Lecture 8.
Nonlinear Device Stamping

Guoyong Shi, PhD
shiguoyong@ic.sjtu.edu.cn
School of Microelectronics
Shanghai Jiao Tong University
Fall 2010

Outline

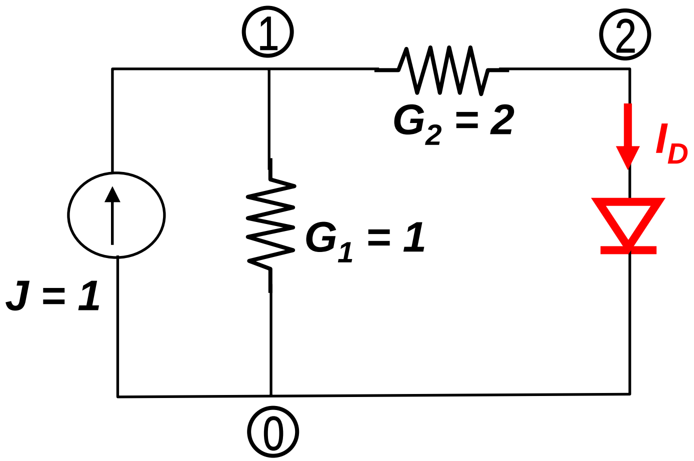
- Solving a nonlinear circuit
- Linearization and Newton-Raphson
- Jacobian
- Stamping nonlinear elements
  - Diode stamp
  - Nonlinear resistor stamp
  - MOS stamp
- Nonlinear transient simulation

A Nonlinear Circuit

The diode equation is nonlinear:

$$i_D = I_{sat}\left(e^{40V_D} - 1\right)$$

assuming $I_{sat} = 1$ A.



Write KCL equations at nodes 1 and 2:

$$\begin{cases} G_1 v_1 + G_2(v_1 - v_2) = 1 & (1) \\ 2(v_2 - v_1) + (e^{40v_2} - 1) = 0 & (2) \end{cases}$$

This part is nonlinear!

Solve the nodal voltages.

Solve by root-finding

$$\begin{cases} G_1 v_1 + G_2(v_1 - v_2) = 1 & \Rightarrow \quad 3v_1 - 2v_2 = 1 \qquad (1) \\ 2(v_2 - v_1) + (e^{40v_2} - 1) = 0 & \qquad (2) \end{cases}$$

From eqn (1): $v_1 = \frac{1}{3} + \frac{2}{3} v_2$

Substitute it into eqn (2): $2(v_2 - v_1) + \left(e^{40v_2} - 1\right) = 0$

$$\Rightarrow \quad f(v_2) := \frac{2}{3} v_2 - \frac{5}{3} + e^{40v_2} = 0$$

We get one nonlinear equation.

Need to find a root for the nonlinear equation: $f(v_2) = 0$

Nonlinear Root Finder

- **A general method for solving nonlinear f(x) = 0 is by iteration.**
- **The iteration is derived from "linearization."**

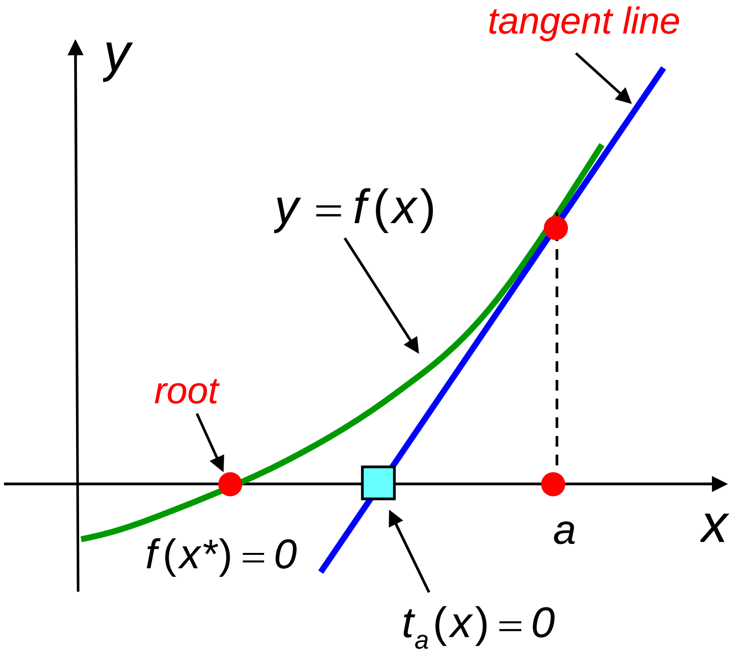
Taylor expand f(x) at a point x = a:

$$f(x) = f(a) + f'(a)(x-a) + \ldots$$

Ignoring the high-order terms,

$$y = t_a(x) = f(a) + f'(a)(x-a)$$

is a tangent line at (a, f(a)).



