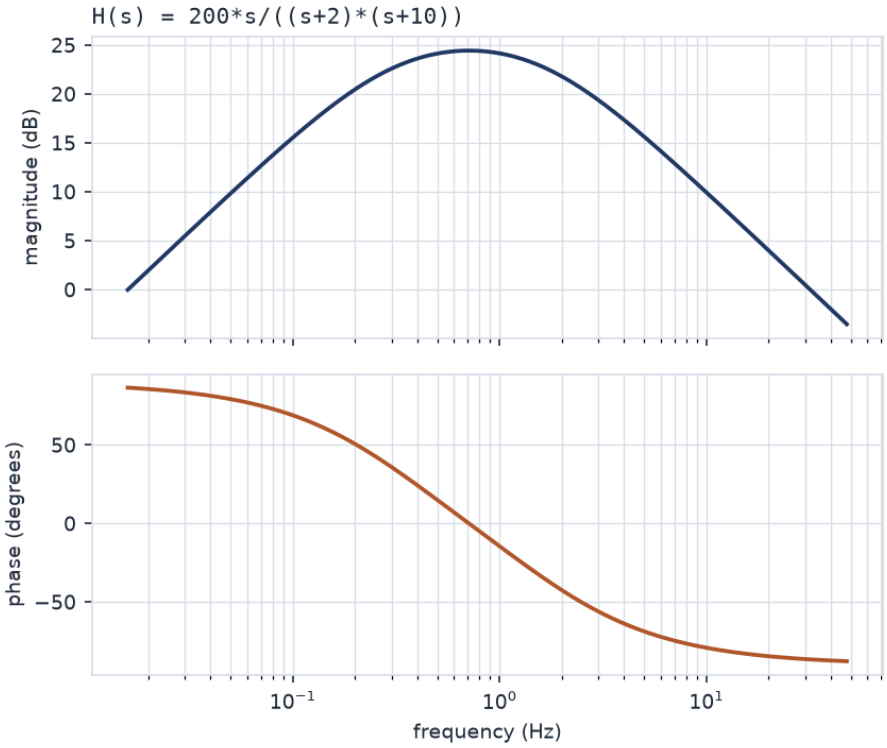
AS7's Practice Problem 14.3

Draw the Bode plots for the given transfer function.

$$H(\omega) = \frac{5\,(j\omega + 2)}{j\omega\,(j\omega + 10)}$$

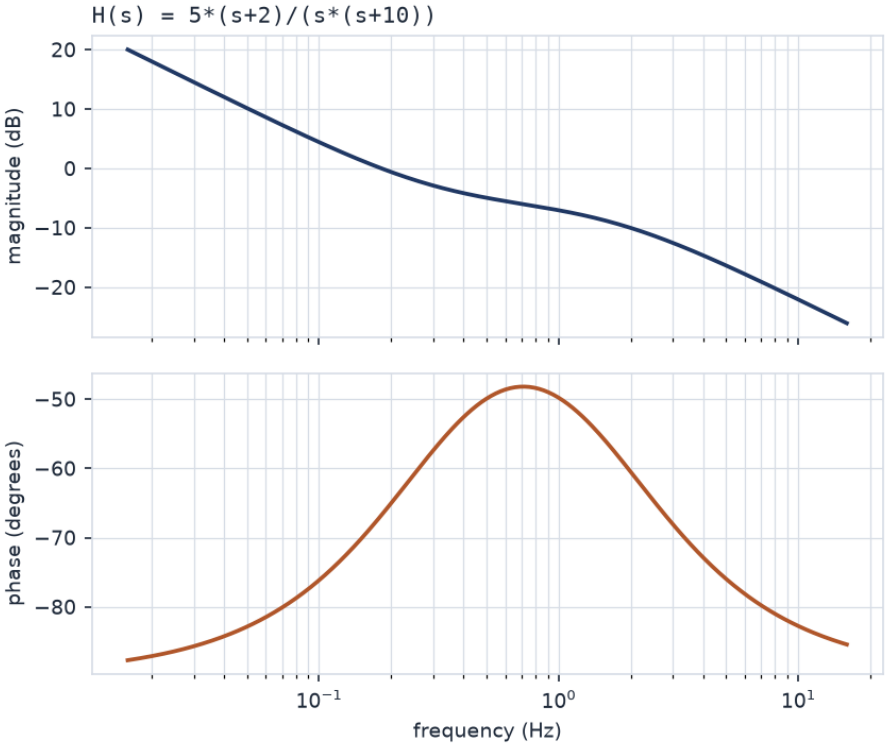
The transfer-function plot type again: write the function in terms of `s` and sweep. The sweep of 0.1 to 100 rad/s is about 0.016 to 16 Hz.

> Transfer function H(s)
> `5*(s+2)/(s*(s+10))`



AS7's Example 14.3: the Bode plot Symbulator draws

- **Start frequency (Hz)** and **End frequency (Hz)**: 0.016 and 16



AS7's Practice Problem 14.3: the Bode plot Symbulator draws

AS7's Example 14.4

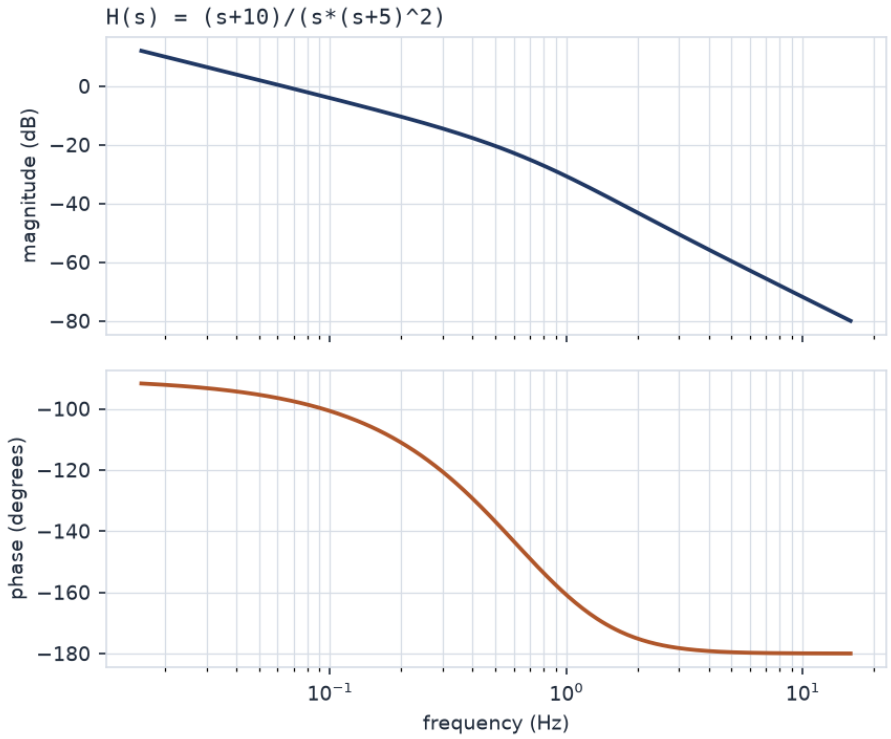
Construct the Bode plots for the given transfer function.

$$H(\omega) = \frac{j\omega + 10}{j\omega\,(j\omega + 5)^2}$$

Solved the same way: the *Bode plot of a transfer function H(s) (FD)* type, with the function written in terms of `s`.

> Transfer function H(s)
> `(s+10)/(s*(s+5)^2)`

- **Start frequency (Hz)** and **End frequency (Hz)**: 0.016 and 16



AS7's Example 14.4: the Bode plot Symbulator draws

AS7's Practice Problem 14.4

Construct the Bode plots for the given transfer function.

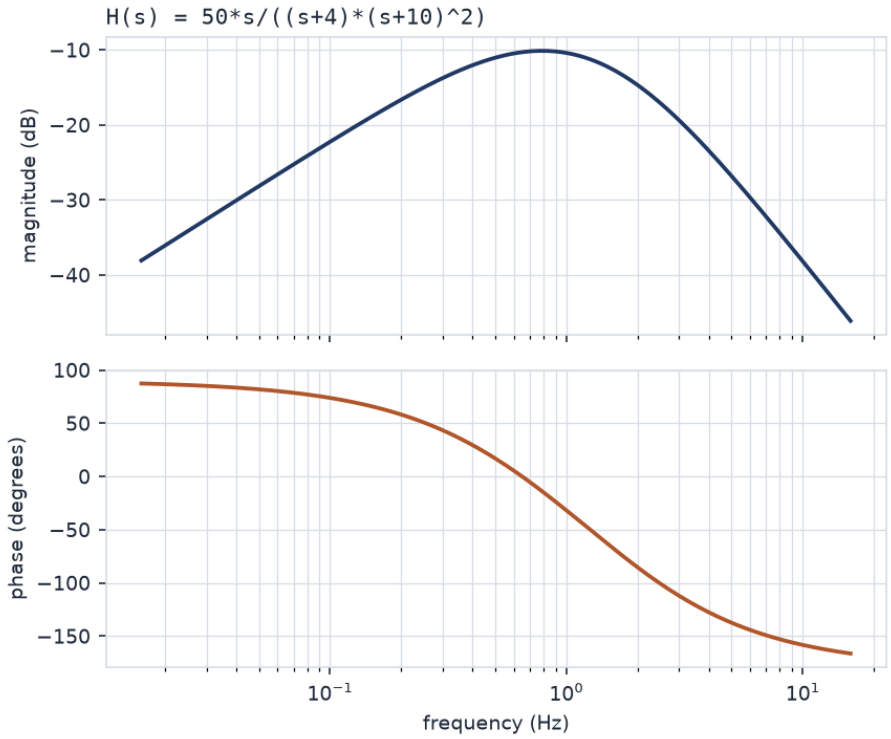
$$H(\omega) = \frac{50\,j\omega}{(j\omega + 4)(j\omega + 10)^2}$$

Symbulator always draws both, magnitude above and phase below, so there is nothing extra to ask for.

Transfer function H(s)

```
50*s/((s+4)*(s+10)^2)
```

- **Start frequency (Hz)** and **End frequency (Hz)**: 0.016 and 16



AS7's Practice Problem 14.4: the Bode plot Symbulator draws

AS7's Example 14.5

Draw the Bode plots for the given transfer function.

$$H(s) = \frac{s+1}{s^2+12s+100}$$

The same transfer-function type, and this one is already written in `s`. If a symbol other than `s` slips in, the error names it.

Transfer function H(s)

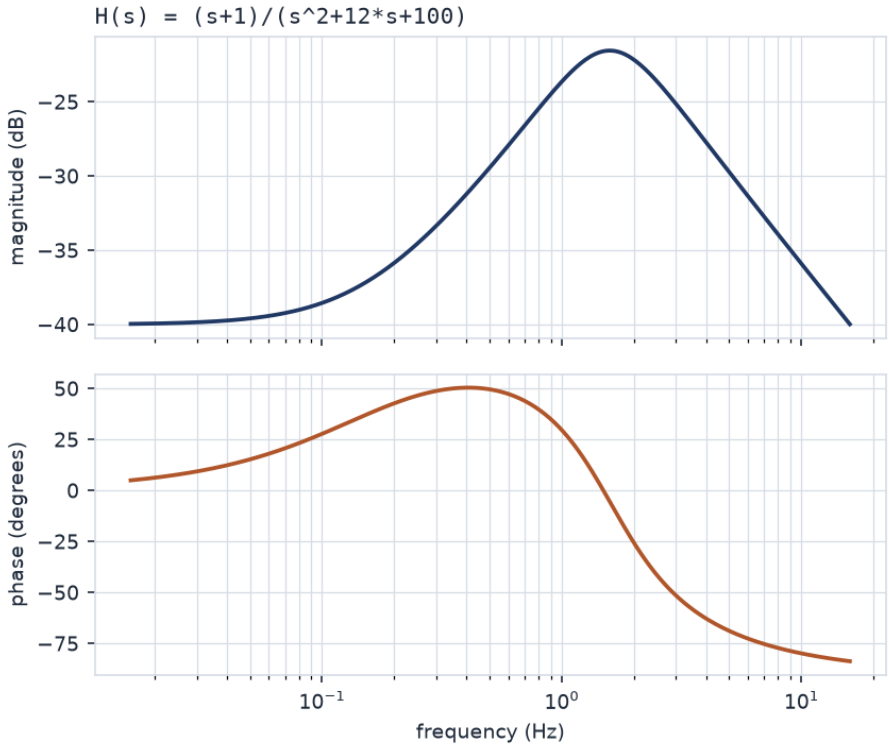
```
(s+1)/(s^2+12*s+100)
```

- **Start frequency (Hz)** and **End frequency (Hz)**: 0.016 and 16

AS7's Practice Problem 14.5

Draw the Bode plots for the given transfer function.

$$H(s) = \frac{10}{s\left(s^2+80s+400\right)}$$



AS7's Example 14.5: the Bode plot Symbulator draws

Nothing new here either: the transfer-function type, the function written in terms of `s`, both curves drawn together.

