EE 105 | Discussion 2

Kieran Peleaux & Qianyi Xie

Discussion Outline

- Overview of SPICE
  - What is SPICE?
  - SPICE Workflow
  - Netlist Syntax
    - Topology & Analysis
  - Running a SPICE Simulation
  - Viewing Simulation Results
  - Beyond the Basics

- Non-ideal op amp practice

What is SPICE?

Simulation

Program

with

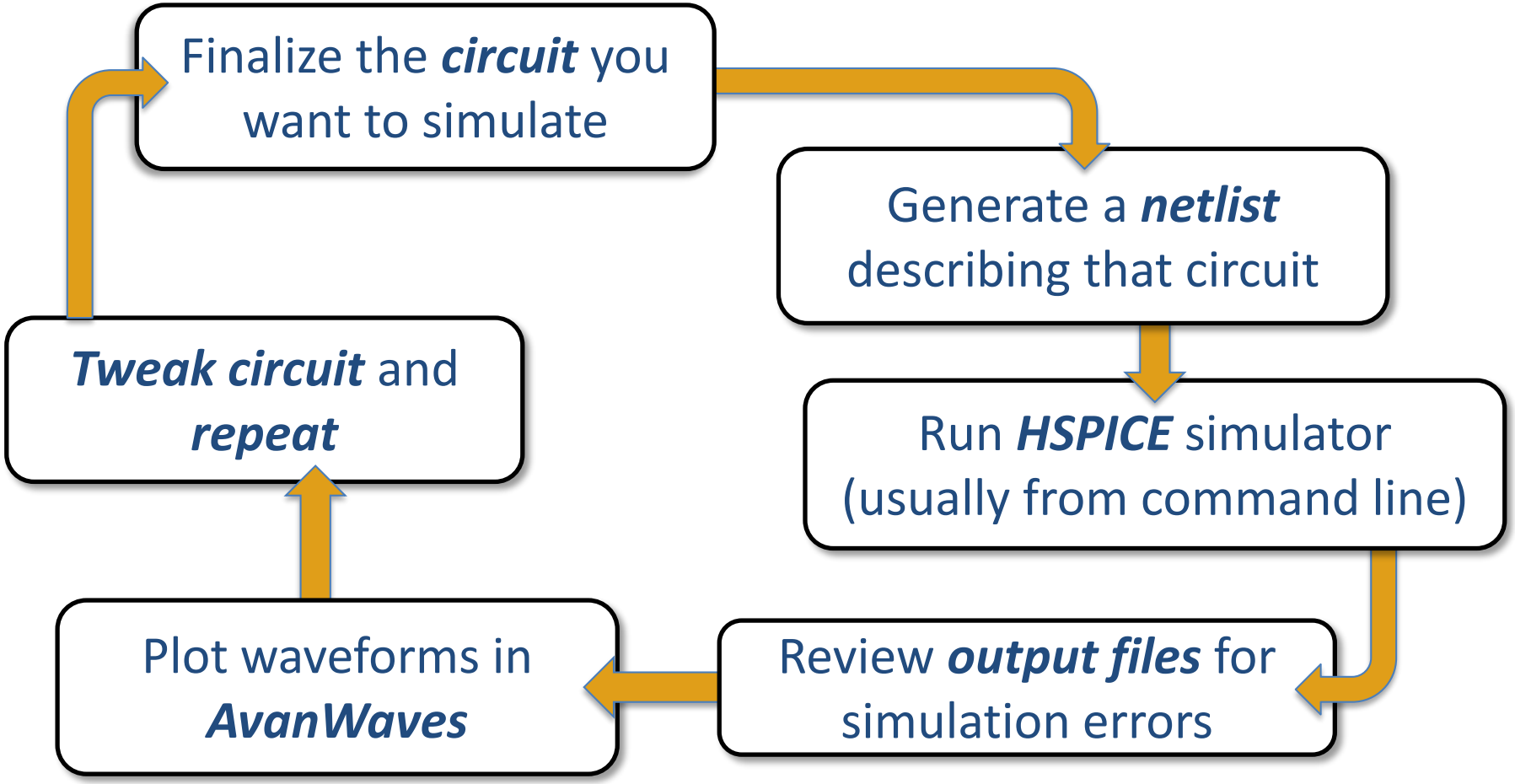
Integrated

Circuit

Emphasis

- Software used for analog circuit simulation (originally intended for developing ICs)
- Developed @ UC Berkeley—version 1 released in 1973 (under public domain)
- Started out as a command-line tool
- Now multiple companies offer their own packaged versions of spice
  - ***LTspice***, ***HSPICE***, *PSpice, TINA-TI*