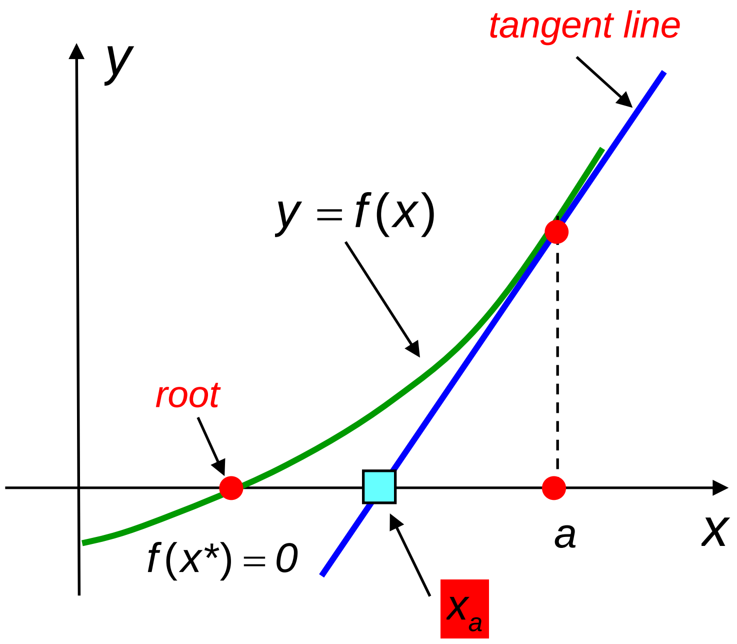
Nonlinear Root Finder

Find the point where the tangent line crosses the x-axis:

$$t_a(x) = f(a) + f'(a)(x-a) = 0$$

$$\Rightarrow \quad x_a = a - [f'(a)]^{-1} f(a)$$

Repeat this process.

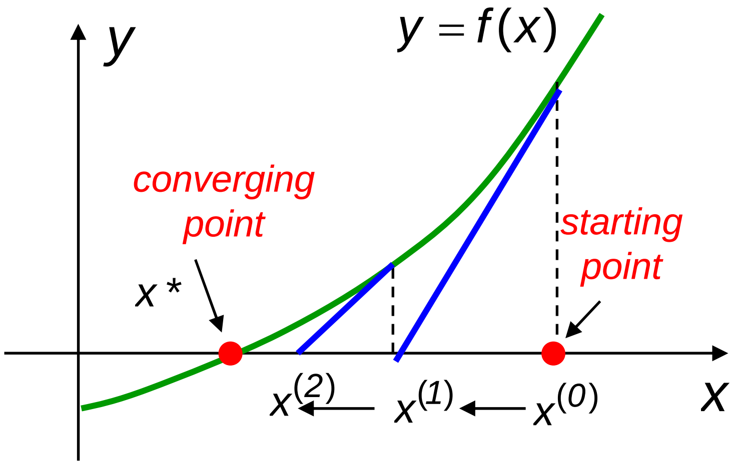


The Newton-Raphson Algorithm

For a scalar nonlinear equation f(x) = 0, the following iteration can find a root:

$$x^{(k+1)} = x^{(k)} - \left[f'(x^{(k)}) \right]^{-1} f(x^{(k)})$$

The initial point $x^{(0)}$ is arbitrary.



We shall discuss the <u>convergence</u> problem later.

Go back to our example ...