> Transfer function H(s)
> `10/(s*(s^2+80*s+400))`

- **Start frequency (Hz)** and **End frequency (Hz)**: 0.016 and 16

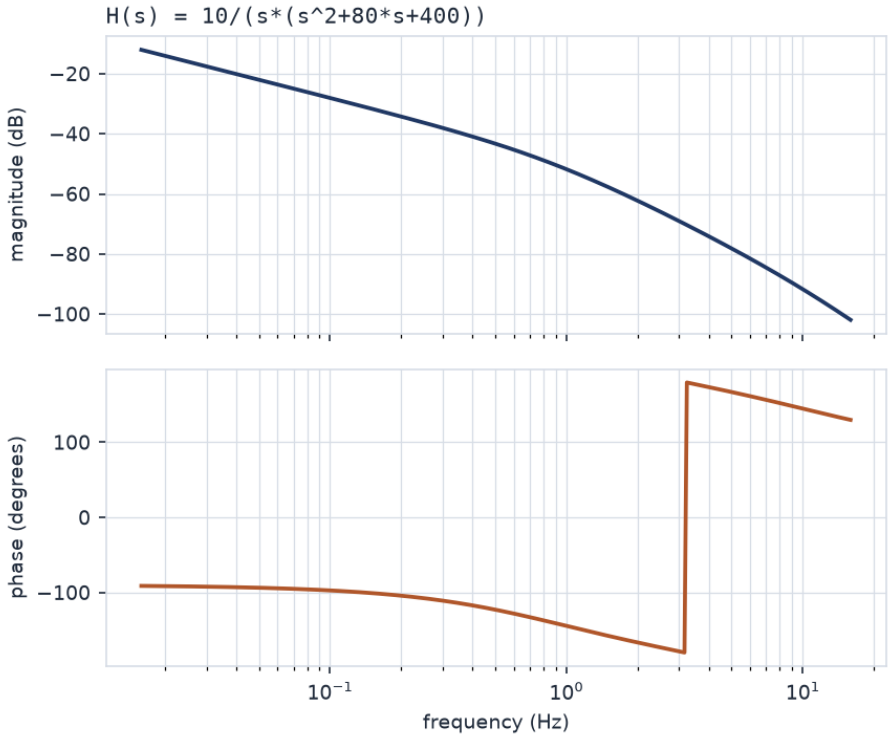
11.2 Resonance

The resonant frequency is the one at which a circuit's reactance vanishes — where the impedance the source sees becomes purely real.

There is no dedicated tool, and none is needed: solve the circuit in AC with `w` in the **ω — angular frequency** box, then put the condition to the **Solve** card.

Two things make it work, and both are easy to leave out:

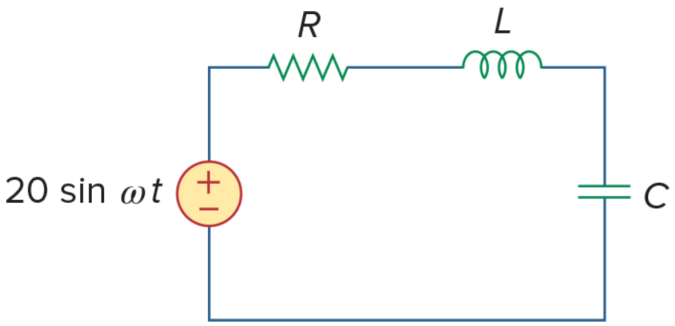
- **Tick *real solutions only*.** Solving `im(ze) = 0` over the complex field returns answers written in terms of `re(w)` and `im(w)` rather than `w` — useless. Declaring the unknown real is what makes it solvable.
- **Add `w > 0` as a condition.** The algebra gives ± the frequency, and only one of the two is a frequency.



AS7's Practice Problem 14.5: the Bode plot Symbulator draws

AS7's Example 14.7

Find the resonance frequency ω_0 and the bandwidth frequencies ω_1 and ω_2.



AS7's Example 14.7

SOLUTION

Circuit Description

```
e,1,0,20
r,1,2,2
l,2,3,1'm
c,3,0,.4'u
```

Put `w` in the ω box and solve in AC. Then, in the **Solve** card:

Equation(s) to solve in terms of the results

```
im(ze) = 0
```

Unknown(s) to solve for

```
w
```

Conditions

```
w>0
```

with **real solutions only** ticked. It gives ω_0 = **50000** rad/s.

The bandwidth frequencies are the half-power points, where the impedance magnitude is √2 times its resistance. Same circuit, different equation:

Equation(s) to solve in terms of the results

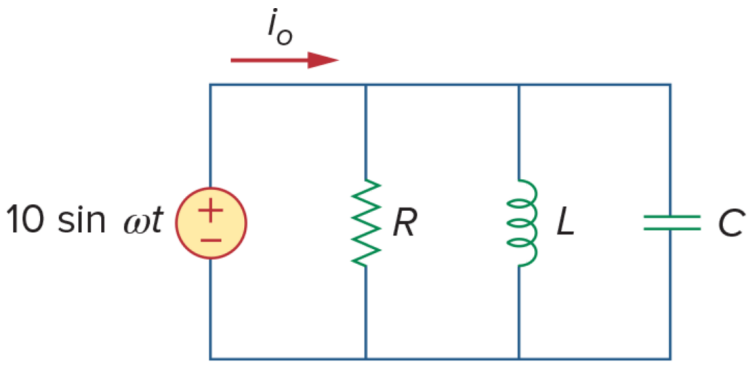
```
abs(ze) = sqrt(2)*2
```

With the condition `w<50000` that gives ω_1 = **49010** rad/s, and with `w>50000` it gives ω_2 = **51010** rad/s.

All three are correct.

AS7's Example 14.8

Find the resonance frequency and the bandwidth frequencies.



AS7's Example 14.8

SOLUTION

Circuit Description

```
e,1,0,10
r,1,0,8'k
l,1,0,.2'm
c,1,0,8'u
```

Trying the same equation as before does not go well here. The peak of resonance is so sharp — a needle — that a numerical search struggles to find its way to the solution. A different condition converges immediately: ask instead for the frequency at which the source's current has no imaginary part.

Equation(s) to solve in terms of the results

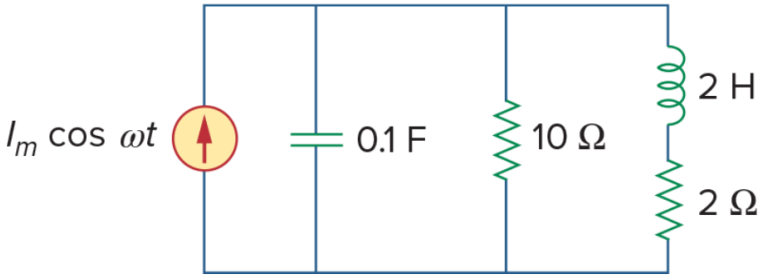
```
im(ie) = 0
```

with `w>0` and **real solutions only**, giving ω_0 = **25000** rad/s.

Symbulator 9 solves this symbolically rather than by searching, so the sharp peak costs it nothing. The alternative equation is still worth knowing: at resonance the source sees a purely real load, so its current is in phase with its voltage.

AS7's Example 14.9

Determine the resonant frequency ω_0 of the circuit.



AS7's Example 14.9

SOLUTION

Circuit Description

```
j,0,1,1
c,1,0,.1
r1,1,0,10
l,1,2,2
r2,2,0,2
```

This time, look for the frequency at which the imaginary part of the voltage at node **1** vanishes.

Equation(s) to solve in terms of the results

```
im(v1) = 0
```

with `w>0` and **real solutions only**.

ω_0 is **2** rad/s. Correct.

LESSON 12

The complex frequency domain

Learn to solve *complex-frequency domain* problems using **FD**. Move between the time domain and the s-domain with the **t2s** and **s2t** shortcuts. Learn to solve transfer function problems.

Every analysis so far has answered in the same domain you asked in. This one does not, and that is the point of it: the complex frequency domain, or s-domain, is where a differential equation becomes an algebraic one. Symbulator will do the algebra; your job is to be clear about which domain each number is in.

12.1 s-domain analysis

Symbulator has an analysis type called *FD — complex frequency domain* that solves a circuit in the domain of complex frequency, where $s = j\omega$.

The rule that matters is this one, and it catches everybody once:

> ⚠ WARNING
>
> **FD is in s. TR is in time.**
>
> The value of a source given to **FD** is read as an s-domain expression. The same value given to **TR** is read as a function of time.
> A source value of 1 in FD is not the same thing as a source value of 1 in TR. In FD it is an impulse; in TR it is a constant. Neither is wrong — they are answers to different questions — but they are not interchangeable, and nothing will warn you.

The same applies to what comes back: **FD answers in s**. Voltages and currents are functions of s, and if you want them in time you have to transform them back.

Resistances are read in the s-domain too, which is more useful than it sounds: it means you can describe a capacitor or an inductor *as an impedance*, writing its value in terms of s, as long as its initial condition is zero.

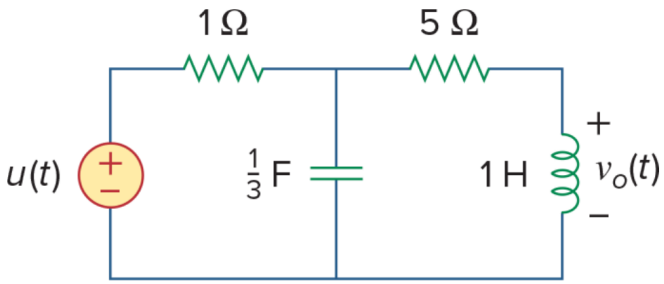
Other than the domain, FD and TR are twins. Capacitors and inductors given in farads and henries are described identically in both, and so is everything else — shorts, op-amps, transformers, all of it.

AS7's Example 16.1

Find $v_o(t)$ in the circuit, assuming zero initial conditions.

SOLUTION

Before we start: this problem does not need the s-domain at all. The source is given as



AS7's Example 16.1

a function of time and the answer is wanted as a function of time, so TR will do it in one step. Notice that the source value is written in the time domain, because that is what TR reads:

Circuit Description

```
e,1,0,u(t)
r1,1,2,1
r2,2,o,5
c,2,0,1/3,0
l,o,0,1,0
```

Run it in TR. The answer is `vo`:

$$v_o(t) = \frac{3\sqrt{2}}{2} e^{-4t} \sin(\sqrt{2}\, t)$$

Now let us do it properly, in the s-domain, which is what you would want if any of the intermediate answers interested you.

The extreme version describes everything as an impedance in s — the source as `1/s`, the capacitor as `3/s`, the inductor as `s`:

Circuit Description

```
e,1,0,1/s
r1,1,2,1
r2,2,o,5
r3,2,0,3/s
r4,o,0,s
```

That was to prove it can be done. In practice the schematic gives you farads and henries, so give Symbulator farads and henries and let it do the converting:

Circuit Description

```
e,1,0,1/s
r1,1,2,1
r2,2,o,5
c,2,0,1/3,0
l,o,0,1,0
```

Choose *FD — complex frequency domain*.

