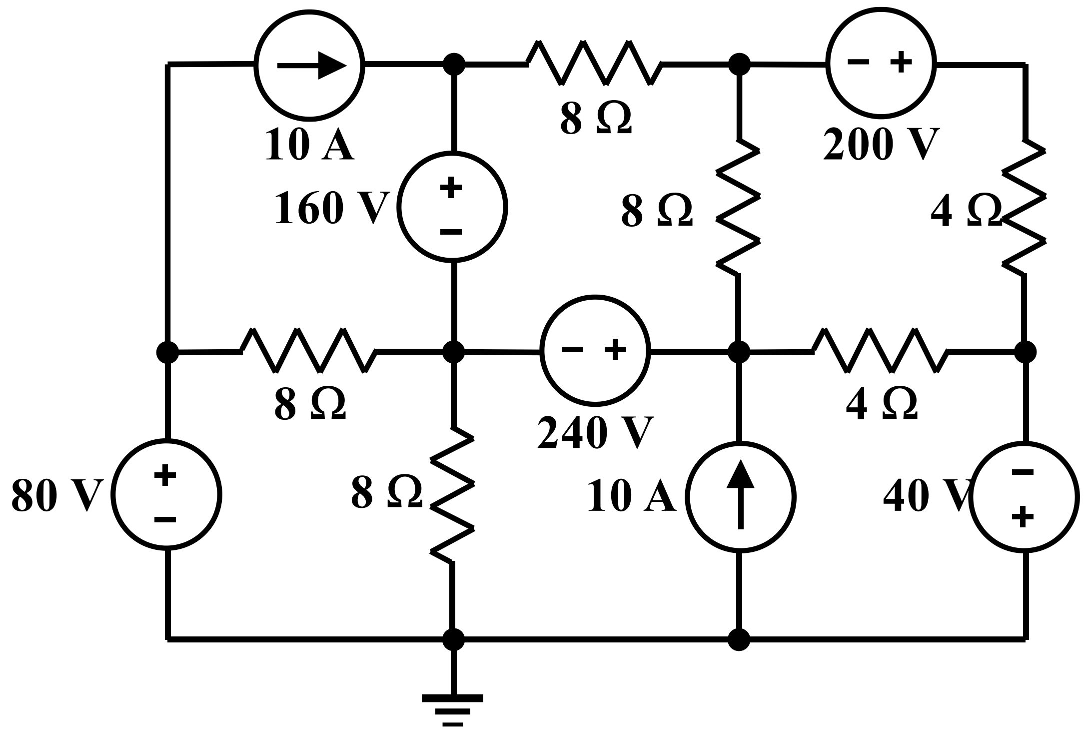
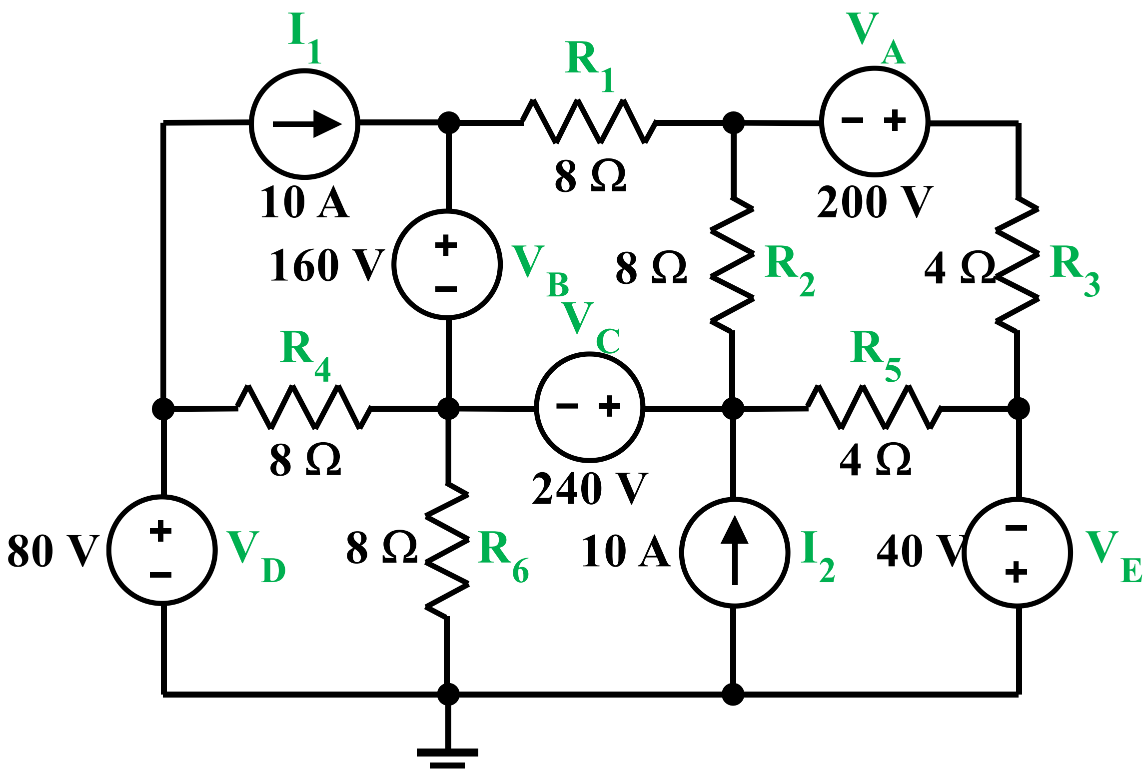
A Detailed Modified Nodal Analysis Example

