EE5780 Advanced VLSI CAD

Lecture 7 Modified Nodal Analysis and SPICE Simulation

Zhuo Feng

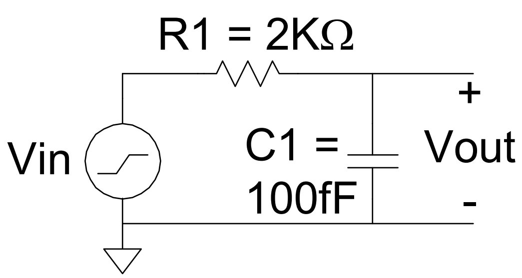
Introduction to SPICE

- **Simulation Program with Integrated Circuit Emphasis**
  - Developed in 1970's at Berkeley
  - Many commercial versions are available
  - HSPICE is a robust industry standard
    - Has many enhancements that we will use
- **Written in FORTRAN for punch-card machines**
  - Circuits elements are called *cards*
  - Complete description is called a SPICE *deck*

Writing Spice Decks

- **Writing a SPICE deck is like writing a good program**
  - Plan: sketch schematic on paper or in editor
    - Modify existing decks whenever possible
  - Code: strive for clarity
    - Start with name, email, date, purpose
    - Generously comment
  - Test:
    - Predict what results should be
    - Compare with actual
    - *Garbage In, Garbage Out!*

Example: RC Circuit

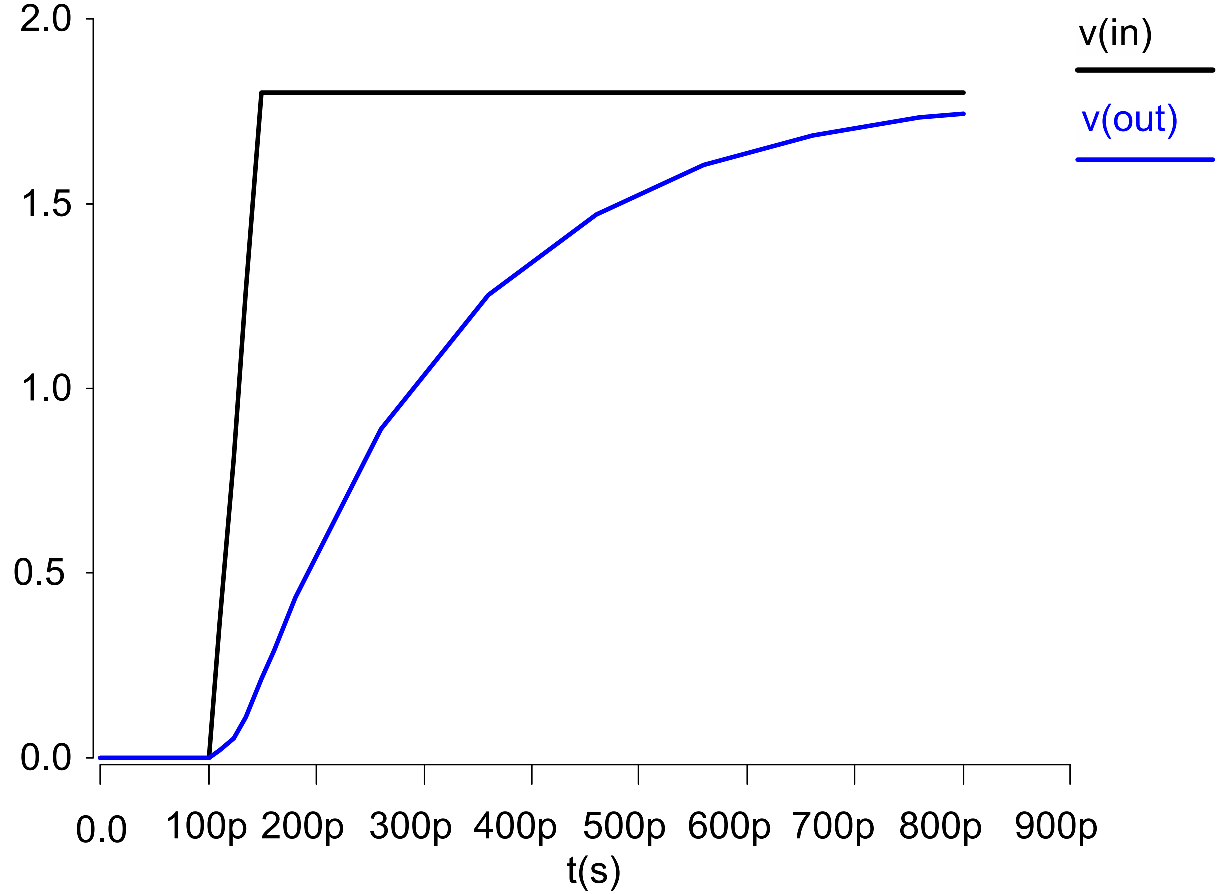



```
* rc.sp
* David_Harris@hmc.edu 2/2/03
* Find the response of RC circuit to rising input

*------------------------------------------------
* Parameters and models
*------------------------------------------------
.option post

*------------------------------------------------
* Simulation netlist
*------------------------------------------------
Vin     in      gnd     pwl     0ps 0 100ps 0 150ps 1.8 800ps 1.8
R1      in      out     2k
C1      out     gnd     100f

*------------------------------------------------
* Stimulus
*------------------------------------------------
.tran 20ps 800ps
.plot v(in) v(out)
.end
```

Result (Textual)

```
legend:
  a: v(in)
  b: v(out)

      time       v(in)
 (ab      )       0.          500.0000m       1.0000        1.5000       2.0000
                      +             +             +             +             +
    0.         0.    -2------+------+------+------+------+------+------+------+-
   20.0000p    0.     2      +      +      +      +      +      +      +      +
   40.0000p    0.     2      +      +      +      +      +      +      +      +
   60.0000p    0.     2      +      +      +      +      +      +      +      +
   80.0000p    0.     2      +      +      +      +      +      +      +      +
  100.0000p    0.     2      +      +      +      +      +      +      +      +
  120.0000p  720.000m +b     +      +     a+      +      +      +      +      +
  140.0000p    1.440  +   b  +      +      +      +      +    a +      +      +
  160.0000p    1.800  +      +b     +      +      +      +      +      +a     +
  180.0000p    1.800  +      +    b +      +      +      +      +      +a     +
  200.0000p    1.800 -+------+------+b-----+------+------+------+------+a-----+-
  220.0000p    1.800  +      +      +    b +      +      +      +      +a     +
  240.0000p    1.800  +      +      +      +b     +      +      +      +a     +
  260.0000p    1.800  +      +      +      +   b  +      +      +      +a     +
  280.0000p    1.800  +      +      +      +     b+      +      +      +a     +
  300.0000p    1.800  +      +      +      +      +b     +      +      +a     +
  320.0000p    1.800  +      +      +      +      +  b   +      +      +a     +
  340.0000p    1.800  +      +      +      +      +    b +      +      +a     +
  360.0000p    1.800  +      +      +      +      +      b      +      +a     +
  380.0000p    1.800  +      +      +      +      +      +b     +      +a     +
  400.0000p    1.800 -+------+------+------+------+------+--b---+------+a-----+-
  420.0000p    1.800  +      +      +      +      +      +   b  +      +a     +
  440.0000p    1.800  +      +      +      +      +      +    b +      +a     +
  460.0000p    1.800  +      +      +      +      +      +     b+      +a     +
  480.0000p    1.800  +      +      +      +      +      +      b      +a     +
  500.0000p    1.800  +      +      +      +      +      +      +b     +a     +
  520.0000p    1.800  +      +      +      +      +      +      +b     +a     +
  540.0000p    1.800  +      +      +      +      +      +      + b    +a     +
  560.0000p    1.800  +      +      +      +      +      +      +  b   +a     +
  580.0000p    1.800  +      +      +      +      +      +      +  b   +a     +
  600.0000p    1.800 -+------+------+------+------+------+------+---b--+a-----+-
  620.0000p    1.800  +      +      +      +      +      +      +   b  +a     +
  640.0000p    1.800  +      +      +      +      +      +      +    b +a     +
  660.0000p    1.800  +      +      +      +      +      +      +    b +a     +
  680.0000p    1.800  +      +      +      +      +      +      +    b +a     +
  700.0000p    1.800  +      +      +      +      +      +      +     b+a     +
  720.0000p    1.800  +      +      +      +      +      +      +     b+a     +
  740.0000p    1.800  +      +      +      +      +      +      +     b+a     +
  760.0000p    1.800  +      +      +      +      +      +      +     b+a     +
  780.0000p    1.800  +      +      +      +      +      +      +      ba     +
  800.0000p    1.800 -+------+------+------+------+------+------+------ba-----+-
                      +             +             +             +             +
```

Result (Graphical)



Sources

- *DC* Source

```
Vdd vdd gnd 2.5
```

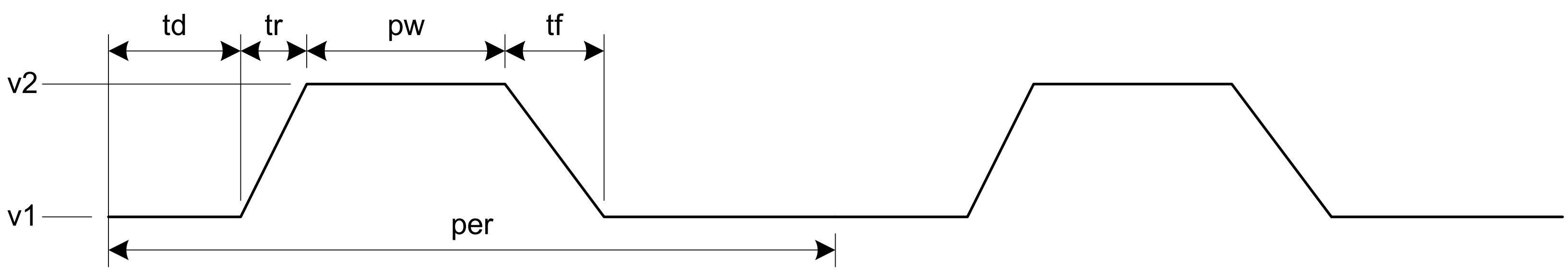
- *Piecewise Linear* Source

```
Vin in gnd pwl 0ps 0 100ps 0 150ps 1.8 800ps 1.8
```

- *Pulsed* Source

```
Vck clk gnd PULSE 0 1.8 0ps 100ps 100ps 300ps 800ps
```

```
PULSE v1 v2 td tr tf pw per
```



SPICE Elements

Letter	Element
R	Resistor
C	Capacitor
L	Inductor
K	Mutual Inductor
V	Independent voltage source
I	Independent current source
M	MOSFET
D	Diode
Q	Bipolar transistor
W	Lossy transmission line
X	Subcircuit
E	Voltage-controlled voltage source
G	Voltage-controlled current source
H	Current-controlled voltage source
F	Current-controlled current source

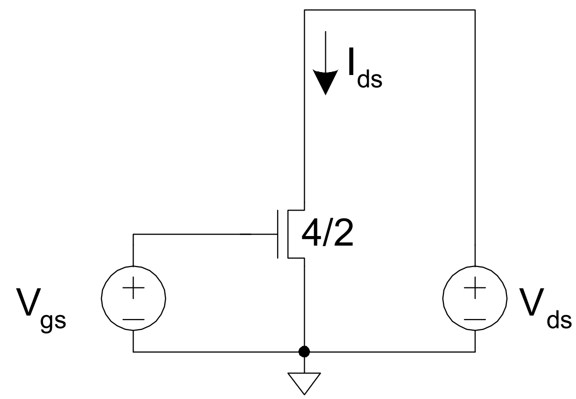
Units

Letter	Unit	Magnitude
a	atto	10^{-18}
f	fempto	10^{-15}
p	pico	10^{-12}
n	nano	10^{-9}
u	micro	10^{-6}
m	mili	10^{-3}
k	kilo	10^{3}
x	mega	10^{6}
g	giga	10^{9}