Both give the same answer, in the s-domain:

$$v_o(s) = \frac{3}{s^2 + 8s + 18}$$

Note the source: `1/s`, not `1`. The unit step is `1/s` in the s-domain, and writing `1` there would have described an impulse instead.

12.2 The t2s and s2t shortcuts

Having to transform every source by hand before typing it, and every answer by hand afterwards, would make the s-domain more trouble than it saves. Symbulator gives you both directions.

`t2s(expression)` converts an expression that is a function of time into its s-domain equivalent, and `s2t(expression)` converts back. Both work wherever you can write an expression: in a source value, in **Evaluate**, and in the **Solve** card.

So the whole problem can be done in the s-domain while you write the source in the domain the schematic gave it to you in:

Circuit Description

```
e,1,0,t2s(u(t))
r1,1,2,1
r2,2,o,5
c,2,0,1/3,0
l,o,0,1,0
```

Run that in FD, then put the answer back into the time domain from **Evaluate**:

Evaluate

```
s2t(vo)
```

The curly-bracket shorthand

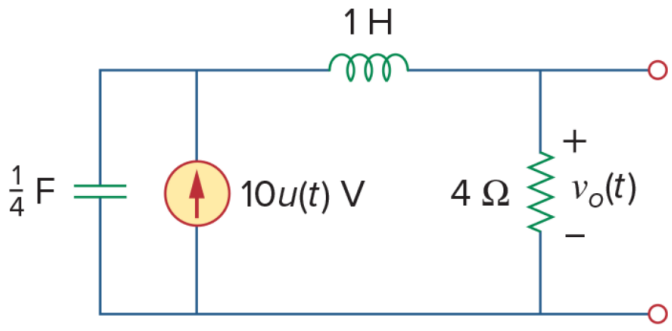
Wrap the value in curly brackets — `{u(t)}` — and FD reads it as `t2s(u(t))`. It works only inside a circuit description, so the answer still needs `s2t(...)` when you ask for it in **Evaluate**.

12.3 Instructive FD problems

AS7's Practice Problem 16.1

Determine $v_o(t)$ in the circuit.

SOLUTION



AS7's Practice Problem 16.1

The same shape as the example above, so the same approach: describe it in FD with the source converted, then transform the answer back.

Circuit Description

```
e,1,0,t2s(u(t))
r1,1,2,6
c,1,0,1/4,0
l,1,o,2,0
r2,o,0,3
```

FD, then `s2t(vo)` in **Evaluate**.

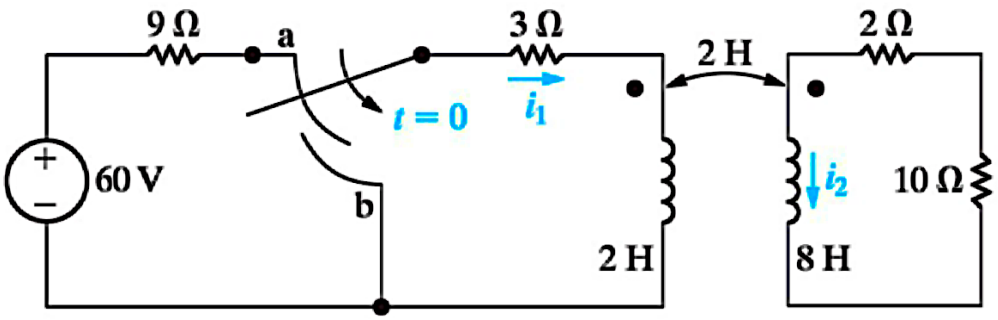
12.4 Transfer function problems

A transfer function is a ratio of two answers, so there is nothing new to learn: solve the circuit with a symbolic source, then divide.

Because both answers are functions of s, so is their ratio — which is exactly what a transfer function is meant to be.

NR11's Example 13.7

Find the transfer function H(s) = V_o/V_s.



NR11's Example 13.7

SOLUTION

Give the source a symbolic value and solve in FD:

Circuit Description

```
e,1,0,vs
r1,1,2,1'k
c,2,0,1'μ
r2,2,o,1'k
```

Solve in FD, then ask **Evaluate** for the ratio:

Evaluate

```
vo/vs
```

The answer is a function of s, as a transfer function should be.

LESSON 13

Two-ports

Find the *two-port equivalent* of a network. Learn how to include *two-ports* in your circuit using the **z**, **y**, **h**, **g**, **a** and **b** elements. Calculate gains with the **gain** mini-tool.

A two-port is a network you have stopped caring about the inside of. What is left is four numbers relating the voltage and current at one pair of terminals to the voltage and current at the other. Symbulator will find those four numbers for you, and will also take them as an element in a larger circuit.

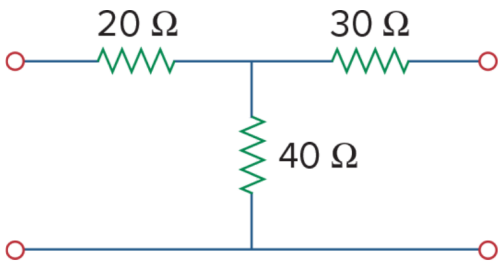
13.1 Find two-port equivalents

Set **Type of analysis** to *Find equivalent* and **Type of equivalent** to *Two-port parameters*. It takes the network and the two **top** nodes of the pair of terminals you want to reduce. The two bottom nodes are ground unless you say otherwise: a port whose lower terminal is not ground is written as a pair, `[top,bottom]`, in its node box — the same spelling a two-port *element* uses for its own ports. A side of the circuit with no path to node **0** at all, the far side of a block or a whole network with no ground, is measured against a reference of its own, and the answers say which node that is.

There are six kinds of parameters — z, y, h, g, a and b — and you choose which you want.

AS7's Example 19.1

Determine the z parameters for the circuit.



AS7's Example 19.1

SOLUTION

First describe the network, exactly as you would describe any circuit:

Circuit Description

```
r1,1,2,20
r2,2,0,40
r3,2,3,30
```

Find equivalent, then *Two-port parameters*, with **Parameters** set to *z — impedance* and **1** and **3** in the two node boxes. In DC.

There is no name to invent: the four come back as `z11`, `z12`, `z21` and `z22`.

Results gives `z11` = **60** Ω, `z12` = **40** Ω, `z21` = **40** Ω and `z22` = **70** Ω.

Correct.

13.2 Use two-ports as elements

A two-port can be an element in a bigger circuit. Its description is three fields: a name whose first letter says which kind it is, then the **top left** node and the **top right** node. Both bottom nodes are ground, always — which is why neither top node may be `0`, and the two may not be the same node; Symbulator stops with a message if they are.

Circuit Description

```
z,1,2
```

That is a z-type two-port between nodes **1** and **2**. An optional fourth term carries its four parameters — see below.

All four terminals. When a bottom node is not ground, write each port as a bracketed pair, top node then bottom node:

Circuit Description

```
z,[1,3],[2,4]
```

That is port 1 between nodes **1** and **3** and port 2 between nodes **2** and **4**; the parameter term, when there is one, follows as before, `z,[1,3],[2,4],[40,20j,30j,50]`. Either node of a pair may be `0`, so `z,[1,0],[2,0]` is the two-node form written out. A two-port conducts nothing from one port to the other, so a side of the circuit with no path of its own to ground is an island: Symbulator takes the port's lower terminal as that side's own reference and says so in a note above the results. A port whose two nodes are the same is refused.

Its answers are the current entering the two-port at each of its live nodes, named with the element and the node — `iz1`, `iz2`, `iz3`, `iz4` for the example above — shown under the element's own name in the results.

Giving it its four parameters

Three possibilities.

Give them in the description, as an optional fourth term — a list of four, in the order 11, 12, 21, 22:

Circuit Description

```
z,1,2,[40,20j,30j,50]
```

Prefer this way: the parameters travel with the circuit, into links, files and every analysis, the equivalent tools included. The entries can be numbers, SI-prefixed values or expressions.

Give them values in Define, and keep the description bare:

Define

```
z11 = 40
z12 = 20j
z21 = 30j
z22 = 50
```

Define any subset and the rest stay symbolic.

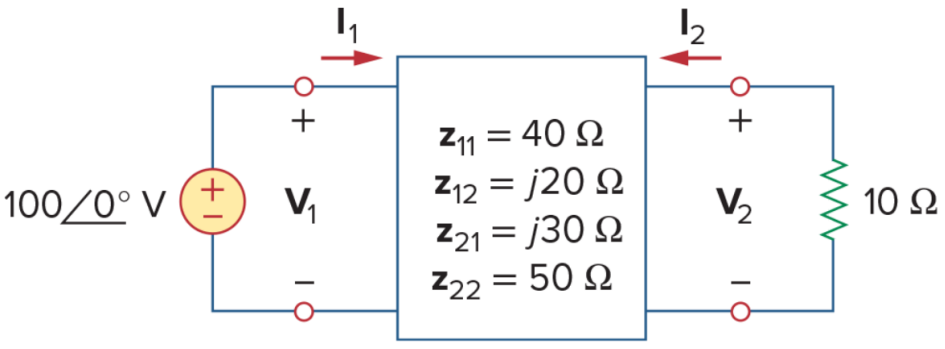
Leave them undefined, in which case they stay symbolic and appear in the answers as `z11`, `z12`, `z21` and `z22`. That is often what you want — the answer as a formula in the parameters. (Pinning them afterwards in **Expert Mode**'s **Add equation(s)** still works too, in a plain solve, and an equation or condition there overrides the description's own values.)

> ⓘ NOTE
>
> **How the variables are named**
>
> The four variables are the element's name plus `11`, `12`, `21`, `22`: a two-port called **z1** owns `z111`...`z122`. Element and node names ignore case, but a variable that names nothing in the circuit is case-sensitive: `za` and `ZA` are two different symbols.

What answers do you get? The current entering each port, named with the two-port and the **node** it sits on — for a two-port **z** on nodes **1** and **2**, that is `iz1` and `iz2`.

AS7's Example 19.2

Find I_1 and I_2 in the circuit.



AS7's Example 19.2

SOLUTION

Circuit Description

```
e,1,0,100
z,1,2,[40,20j,30j,50]
r,2,0,10
```

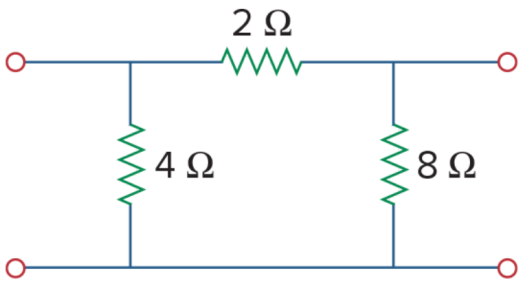
The four parameters ride in the description's fourth term. In AC.

`aa(iz1)` reads **2**∠**0**° A and `aa(iz2)` reads **1**∠**-90**° A.

13.3 Instructive two-port problems

AS7's Example 19.3

Obtain the y parameters for the network.



AS7's Example 19.3

SOLUTION

Circuit Description

```
r1,1,0,4
r2,1,2,2
r3,2,0,8
```

Find equivalent, *Two-port parameters*, *y — admittance*, nodes **1** and **2**, DC.

3/4, **-1/2**, **-1/2**, **5/8**, all in S.

Correct.

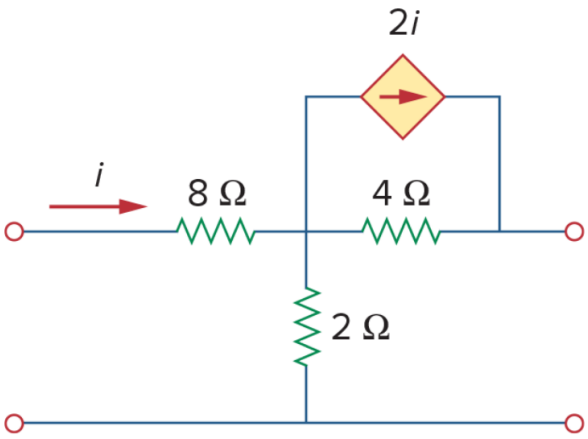
AS7's Example 19.4

Determine the y parameters for the two-port shown. Note the dependent source.