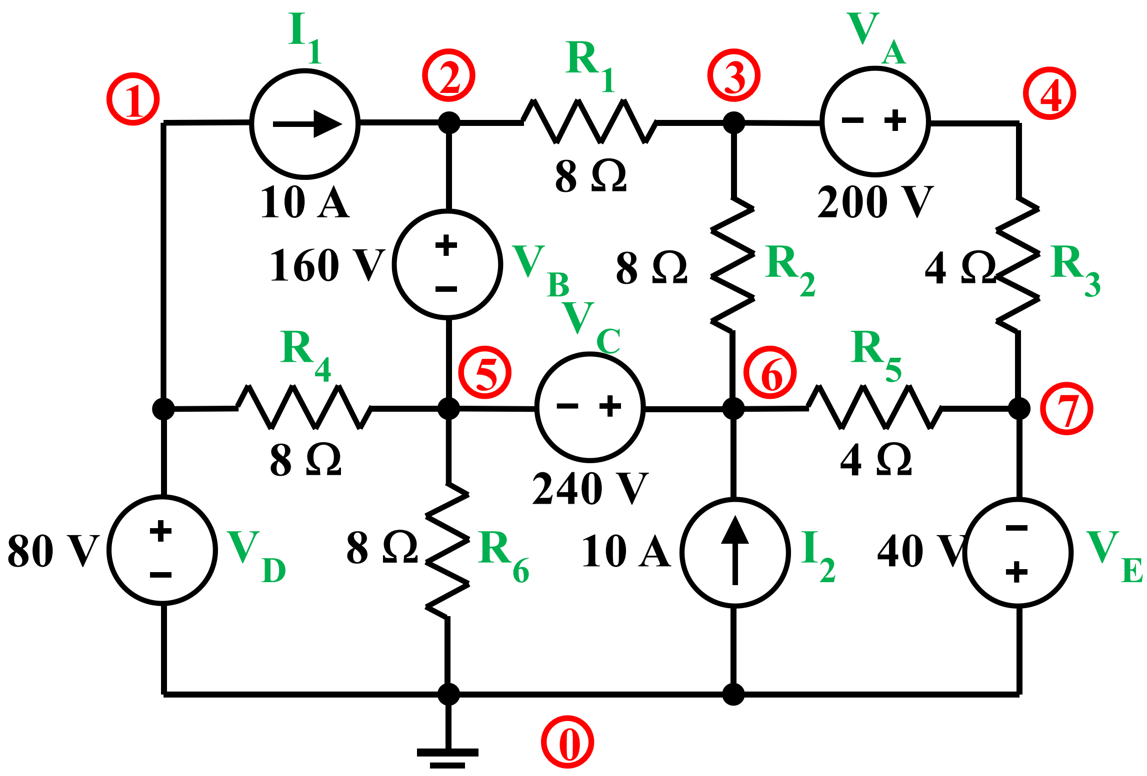
Given the circuit shown:



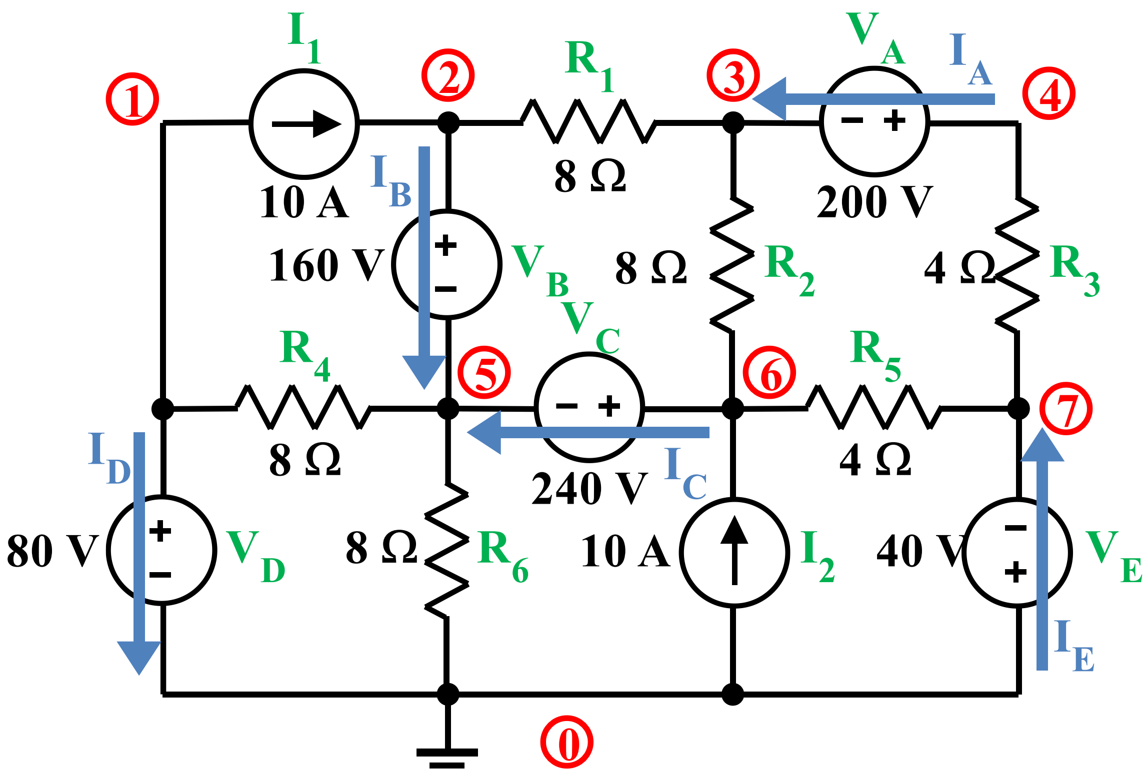
Label the components (**shown here in green**):



Label the nodes (**shown here in red**):



Define a current through each independent voltage source (must satisfy the Passive Sign Convention, **shown here in blue**).



Write a KCL equation at each node (except the reference node), with independent current sources on the right-hand side. Currents leaving the node are taken as positive.

Node 1: $\frac{V_1 - V_5}{R_4} + I_D = -I_1$

Node 2: $\frac{V_2 - V_3}{R_1} + I_B = I_1$

Node 3: $\frac{V_3 - V_2}{R_1} + \frac{V_3 - V_6}{R_2} - I_A = 0$

Node 4: $I_A + \frac{V_4 - V_7}{R_3} = 0$

Node 5: $-I_B + \frac{V_5 - V_1}{R_4} + \frac{V_5}{R_6} - I_C = 0$

Node 6: $I_C + \frac{V_6 - V_3}{R_2} + \frac{V_6 - V_7}{R_5} = I_2$

Node 7: $\frac{V_7 - V_4}{R_3} + \frac{V_7 - V_6}{R_5} - I_E = 0$

Node 0: $-I_D - \frac{V_5}{R_6} + I_E = -I_2$ This equation is redundant and as noted above, by convention, is normally not included

We will ignore it in the following discussion.