Ex: 100 femptofarad capacitor = 100fF, 100f, 100e-15

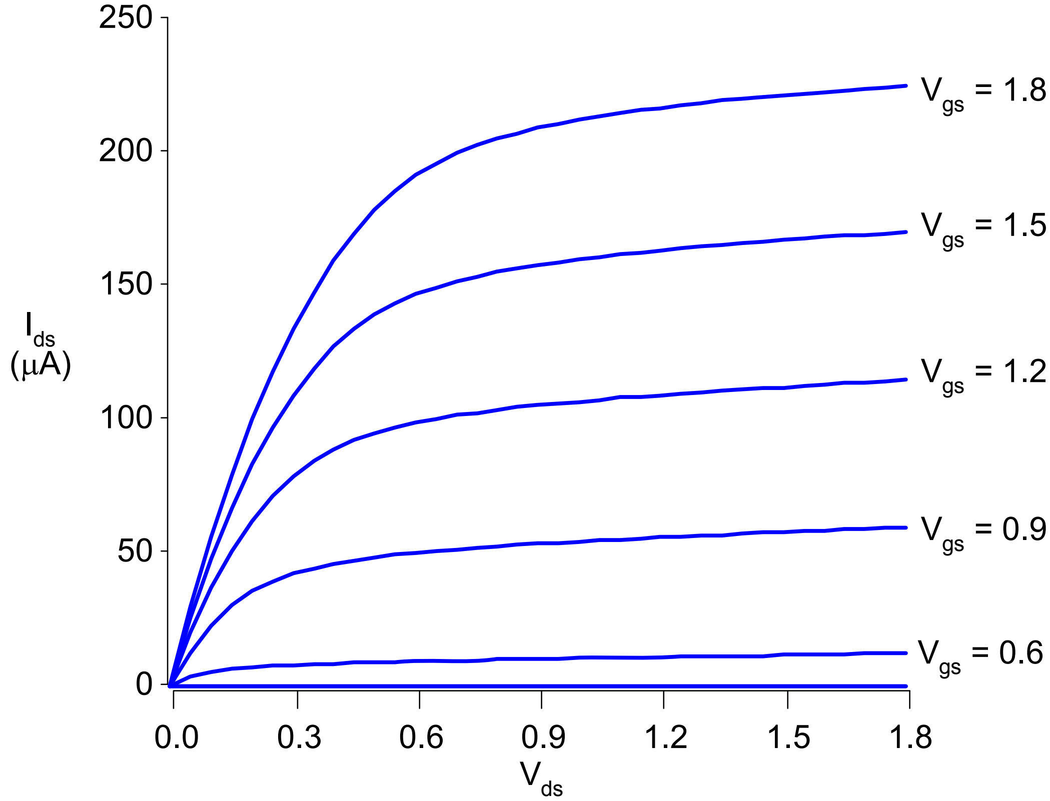
DC Analysis

```
* mosiv.sp

*------------------------------------------------
* Parameters and models
*------------------------------------------------
.include '../models/tsmc180/models.sp'
.temp 70
.option post

*------------------------------------------------
* Simulation netlist
*------------------------------------------------
*nmos
Vgs     g     gnd    0
Vds     d     gnd    0
M1      d     g      gnd    gnd    NMOS    W=0.36u  L=0.18u

*------------------------------------------------
* Stimulus
*------------------------------------------------
.dc Vds 0 1.8 0.05 SWEEP Vgs 0 1.8 0.3
.end
```



I-V Characteristics

- **NMOS I-V**
  - V_{gs} dependence
  - Saturation



MOSFET Elements

M element for MOSFET

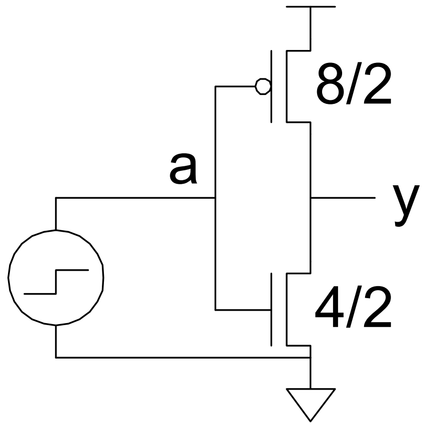
Mname drain gate source body type

+ W=<width> L=<length>

+ AS=<area source> AD = <area drain>

+ PS=<perimeter source> PD=<perimeter drain>

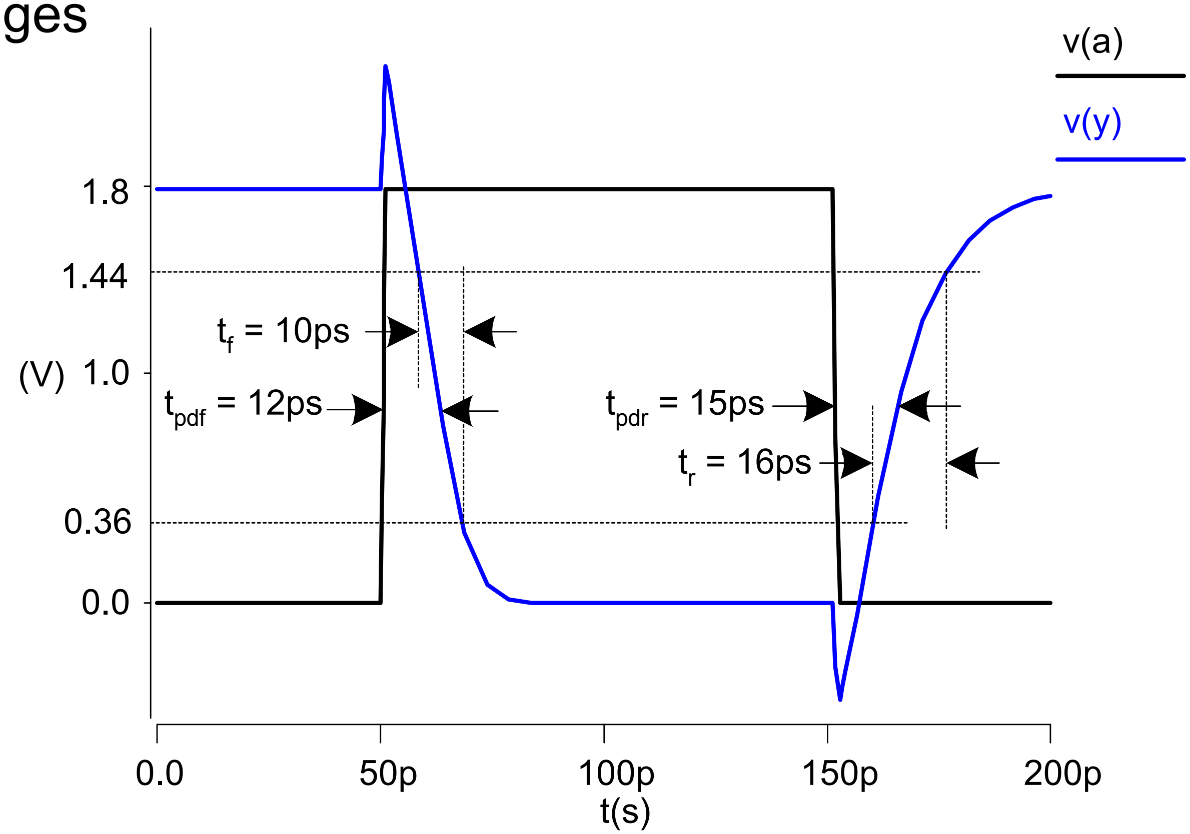
Transient Analysis




```
* inv.sp

* Parameters and models
*----------------------------------------------------
.param SUPPLY=1.8
.option scale=90n
.include '../models/tsmc180/models.sp'
.temp 70
.option post

* Simulation netlist
*----------------------------------------------------
Vdd     vdd     gnd     'SUPPLY'
Vin     a       gnd     PULSE   0 'SUPPLY' 50ps 0ps 0ps 100ps 200ps
M1      y       a       gnd     gnd     NMOS    W=4     L=2
+ AS=20 PS=18 AD=20 PD=18
M2      y       a       vdd     vdd     PMOS    W=8     L=2
+ AS=40 PS=26 AD=40 PD=26

* Stimulus
*----------------------------------------------------
.tran 1ps 200ps
.end
```

Transient Results

- **Unloaded inverter**
  - Overshoot
  - Very fast edges

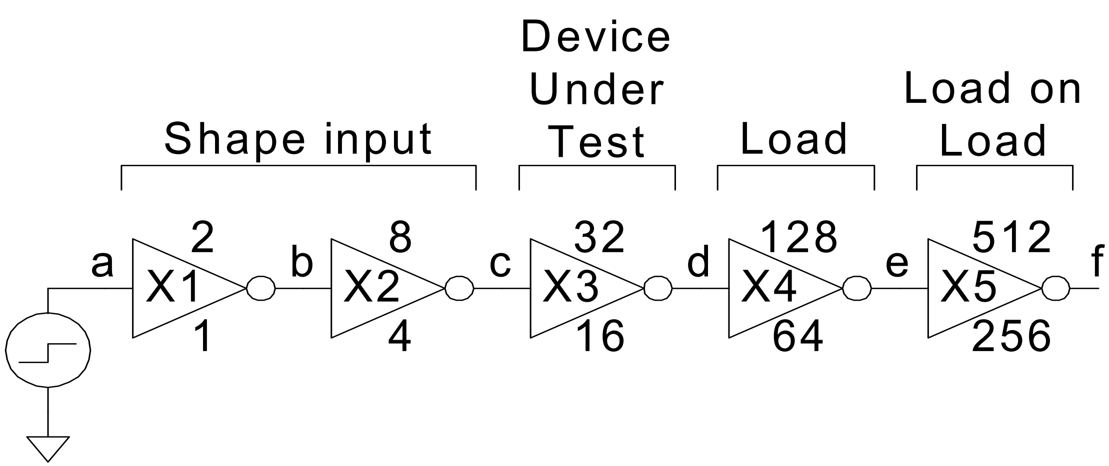


Subcircuits

- **Declare common elements as subcircuits**

```
.subckt inv a y N=4 P=8
M1 y a gnd gnd NMOS W='N' L=2
+ AS='N*5' PS='2*N+10' AD='N*5' PD='2*N+10'
M2 y a vdd vdd PMOS W='P' L=2
+ AS='P*5' PS='2*P+10' AD='P*5' PD='2*P+10'
.ends
```

- **Ex: Fanout-of-4 Inverter Delay**
  - Reuse inv
  - Shaping
  - Loading



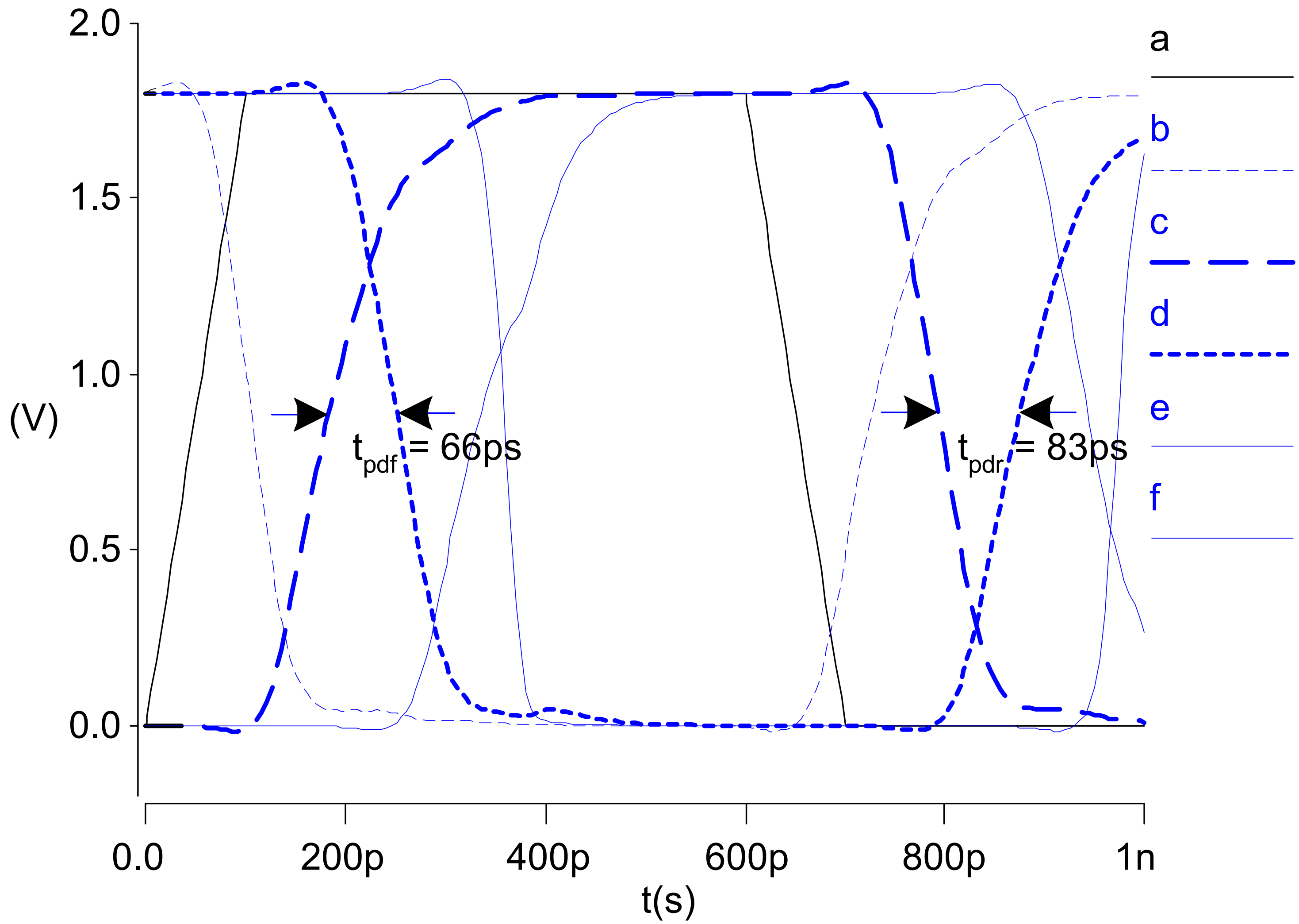
FO4 Inverter Delay

```
* fo4.sp

* Parameters and models
*----------------------------------------------------------------------
.param SUPPLY=1.8
.param H=4
.option scale=90n
.include '../models/tsmc180/models.sp'
.temp 70
.option post

* Subcircuits
*----------------------------------------------------------------------
.global vdd gnd
.include '../lib/inv.sp'

* Simulation netlist
*----------------------------------------------------------------------
Vdd     vdd     gnd     'SUPPLY'
Vin     a       gnd     PULSE   0 'SUPPLY' 0ps 100ps 100ps 500ps 1000ps
X1      a       b       inv                * shape input waveform
X2      b       c       inv     M='H'      * reshape input waveform
```