SPICE Workflow



What is a Netlist?

- A simple text file that contains a ***circuit description*** and ***analysis options***
- Different circuit elements specified by unique commands
- Circuit is topology defined by:
  - giving each node a unique name
  - assigning elements between these nodes

Netlist Syntax

- Filename ends in .sp
  - E.g., `mycircuit.sp`
- First line is always a comment!
- Not case sensitive
  - `vs = Vs = VS = vS`
- Last line must be `.end`
- Other than first/last line, order doesn't matter

```
1  EE105 SPICE Tutorial Example 1 - Simple RC Circuit
2  vs vs gnd PWL(0s 0V 5ms 0V 5.001ms 5V 10ms 5V)
3  r1 vs vo  1k
4  c1 vo gnd 1uF
5  .tran 0.01ms 10ms
6  .option post=2 nomod
7  .end
```

Netlist Syntax | Topology

- Each line denotes a different circuit element
  - The 1st term defines the type of element and assigns it an arbitrary number/letter,
  - The 2nd term is the first node the element is connected to
  - The 3rd term is the second node the element is connected to (more nodes will follow for devices with >2 terminals)
  - The last term is the element value (can use prefixes `f`, `p`, `n`, `u`, `m`, `k`, `meg`, `giga`, & `tera` to denote magnitude)

```
1  EE105 SPICE Tutorial Example 1 - Simple RC Circuit
2  vs vs gnd PWL(0s 0V 5ms 0V 5.001ms 5V 10ms 5V)
3  r1 vs vo  1k
4  c1 vo gnd 1uF
5  .tran 0.01ms 10ms
6  .option post=2 nomod
7  .end
```

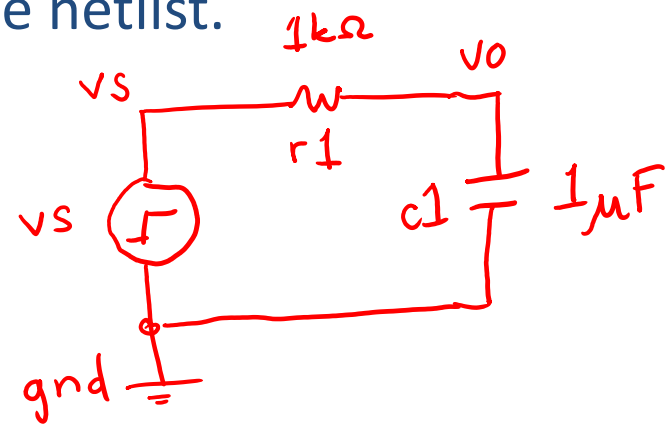
Netlist Syntax | Topology

- Circuit nodes & elements can have the same name
- `gnd` is a standard name used to reference ground (can also use `0`)
- Order of nodes matters for things like sources!

```
1  EE105 SPICE Tutorial Example 1 - Simple RC Circuit
2  vs vs gnd PWL(0s 0V 5ms 0V 5.001ms 5V 10ms 5V)
3  r1 vs vo  1k
4  c1 vo gnd 1uF
5  .tran 0.01ms 10ms
6  .option post=2 nomod
7  .end
```

Netlist Syntax | Topology

- What does this circuit look like? Draw using labels that match the netlist.




```
1  EE105 SPICE Tutorial Example 1 - Simple RC Circuit
2  vs vs gnd PWL(0s 0V 5ms 0V 5.001ms 5V 10ms 5V)
3  r1 vs vo  1k
4  c1 vo gnd 1uF
5  .tran 0.01ms 10ms
6  .option post=2 nomod
7  .end
```