Write a constraint equation for each voltage source, with independent voltage sources on the right-hand side:

V_A: $V_4 - V_3 = V_A$

V_B: $V_2 - V_5 = V_B$

V_C: $V_6 - V_5 = V_C$

V_D: $V_1 = V_D$

V_E: $-V_7 = V_E$

Express ***all*** equations in the combined matrix form shown below. (Blank spaces are zeros, not shown for easier visualization.)

		Node Voltages							Voltage-Source Currents								
		V_1	V_2	V_3	V_4	V_5	V_6	V_7	I_A	I_B	I_C	I_D	I_E				
Node Numbers	N_1	$\frac{1}{R_4}$				$-\frac{1}{R_4}$						1			V_1		$-I_1$
	N_2		$\frac{1}{R_1}$	$-\frac{1}{R_1}$						1					V_2		I_1
	N_3		$-\frac{1}{R_1}$	$\frac{1}{R_1}+\frac{1}{R_2}$			$-\frac{1}{R_2}$		-1						V_3		0
	N_4				$\frac{1}{R_3}$			$-\frac{1}{R_3}$	1						V_4		0
	N_5	$-\frac{1}{R_4}$				$\frac{1}{R_4}+\frac{1}{R_6}$				-1	-1				V_5		0
	N_6			$-\frac{1}{R_2}$			$\frac{1}{R_2}+\frac{1}{R_5}$	$-\frac{1}{R_5}$			1				V_6		I_2
	N_7				$-\frac{1}{R_3}$		$-\frac{1}{R_5}$	$\frac{1}{R_3}+\frac{1}{R_5}$					-1	$\times$	V_7	$=$	0
Voltage Sources	V_A			-1	1										I_A		V_A
	V_B		1			-1									I_B		V_B
	V_C					-1	1								I_C		V_C
	V_D	1													I_D		V_D
	V_E							-1							I_E		V_E

Alternatively, to do this analysis with SCAM:

Create a SPICE-like netlist file called “My_Example.cir” and save it in the MATLAB path. (Note that there is no title line, no .end line, and no analysis command.)

```
I1 1 2 10
R1 2 3 8
VA 4 3 200
VB 2 5 160
R2 3 6 8
R3 4 7 4
R4 1 5 8
VC 6 5 240
R5 6 7 4
VD 1 0 80
R6 5 0 8
I2 0 6 10
VE 0 7 40
```