FO4 Inverter Delay Cont.

```
X3      c       d       inv     M='H**2' * device under test
X4      d       e       inv     M='H**3' * load
x5      e       f       inv     M='H**4' * load on load

* Stimulus
*-----------------------------------------------------------------------
.tran 1ps 1000ps
.measure tpdr                               * rising prop delay
+     TRIG v(c)  VAL='SUPPLY/2' FALL=1
+     TARG v(d)  VAL='SUPPLY/2' RISE=1
.measure tpdf                               * falling prop delay
+     TRIG v(c)  VAL='SUPPLY/2' RISE=1
+     TARG v(d)  VAL='SUPPLY/2' FALL=1
.measure tpd param='(tpdr+tpdf)/2'          * average prop delay
.measure trise                                      * rise time
+       TRIG v(d)       VAL='0.2*SUPPLY' RISE=1
+       TARG v(d)       VAL='0.8*SUPPLY' RISE=1
.measure tfall                                      * fall time
+       TRIG v(d)       VAL='0.8*SUPPLY' FALL=1
+       TARG v(d)       VAL='0.2*SUPPLY' FALL=1
.end
```

FO4 Results


2.0
1.5
1.0
0.5
0.0
(V)
t_{pdf} = 66ps
t_{pdr} = 83ps
a
b
c
d
e
f
0.0
200p
400p
600p
800p
1n
t(s)

Power Measurement

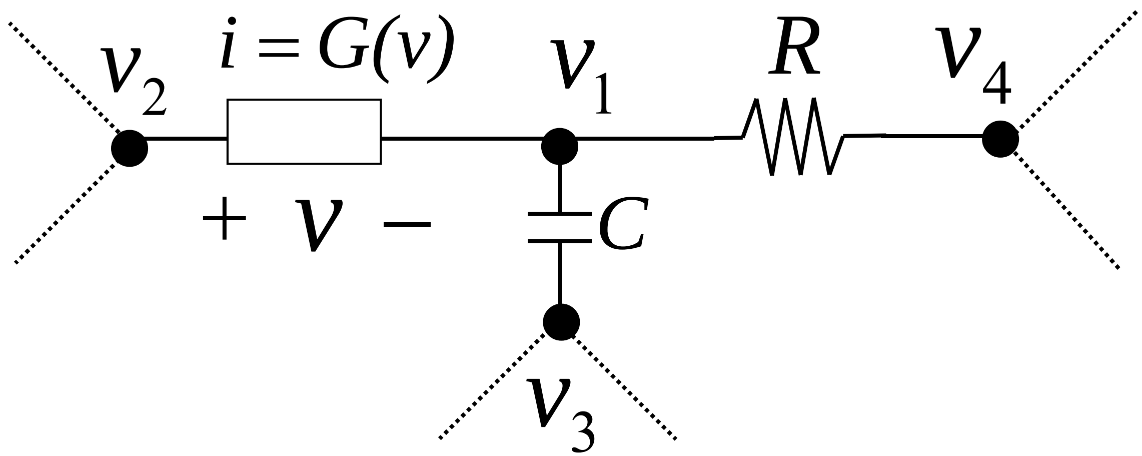
- **HSPICE can measure power**
  - ► Instantaneous P(t)
  - ► Or average P over some interval

```
.print P(vdd)
.measure pwr AVG P(vdd) FROM=0ns TO=10ns
```

- **Power in single gate**
  - ► Connect to separate V_{DD} supply
  - ► Be careful about input power

How is SPICE simulator created?

- N equations in terms of N unknown Node voltages
- More generally using modified nodal analysis



Time Domain Equations at node 1:

$$C\frac{d(v_1 - v_3)}{dt} + \frac{(v_1 - v_4)}{R} - G(v_2 - v_1) = 0$$

- If we do this for all N nodes:

$$F(\dot{\vec{x}}(t),\vec{x}(t),\vec{u}(t))=0 \qquad \vec{x}(0) = \vec{X}$$

$\vec{x}(t) =$ N dimensional vector of unknown node voltages

$\vec{u}(t) =$ vector of independent sources

$F =$ nonlinear operator

- Closed form solution is not possible for arbitrary order of differential equations

- We must approximate the solution of:

$$F(\dot{\vec{x}}(t), \vec{x}(t), \vec{u}(t)) = 0 \qquad \vec{x}(0) = \vec{X}$$

- This is facilitated in SPICE via numerical solutions

■ Basic circuit analyses

- ► (Nonlinear) DC analysis
  - ▼ Finds the DC operating point of the circuit
  - ▼ Solves a set of nonlinear algebraic eqns

- ► AC analysis
  - ▼ Performs frequency-domain small-signal analysis
  - ▼ Require a preceding DC analysis
  - ▼ Solves a set of complex linear eqns

- ► (Nonlinear) transient analysis
  - ▼ Computes the time-domain circuit transient response
  - ▼ Solves a set of nonlinear different eqns
  - ▼ Converts to a set nonlinear algebraic of eqns using numerical integration

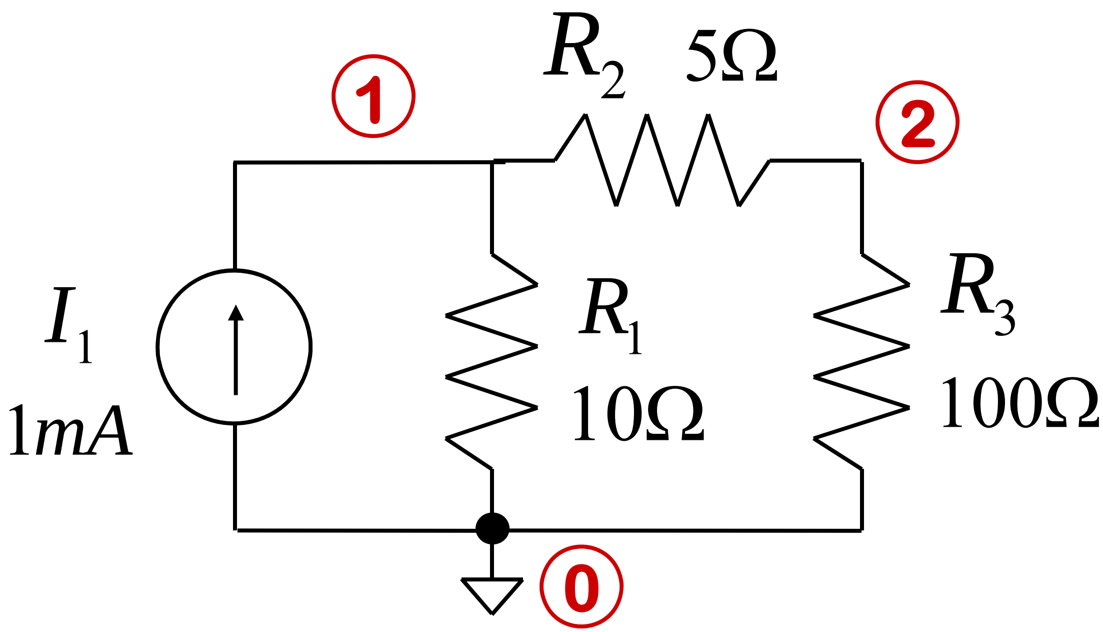
- **SPICE offers practical techniques to solve circuit problems in time & freq. domains**
  - Interface to device models
    - Transistors, diodes, nonlinear caps etc
  - Sparse linear solver
  - Nonlinear solver – Newton-Raphson method
  - Numerical integration
  - Convergence & time-step control

- Circuit equations are usually formulated using
  - ▶ Nodal analysis
    - ▼ N equations in N nodal voltages
  - ▶ Modified analysis
    - ▼ Circuit unknowns are nodal voltages & some branch currents
    - ▼ Branch current variables are added to handle
      - Voltages sources
      - Inductors
      - Current controlled voltage source etc
- Formulations can be done in both time and frequency

How do we set up a matrix problem given a list of linear(ized) circuit elements?

Similar to reading a netlist for a linear circuit:

* Element Name	From	To	Value
I_1	0	1	$1mA$
R_1	1	0	10Ω
R_2	1	2	5Ω
R_3	2	0	100Ω



The nodal analysis matrix equations are easily constructed via KCL at each node:

$$Y\vec{v} = \vec{J}$$

- **Naïve approach**
  - a) Write down the KCL eqn for each node
  - b) Combine all of them to a get N eqns in N node voltages

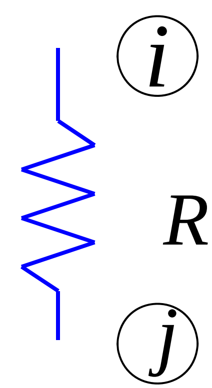
- **Intuitive for hand analysis**

- **Computer programs use a more convenient "element" centric approach**
  - Element stamps

Instead of converting the netlist into a graph and writing KCL eqns, *stamp* in elements one at a time:

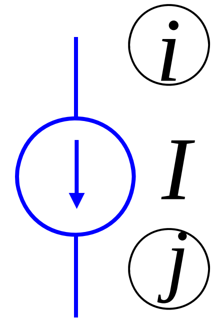
Stamps: add to existing matrix entries

$$
Y \quad
\begin{array}{cc}
\begin{bmatrix} \frac{1}{R} & -\frac{1}{R} \\ -\frac{1}{R} & \frac{1}{R} \end{bmatrix} &
\begin{array}{l} \leftarrow \text{From row } i \\ \leftarrow \text{To row } j \end{array} \\
\begin{array}{cc} \uparrow & \uparrow \\ \text{From col. } i & \text{To col. } j \end{array} &
\end{array}
\longleftrightarrow
$$



- RHS $\vec{J}$ of equations are stamped in a similar way:

$$\begin{array}{ll} \begin{bmatrix} -I \\ I \end{bmatrix} & \begin{array}{l} \leftarrow \text{From row } i \\ \leftarrow \text{To row } j \end{array} \end{array} \longleftrightarrow$$

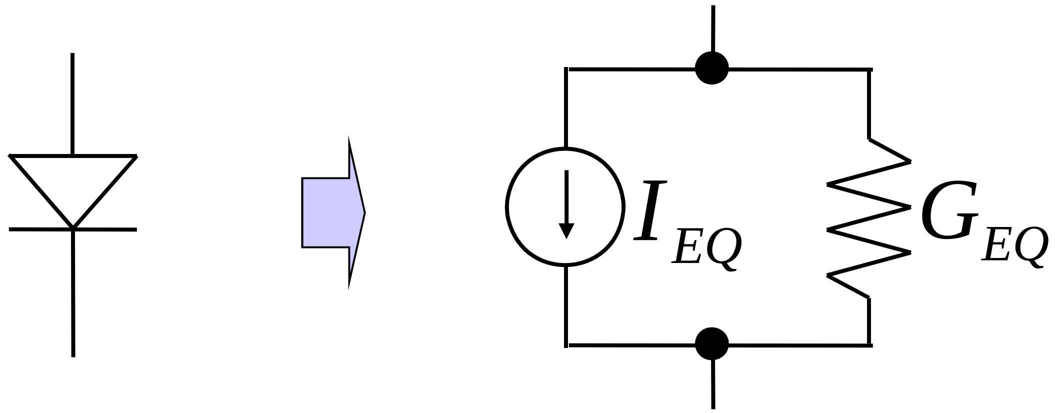


- Stamping our simple example one element at a time:

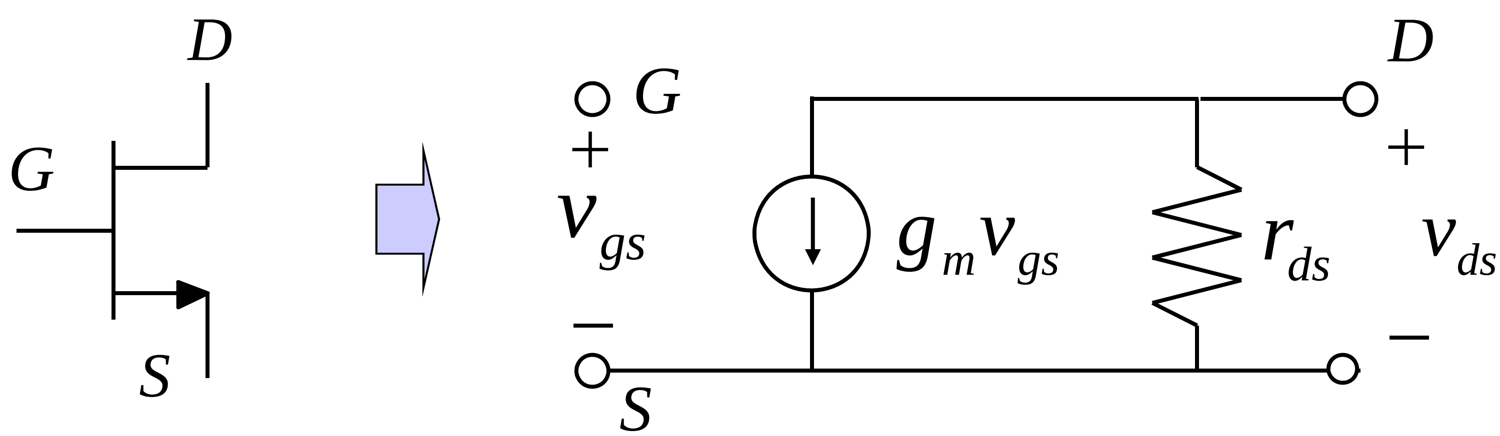
I_1	0	1	$1\,mA$
R_1	1	0	$10\ \Omega$
R_2	1	2	$5\,\Omega$
R_3	2	0	$100\ \Omega$

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

- We know that nonlinear elements are first converted to linear components, then stamped