SOLUTION

Circuit Description

```
r1,1,2,8
r2,2,0,2
r3,2,3,4
```



AS7's Example 19.4

```
j,2,3,2*ir1
```

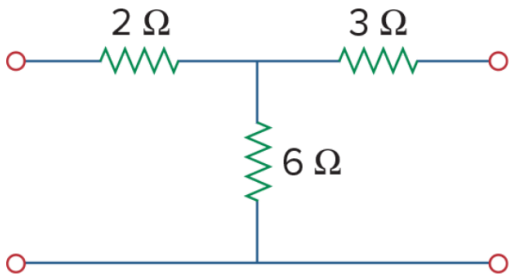
y, nodes **1** and **3**, DC. A two-port equivalent works perfectly well around a controlled source.

3/20, **-1/20**, **-1/4**, **1/4**, all in S.

Correct.

AS7's Example 19.5

Find the hybrid parameters for the two-port network.



AS7's Example 19.5

SOLUTION

Circuit Description

```
r1,1,2,2
r2,2,0,6
r3,2,3,3
```

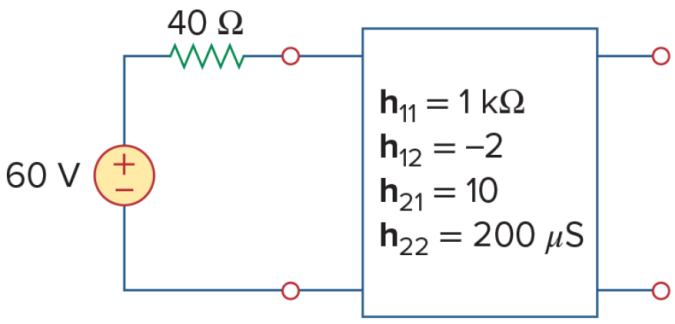
h – *hybrid*, nodes **1** and **3**, DC.

4, **2/3**, **-2/3**, **1/9**.

Correct.

AS7's Example 19.6

Determine the Thévenin equivalent at the output port of the circuit, whose h parameters are 1000 Ω, −2, 10 and 2×10⁻⁴ S.



AS7's Example 19.6

SOLUTION

Circuit Description

```
e,1,0,60
r,1,2,40
h,2,3,[1000,-2,10,2e-4]
```

Find equivalent, *Thévenin / Norton*, nodes **3** and **0**, DC. The parameters ride in the description, so the answers come back as numbers.

Leave the fourth term off — `h,2,3` alone — and the same run answers with formulas instead:

Thévenin voltage

$$v_{th} = \frac{-60\,h_{21}}{h_{11}h_{22} - h_{12}h_{21} + 40\,h_{22}}$$

which is useful in its own right: put `vth` in **Evaluate** with the four parameters in its **Conditions** box, and the same number falls out.

`vth` comes back **-29.69** V and `req` **51.46** Ω.

Correct.

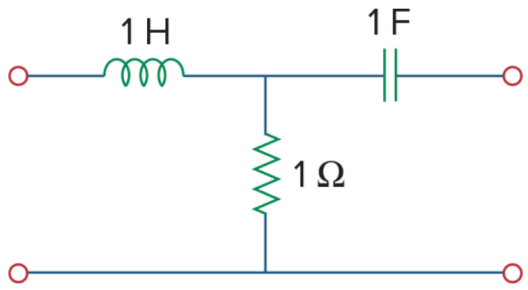
AS7's Example 19.7

Find the g parameters as functions of s for the circuit.

SOLUTION

Circuit Description

```
l,1,2,1
r,2,0,1
c,2,3,1
```



AS7's Example 19.7

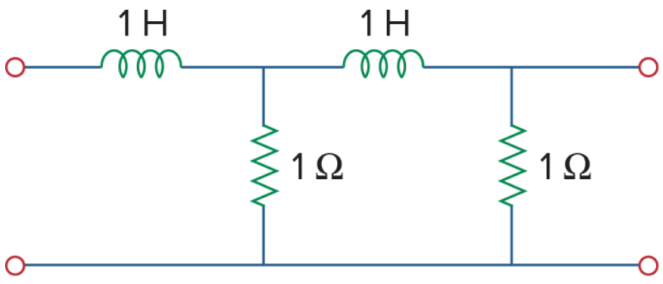
g — inverse hybrid, nodes **1** and **3**, and this time in **FD**, since the question asks for functions of s.

The four come back as

$$g_{11} = \frac{1}{s+1}, \quad g_{12} = -\frac{1}{s+1},$$
$$g_{21} = \frac{1}{s+1}, \quad g_{22} = \frac{s^2+s+1}{s(s+1)}.$$

AS7's Practice Problem 19.7

For the ladder network, determine the g parameters in the s domain.



AS7's Practice Problem 19.7

SOLUTION

Circuit Description

```
l1,1,2,1
r1,2,0,1
l2,2,3,1
r2,3,0,1
```

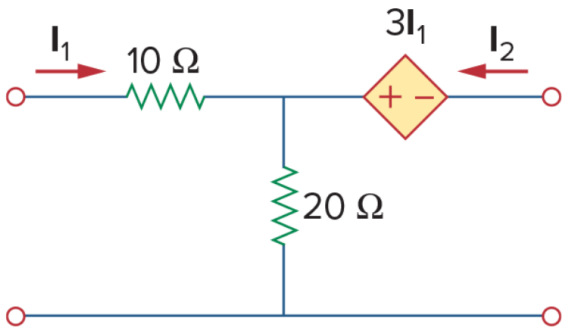
g, nodes **1** and **3**, FD. The four come back as

$$g_{11} = \frac{s+2}{s^2+3s+1}, \quad g_{12} = -\frac{1}{s^2+3s+1},$$
$$g_{21} = \frac{1}{s^2+3s+1}, \quad g_{22} = \frac{s(s+2)}{s^2+3s+1}.$$

Correct.

AS7's Example 19.8

Find the transmission parameters for the two-port network.



AS7's Example 19.8

SOLUTION

Circuit Description

```
r1,1,2,10
r2,2,0,20
e,2,3,3*ir1
```

a — transmission, nodes **1** and **3**, DC.

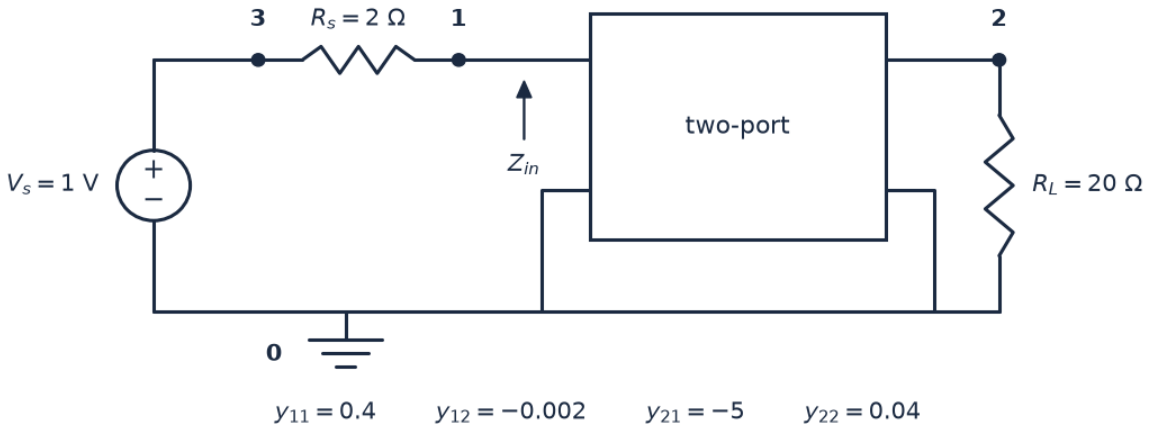
1.765, **15.29**, **0.0588**, **1.176**.

Correct.

13.4 The gain mini-tool

Network problems ask for gains: voltage gain, current gain, power gain, and the impedance seen at the input. Symbulator has a mini-tool for that — the third in the **Mini-Tools** card Lesson 7 (page 206) introduced: choose *gain*.

It wants four values — the voltage and current at the input, then the voltage and current at the output — and answers with all four figures at once.



Gain Example 1

Gain Example 1

For the circuit, find G_v, G_i, G_p and Z_{in}. The two-port has y parameters y_{11} = 0.4 S, y_{12} = −0.002 S, y_{21} = −5 S and y_{22} = 0.04 S.

SOLUTION

The source is 1 V because its value does not matter to a gain — it cancels. A symbolic `vs` would do just as well.

Circuit Description

```
es,3,0,1
rs,3,1,2
rl,2,0,20
y,1,2,[0.4,-0.002,-5,0.04]
```

Solve in DC, then open **Mini-Tools**, choose *gain*, and give it the four:

v1

```
v1
```

i1

```
iy1
```

v2

```
v2
```

i2

```
iy2
```

The voltage gain Av is **55.56**, the current gain Ai **-9.615**, the power gain A_p **534.2**, and the input impedance Z_i **3.462** Ω.

These are correct.

Gain Example 2

Find G_v, G_i, G_p and Z_{in} for a two-port with z parameters z_{11} = 4 Ω, z_{12} = 1.5 Ω, z_{21} = 10 Ω and z_{22} = 3 Ω, driven by a source V_s with 5 Ω in series and loaded with 2 Ω.

There is no picture of this one — the description below is the whole circuit.

SOLUTION

The same shape as before, with z parameters instead of y.

Circuit Description

```
es,3,0,1
rs,3,1,5
rl,2,0,2
z,1,2,[4,1.5,10,3]
```

Solve in DC, then open **Mini-Tools**, choose *gain*, and give it the four:

v1

```
v1
```

i1

```
iz1
```

v2

```
v2
```

i2

```
iz2
```

The voltage gain Av is **4**, the current gain Ai **-2**, the power gain A_p **8**, and the input impedance Z_i **1** Ω.

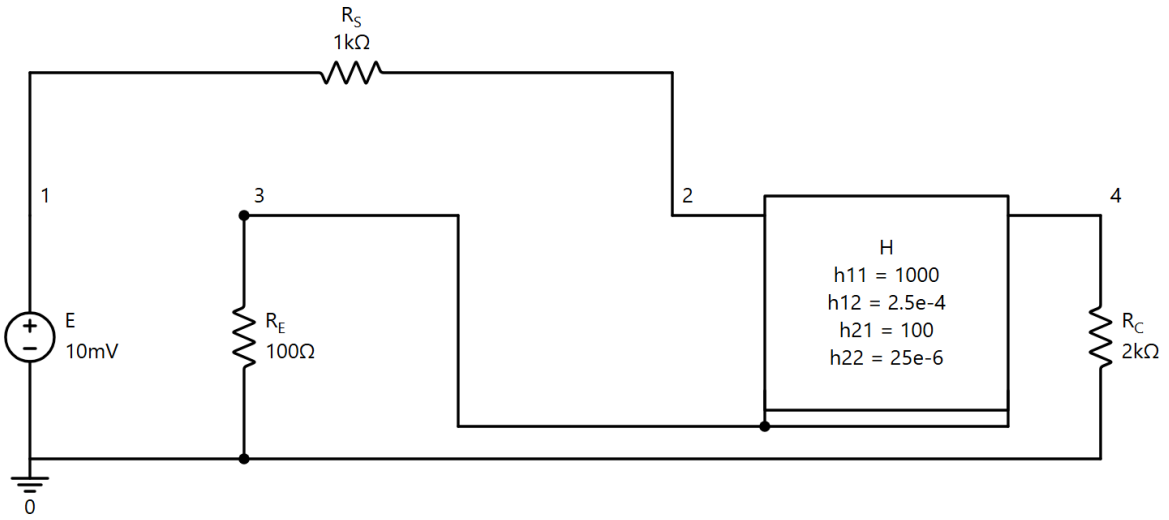
These are correct.

An h-parameter model with a resistor under its common terminal

A transistor stage described by its hybrid parameters, h_{ie} = 1 kΩ, h_{re} = 2.5 × 10^{-4}, h_{fe} = 100 and h_{oe} = 25 μS, is driven by a 10 mV source through a 1 kΩ resistor and loaded by 2 kΩ. Its common terminal is not grounded: a 100 Ω resistor sits between it and ground. Find the output voltage, and compare it with the same stage with the common terminal grounded.