Netlist Syntax | Analysis

- Line 5 tells HSPICE to perform a transient analysis from time $t = 0$ ms to $t = 10$ ms in steps of 10 µs
- Line 6 tells HSPICE to generate waveform files necessary for viewing in *awaves* while not including model info in the output

```
1  EE105 SPICE Tutorial Example 1 - Simple RC Circuit
2  vs vs gnd PWL(0s 0V 5ms 0V 5.001ms 5V 10ms 5V)
3  r1 vs vo  1k
4  c1 vo gnd 1uF
5  .tran 0.01ms 10ms
6  .option post=2 nomod
7  .end
```

Simulating in HSPICE

- To simulate the circuit, simply run the command below in a UNIX terminal

```
hspice mycircuit.sp > mycircuit.lis
```

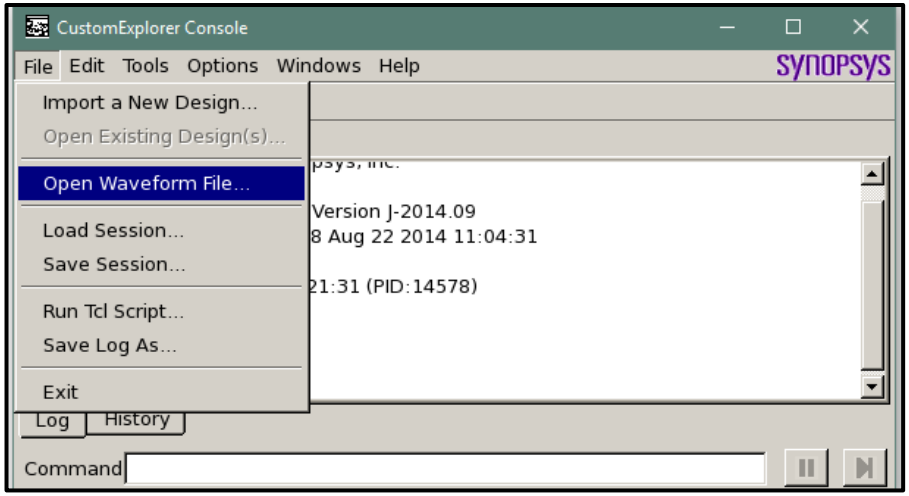
- This will run the simulation and store the outputs in `mycircuit.lis`
- You can open `mycircuit.lis` and check for errors/operating points, but most of the time we'll be interested in looking at plots of voltages and currents

Viewing Simulation Results

- To run *awaves*, use the command below (make sure you have X11 enabled!)