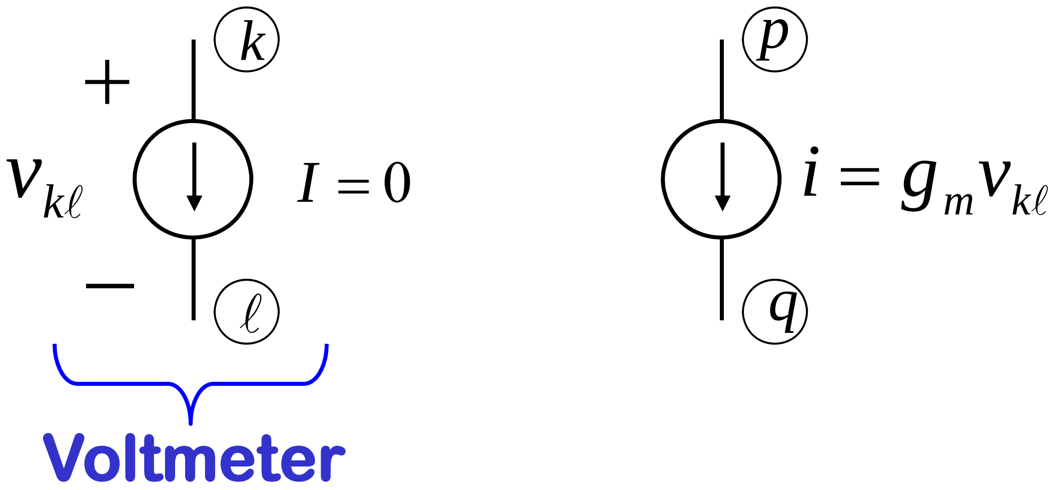


- For 3 & 4 terminal elements we know that the linearized models have linear controlled sources



- We can stamp in MOSFETs in terms of a complete stamp, or in terms of simpler element stamps

Voltage controlled current source

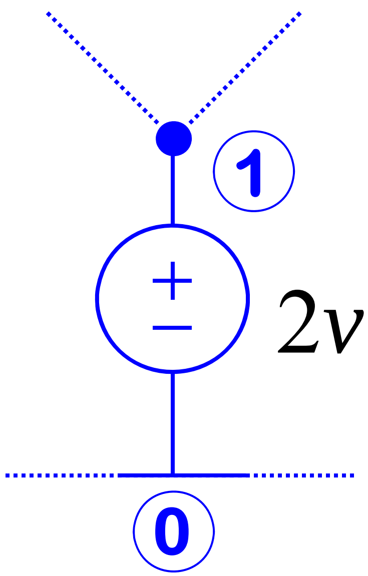


$$
\begin{array}{cc}
\begin{bmatrix} g_m & -g_m \\ -g_m & g_m \end{bmatrix} & \begin{array}{l} \leftarrow \textbf{row } p \\ \leftarrow \textbf{row } q \end{array} \\
\begin{array}{cc} \uparrow & \uparrow \\ \textbf{col } k & \textbf{col } \ell \end{array} &
\end{array}
$$

Large value that does not fall on diagonal of Y!

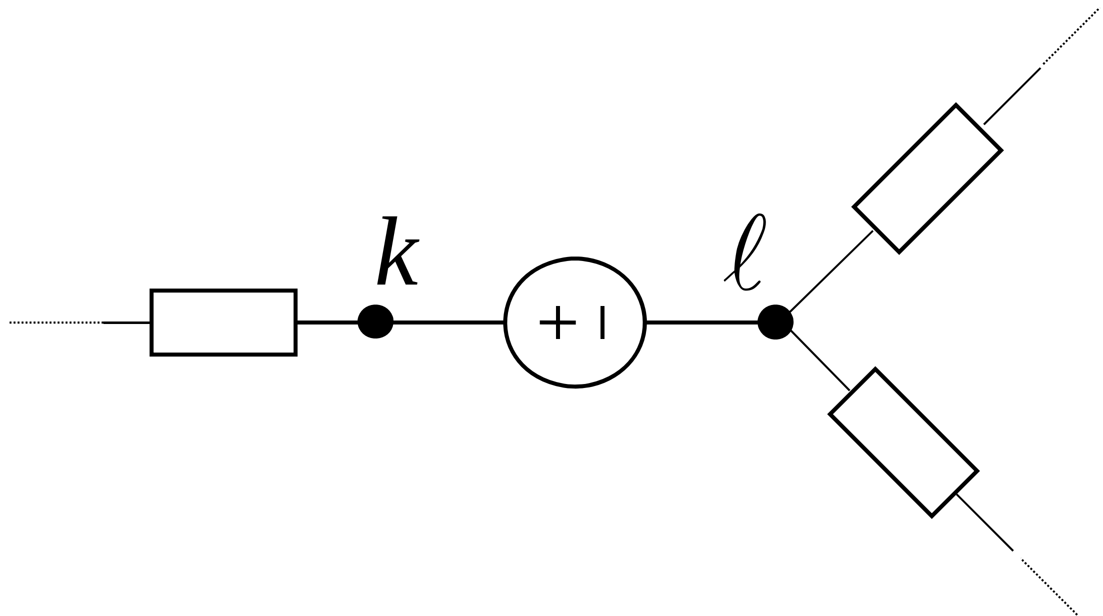
- All other types of controlled sources include voltage sources

- Voltage sources are inherently incompatible with nodal analysis
- Grounded voltages sources are easily accommodated



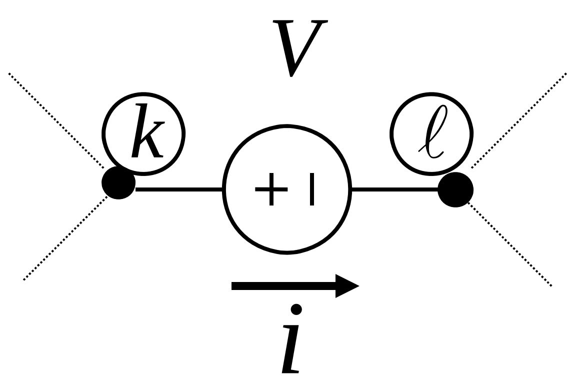
$$\begin{bmatrix} 1 & \cdots & \\ \vdots & & \\ & & \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} 2 \\ \vdots \\ \end{bmatrix}$$

- But a voltage source in between nodes is more difficult



- Node voltages k and ℓ are not independent

- We no longer have N independent node voltage variables
- So we can potentially eliminate one equation and one variable (section 2.3 of reference [1])
- But the more popular solution is <u>modified</u> nodal analysis (MNA)



Create one <u>extra</u> variable and one extra equation

- **Extra variable: voltage source current**
- **Allows us to write KCL at nodes** k **and** ℓ
- **Extra equation**

$$v_k - v_\ell = V$$

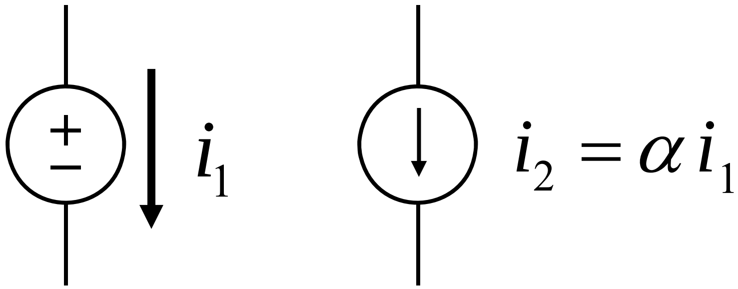
- **Advantage: now have an easy way of printing current results - - ammeter**

Voltage source stamp:

$$
\begin{array}{ll}
\textbf{row } k \\
\textbf{row } \ell \\
\textbf{row } N+1
\end{array}
\left[\begin{array}{cc|c}
 & & 1 \\
 & & -1 \\
\hline
1 & -1 & 0
\end{array}\right]
\left[\begin{array}{c}
 \\
 \\
\hline
i
\end{array}\right]
=
\left[\begin{array}{c}
 \\
 \\
\hline
V
\end{array}\right]
$$

$$
\begin{array}{ccc}
\textbf{col} & \textbf{col} & \textbf{col} \\
k & \ell & N+1
\end{array}
$$

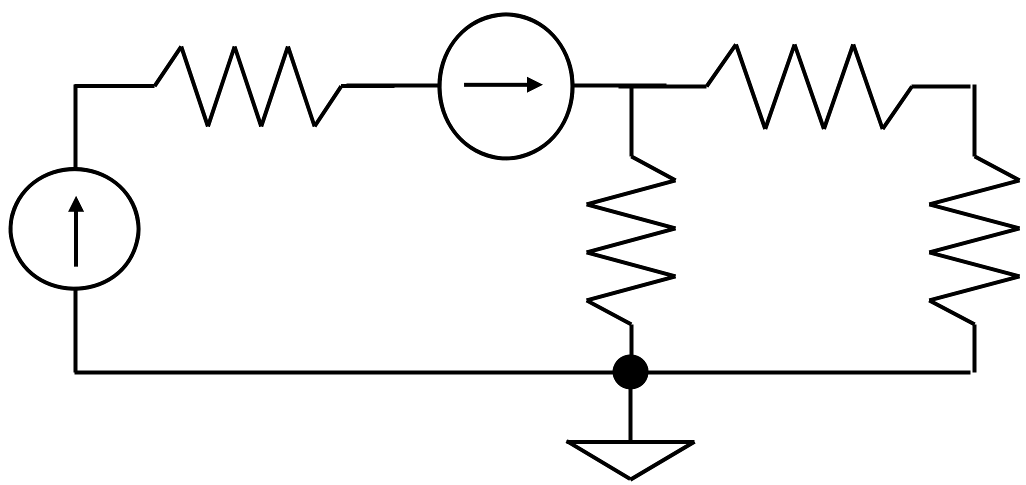
- **Current-controlled current source (e.g. BJT) has to stamp in an ammeter and a controlled current source**



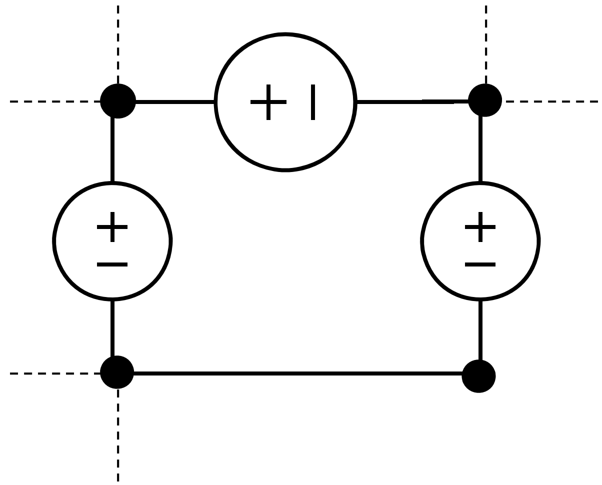
In general, we would not blindly build the matrix from an input netlist and then attempt to solve it

Various illegal ckts are possible:

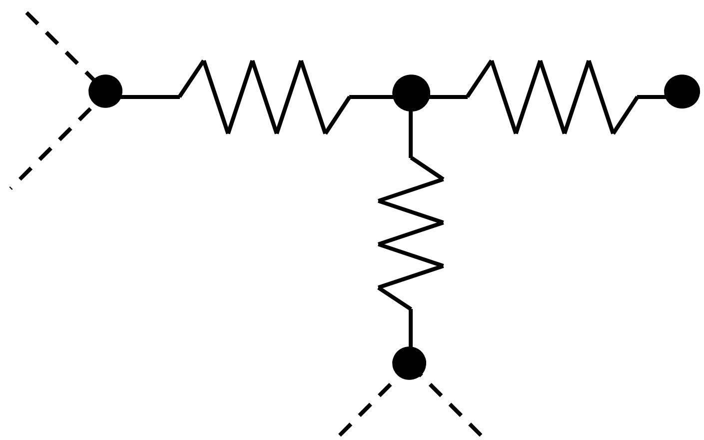
Cutsets of current sources



Loops of voltage sources



Dangling nodes



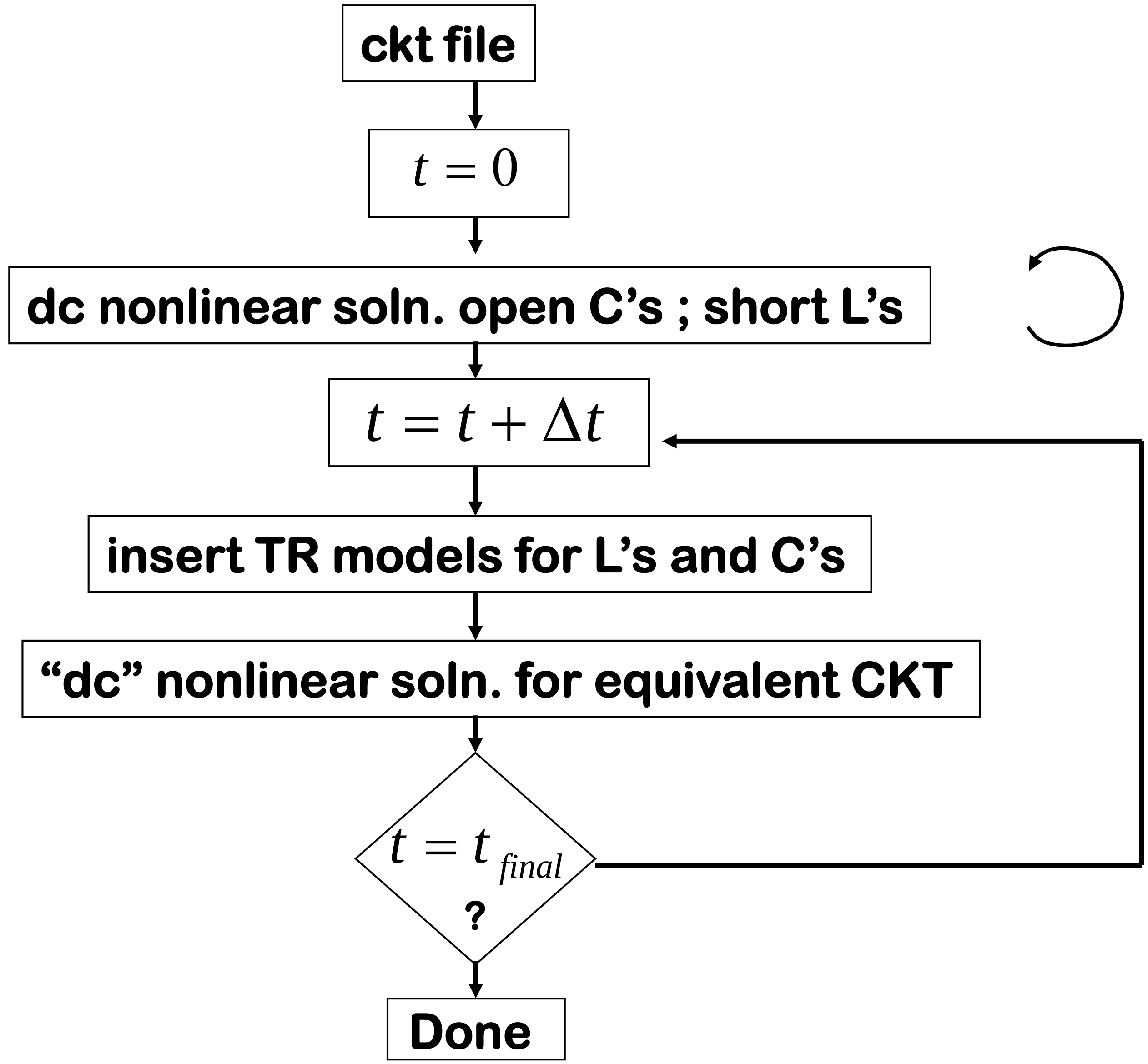
- **Once we efficiently formulate MNA equations, an efficient solution to $Y\vec{v} = \vec{J}$ is even more important**

- **For large ckts the matrix is really sparse**
  - Number of entries in Y is a function of number of elements connected to the corresponding node

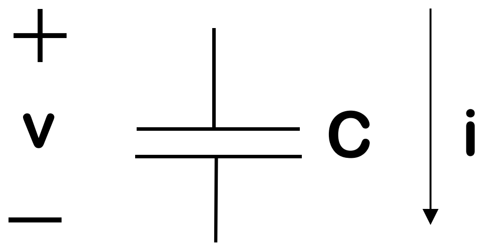
- **Inverting a sparse matrix is never a good idea since the inverse is not sparse!**

- **Instead direct solution methods employ Gaussian Elimination or LU factorization**

- TR analysis flow (Chap. 4 of ref. 2)



■ One-step integration approximation

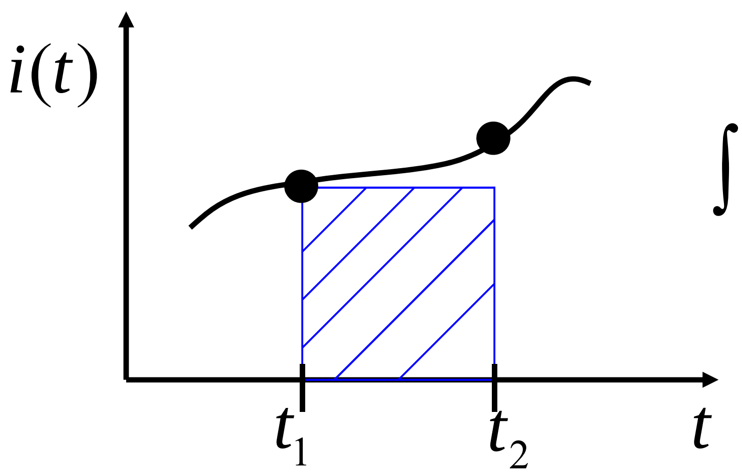


$$i = C\frac{dv}{dt}$$

$$v(t+\Delta t) = v(t) + \frac{1}{C}\int_{t}^{t+\Delta t} i(\tau)d\tau$$

$$\int_{i}^{t+\Delta t} i(\tau)d\tau \approx \begin{cases} \Delta t \cdot i(t) & \textbf{Forward Euler (FE)} \\ \Delta t \cdot i(t+\Delta t) & \textbf{Backward Euler (BE)} \\ \frac{\Delta t}{2}\left[i(t) + i(t+\Delta t)\right] & \textbf{Trapezoidal (TR)} \end{cases}$$

- **FE is explicit, and no nonlinear iterations are required**
  - ► Extremely difficult to use in practice



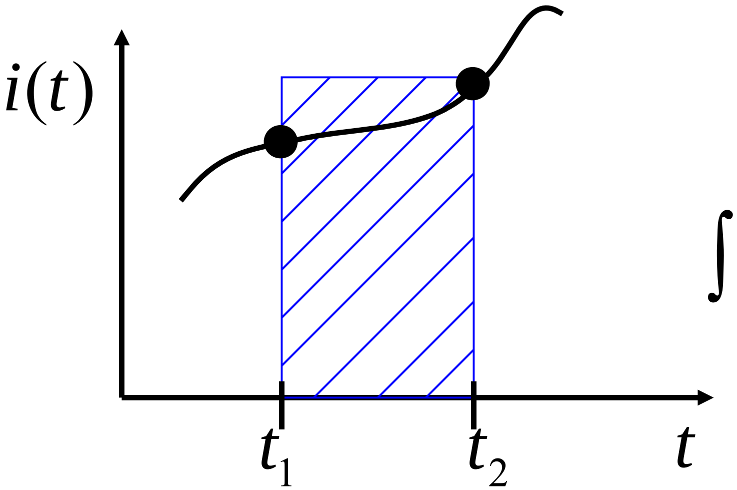
Forward Euler

$$\int_{t_1}^{t_2} i(t)dt \cong \Delta t \cdot i(t_1)$$

$$\Delta t = t_2 - t_1$$

BE is implicit

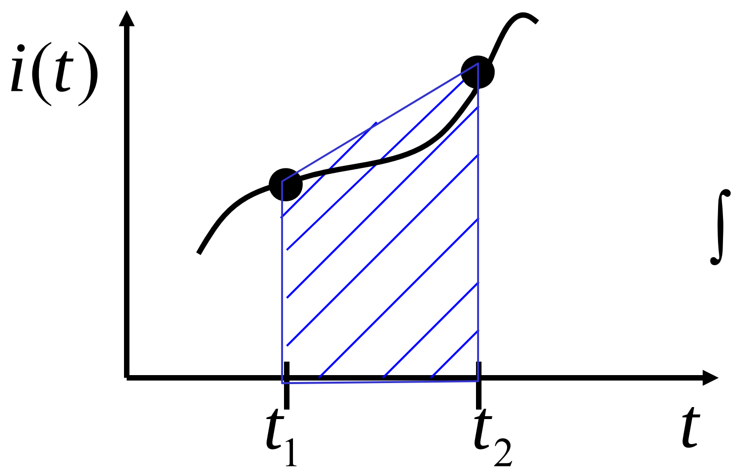
- Much more robust than FE
- Can also make unstable responses appear stable



Backward Euler

$$\int_{t_1}^{t_2} i(t)dt \cong \Delta t \cdot i(t_2)$$

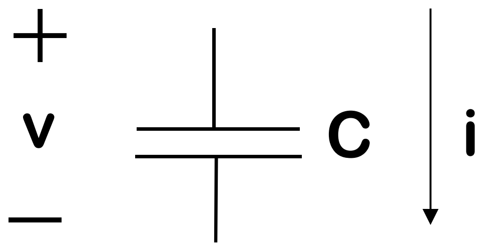
- **TR is implicit too**
  - Works similarly to BE
  - Incurs less error



Trapezoidal

$$\int_{t_1}^{t_2} i(t)dt \cong \frac{\Delta t}{2} \cdot \left(i(t_1) + i(t_2)\right)$$

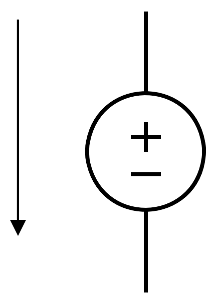
■ FE Capacitor Companion Model



$$i = C\frac{dv}{dt}$$

$$\int_{t}^{t+\Delta t} i(\tau)d\tau \approx \Delta t \cdot i(t)$$

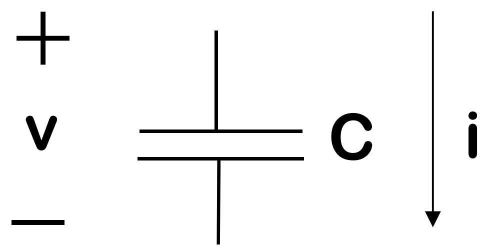
$i(t+\Delta t)$



$$v(t+\Delta t) \approx v(t) + \frac{\Delta t}{C} i(t)$$

BE Capacitor Companion Model

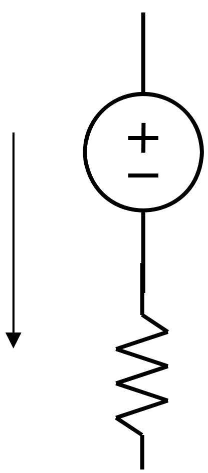
- Thevenin



$$i = C\frac{dv}{dt}$$

$$\int_{t}^{t+\Delta t} i(\tau)d\tau \approx \Delta t \cdot i(t+\Delta t)$$

$i(t+\Delta t)$

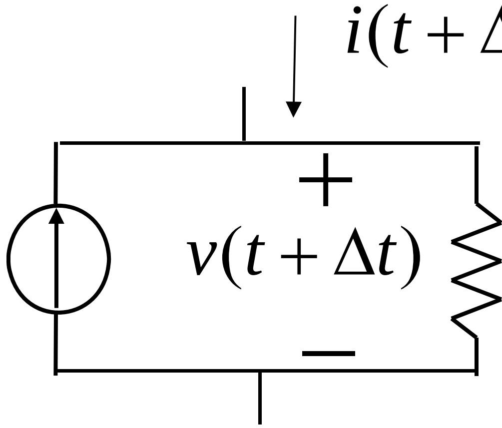


$$V_{eq} = v(t)$$

$$R_{eq} = \frac{\Delta t}{C}$$

BE Capacitor Companion Model

- Norton

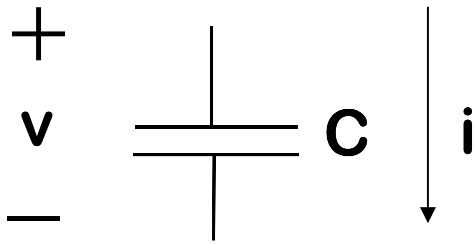


$$I_{eq} = \frac{C}{\Delta t} v(t)$$

$$G_{eq} = \frac{C}{\Delta t}$$

TR Capacitor Companion Model

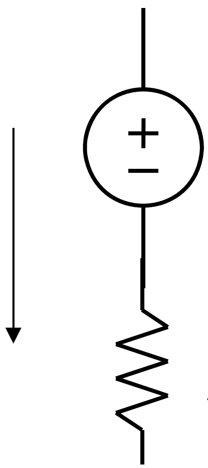
- Thevenin



$$i = C\frac{dv}{dt}$$

$$\int_{t}^{t+\Delta t} i(\tau)d\tau \approx \frac{\Delta t}{2}\cdot\left(i(t) + i(t+\Delta t)\right)$$

$$i(t+\Delta t)$$



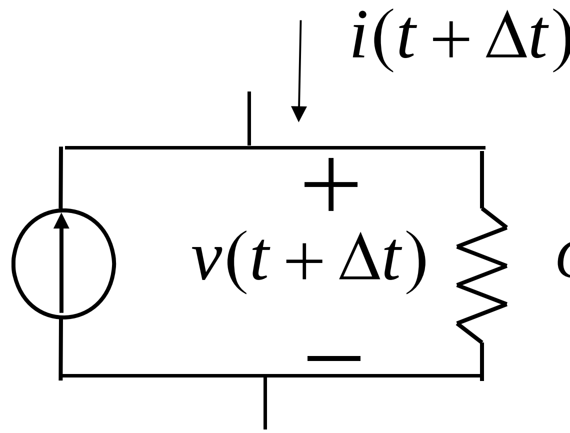
$$V_{eq} = v(t) + \frac{\Delta t}{2C} i(t)$$

$$R_{eq} = \frac{\Delta t}{2C}$$

TR Capacitor Companion Model

- Norton

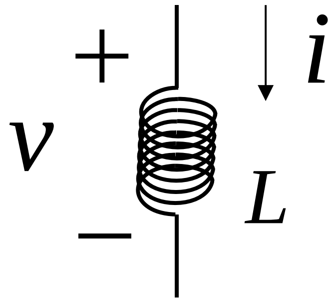
$$I_{eq} = \frac{2C}{\Delta t} v(t) + i(t)$$



$$G_{eq} = \frac{2C}{\Delta t}$$

TR Inductor Companion Model

- Norton

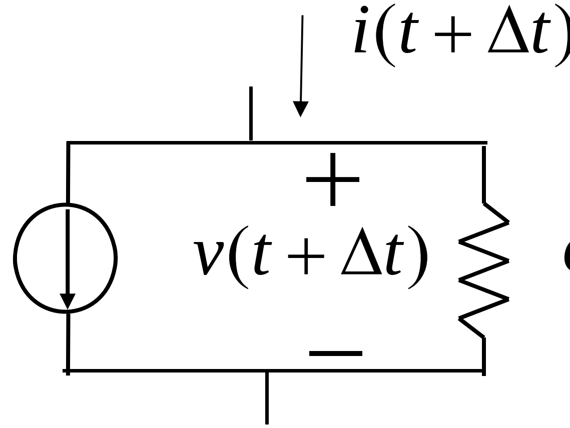


$$v = \frac{Ldi}{dt}$$

$$i(t+\Delta t) = i(t) + \frac{1}{L}\int v(\tau)d\tau$$

$$\int_{t}^{t+\Delta t} v(\tau)d\tau \approx \frac{\Delta t}{2}\cdot\big(v(t)+v(t+\Delta t)\big)$$

$$I_{eq} = \frac{\Delta t}{2L}v(t) + i(t)$$

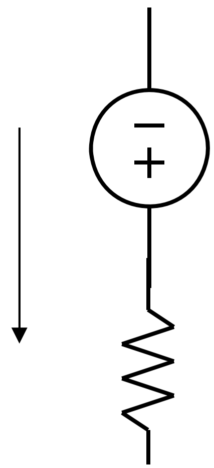


$$G_{eq} = \frac{\Delta t}{2L}$$

TR Inductor Companion Model

- Thevenin

$$i(t+\Delta t)$$

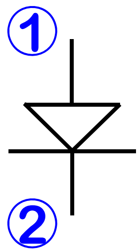


$$V_{eq} = v(t) + \frac{2L}{\Delta t} i(t)$$

$$R_{eq} = \frac{2L}{\Delta t}$$

- **Nonlinear DC analysis (Chapter 10.6 – 10.8 of Ref. 2)**
  - Store device equations and their partial derivatives w.r.t. branch voltages for efficient N-R procedure:
    - Insert linearized models into MNA formulation (first order Taylor series at operating point)
    - Solve the linear ckt to complete one N-R iteration
    - Use the solution as operating pt. for next linearization step