SOLUTION

The two-node form cannot describe this: it grounds both bottom terminals, and the whole point is the resistor under them. With the four-terminal form each port is a pair, and both pairs share node **3**, the common terminal.



An h-parameter model with a resistor under its common terminal, drawn by Symbulator

Circuit Description

```
e,1,0,0.01
rs,1,2,1000
h,[2,3],[4,3],[1000,2.5e-4,100,25e-6]
re,3,0,100
rc,4,0,2000
```

DC. **Results** gives v4 = **-0.1645** V, a voltage gain of **-16.45**, and under **h** the three currents entering the block: ih2 = **0.8657** μA at the input, ih4 = **82.25** μA at the output, and ih3 = **-83.12** μA at the common terminal — the sum of the other two, leaving the block, which is the current through the 100 Ω resistor.

Ground the common terminal instead, `h,2,4,[1000,2.5e-4,100,25e-6]` with no `re`, and v4 is **-0.9756** V, a gain of **-97.56**. The resistor under the common terminal costs a factor of six in gain, which is what emitter degeneration does.

A z-block with its second port lifted off ground

A z-parameter block, $\boldsymbol{z}$ = [100, 10, 20, 50], is driven at its first port by a 10 V source. Its second port feeds a 200 Ω load, but the port's lower terminal is not ground: it reaches ground through a 20 Ω resistor. Find the currents into the block and the voltage at its lower terminal.

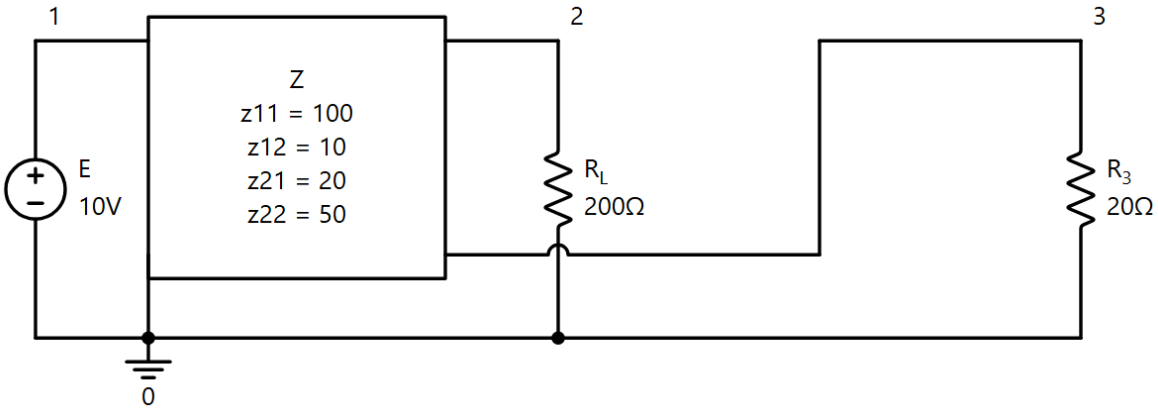
SOLUTION

Port 1 is the pair `[1,0]`, grounded as before; port 2 is the pair `[2,3]`.

Circuit Description

```
e,1,0,10
z,[1,0],[2,3],[100,10,20,50]
rl,2,0,200
r3,3,0,20
```

DC. **Results** gives, under **z**,



A z-block with its second port lifted off ground, drawn by Symbulator

current into port at node 1

$$i_{z1} = \frac{27}{268}$$

current into port at node 2

$$i_{z2} = -\frac{1}{134}$$

current into port at node 3

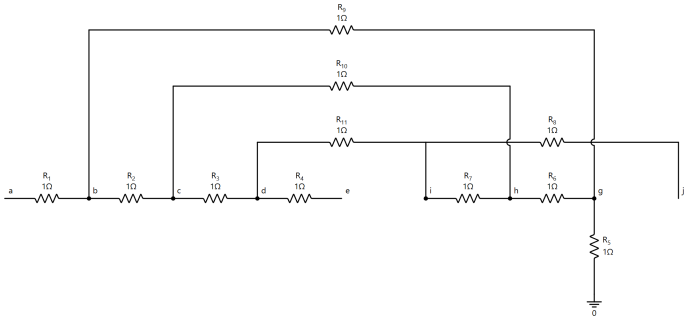
$$i_{z3} = \frac{1}{134}$$

and `v3` is **-10/67** V. The second port's current leaves the block at node **2**, goes down through the load to ground, and comes back up through the 20 Ω resistor into node **3** — which is why `iz2` and `iz3` are equal and opposite, and why node **3** sits below ground.

Three problems from Alexander and Sadiku's Chapter 19 in which **no port is grounded** — the case the pair spelling exists for. In each, grounding the bottoms instead would solve a different circuit and hand back plausible, wrong numbers.

AS7's Problem 19.2

Determine the equivalent impedance parameters of the ladder: four 1 Ω resistors in the upper rail, four in the lower, three 1 Ω rungs.



AS7's Problem 19.2, drawn by Symbulator with **f** as the reference the tool takes

SOLUTION

Neither rail is ground, so each port is a pair: port 1 is `[a,f]`, port 2 is `[e,j]`.

Circuit Description

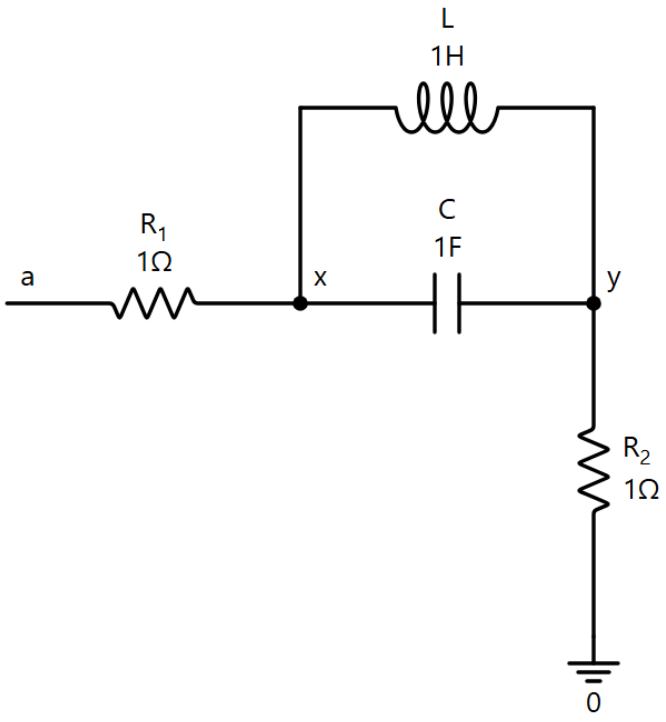
```
r1,a,b,1
r2,b,c,1
r3,c,d,1
r4,d,e,1
r5,f,g,1
r6,g,h,1
r7,h,i,1
r8,i,j,1
r9,b,g,1
r10,c,h,1
r11,d,i,1
```

z — impedance, with `[a,f]` and `[e,j]` in the two node boxes. DC. **Results** gives `z11` = `z22` = **2.733** Ω and `z12` = `z21` = **0.06667** Ω — exactly 41/15 and 1/15. Symmetric, as the drawing is, and small in transfer: three shunt ohms bleed nearly everything to the far rail.

The network has no node **0** anywhere. The tool takes each port's lower terminal as the reference for its own measurement, which is what the definition of the parameters does. Type **a** and **e** instead, with the lower rail grounded, and the answer is 11/5 and 3/5 — the parameters of a different circuit, the lower rail's resistors shorted out.

AS7's Problem 19.19

Determine the y parameters of the two-port in terms of s: 1 Ω in each rail, then a 1 F capacitor and a 1 H inductor side by side between the rails.



AS7's Problem 19.19, drawn by Symbulator with **f** as the reference the tool takes

SOLUTION

Port 2 is the pair of junctions, `[x,y]`; port 1 the pair of outer ends, `[a,f]`. Neither is grounded.

Circuit Description

```
r1,a,x,1
r2,f,y,1
c,x,y,1
l,x,y,1
```

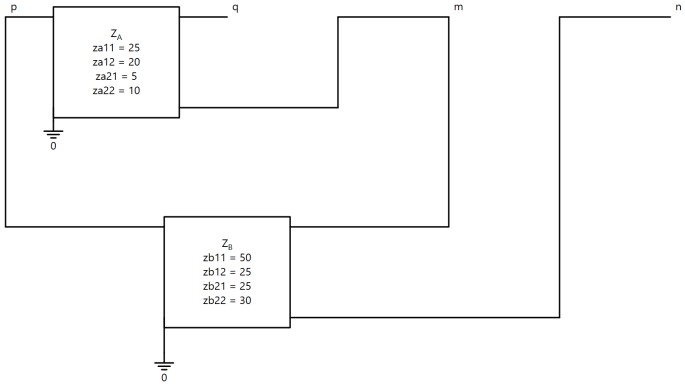
y — admittance, FD. The four come back as

$$y_{11} = \frac{1}{2}, \qquad y_{12} = -\frac{1}{2},$$
$$y_{21} = -\frac{1}{2}, \quad y_{22} = s + \frac{1}{2} + \frac{1}{s},$$

all in siemens. With port 2 shorted the capacitor and inductor are shorted with it, so port 1 sees the two series ohms and returns all of that current through the short; with port 1 shorted, port 2 sees the capacitor and inductor in parallel, `s + 1/s`, plus the two ohms in series through the short.

AS7's Problem 19.70

Two two-ports given by their z parameters, $\boldsymbol{z_A}$ = [25, 20, 5, 10] Ω and $\boldsymbol{z_B}$ = [50, 25, 25, 30] Ω, are joined in a parallel-series connection: their input ports in parallel, their output ports in series. Find the g parameters of the whole.



AS7's Problem 19.70, drawn by Symbulator

SOLUTION

Type each block as a four-terminal `z` element with its parameters. The inputs share the pair `[p,0]`; block A's lower output terminal, **m**, is block B's upper one; the output port is **q** to **n**.

Circuit Description

```
za,[p,0],[q,m],[25,20,5,10]
zb,[p,0],[m,n],[50,25,25,30]
```

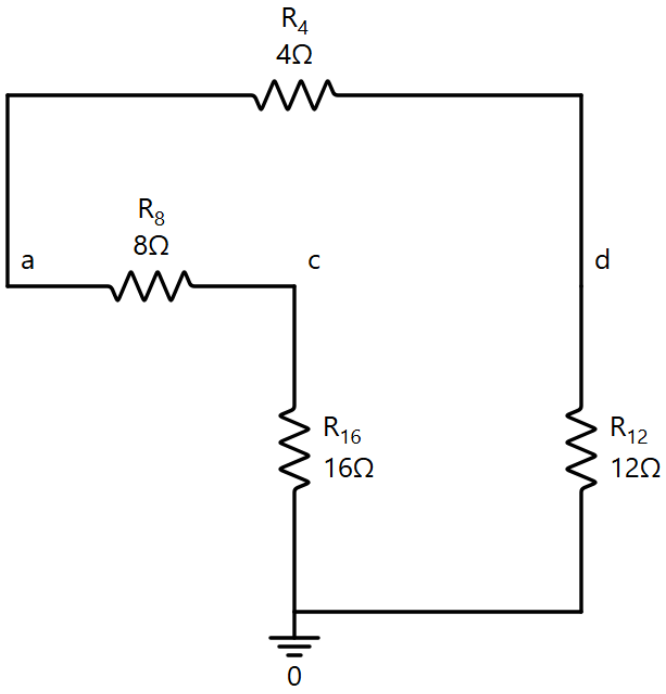
g — inverse hybrid, with **p** and `[q,n]` in the node boxes. DC. **Results** gives `g11` = **0.06** S, `g12` = **-1.3**, `g21` = **0.7** and `g22` = **23.5** Ω.

That is the sum of the two blocks' own g matrices — each block's g follows from its z, and a parallel-series connection adds them — so the tool has done the textbook's rule for you.