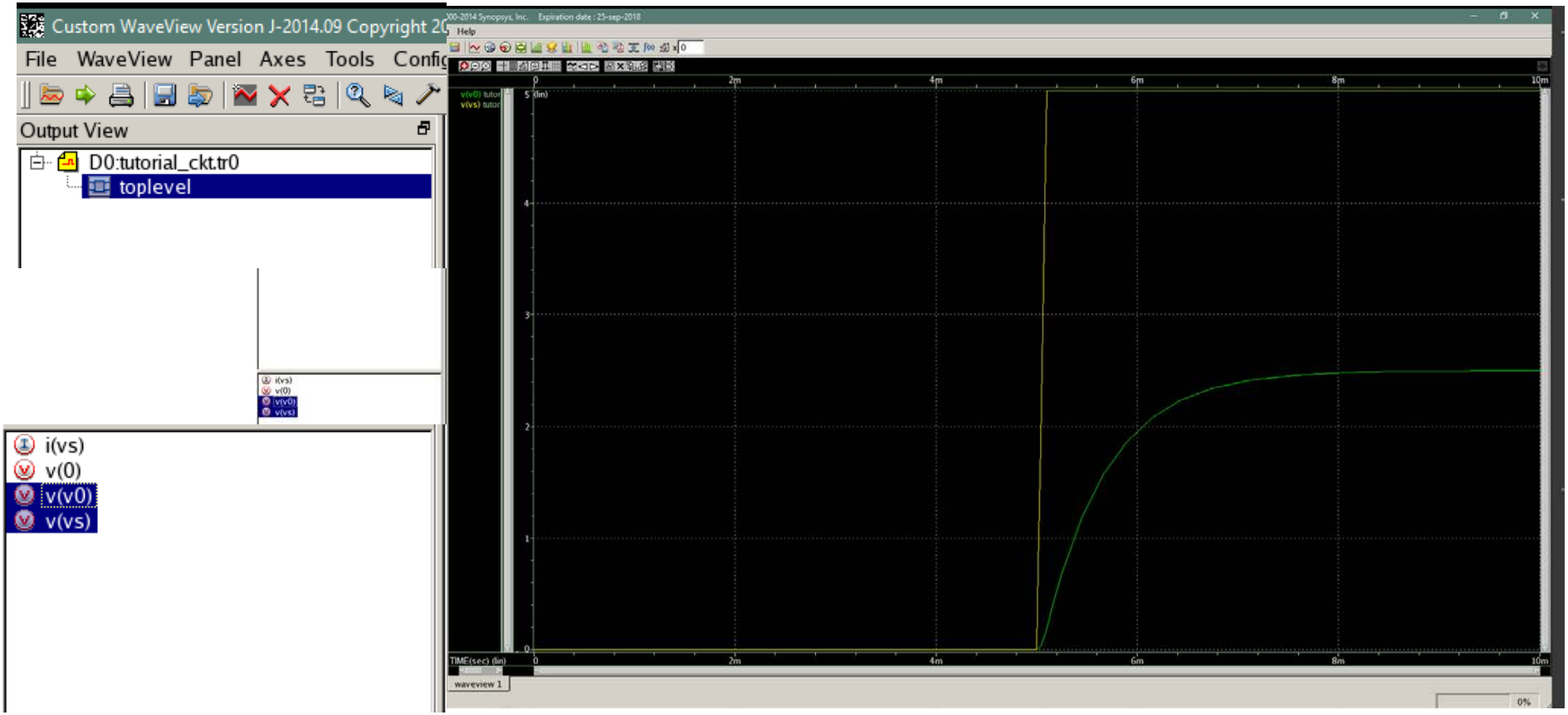
```
awaves &
```

- This will run the *awaves* software and leave your terminal free to use
- Click “Open Waveform File” & navigate to the file called `mycircuit.tr0`



Viewing Simulation Results

- This will open up the Custom WaveView window, where you can add traces to view



Beyond the Basics

- Can perform many different type of analyses
  - AC (`.ac`), DC (`.dc`), transfer function (`.tf`), DC operating point (`.op`)
- For nonlinear devices (diodes, MOSFETs), must define a model to specify device parameters
  - Model names cannot start w/ a number!

```
1  EE105 SPICE Tutorial Example 2 - Simple Diode Circuit
2  .model tut_diode d (is=1e-14 vj=0.6 rs=10)
3  vs vs gnd 5V
4  rs vs vd  5k
5  d1 vd gnd tut_diode
6  .op
7  .end
```

Non-ideal Op Amps | Finite Gain

- Find an expression for the closed loop gain, G assuming non-infinite open loop gain, A