▶ Diode equations and companion model for N-R



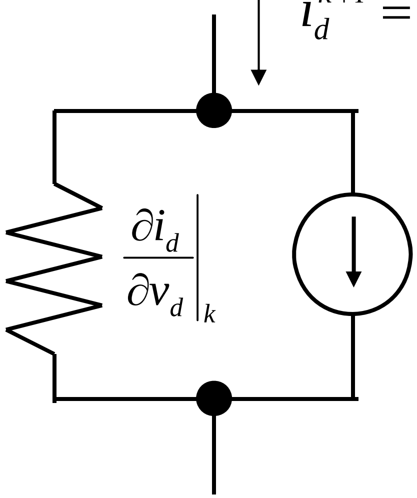
$$v_d = v_1 - v_2$$

$$i_d = I_s\left(e^{v_d/V_T} - 1\right)$$

$$\frac{\partial i_d}{\partial v_d} = \frac{I_s}{V_T} e^{v_d/V_T}$$

Stored Model Eqns.

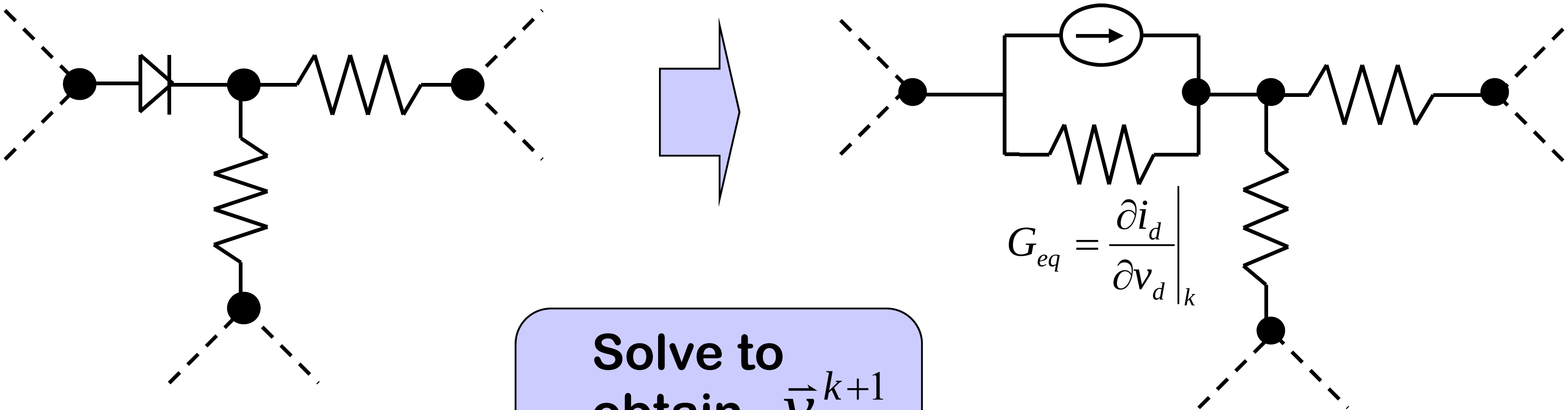
$$i_d^{k+1} = i_d^k + \left.\frac{\partial i_d}{\partial v_d}\right|_k \cdot \left(v_d^{k+1} - v_d^k\right)$$



$$i_d^k + \left.\frac{\partial i_d}{\partial v_d}\right|_k \left(-v_d^k\right)$$

- Diodes are modeled by Norton equivalent companion models

$$I_{eq} = i_d^K + \frac{\partial i_d}{\partial v_d}\bigg|_k (-v_d^k)$$



$$G_{eq} = \frac{\partial i_d}{\partial v_d}\bigg|_k$$

Solve to obtain $\vec{V}^{k+1}$ and so on…

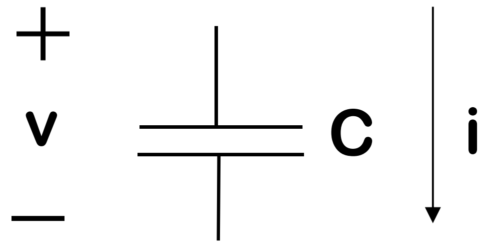
- Some sort of voltage limiting scheme is required to make N-R iterations robust

■ Recap: linear transient analysis

- ► Replace each of C's and L's by a companion model – numerical integration
  - ▼ Forward Euler, Backward Euler, Trapezoidal
  - ▼ Norton or Thevenin models
- ► Solve the equivalent linear circuit at the current time step
- ► Update all the companion models and move to the next time step

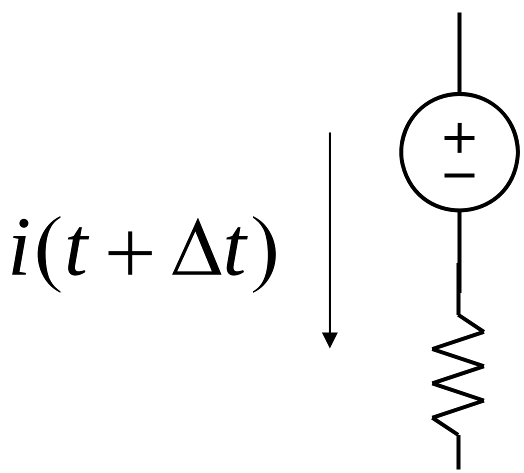
Example: BE capacitor companion models

- Thevenin



$$i = C\frac{dv}{dt}$$

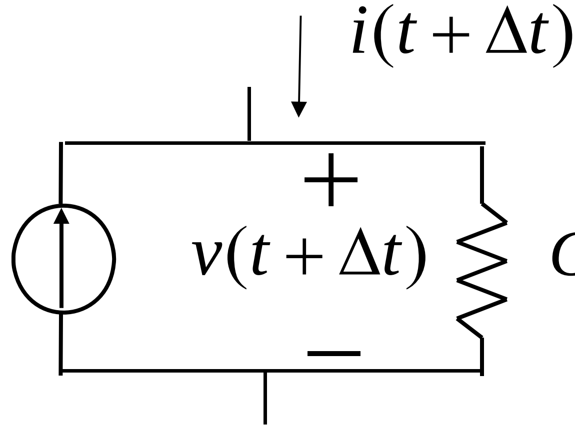
$$\int_{t}^{t+\Delta t} i(\tau)d\tau \approx \Delta t \cdot i(t+\Delta t)$$



$$V_{eq} = v(t)$$

$$R_{eq} = \frac{\Delta t}{C}$$

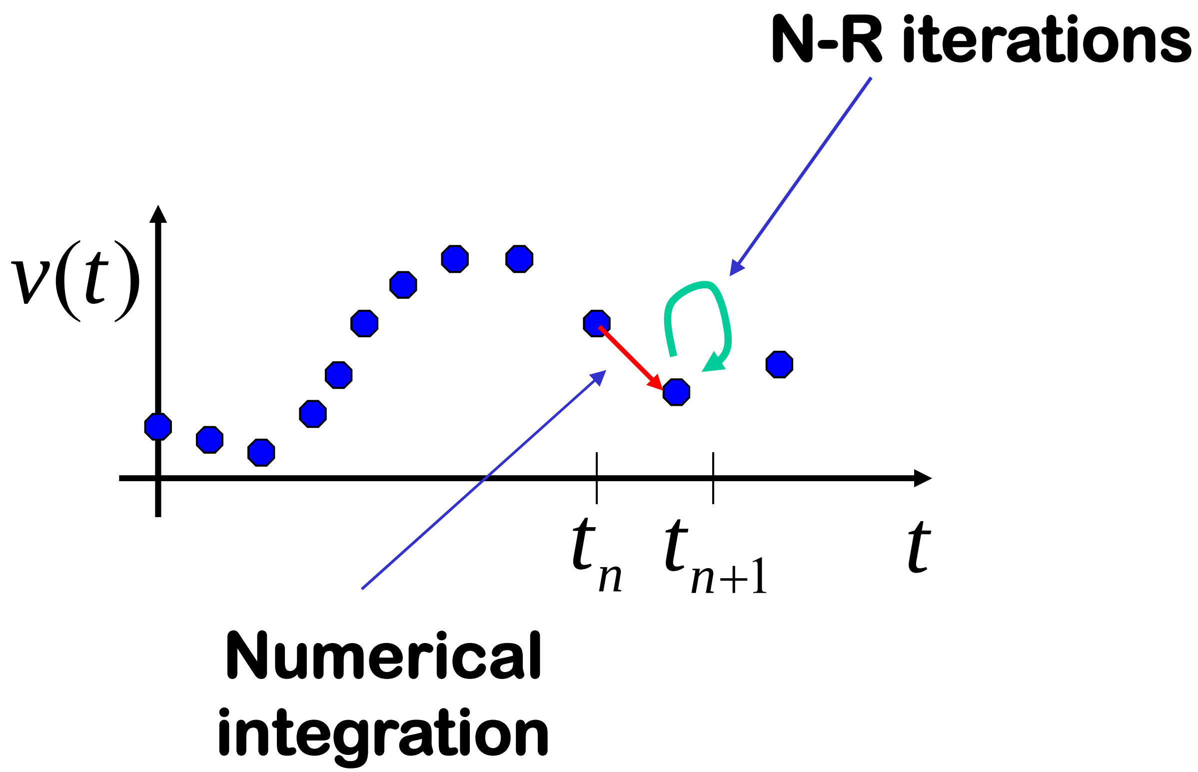
►Norton



$$I_{eq} = \frac{C}{\Delta t} v(t)$$

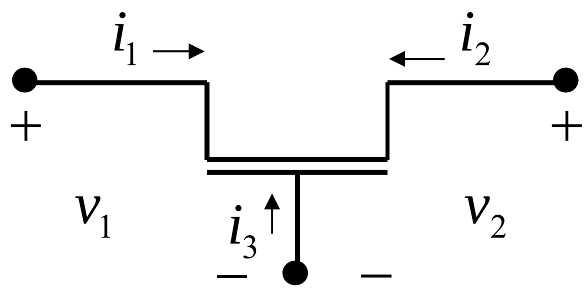
$$G_{eq} = \frac{C}{\Delta t}$$

- We combine the above two in nonlinear transient analysis



- **Start from some initial condition at time t_0**
- **Move to the next time step by replacing all the C's & L's by a companion model**
  - Solve the equivalent nonlinear DC problem at the new time point
    - Use the solution at the previous time step as the initial guess for N-R
    - Iterate till convergence
- **Repeat till reaching the ending time**
- **Nonlinear dynamic elements need to be handled more carefully**
  - Charge conservation – more on this later

- What about nonlinear elements with more than 2 terminals?