Then, in MATLAB do the following:

```
>> fname="My_Example.cir";
>> ISU_scam

Started -- please be patient.

Netlist:
I1 1 2 10
R1 2 3 8
VA 4 3 200
VB 2 5 160
R2 3 6 8
R3 4 7 4
R4 1 5 8
VC 6 5 240
R5 6 7 4
```

```
VD 1 0 80
R6 5 0 8
I2 0 6 10
VE 0 7 40

The A matrix:
[ 1/R4,     0,            0,     0,       -1/R4,           0,           0,  0,  0,  0, 1,  0]
[    0,  1/R1,       -1/R1,     0,           0,           0,           0,  0,  1,  0, 0,  0]
[    0, -1/R1, 1/R1 + 1/R2,     0,           0,       -1/R2,           0, -1,  0,  0, 0,  0]
[    0,     0,            0,  1/R3,           0,           0,       -1/R3,  1,  0,  0, 0,  0]
[-1/R4,     0,            0,     0, 1/R4 + 1/R6,           0,           0,  0, -1, -1, 0,  0]
[    0,     0,       -1/R2,     0,           0, 1/R2 + 1/R5,       -1/R5,  0,  0,  1, 0,  0]
[    0,     0,            0, -1/R3,           0,       -1/R5, 1/R3 + 1/R5,  0,  0,  0, 0, -1]
[    0,     0,          -1,     1,           0,           0,           0,  0,  0,  0, 0,  0]
[    0,     1,            0,     0,          -1,           0,           0,  0,  0,  0, 0,  0]
[    0,     0,            0,     0,          -1,           1,           0,  0,  0,  0, 0,  0]
[    1,     0,            0,     0,           0,           0,           0,  0,  0,  0, 0,  0]
[    0,     0,            0,     0,           0,           0,          -1,  0,  0,  0, 0,  0]


The x vector:
 v_1
 v_2
 v_3
 v_4
 v_5
 v_6
 v_7
I_VA
I_VB
I_VC
I_VD
I_VE


The z vector:
-I1
```

```
 I1
  0
  0
  0
 I2
  0
 VA
 VB
 VC
 VD
 VE
 The matrix equation:
                   I_VD + v_1/R4 - v_5/R4 == -I1
                    I_VB + v_2/R1 - v_3/R1 == I1
 v_3*(1/R1 + 1/R2) - v_2/R1 - v_6/R2 - I_VA == 0
                     I_VA + v_4/R3 - v_7/R3 == 0
   v_5*(1/R4 + 1/R6) - I_VC - v_1/R4 - I_VB == 0
I_VC - v_3/R2 - v_7/R5 + v_6*(1/R2 + 1/R5) == I2
 v_7*(1/R3 + 1/R5) - v_4/R3 - v_6/R5 - I_VE == 0
                                 v_4 - v_3 == VA
                                 v_2 - v_5 == VB
                                 v_6 - v_5 == VC
                                       v_1 == VD
                                      -v_7 == VE


The solution:

v_1 == VD

v_2 == (R1*R2*R4*R5*VB - R2*R4*R5*R6*VA - R1*R4*R5*R6*VA + R1*R2*R4*R6*VB + R1*R3*R4*R5*VB + R1*R2*R5*R6*VB
+ R1*R3*R4*R6*VB + R2*R3*R4*R5*VB + R1*R3*R5*R6*VB + R2*R3*R4*R6*VB + R1*R4*R5*R6*VB + R2*R3*R5*R6*VB -
R1*R2*R4*R6*VC - R1*R3*R4*R6*VC - R2*R3*R4*R6*VC - R1*R4*R5*R6*VC + R1*R2*R5*R6*VD + R1*R3*R5*R6*VD +
R2*R3*R5*R6*VD - R1*R2*R4*R6*VE - R1*R3*R4*R6*VE - R2*R3*R4*R6*VE - R1*R4*R5*R6*VE - R2*R4*R5*R6*VE +
I1*R1*R2*R4*R5*R6 + I1*R1*R3*R4*R5*R6 + I2*R1*R2*R4*R5*R6 + I1*R2*R3*R4*R5*R6 + I2*R1*R3*R4*R5*R6 +
I2*R2*R3*R4*R5*R6)/(R1*R2*R4*R5 + R1*R2*R4*R6 + R1*R3*R4*R5 + R1*R2*R5*R6 + R1*R3*R4*R6 + R2*R3*R4*R5 +
R1*R3*R5*R6 + R2*R3*R4*R6 + R1*R4*R5*R6 + R2*R3*R5*R6 + R2*R4*R5*R6)
```

v_3 == -(R1*R2*R4*R5*VA + R1*R2*R4*R6*VA + R1*R2*R5*R6*VA + R1*R4*R5*R6*VA + R2*R4*R5*R6*VA - R2*R3*R4*R5*VB
- R2*R3*R4*R6*VB - R2*R3*R5*R6*VB - R1*R3*R4*R5*VC - R1*R3*R5*R6*VC + R2*R3*R4*R6*VC - R1*R3*R5*R6*VD -
R2*R3*R5*R6*VD + R1*R2*R4*R5*VE + R1*R2*R4*R6*VE + R1*R2*R5*R6*VE + R1*R3*R4*R6*VE + R2*R3*R4*R6*VE +
R1*R4*R5*R6*VE + R2*R4*R5*R6*VE - I1*R1*R3*R4*R5*R6 - I1*R2*R3*R4*R5*R6 - I2*R1*R3*R4*R5*R6 -
I2*R2*R3*R4*R5*R6)/(R1*R2*R4*R5 + R1*R2*R4*R6 + R1*R3*R4*R5 + R1*R2*R5*R6 + R1*R3*R4*R6 + R2*R3*R4*R5 +
R1*R3*R5*R6 + R2*R3*R4*R6 + R1*R4*R5*R6 + R2*R3*R5*R6 + R2*R4*R5*R6)