We would like to solve : $f(v_2) = \frac{2}{3}v_2 - \frac{5}{3} + e^{40v_2} = 0$

by the iteration $v_2^{(k+1)} = v_2^{(k)} - \left[f'(v_2^{(k)})\right]^{-1} f(v_2^{(k)})$

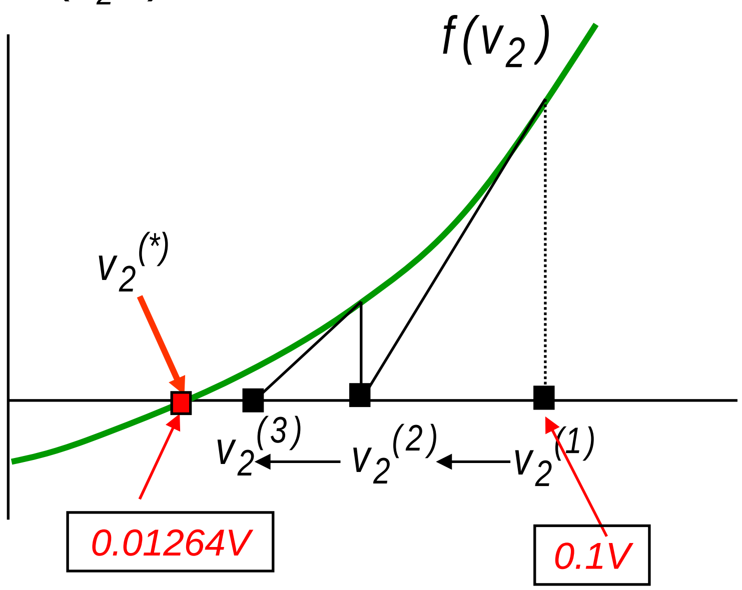
The derivative is: $f'(v_2) = \frac{2}{3} + 40e^{40v_2}$

Choose $v_2^{(0)} = 0.1\,V$

we get the following iterations:

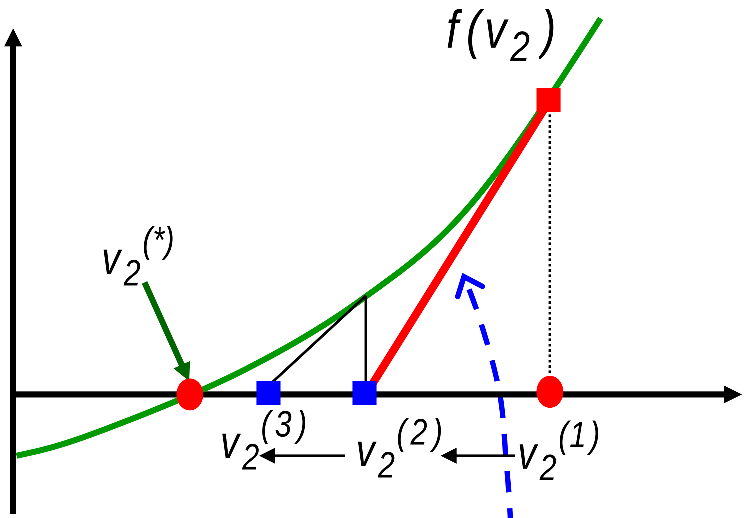
$v_2^{(1)} = 0.07574, \cdots, \quad v_2^{(4)} = 0.01883, \cdots$

$v_2^{(7)} \approx v_2^{(8)} = 0.012644, \cdots$ (converging ...)



Newton-Raphson Iteration (recap)

1. Choose an initial voltage;
2. Follow the tangent line to get the zero-crossing point;
3. Repeat the process.



Formal Algorithm:

$v_2^{(2)}$ *satisfies*

Linearization →

$$0 = f\left(v_2^{(1)}\right) + f'\left(v_2^{(1)}\right)\left(v_2^{(2)} - v_2^{(1)}\right)$$

N.R. Iteration: →

$$v_2^{(2)} = v_2^{(1)} - \left[f'(v_2^{(1)})\right]^{-1} f\left(v_2^{(1)}\right)$$

Multiple Nonlinear Equations

- **How to solve *n* variables from *n* equations, with at least one nonlinear;**
- **i.e., how to solve *f(x) = 0* where both f and x are vectors?**

We have a system of nonlinear multivariate equations: $f_j(x_1,\cdots,x_n)=0; \qquad j=1,\cdots,n$