- Nonlinear equations:

$$i_1 = g_1(v_1, v_2) \qquad i_2 = g_2(v_1, v_2)$$

$$i_3 = -i_1 - i_2$$

Port Equations

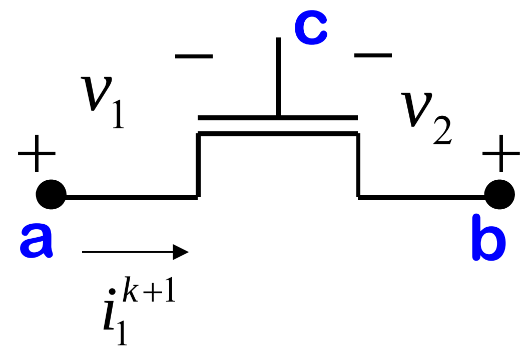
- Once again, model by first 2 terms of Taylor series

$$i_1^{k+1} = i_1^k + \Delta i_1^k = g_1(v_1^k, v_2^k) + \left.\frac{\partial g_1}{\partial v_1}\right|_k \cdot \Delta v_1 + \left.\frac{\partial g_1}{\partial v_2}\right|_k \cdot \Delta v_2$$

$$i_2^{k+1} = i_2^k + \Delta i_2^k = g_2(v_1^k, v_2^k) + \left.\frac{\partial g_2}{\partial v_1}\right|_k \cdot \Delta v_1 + \left.\frac{\partial g_2}{\partial v_2}\right|_k \cdot \Delta v_2$$

$$\begin{aligned} i_3^{k+1} = i_3^k + \Delta i_3^k &= -g_1(v_1^k, v_2^k) - g_2(v_1^k, v_2^k) \\ &+ \left.\left(\frac{-\partial g_1}{\partial v_1} - \frac{\partial g_2}{\partial v_1}\right)\right|_k \Delta v_1 + \left.\left(\frac{-\partial g_1}{\partial v_2} - \frac{\partial g_2}{\partial v_2}\right)\right|_K \Delta v_2 \end{aligned}$$

3-Terminal MOSFET Stamp:



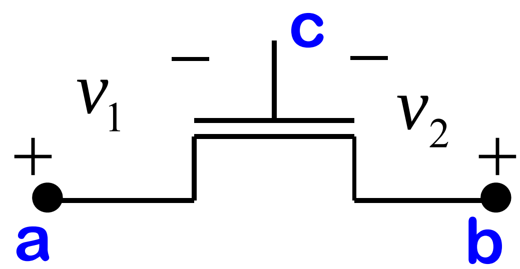
- **Consider the current contribution at each node**
  - **Note that we must translate port voltages to node voltages**

$$v_1 = v_a - v_c$$
$$v_2 = v_b - v_c$$

$$i_1^{k+1} = i_1^k + \Delta i_1^k = g_1(v_1^k, v_2^k) + \left.\frac{\partial g_1}{\partial v_1}\right|_k \cdot \Delta v_1 + \left.\frac{\partial g_1}{\partial v_2}\right|_k \cdot \Delta v_2$$

$$i_1^{k+1} = i_1^k + \Delta i_1^k = g_1(v_1^k, v_2^k) + \left.\frac{\partial g_1}{\partial v_1}\right|_k \cdot \Delta v_a + \left.\frac{\partial g_1}{\partial v_2}\right|_k \cdot \Delta v_b - \left(\left.\frac{\partial g_1}{\partial v_1}\right|_k + \left.\frac{\partial g_1}{\partial v_2}\right|_k \right) \cdot \Delta v_c$$

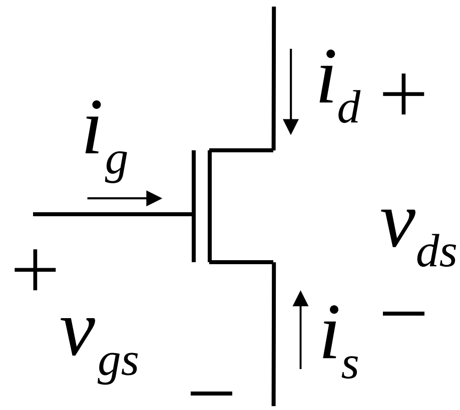
3-Terminal MOSFET Stamp:



- Most solvers are set up to solve for $\Delta\vec{v}$ instead of $\vec{v}^{\,k+1}$

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \quad a & & \quad b & & \quad c \end{array} \\
\begin{array}{c} a \\ \\ b \\ \\ c \end{array} &
\begin{bmatrix}
\dfrac{\partial g_1^k}{\partial v_1} & \cdots & \dfrac{\partial g_1^k}{\partial v_2} & \cdots & \left(\dfrac{-\partial g_1^k}{\partial v_1}-\dfrac{\partial g_1^k}{\partial v_2}\right) \\
\vdots & & \vdots & & \vdots \\
\dfrac{\partial g_2^k}{\partial v_1} & \cdots & \dfrac{\partial g_2^k}{\partial v_2} & \cdots & \left(\dfrac{-\partial g_2^k}{\partial v_1}-\dfrac{\partial g_2^k}{\partial v_2}\right) \\
\vdots & & \vdots & & \vdots \\
\left(\dfrac{-\partial g_1^k}{\partial v_1}-\dfrac{\partial g_2^k}{\partial v_1}\right) & \cdots & \left(\dfrac{-\partial g_1^k}{\partial v_2}-\dfrac{\partial g_2^k}{v_2}\right) & \cdots & \left\{\dfrac{\partial g_1^k}{\partial v_1}+\dfrac{\partial g_1^k}{\partial v_2}+\dfrac{\partial g_2^k}{\partial v_1}-\dfrac{\partial g_2^k}{\partial v_2}\right\}
\end{bmatrix}
\end{array}
\overset{\Delta\vec{v}}{\begin{bmatrix} \vdots \\ \Delta v_a \\ \vdots \\ \Delta v_b \\ \vdots \\ \Delta v_c \end{bmatrix}}
=
\begin{bmatrix} \vdots \\ -g_1(v_1^k, v_2^k) \\ \vdots \\ -g_2(v_1^k, v_2^k) \\ \vdots \\ g_1(v_1^k, v_2^k) \\ +g_2(v_1^k, v_2^k) \end{bmatrix}
$$

MOSFETs (3 terminal)



Triode Region

$$i_{ds} = \beta\left[(v_{gs} - v_{TH})v_{ds} - \frac{v_{ds}^{2}}{2}\right](1 + \lambda v_{ds})$$

$$v_{ds} < v_{gs} - v_{TH} \qquad \beta = \mu C_{ox}\frac{W}{L}$$

$$v_{gs} - v_{TH} > 0$$

Saturation Region

$$i_{ds} = \frac{\beta}{2}\left[(v_{gs} - v_{TH})^2\right]\cdot(1 + \lambda v_{ds})$$

$$v_{ds} > v_{gs} - v_{TH}$$

Cutoff Region

$$i_{ds} = 0 \qquad\qquad v_{gs} < v_{TH}$$

dc port equations:

$$i_g = 0$$

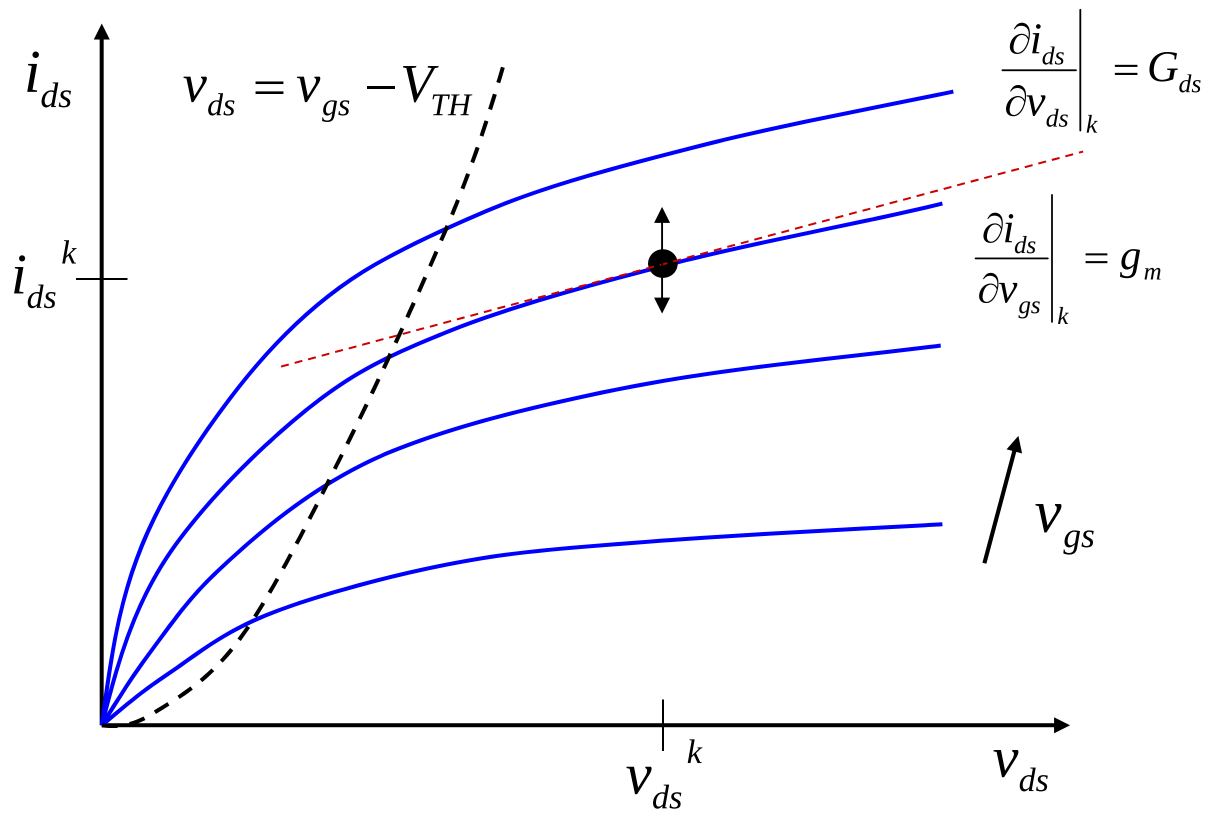
$$i_d = i_{ds}(v_{gs}, v_{ds})$$

$$i_s = -i_d = -i_{ds}(v_{gs}, v_{ds})$$

$$i_d^{k+1} = i_d^k + \Delta i_d^k = i_{ds}(v_{gs}{}^k, v_{ds}{}^k) +$$

$$\left.\frac{\partial i_{ds}}{\partial v_{gs}}\right|_k \cdot (v_{gs}{}^{k+1} - v_{gs}{}^k)$$

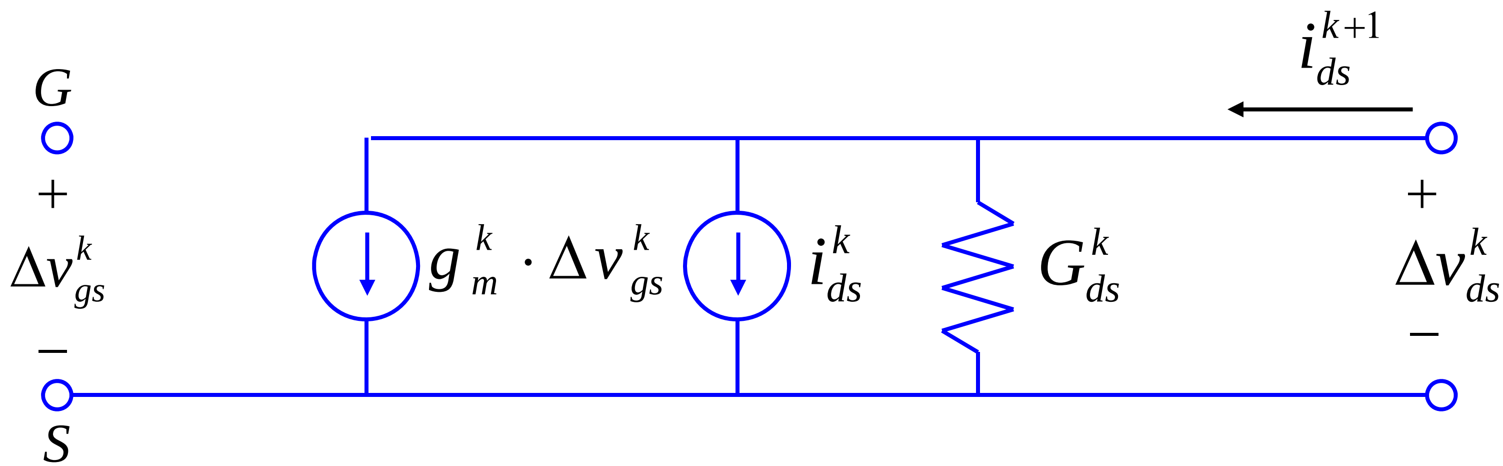
$$\left.\frac{\partial i_{ds}}{\partial v_{ds}}\right|_k \cdot (v_{ds}{}^{k+1} - v_{ds}{}^k)$$



$$i_{ds}^{k+1} = i_{ds}^{k} + g_m^k \cdot \Delta v_{gs}^k + G_{ds}^k \cdot \Delta v_{ds}^k$$

Equivalent ckt model for N-R

$$i_{ds}^{k+1} = i_{ds}^{k} + g_m \Delta v_{gs}^{k} + G_{ds}^{k} \cdot \Delta v_{ds}^{k}$$



$$\Delta v_{gs}^{k} = \left(v_{gs}^{k+1} - v_{gs}^{k}\right)$$

$$\Delta v_{ds}^{k} = \left(v_{ds}^{k+1} - v_{ds}^{k}\right)$$

- Could also build models to solve for v_{gs}^{k+1} and v_{ds}^{k+1} directly

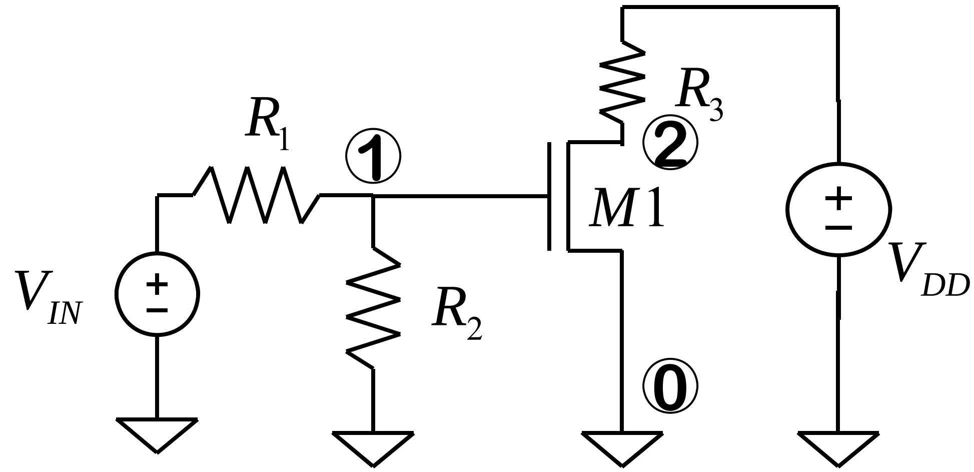
- Stamping in terms for all of these 2-terminal elements is equivalent to applying MOSFET stamp