v_4 == (R1*R3*R4*R5*VA + R1*R3*R4*R6*VA + R2*R3*R4*R5*VA + R1*R3*R5*R6*VA + R2*R3*R4*R6*VA + R2*R3*R5*R6*VA
+ R2*R3*R4*R5*VB + R2*R3*R4*R6*VB + R2*R3*R5*R6*VB + R1*R3*R4*R5*VC + R1*R3*R5*R6*VC - R2*R3*R4*R6*VC +
R1*R3*R5*R6*VD + R2*R3*R5*R6*VD - R1*R2*R4*R5*VE - R1*R2*R4*R6*VE - R1*R2*R5*R6*VE - R1*R3*R4*R6*VE -
R2*R3*R4*R6*VE - R1*R4*R5*R6*VE - R2*R4*R5*R6*VE + I1*R1*R3*R4*R5*R6 + I1*R2*R3*R4*R5*R6 + I2*R1*R3*R4*R5*R6
+ I2*R2*R3*R4*R5*R6)/(R1*R2*R4*R5 + R1*R2*R4*R6 + R1*R3*R4*R5 + R1*R2*R5*R6 + R1*R3*R4*R6 + R2*R3*R4*R5 +
R1*R3*R5*R6 + R2*R3*R4*R6 + R1*R4*R5*R6 + R2*R3*R5*R6 + R2*R4*R5*R6)

v_5 == -(R6*(R1*R4*R5*VA + R2*R4*R5*VA + R2*R4*R5*VB + R1*R2*R4*VC + R1*R3*R4*VC + R2*R3*R4*VC + R1*R4*R5*VC
- R1*R2*R5*VD - R1*R3*R5*VD - R2*R3*R5*VD + R1*R2*R4*VE + R1*R3*R4*VE + R2*R3*R4*VE + R1*R4*R5*VE +
R2*R4*R5*VE - I1*R1*R2*R4*R5 - I1*R1*R3*R4*R5 - I2*R1*R2*R4*R5 - I1*R2*R3*R4*R5 - I2*R1*R3*R4*R5 -
I2*R2*R3*R4*R5))/(R1*R2*R4*R5 + R1*R2*R4*R6 + R1*R3*R4*R5 + R1*R2*R5*R6 + R1*R3*R4*R6 + R2*R3*R4*R5 +
R1*R3*R5*R6 + R2*R3*R4*R6 + R1*R4*R5*R6 + R2*R3*R5*R6 + R2*R4*R5*R6)

v_6 == (R1*R2*R4*R5*VC - R2*R4*R5*R6*VA - R2*R4*R5*R6*VB - R1*R4*R5*R6*VA + R1*R3*R4*R5*VC + R1*R2*R5*R6*VC
+ R2*R3*R4*R5*VC + R1*R3*R5*R6*VC + R2*R3*R5*R6*VC + R2*R4*R5*R6*VC + R1*R2*R5*R6*VD + R1*R3*R5*R6*VD +
R2*R3*R5*R6*VD - R1*R2*R4*R6*VE - R1*R3*R4*R6*VE - R2*R3*R4*R6*VE - R1*R4*R5*R6*VE - R2*R4*R5*R6*VE +
I1*R1*R2*R4*R5*R6 + I1*R1*R3*R4*R5*R6 + I2*R1*R2*R4*R5*R6 + I1*R2*R3*R4*R5*R6 + I2*R1*R3*R4*R5*R6 +
I2*R2*R3*R4*R5*R6)/(R1*R2*R4*R5 + R1*R2*R4*R6 + R1*R3*R4*R5 + R1*R2*R5*R6 + R1*R3*R4*R6 + R2*R3*R4*R5 +
R1*R3*R5*R6 + R2*R3*R4*R6 + R1*R4*R5*R6 + R2*R3*R5*R6 + R2*R4*R5*R6)