Use multivariate Taylor expansion for each $f_j(x)$:

$$f_j(x_1,\cdots,x_n)=f_j(a_1,\cdots,a_n)+\left(\frac{\partial f_j(a)}{\partial x_1},\cdots,\frac{\partial f_j(a)}{\partial x_n}\right)\begin{pmatrix}x_1-a_1\\ \vdots\\ x_n-a_n\end{pmatrix}+\cdots$$

linearized part

Linearization

Putting n Taylor expansions together:

$$f_1(x_1,\cdots,x_n)=f_1(a_1,\cdots,a_n)+\left(\frac{\partial f_1(a)}{\partial x_1},\cdots,\frac{\partial f_1(a)}{\partial x_n}\right)\begin{pmatrix}x_1-a_1\\ \vdots\\ x_n-a_n\end{pmatrix}+\cdots$$

$$\vdots$$

$$f_j(x_1,\cdots,x_n)=f_j(a_1,\cdots,a_n)+\left(\frac{\partial f_j(a)}{\partial x_1},\cdots,\frac{\partial f_j(a)}{\partial x_n}\right)\begin{pmatrix}x_1-a_1\\ \vdots\\ x_n-a_n\end{pmatrix}+\cdots$$

$$\vdots$$

$$f_n(x_1,\cdots,x_n)=f_n(a_1,\cdots,a_n)+\left(\frac{\partial f_n(a)}{\partial x_1},\cdots,\frac{\partial f_n(a)}{\partial x_n}\right)\begin{pmatrix}x_1-a_1\\ \vdots\\ x_n-a_n\end{pmatrix}+\cdots$$

Linearization in Matrix-Vector Form

$$\mathbf{x} = (x_1, \cdots, x_n)^T$$

$$\mathbf{a} = (a_1, \cdots, a_n)^T$$

are vectors.

$$\begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} f_1(\mathbf{a}) \\ \vdots \\ f_n(\mathbf{a}) \end{pmatrix} + \begin{pmatrix} \frac{\partial f_1(\mathbf{a})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{a})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\mathbf{a})}{\partial x_1} & \cdots & \frac{\partial f_n(\mathbf{a})}{\partial x_n} \end{pmatrix} \begin{pmatrix} x_1 - a_1 \\ \vdots \\ x_n - a_n \end{pmatrix} + \cdots$$

Jacobian Matrix

In compact form:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{a}) + \frac{\partial \mathbf{f}(\mathbf{a})}{\partial \mathbf{x}} (\mathbf{x} - \mathbf{a}) + \cdots$$

Jacobian Matrix

$$y = f(x) \quad \Rightarrow \quad \frac{\partial f(a)}{\partial x} = \left(\frac{\partial f_i(a)}{\partial x_j}\right)_{nxn} = \begin{pmatrix} \frac{\partial f_1(a)}{\partial x_1} & \cdots & \frac{\partial f_1(a)}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(a)}{\partial x_1} & \cdots & \frac{\partial f_n(a)}{\partial x_n} \end{pmatrix}$$

Linearized equation
at x = a

$$f(x) = 0 \quad \Rightarrow \quad f(a) + \left[\frac{\partial f(a)}{\partial x}\right](x - a) = 0$$

$$\text{Let} \quad A = \frac{\partial f(a)}{\partial x}; \qquad b = -f(a) \quad \Rightarrow \quad A(x - a) = b$$

Solve (x-a).

Multivariable Newton-Raphson

Given

$$\boldsymbol{f}(x_1,\cdots,x_n)=\begin{pmatrix} f_1(x_1,\cdots,x_n) \\ \vdots \\ f_n(x_1,\cdots,x_n) \end{pmatrix}$$

Denote the "Jacobian matrix"

$$\boldsymbol{J}(\boldsymbol{x})=\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}=\left(\frac{\partial f_i(\boldsymbol{x})}{\partial x_j}\right)_{n\times n}=\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

$$\boldsymbol{f}(\boldsymbol{x}^{(k)})+\left[\frac{\partial \boldsymbol{f}(\boldsymbol{x}^{(k)})}{\partial \boldsymbol{x}}\right](\boldsymbol{x}^{(k)}-\boldsymbol{a})=0$$

$$x^{(k+1)}=x^{(k)}-\left[\,J(v^{(k)})\,\right]^{-1}f(x^{(k)})$$

NR Iteration

Summary

- **Nonlinear equations** ***f(x) = 0*** **are solved by** ***linearization*****.**
- **Once again, we are solving Ax = b.**
- **But have solve Ax = b for many times until convergence.**

$$x^{(k+1)} = x^{(k)} - \left[J(v^{(k)}) \right]^{-1} f(x^{(k)})$$

$$\left[J(x^{(k)}) \right] \left(x^{(k+1)} - x^{(k)} \right) = -f(x^{(k)})$$

Simulator Perspective

- **In a simulator implementation, we do NOT formulate nonlinear equations then solve.**
- **Rather, we do nonlinear element stamping following the principle of "Newton-Raphson Iteration".**
- **In transient simulation, we'll have two loops:**
  - **The outer loop is for time advancing;**
  - **The inner loop is for NR iteration.**