The output side has no path to node **0**: a block conducts nothing across its ports, so **q**, **m** and **n** are an island. The tool takes **n**, the output port's lower terminal, as that side's reference. Solve the same description with a 1 V source at **p** and the answers carry a note saying which node was taken as 0, and `v_m` reads 0.

AS7's Problem 19.17

Determine the z and y parameters of the lattice: 8 Ω across the top, 12 Ω across the bottom, 4 Ω and 16 Ω on the two diagonals.



AS7's Problem 19.17, drawn by Symbulator with **f** as the reference the tool takes

SOLUTION

Both ports float: `[a,f]` on the left, `[c,d]` on the right.

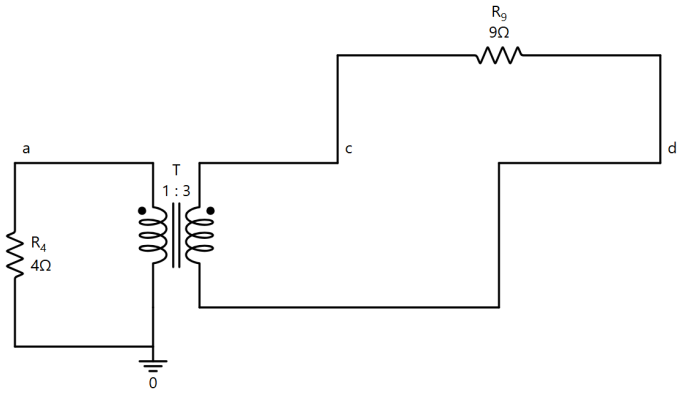
> Circuit Description
> ```
> r8,a,c,8
> r12,f,d,12
> r4,a,d,4
> r16,f,c,16
> ```

z — impedance, DC. **Results** gives `z11` = **9.6** Ω, `z12` = `z21` = **-0.8** Ω and `z22` = **8.4** Ω. The first is the two paths from **a** to **f** with port 2 open, 24 Ω and 16 Ω, in parallel; the last the two from **c** to **d**, 12 Ω and 28 Ω.

Run it again with *y — admittance* and the four are **0.105** S, **0.01** S, **0.01** S and **0.12** S — the inverse of the z matrix, as they must be.

AS7's Problem 19.63

Determine the z parameters of an ideal 1:3 transformer with 4 Ω across its primary and 9 Ω across its secondary.



AS7's Problem 19.63, drawn by Symbulator with **b** as the reference the tool takes

SOLUTION

The ports are the two windings' own terminal pairs, so the transformer is written in its four-terminal form and neither port is grounded.

> Circuit Description
>
> ```
> r4,a,b,4
> t,[a,b],[c,d],[1,3]
> r9,c,d,9
> ```

z — impedance, with `[a,b]` and `[c,d]` in the node boxes. DC. **Results** gives `z11` = **0.8** Ω, `z12` = `z21` = **2.4** Ω and `z22` = **7.2** Ω: 4 Ω in parallel with the 9 Ω referred to the primary, 9/3² = 1 Ω; 9 Ω in parallel with the 4 Ω referred to the secondary, 36 Ω; and the transfer terms three times and a third of those.

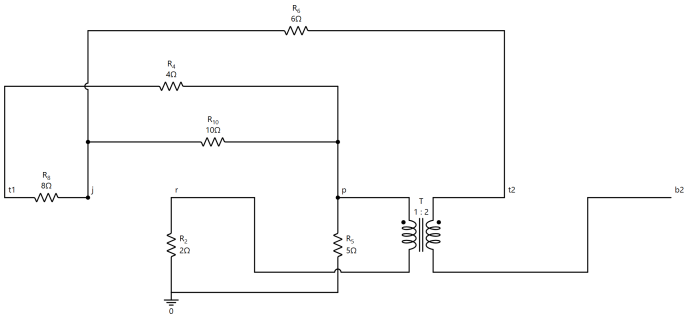
Each side is an island — a transformer conducts nothing across — so the tool takes **b** and **d** as the two references, one per side.

AS7's Problem 19.71

Determine the z parameters of the network of Figure 19.118: 8 Ω and 6 Ω along the top with 10 Ω hanging from their junction, 4 Ω from the upper-left terminal to the foot of that 10 Ω, a 5 Ω and a 2 Ω below, and a 1:2 transformer whose secondary is the right-hand port.

SOLUTION

As read from the figure: the upper-left terminal **t1** feeds the 8 Ω and, through the 4 Ω, the node **p** where the 10 Ω, the 5 Ω and the primary's top meet; the lower-left terminal is **q**, the foot of the 5 Ω, which reaches the primary's bottom **r** through the 2 Ω; the secondary runs from the upper-right terminal **t2**, beyond the 6 Ω, to the lower-right terminal **b2**,



AS7's Problem 19.71 as read from the book's drawing, drawn by Symbulator with **q** as the reference the tool takes

both dots up.

Circuit Description

```
r8,t1,j,8
r10,j,p,10
r4,t1,p,4
r5,p,q,5
r2,q,r,2
r6,j,t2,6
t,[p,r],[t2,b2],[1,2]
```

z — impedance, with `[t1,q]` and `[t2,b2]` in the node boxes. DC. **Results** gives `z11` = **8.273** Ω (exactly 91/11), `z12` = `z21` = **10** Ω and `z22` = **28** Ω.

Working with input files

TECH NOTE A

Load the tutorial's built-in examples, save your own circuits as entries, and download an input file to your device. Plus everything an entry remembers, so you can pick up where you left off.

Everything you type into Symbulator — circuit, analysis, settings, the **Evaluate** and **Solve** cards — can be saved and reused. Two words matter here.

- An **entry** is one named set of inputs: a circuit, the analysis to run on it, and the settings.
- An **input file** is a named collection of entries, such as *Circuits for Lesson 1 of the Tutorial*.

Input files are plain text with the extension `.cir`. They are optional. The buttons to load, download and create them are in the **Input File** card at the top of the app.

Loading input files

Symbulator comes with an input file for each lesson of this tutorial, holding an entry for every simulation in it. A problem that takes more than one simulation has an entry for each, so a lesson file may hold more entries than the lesson has problems. With them you can follow the whole tutorial without typing a circuit.

Built-in examples lists them; click a title to load it. These files are **read-only**: change a loaded one as much as you like, but you cannot write back to it.

To open your own input file, click **Upload** and choose the `.cir` file. The dropdown beneath then reads *Entries in* and the file's name. Choosing an entry fills in every box on the page. Solve it as if you had just typed it.

Saving an entry while a read-only file is loaded, or clicking **Create new**, starts a blank input file of your own.

Saving your work

Two links under the circuit box appear when they apply:

- **Save inputs to new entry** adds what is on screen as a new entry and asks for a name.
- **Update inputs in this entry** writes your changes back into the loaded entry. It appears once you have changed something.

Beside the dropdown, **Rename** and **Delete** act on the entry itself. Neither is offered for

the built-in examples, since they are read-only.

> ⚠ WARNING
>
> **Nothing reaches your disk until you press Download**
>
> Saving an entry writes it into the file open **in your browser**. Only **Download** produces a `.cir` on your device.
> Your work is mirrored into the browser's storage, so closing the tab does not lose it, and you are warned before anything replaces a file with entries you have not downloaded. Still, download when you are done.

What an entry remembers

An entry keeps everything you typed, not just the circuit: the analysis and what it needs (the frequency, the nodes of an equivalent, the kind of two-port), the **Expert Mode** equations, unknowns and conditions, everything in **Settings**, **Evaluate** and its **Conditions**, the **Solve** card, the **Plotting Tools** inputs, and a note of your own shown when the entry is loaded. Load one a year later and you are back where you left off.

The **About input file (.cir) format** section inside the **Input File** card explains what input files carry, in detail.

Limiting the results to save time

TECH NOTE B

A transient run returns every voltage, current and power in the circuit, and each one costs an inverse Laplace transform. Ask for only the answers you want and the wait gets shorter.

Unless you say otherwise, Symbulator gives you the whole set of answers: the voltage at every node, and the voltage drop, current and power in every element. That is usually what you want, and for a direct current or an alternating current analysis it costs almost nothing.

Transient analysis is the exception. Each TR answer is found by inverting a Laplace transform, and that is the slow step. A circuit with a dozen elements has several dozen answers, and you are very often after one of them.

Asking for a few answers

Choose **TR** in the analysis menu, and a question appears in **Settings**:

> ***Do you want to limit the results to save time?***

Tick it, and a field opens. List the answers you want, separated by commas:

> What results are you after? List the variables here
> v2

Every answer name Symbulator knows is allowed here — `v2` for the voltage at node **2**, `ir1` for the current through **r1**, `pr3` for the power in **r3**.

Ask for one answer instead of a dozen and you skip a dozen inverse transforms. On a small circuit the difference is not worth the tick; on a large one, or a symbolic one, it is the easiest speed-up there is.

> TIP
> **It changes what is computed, not what is correct**
> Limiting the results does not approximate anything. The answers you ask for are the same answers you would have got without the tick — there are simply fewer of them.

When not to bother

If you are exploring, leave it alone. The whole answer set is the point of Symbulator: you run a circuit and read what happened everywhere in it, which is exactly what you cannot do on paper. Reach for this when a particular run is slow enough to interrupt you, and not before.

Underscores, and subscripts in your answers

TECH NOTE C

An underscore does two jobs. In an answer's name it is optional, so `ir1` and `i_r1` mean the same current. In a symbolic value it is not: it puts what follows into a subscript, and makes an answer read the way a textbook prints it.

The underscore does two different jobs in Symbulator, and it is worth knowing which is which. In the name of an **answer** it is optional and changes nothing. In the **value** you give an element, it changes how the answer is typeset.

In an answer's name, it is optional

You may write the name of any of Symbulator's answers in two ways: with or without an underscore. `ir1` and `i_r1` are the same current, `v2` and `v_2` the same voltage; capitals make no difference either.

Version 9 added the longer spelling so that a name written out by a machine reads back without ambiguity; the shorter one is what the calculators used, and it is what this tutorial uses throughout. Neither is more correct than the other, and you can mix them freely — wherever you type an answer's name, in **Evaluate**, in **Solve**, in a condition, or (in the case of dependent sources) as part of an element's value, Symbulator recognises both formats.

In a symbolic value, it makes a subscript

Textbooks set their symbols with subscripts: v_{cc}, r_b, i_{rb}. Symbolic answers can be made to read the same way, and the trick is very simple: **put an underscore in your symbolic variable names.** The text after the underscore is set as a subscript.

That is all there is to it. `v_cc` prints as v_{cc}, `r_e1` as r_{e1}. It costs nothing, it changes no answer, and it makes a symbolic result that much easier to compare with the page you are working from.

One difference from the section above, and it is the only trap here. In an answer's name the underscore is optional; in a symbolic variable it is *part of the name*. Rename `vcc` to `v_cc` everywhere and nothing changes but the typesetting — but write `v_cc` in the circuit and `vcc` in a condition and you have named two variables, not one, and Symbulator will treat them as two. Pick a spelling for each symbol and stay with it.

TR5's Example 4.5 (Symbolic)

The same circuit twice: once with plain names, once with subscripted ones, so the difference is only in how the answer reads.

Given this circuit description, with plain names:

Circuit Description

```
e1,1,0,vcc
rb,1,b,rb
e2,e,b,vγ
re1,e,0,re1
rc,1,c,rc
j,c,e,β*irb
```

the current through **rb** comes back as

current through rb

$$i_{rb} = \frac{vcc + v\gamma}{r_{e1}\,\beta + r_{e1} + rb}$$

One subscript arrives without being asked for: **re1** prints as $\boldsymbol{r_{e1}}$, because trailing digits are set as a subscript anyway. The underscore is what gets you the rest.

Run the same circuit with underscores in the symbolic values instead:

Circuit Description

```
e1,1,0,v_cc
rb,1,b,r_b
e2,e,b,v_γ
re1,e,0,r_e1
rc,1,c,r_c
j,c,e,β*irb
```

and the answer is typeset the way the book prints it:

current through rb

$$i_{rb} = \frac{v_{cc} + v_\gamma}{r_b + r_{e1}\,\beta + r_{e1}}$$

The two answers are the same expression. Only their appearance differs.