v_7 == -VE

I_VA == -(R1*R4*R5*VA + R1*R4*R6*VA + R2*R4*R5*VA + R1*R5*R6*VA + R2*R4*R6*VA + R2*R5*R6*VA + R2*R4*R5*VB +
R2*R4*R6*VB + R2*R5*R6*VB + R1*R4*R5*VC + R1*R5*R6*VC - R2*R4*R6*VC + R1*R5*R6*VD + R2*R5*R6*VD +
R1*R4*R5*VE + R2*R4*R5*VE + R1*R5*R6*VE + R2*R5*R6*VE + I1*R1*R4*R5*R6 + I1*R2*R4*R5*R6 + I2*R1*R4*R5*R6 +
I2*R2*R4*R5*R6)/(R1*R2*R4*R5 + R1*R2*R4*R6 + R1*R3*R4*R5 + R1*R2*R5*R6 + R1*R3*R4*R6 + R2*R3*R4*R5 +
R1*R3*R5*R6 + R2*R3*R4*R6 + R1*R4*R5*R6 + R2*R3*R5*R6 + R2*R4*R5*R6)

I_VB == (R2*R4*R6*VC - R2*R4*R6*VA - R2*R5*R6*VA - R2*R4*R5*VB - R2*R4*R6*VB - R3*R4*R5*VB - R2*R5*R6*VB -

```
R3*R4*R6*VB - R3*R5*R6*VB - R4*R5*R6*VB - R2*R4*R5*VA + R3*R4*R5*VC + R3*R4*R6*VC + R3*R5*R6*VC +
R4*R5*R6*VC - R2*R5*R6*VD - R2*R4*R5*VE - R2*R5*R6*VE + I1*R1*R2*R4*R5 + I1*R1*R2*R4*R6 + I1*R1*R3*R4*R5 +
I1*R1*R2*R5*R6 + I1*R1*R3*R4*R6 + I1*R2*R3*R4*R5 + I1*R1*R3*R5*R6 + I1*R2*R3*R4*R6 + I1*R1*R4*R5*R6 +
I1*R2*R3*R5*R6 - I2*R2*R4*R5*R6)/(R1*R2*R4*R5 + R1*R2*R4*R6 + R1*R3*R4*R5 + R1*R2*R5*R6 + R1*R3*R4*R6 +
R2*R3*R4*R5 + R1*R3*R5*R6 + R2*R3*R4*R6 + R1*R4*R5*R6 + R2*R3*R5*R6 + R2*R4*R5*R6)

I_VC == -(R1*R4*R5*VA + R1*R5*R6*VA - R2*R4*R6*VA - R2*R4*R6*VB - R3*R4*R5*VB - R3*R4*R6*VB - R3*R5*R6*VB -
R4*R5*R6*VB + R1*R2*R4*VC + R1*R3*R4*VC + R1*R2*R6*VC + R2*R3*R4*VC + R1*R3*R6*VC + R1*R4*R5*VC +
R2*R3*R6*VC + R1*R5*R6*VC + R2*R4*R6*VC + R3*R4*R5*VC + R3*R4*R6*VC + R3*R5*R6*VC + R4*R5*R6*VC +
R1*R2*R6*VD + R1*R3*R6*VD + R2*R3*R6*VD + R1*R5*R6*VD + R1*R2*R4*VE + R1*R3*R4*VE + R1*R2*R6*VE +
R2*R3*R4*VE + R1*R3*R6*VE + R1*R4*R5*VE + R2*R3*R6*VE + R1*R5*R6*VE + I1*R1*R2*R4*R6 - I2*R1*R2*R4*R5 +
I1*R1*R3*R4*R6 - I2*R1*R3*R4*R5 + I1*R2*R3*R4*R6 - I2*R1*R2*R5*R6 - I2*R2*R3*R4*R5 + I1*R1*R4*R5*R6 -
I2*R1*R3*R5*R6 - I2*R2*R3*R5*R6 - I2*R2*R4*R5*R6)/(R1*R2*R4*R5 + R1*R2*R4*R6 + R1*R3*R4*R5 + R1*R2*R5*R6 +