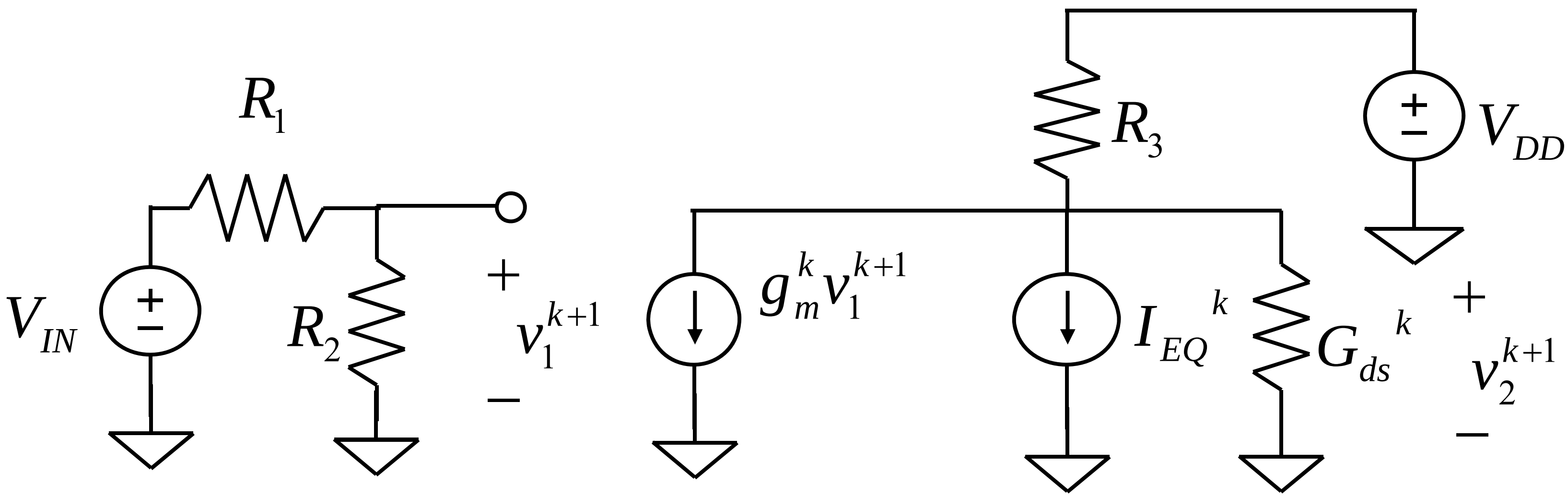
- Note the large off-diagonal terms that are created by g_m's

Simple Example

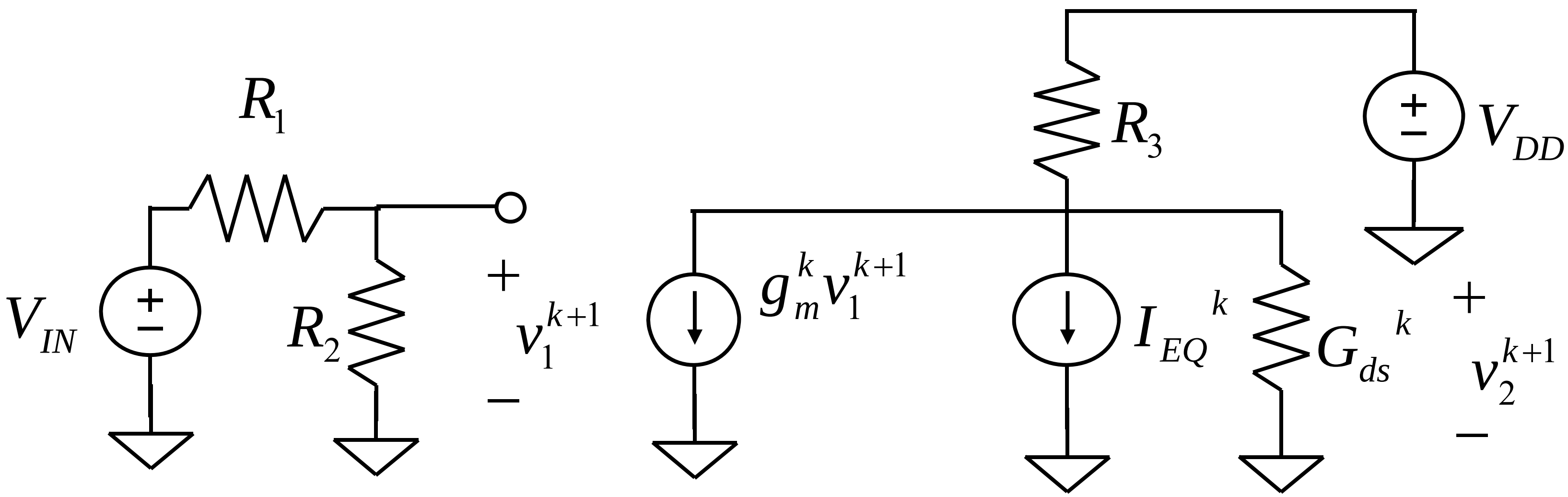
$$g_m^k = \left.\frac{\partial i_{ds}}{\partial v_{gs}}\right|_k \qquad G_{ds}^k = \left.\frac{\partial i_{ds}}{\partial v_{ds}}\right|_k$$

$$I_{EQ}^k = i_{ds}^k + g_m^k\left(-v_{gs}^k\right) + G_{ds}^k\left(-v_{ds}^k\right)$$





- Now formulate the nodal equations for this linearized equivalent ckt



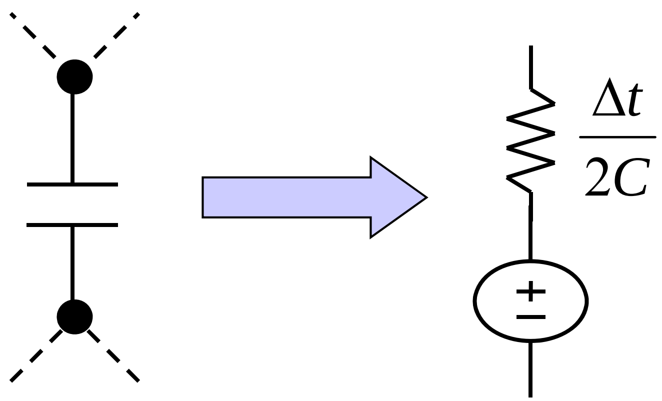
- g_m, G_{ds} and I_{EQ} values change at each N-R iteration

- Damping methods control the N-R convergence
  - More of a problem for BJTS

- In general, we would include the 4-th terminal of the MOSFET (body effect)

- We use dc nonlinear algorithms find solution at t=0 AND for all timepoints

- Energy storage elements are properly considered by numerical integration

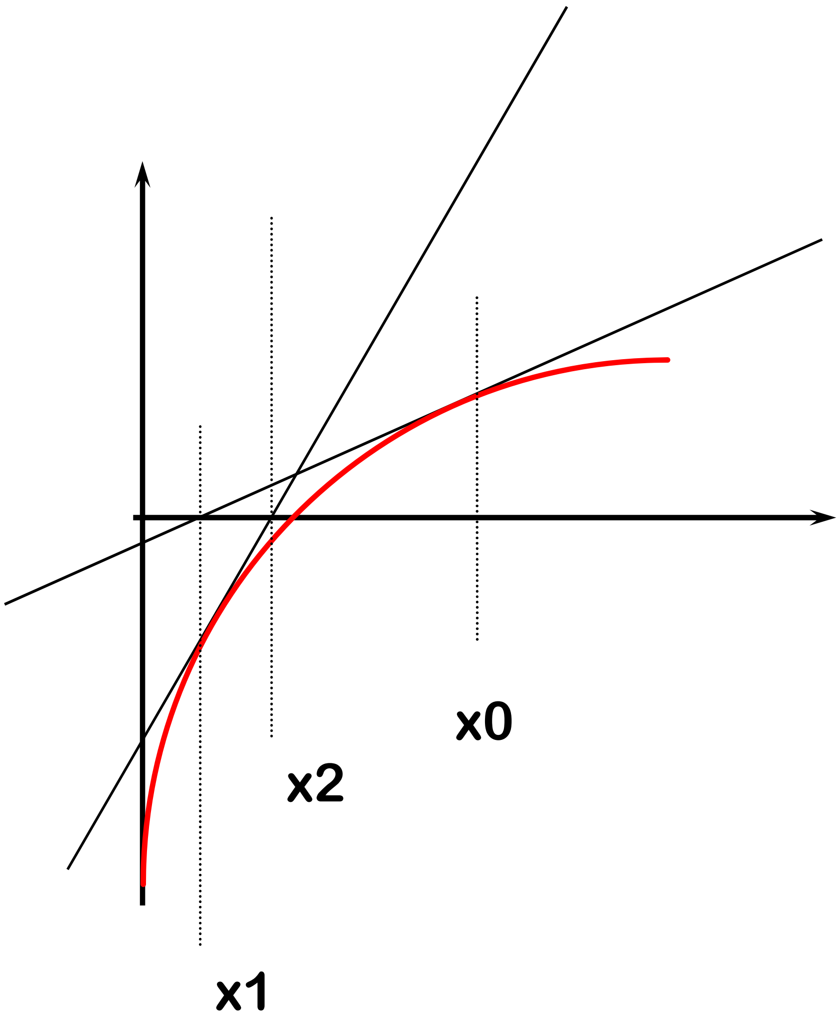


- How about nonlinear capacitors?

- **Conventional NL device models**
  - $i_{ds} = f(v_{ds}, v_{gs}, v_{sb})$ -- 30+ parameters
  - Accurate evaluation can be very expensive
- **NR requires full evaluation of models**
  - Derivatives required for Jacobian
    - **Expensive to evaluate -- can consume 50-80% of computation time**
- **SC-based approach**
  - No derivatives to re-organize – approximate Jacobian
  - Will work with table models, measured data
  - Reduced model evaluation time, but is based on:
    - **Slower convergence (depending on the J_{approx})**
    - **Matrix update required for varying timesteps**
    - **Selection of representative linear model in J_{approx}**

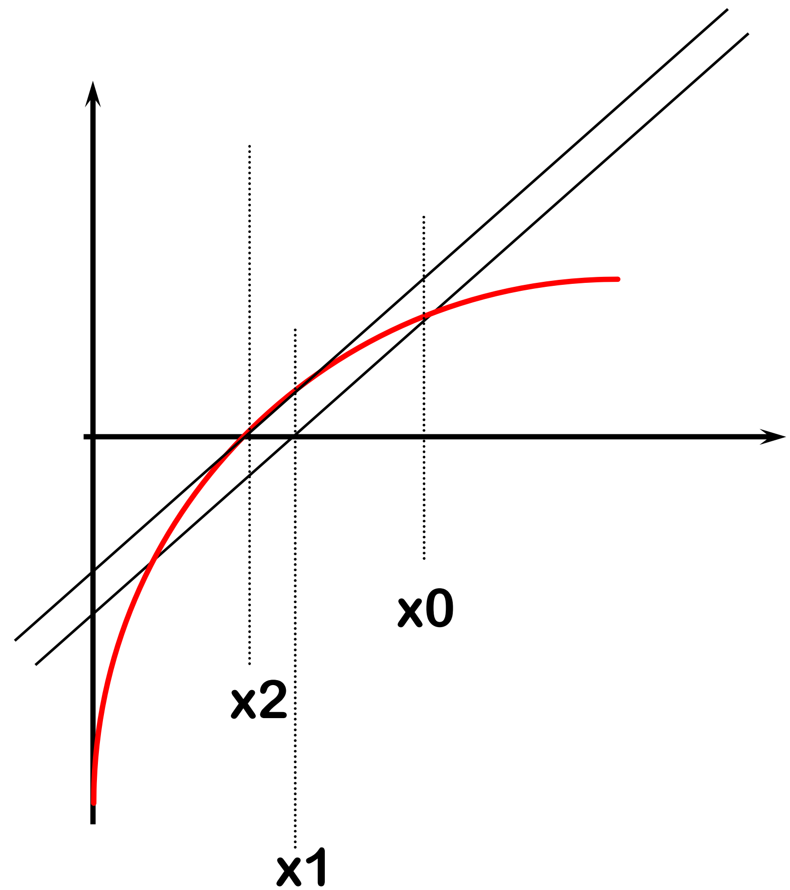
Newton-Raphson

- Good convergence properties
- Reliable when implemented with a step-limiting mechanism (damped)
- Requires explicit differentiation
- n-D case: Jacobian Matrix stamping (update) and re-factorization



Successive Chord

- J(x) represents a fixed gradient (Jacobian)
- No explicit differentiation
- Reliable when implemented with a step-limiting mechanism (damped)
- n-D case: Single Jacobian factorization improvement by optional updates



- NR has varying R, I
- SC has fixed R

- Linearized network changes for each NR step, i.e. $J(x_n)$

- Single linearized network for SC steps, J_{approx}