When a load is not the only thing attached

TECH NOTE D

The load formulas assume a load and nothing else. When something more is connected to the equivalent, Symbulator will write the equivalent circuit into the input fields for you, ready to be added to.

Lesson 4's load answers — `irl`, `vrl`, `prl` and `pmax` — are formulas, and they hold only when a load is the one thing connected to the equivalent circuit. Connect anything else to those terminals and none of them applies any more, and the problem has to be run as a circuit again.

Symbulator will write that circuit for you, so you are adding to it rather than typing it out.

The button

With the load question ticked, a button appears at the foot of **Results**, under the equivalent's answers: **Load circuit equivalent?** It is live once the equivalent has been found, and goes dead again if you change any input until you run again.

Press it and it warns you first: the equivalent circuit will overwrite the **Circuit Description**, **Define** and **Expert Mode** fields and switch the analysis to *Solve circuit*, so a circuit you have not saved yet can be saved before it goes. Proceed, and the description becomes

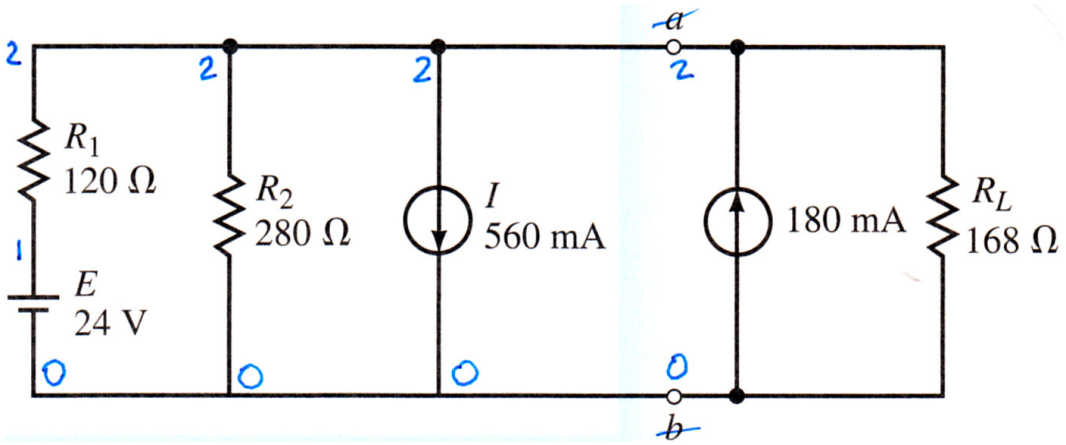
> Circuit Description
>
> ```
> jN,0,n,iNo
> rE,n,0,rEq
> rL,n,0,load
> ```

with **Define** holding the values of `iNo` and `rEq` that **Results** found, exact rather than rounded, and a value for `load` if you had given it one. It is the Norton equivalent connected, between nodes **n** and **0**, to a load called **rL** with the symbolic value **load**, in ohms. You can use it as a starting point.

An example

RM3's Example 9-8

Find the Norton equivalent of the circuit left of a-b; then find the current through R_L.



RM3's Example 9-8

SOLUTION

Let's first find the circuit equivalent:

Circuit Description

```
e,1,0,24
r1,1,2,120
r2,2,0,280
j,2,0,560'm
```

Find equivalent, *Thévenin / Norton*, nodes **2** and **0**, in DC, with the load question ticked. **Results** gives:

Norton current

$$i_{no} = -0.36 \text{ A}$$

equivalent resistance

$$R_{eq} = 84\ \Omega$$

The Norton current is reported in the direction it actually flows, from the first node to the second. Carry the sign through and the load current below comes out the same.

Now to the second part of the question. In order to find the current through R_L, we cannot use the load expressions, because now the load is not the only thing connected to the terminals of the equivalent: there is also a current source. We have to run a new simulation.

The fastest way is the button above. Proceed past its warning and the description is replaced by the equivalent, with `iNo` and `rEq` already in **Define**:

Circuit Description

```
jN,0,n,iNo
rE,n,0,rEq
rL,n,0,load
```

We change the value of the load to 168 Ω, and add the 180 mA source flowing from node **0** to node **n**. Then we run a DC simulation and ask for the current in the load:

Circuit Description

```
jN,0,n,iNo
rE,n,0,rEq
rL,n,0,168
j,0,n,180'm
```

The analysis is already *Solve circuit*; run it in DC, with **SI prefixes** ticked in **Settings**. The current through **rl**, `irl`, reads **-60** mA.

Correct: there is a current of 60 mA flowing through R_L from 0 to n.

Using the equivalent circuit description is meant to save you time. If you find it confusing to use, just don't use it.

The SI prefixes Symbulator accepts

TECH NOTE E

Every prefix Symbulator takes, from peta down to atto, with its shorthand and its factor — and the two places where case matters.

An SI prefix in an element's value, preceded by an apostrophe, multiplies that value by the corresponding factor, exactly. `8'k` is 8000, and it is 8000 exactly, not 8000 to some number of figures.

Eleven prefixes are accepted:

Prefix	Shorthand	Factor
peta	`'P`	10^{15}
tera	`'T`	10^{12}
giga	`'G`	10^{9}
mega	`'M`	10^{6}
kilo	`'k` or `'K`	10^{3}
milli	`'m`	10^{-3}
micro	`'u` or `'μ`	10^{-6}
nano	`'n`	10^{-9}
pico	`'p`	10^{-12}
femto	`'f`	10^{-15}
atto	`'a`	10^{-18}

They work in the value of every element, not only resistors.

> ⚠ WARNING
>
> **Case matters**
>
> Kilo is the one prefix you may write either way, `'k` or `'K`. Every other prefix is case-sensitive, and two of them are a factor of 10^9 apart: `'m` is milli and `'M` is mega. Micro is the other one with a choice: `'u`, or either of the two mu characters a keyboard may give you, which look alike and mean the same thing here. All of them are lower case.

Exact, and why that matters

A prefixed value is exact, and so is a plain integer. A decimal point or an exponent makes a value approximate instead. So an 8 kΩ resistor can be written four ways that are not quite the same thing:

- `8000` — exact
- `8'k` — exact
- `8000.` — approximate
- `8E3` — approximate

The difference does not show until an answer is symbolic or a division does not come out even. `8000` and `8'k` let Symbulator keep a current as 3/500 A; `8000.` and `8E3` make it 0.006 A from the start.

Prefixes in the answers, not just the values

The table above is about what you *type*. Symbulator will also write its answers with prefixes: tick **Use SI prefixes in answers** in **Settings**, and a current of 3/500 A reads 6 mA instead.

Ticking it moves **Rounding** off *exact* by itself, since a prefixed value is a decimal, and choosing *exact* again switches the prefixes off. Symbulator says so on screen when it does.

Every control in the Settings card

TECH NOTE F

Rounding, the four Display ticks and the AC power convention, each with what it changes, what it does not, and why two of them appear only in AC.

Nothing in **Settings** changes an answer. Every control there changes how an answer is *written* — how many figures it carries, whether it wears a unit, whether a complex number is shown as a phasor. The one exception is the AC power convention at the foot of the card, which changes what the average power in an AC circuit *means*, and is called out below.

The lessons reach for these controls as they need them. This note is the whole card in one place, for when you want to know what a tick does without hunting for the lesson that first used it.

Rounding

A menu of four, most exact first, with a second box for **n** that appears only when the choice needs one.

Choice	What you get
exact	The symbolic answer, unrounded: a current comes back as 3/500 A
exact and approx to n digits	Both, side by side, for any answer that is a pure number
approx to n digits	The decimal alone, to **n** significant figures
approx (full precision)	The decimal alone, at full working precision

n takes 2 to 12 and starts at 4. A symbolic answer has nothing to approximate and is left exact under every choice, so *exact and approx* shows one value for `x*vin/(r1 + x)` and two for `0.006`.

The rounding is done in decimal, not in binary, so the last digit is the one you would write by hand.

Display

Four ticks, and they are independent of each other except where noted.

Show units in answers — V, A, W, VA, Ω. On by default. Quantities with no unit to show, such as a power factor or a gain, stay bare whatever this says.

Use SI prefixes in answers — 0.002 becomes 2 m, 1200 becomes 1.2 k. Off by default.

It applies to numeric answers only; a symbolic answer is left as it is. “SI prefixes” (in the Manual) has the full table, and the same shorthand for values you type. Either mu works for micro, whichever your keyboard gives you.

> NOTE
>
> **SI prefixes and *exact* cannot both apply**
>
> A prefixed value is a decimal, so ticking the prefixes moves **Rounding** off *exact* by itself, and choosing *exact* again switches the prefixes off. Whichever you touched last wins, the other visibly changes, and the card says so on screen rather than leaving you to notice.

Show AC answers as polar phasors — 3 + 4j becomes 5∠53.13°. Average power stays real either way, being a real quantity. This tick appears only in AC: in DC every answer is real, and in FD and TR the answers are functions of *s* or *t*, with no angle to take.

Show equations — adds an **Equations** card listing the system the solver assembled. Off by default, because on anything larger than a teaching circuit that system is a page of algebra. The card arrives open when it appears.

AC power convention

RMS phasors. Off means peak amplitude — the ÷2 convention. It affects AC power only, so the row appears in AC and nowhere else.

This is the one setting on the card that changes an answer rather than its appearance. For the same phasor magnitudes, RMS reports twice what peak does — a source of 10 V across a 5 Ω resistor gives 10 W with the tick off and 20 W with it on — and the answer is labelled *average power* under the peak convention and *power consumed* under RMS. Set it to match the book you are working from before comparing numbers.

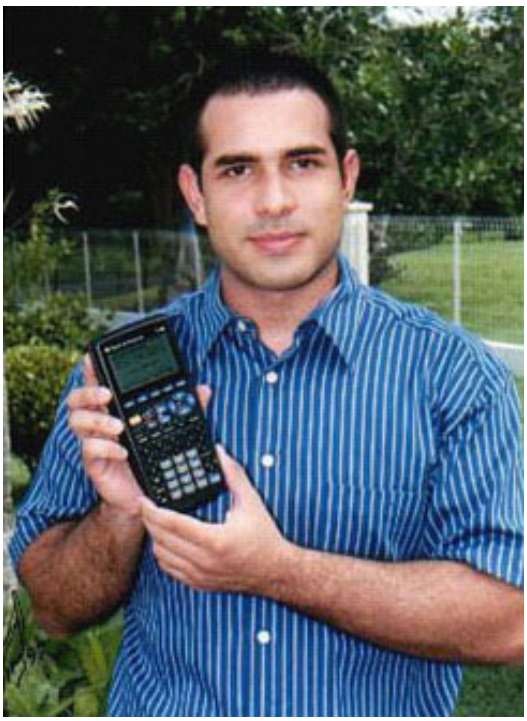
> TIP
>
> **A hidden setting keeps its value**
>
> The two AC controls disappear outside AC rather than greying out, and neither is cleared while it is hidden. Set a circuit up in AC with RMS phasors on, run it in DC, come back to AC, and the tick is where you left it — it travels with the entry when you save one, too.

Roll the credits

If you are interested in the people behind Symbulator, here's who made it and who helped, plus some acknowledgements. And some glowing reviews received over the decades.

Who made Symbulator

Symbulator and its documentation, in all its versions up to this date, have been made by me, **Roberto Perez-Franco**. I started coding it as an engineering student at the Technological University of Panama (Universidad Tecnológica de Panamá) in Azuero, and the timeline below has the dates.