R1*R3*R4*R6 + R2*R3*R4*R5 + R1*R3*R5*R6 + R2*R3*R4*R6 + R1*R4*R5*R6 + R2*R3*R5*R6 + R2*R4*R5*R6)

I_VD == -(R1*R5*R6*VA + R2*R5*R6*VA + R2*R5*R6*VB + R1*R2*R6*VC + R1*R3*R6*VC + R2*R3*R6*VC + R1*R5*R6*VC +
R1*R2*R5*VD + R1*R2*R6*VD + R1*R3*R5*VD + R1*R3*R6*VD + R2*R3*R5*VD + R2*R3*R6*VD + R1*R5*R6*VD +
R2*R5*R6*VD + R1*R2*R6*VE + R1*R3*R6*VE + R2*R3*R6*VE + R1*R5*R6*VE + R2*R5*R6*VE + I1*R1*R2*R4*R5 +
I1*R1*R2*R4*R6 + I1*R1*R3*R4*R5 + I1*R1*R3*R4*R6 + I1*R2*R3*R4*R5 + I1*R2*R3*R4*R6 - I2*R1*R2*R5*R6 +
I1*R1*R4*R5*R6 - I2*R1*R3*R5*R6 + I1*R2*R4*R5*R6 - I2*R2*R3*R5*R6)/(R1*R2*R4*R5 + R1*R2*R4*R6 + R1*R3*R4*R5
+ R1*R2*R5*R6 + R1*R3*R4*R6 + R2*R3*R4*R5 + R1*R3*R5*R6 + R2*R3*R4*R6 + R1*R4*R5*R6 + R2*R3*R5*R6 +
R2*R4*R5*R6)
                                                                          I_VE == -(R1*R4*R5*VA + R2*R4*R5*VA
+ R1*R5*R6*VA + R2*R5*R6*VA + R2*R4*R5*VB + R2*R5*R6*VB + R1*R2*R4*VC + R1*R3*R4*VC + R1*R2*R6*VC +
R2*R3*R4*VC + R1*R3*R6*VC + R1*R4*R5*VC + R2*R3*R6*VC + R1*R5*R6*VC + R1*R2*R6*VD + R1*R3*R6*VD +
R2*R3*R6*VD + R1*R5*R6*VD + R2*R5*R6*VD + R1*R2*R4*VE + R1*R3*R4*VE + R1*R2*R6*VE + R2*R3*R4*VE +
R1*R3*R6*VE + R1*R4*R5*VE + R2*R3*R6*VE + R2*R4*R5*VE + R1*R5*R6*VE + R2*R5*R6*VE + I1*R1*R2*R4*R6 +
I1*R1*R3*R4*R6 + I2*R1*R2*R4*R6 + I1*R2*R3*R4*R6 + I2*R1*R3*R4*R6 + I1*R1*R4*R5*R6 + I2*R2*R3*R4*R6 +
I1*R2*R4*R5*R6 + I2*R1*R4*R5*R6 + I2*R2*R4*R5*R6)/(R1*R2*R4*R5 + R1*R2*R4*R6 + R1*R3*R4*R5 + R1*R2*R5*R6 +
R1*R3*R4*R6 + R2*R3*R4*R5 + R1*R3*R5*R6 + R2*R3*R4*R6 + R1*R4*R5*R6 + R2*R3*R5*R6 + R2*R4*R5*R6)


Elapsed time is 4.00561 seconds.
```

To get numerical values, try things like:

```
>> eval(z)

ans =

   -10
    10
     0
     0
     0
    10
     0
   200
   160
   240
    80
    40
```

or

```
>> eval(I_VC)

ans =

   -45
```