$G \rightarrow v_o/v_i$

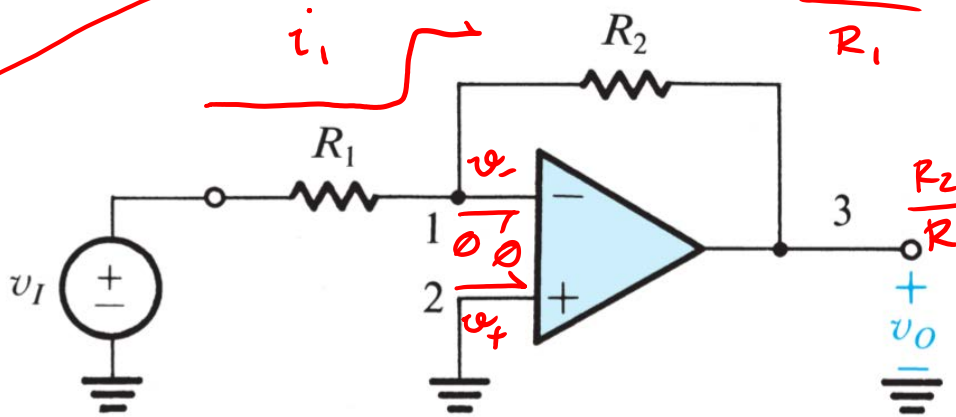
$A_{cl,ideal} = -R_2/R_1$

$v_o = A(v_+ - v_-)$

$v_o = A(0 - v_-)$

$v_- = -\frac{v_o}{A}$

$i_1 = \frac{(v_i - v_-)}{R_1} = \frac{(v_- - v_o)}{R_2}$



$\frac{R_2}{R_1}\left(v_i + \frac{v_o}{A}\right) = -v_o\left(1 + \frac{1}{A}\right)$

$\frac{R_2}{R_1} v_i = -v_o\left(1 + \frac{1}{A} + \frac{R_2}{R_1}\cdot\frac{1}{A}\right)$

$$\frac{v_o}{v_i} = \frac{(-R_2/R_1)}{1 + \frac{(R_2/R_1 + 1)}{A}}$$

as $A \rightarrow \infty$

$G \rightarrow -R_2/R_1$

Non-ideal Op Amps | Finite Gain

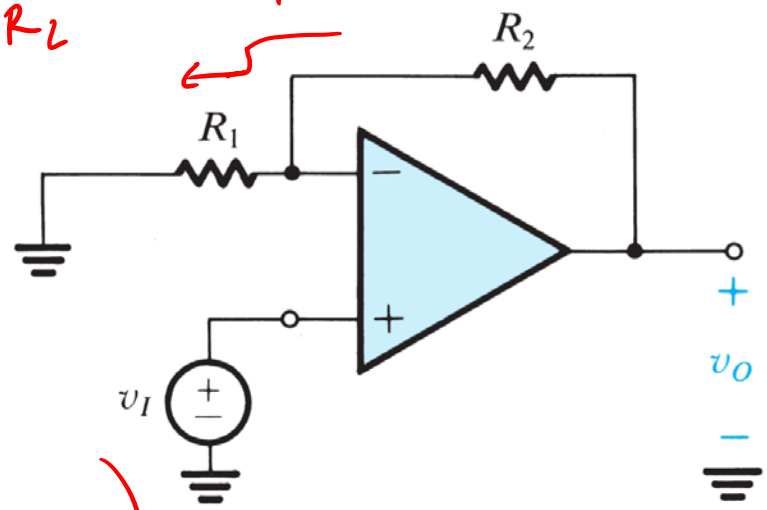
- Find an expression for the closed loop gain, G assuming non-infinite open loop gain, A

$$v_o = (v_+ - v_-)A \quad \{v_+ = v_i\} \Longrightarrow v_o = (v_i - v_-)A$$

$$\frac{v_-}{R_1} = i_1 = \frac{v_o - v_-}{R_2}$$

$$v_- = v_i - \frac{v_o}{A}$$

$$v_- = \frac{v_o}{(1 + R_2/R_1)}$$



$$v_i - \frac{v_o}{A} = \frac{v_o}{1 + R_2/R_1}$$

$$v_i = v_o\left(\frac{1}{A} + \frac{1}{1 + R_2/R_1}\right)$$

$$\frac{v_o}{v_i} = \frac{(1 + R_2/R_1)}{1 + \frac{(1 + R_2/R_1)}{A}}$$

as $A \to \infty$

$$G \to 1 + \frac{R_2}{R_1}$$

Non-ideal Op Amps | Finite Gain

- If $R_1 = 1k\Omega$, $R_2 = 9k\Omega$, and $A = 10^3$, find the percent deviation from the ideal case