Roberto in 2000, with a TI-89

A timeline

Date	Event
30 March 1999	The first line of code, at the Technological University of Panama in Azuero.
2 April 1999	Version 0.92, the first public release, goes onto the Internet.
29 May 1999	Version 0.994 — the equivalent of version 1.
19 August 1999	Version 2, presented in person for the first time, at UTP Azuero.
October 1999	The Elect SCS plug-in. The program is still called SCS.
November 1999	The paper is written.
Late 1999 or early 2000	Version 3.
14 May 2000	The Power Tools plug-in. The program is now called Symbulator.
October 2000	First place, IEEE Region 9 Student Paper Competition.
December 2000	The paper is published in Panama, in the bulletin of the IEEE section.
9 January 2001	Version 4.
30 March 2001	Version 5, called Q.
9 July 2001	The thesis is defended, and graded 100 out of 100.
September 2001	The paper is published in Barcelona, in the magazine of the IEEE student branch.
June 2013	Version 6, written at MIT.
July 2023	Versions 7 and 8, written in Melbourne.
13 August 2026	The symbulator package goes onto PyPI.
28 August 2026	Version 9, a public beta.
1 October 2026	Versions 9 and 8 leave beta.

What Symbulator did for me

I've never made any money from Symbulator: it has always been and will remain free of charge. But I have benefited from Symbulator in other ways. Thanks in large part to Symbulator, I landed my first job, received a Distinguished Alum award from UTP in 2008 and an Outstanding Young Person (TYOP) award from the Panamanian chapter of Junior Chamber International (JCI), and won a Fulbright Scholarship that took me to a Master of Engineering and later a PhD in Engineering Systems at the Massachusetts Institute of Technology. MIT then hired me, and I lived in Boston for almost twelve years. I now live in Melbourne, Australia. You can say that writing Symbulator changed my life. Beyond my lifetime, Symbulator may preserve a part of my mind as it solves circuits for new generations, the same way a composer lives in the tunes or a writer in the words they conceived.

Dedication

Two people made Symbulator possible, with their love and support.

The first is my wife **Monica**, who had the patience to live with a cyborg for all the years my mind was focused on coding and documenting. *Mi reina*, my attention was elsewhere, but my heart was always with you.

The second is my late father **Tito**, who invested time and money in my education and bought me every gadget and every book I needed or wanted for my studies, with a blind faith that one day something good would come of it. I'm happy to know that I made you proud, Dad, and I miss you every day.

How to reach me

If you liked Symbulator, let me know. My email is my given name, then my last name with a hyphen, as a dot-com address, like so: rxxxxxx@pxxxx-fxxxxx.cxx

Acknowledgements

Human collaborators

Over the past decades, many people have made Symbulator better through their suggestions and corrections, on both the software and the documentation. Many were complete strangers at first and became dear friends through our exchanges. Others were dear friends that became collaborators.

Versions 1 through 6 received feedback and suggestions from José Vega (Panama), Tim Hutcheson (USA), Lars Frederiksen (Denmark), Joe Riel (USA), Arne Harstad (Norway), Erwin Baert (Belgium), Charles 'Chuck' Ware (USA), Doug Burkett (USA), Reinhard Willinski (Germany), Kamil Malinski (Austria), Jake Adams (USA), Daniele Martini (Italy), Rozgonyi Szabolcs (Hungary), Michael Rans (UK), Alex Astashyn (Russia), Al Charpentier (USA), Nevin McChesney (USA), Ivan Oro Yu (Panama), Pepe Iborra (Spain), and Dave Conklin (USA). **Versions 7 and 8** received feedback and suggestions from Carlos Perez Ortega (Chile) and Qifan Wang (China).

Version 9 benefited from feedback and excellent suggestions from my dear old friend Antony García (Panama), who is not only one hell of an engineer and a consummate circuit theorist, but who also taught the same Circuits class at UTP as Prof. Eliane Boulet for many years. I met Antony in May 2013, while he was a student at UTP Azuero in the same major that I had completed more than a dozen years earlier. I was immediately struck by his brilliance, and we became friends. In August 2026, I shared an early version of Symbulator 9 with Antony while he was a PhD candidate in the ECE Department at WPI (MA, USA). Antony generously provided me with a series of fantastic recommendations — from his vantage point as both a student and a teacher of circuit theory — regarding features he would like to see in Symbulator. I worked with Claude over a few weeks in September to turn them into reality.

My thanks to all of them. I truly am in their debt.

AI collaborator

I also want to acknowledge that porting Symbulator to Python/SymPy, creating its interface and websites and updating its documentation, would not have been possible without the coding prowess of Anthropic's artificial intelligence assistant. Claude allowed me to do in two weeks something that I was unable to find the time to do in two decades: to port Symbulator to Python.

> ⚠ WARNING
>
> **Use AI responsibly**
>
> AI is not without risks, and needs careful policies and regulatory oversight on fronts like ethics, systemic biases, its impact on labour, and its voracious use of water and energy.

I believe there is a right way to deploy this technology, and — after working on Symbulator 9 with Claude — I am now convinced that AI will revolutionise the way we work. We just need to do it the right way.

Software I relied on

Symbulator 9 runs on **Python** and is powered by the algebraic capabilities of **SymPy** and the numerical prowess of **NumPy**. I am thankful to the creators of such wonderful open-source software, and to the maintainers who keep their development going. I am proud to honour the open-source community by making Symbulator free and open-source software, from 2026.

Throughout its development, Symbulator 9's answers were checked against **ahkab**, an independent circuit simulator written in Python by Giuseppe Venturini. Every circuit that Symbulator's SPICE translator exports is solved a second time by ahkab, and the two must agree before a release goes out — a habit that has caught more than one convention I had taken for granted. None of ahkab's code is part of Symbulator, and it is not needed to run it: it is a second opinion, kept at arm's length, and a valuable one. My thanks to Giuseppe for building such a tool and sharing it freely.

Problem credits

To make this book useful to actual students, the problems were chosen from widely used textbooks on electric circuit analysis, and their circuit diagrams are reproduced here as they appear in the source.

Code	Textbook
AS2	*Fundamentals of Electric Circuits* (2nd ed.), Charles K. Alexander and Matthew N. O. Sadiku, McGraw-Hill, 2004
AS5	*Fundamentals of Electric Circuits* (5th ed.), Alexander and Sadiku, McGraw-Hill, 2013
AS7	*Fundamentals of Electric Circuits* (7th international ed.), Alexander and Sadiku, McGraw-Hill, 2020
B11	*Introductory Circuit Analysis* (11th ed.), Robert L. Boylestad, Pearson – Prentice Hall, 2007
Bo2	*Elementary Linear Circuit Analysis* (2nd ed.), Leonard S. Bobrow, Oxford, 1987
HK5	*Engineering Circuit Analysis* (5th ed.), William H. Hayt Jr. and Jack E. Kemmerly, McGraw-Hill, 1993
NR9	*Electric Circuits* (9th ed.), James W. Nilsson and Susan A. Riedel, Prentice Hall, 2011
NR11	*Electric Circuits* (11th ed.), Nilsson and Riedel, Pearson, 2020
NR12	*Electric Circuits* (12th ed.), Nilsson and Riedel, Pearson, 2023
RM3	*Circuit Analysis: Theory and Practice* (3rd ed.), Allan H. Robbins and Wilhelm C. Miller, Thomson – Delmar Learning, 2004
TR5	*The Analysis and Design of Linear Circuits* (5th ed.), Roland E. Thomas and Albert J. Rosa, Wiley, 2006

For the purpose of teaching students how to use Symbulator, these diagrams are reproduced under the principle of fair use. No copyright infringement is intended.

Praise received

Over the past quarter of a century, the words of encouragement I received from users of Symbulator were my incentive to keep working on it. Here is a little of that precious collection.

> *The masterpiece of TI-89 programming.*
>
> — **Alex Astashyn**, EE+CoE+CoS major, Polytechnic University, Brooklyn, New York, USA

> *Symbulator is and always will be the best program for TI-89/92/V200.*
>
> — **Charles Ware**, EE major, Columbus, Ohio, USA

> *Roberto, your program Symbulator is by itself a reason to buy the calc. It's a masterpiece of programming.*
>
> — **Pepe Iborra**, Telecommunications, Spain

> *The program is really a masterpiece. It gives you everything you need, very accurate, very fast. It's really what every Electrical Engineer dreams of. It's so easy to use and*

it does everything! DC, AC, Transient analysis, Thévenin, Bode plots... everything! I'm really amazed.

— **Nikolaos Trichakis**, EE major, Aristotle University of Thessaloniki, Greece

I wish I had it while going through school.

— **Joe Riel**, BSEE, creator of Syrup (a symbolic circuit simulator for Maple V), University of California, Irvine, USA

Congratulations for your program: it's simply fantastic. Symbulator is worth the cost of a TI-89!

— **Pier Giorgio Raponi**, engineering student, Pisa, Italy

You've made an excellent program. Symbulator is really the best circuit simulator ever made for a calculator.

— **Rui Sebastião**, EE major, University of Coimbra, Portugal

*Symbulator is probably the most useful program ever made for a TI calculator. I am an electronics student and even paid for the EE*Pro software, yet I find myself using Symbulator more because it is much better at what it does.*

— **Josh Cunningham**, electronics major

Symbulator, by far, has been the best tool for EE I have come across yet. [...] It is truly an impressive piece of work. You should be proud. It certainly is helping me to focus on my studying of EE instead of trying to remember all the math.

— **Jim Wachtel**, EE, Process Control & Electrical Co., St. Louis, Missouri, USA

Licence

From June 2013, Symbulator was offered under a Creative Commons BY-NC-SA licence. Since 2026 it is open source under the MIT licence, which supersedes it. If you want to port it to another platform, you are free to do so, provided the copyright notice and the attribution of authorship travel with it.

Index

aa mini-tool, 208, 210
AC analysis, 206
analyses (DC, AC, TR, FD), 6
angular frequency (ω), 206
answer names, 12, 298
autotransformer, 251
average power, 208

Bode plot, 263
Bode plot of a transfer function H(s), 264
brackets, 46
built-in examples, 8, 294
By-Hand Equations card, 17

capacitor, 153
case sensitivity, 32
circuit description, 9
complex power, 208, 226
Conditions (Evaluate), 85
conductance, 43
coupling coefficient (k), 244, 247
curly-bracket shorthand, 274
current source, 42

DC analysis, 9
delta-connected sources, 238
dependent source, 47, 214
dot convention, 245

element types, 6
Equations card, 17, 306
equivalent circuit button, 86, 300
equivalent resistance, 82
Evaluate card, 17
exact and approximate values, 303
Expert Mode, 36

floating section (island), 249, 278
four-terminal form, 249, 278
frequency domain, 272

gain mini-tool, 284
getting Symbulator (online, offline, local), 7
ground node, 11

impedance, 207
impulse δ(t), 155, 272
inductor, 154
initial conditions, 153
input files (.cir), 10, 294

Laplace transform, 274
limiting the results (TR), 156, 296
load problems (irl, vrl, prl), 85, 300

maximum power transfer, 86, 223
mesh current direction, 18
Mini-Tools card, 207
MIT licence, 7
multiple solutions, 37
mutual inductance, 244

names of elements and nodes, 10
nodal and mesh analysis, 17
nonlinear devices, 7
Norton equivalent, 83
numerical-from-symbolic problems, 32

operational amplifier, 119

parallel resistors, 44
passive circuit, 83
pf mini-tool, 224
phasor, 206
plots (Plotting Tools card), 156
polar phasors, 208, 306
poles and zeros, 207
power consumed and delivered, 13
power factor, 224
pr function, 44
pz mini-tool, 207

reactances given in ohms, 207
real solutions only, 268
reporting problems, 7
resistance seen by a source, 13
resistor, 10
resonance, 268

Results card, 11
RMS, 223, 306
Rounding setting, 11, 305

s-domain, 272
separators (newline or colon), 10
Settings card, 12, 305
short circuit, 80
SI prefixes, 11, 303, 304
sign conventions, 16
sinusoids, 206
Solve card, 35
split view, 8
subscripts in answers, 298
supernodes and supermeshes, 17
switched circuits (two runs), 156
symbolic circuit, 32

t2s and s2t, 274
three-phase circuits, 230
Thévenin equivalent, 83
transfer function, 275
transformer, 248
transformer polarity, 249
transient analysis (TR), 154
two-port, 277, 278
two-port parameters (z, y, h, g, a, b), 277

unbalanced system, 233
underscore, 12, 298
unit step u(t), 155, 274

voltage source, 12