Nonlinear Element Stamping

- **Nonlinear devices are stamped by the principle of "linearization".**

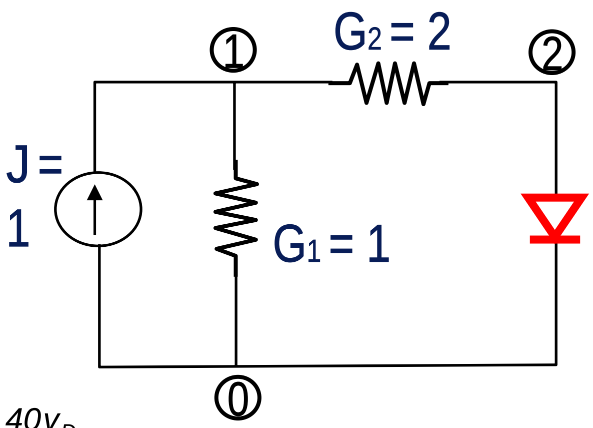
- **Recall the Newton-Raphson iteration:**

$$\left[J(v^{(k)}) \right]\left(x^{(k+1)} - x^{(k)} \right) = -f(x^{(k)})$$

$$\Rightarrow \quad \underbrace{\left[J(v^{(k)}) \right]}_{\mathbf{A}} x^{(k+1)} = \underbrace{\left[J(v^{(k)}) \right] x^{(k)} - f(x^{(k)})}_{\mathbf{RHS}}$$

In Spice simulator, matrix A and RHS are filled up dynamically via element-by-element stamping (both linear and nonlinear).

Back to our example again ...



We'll work it out again by stamping.

Idea: Linearize the diode equation

at the previously solved voltage:

$$i_D = e^{40V_D} - 1 \quad \Rightarrow \quad \frac{\partial i_D}{\partial v_D} = 40e^{40v_D}$$

$$i_D^{(k+1)} \approx i_D^{(k)} + \left(\frac{\partial i_D}{\partial v_D}\right)^{(k)} \left(v_D - v_D^{(k)}\right)$$

(linearization point)

$$= \left(\frac{\partial i_D}{\partial v_D}\right)^{(k)} v_D + \left[i_D^{(k)} - \left(\frac{\partial i_D}{\partial v_D}\right)^{(k)} v_D^{(k)} \right]$$

Linearized current equation

equiv G

equiv current

Companion Network

$$i^{(k+1)} = i^{(k)} + \frac{\partial g(v^{(k)})}{\partial v}\left(v^{(k+1)} - v^{(k)}\right)$$

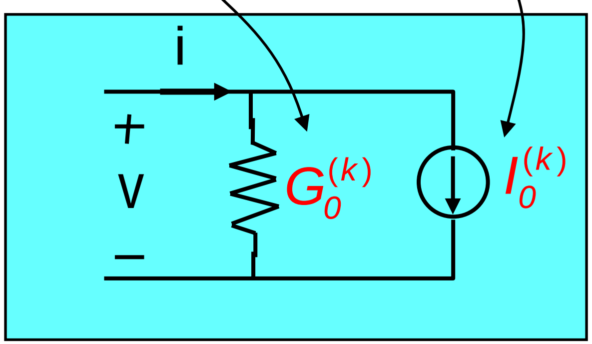
$$= G_0^{(k)}\, v^{(k+1)} + I_0^{(k)}$$

I-V relation: *i = g(v)*

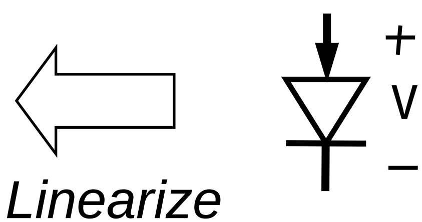
where

This term is zero for linear devices, .

$$G_0^{(k)} = \frac{\partial g(v^{(k)})}{\partial v}, \quad I_0^{(k)} = i^{(k)} - G_0^{(k)} v^{(k)}$$



Companion network (linearized)



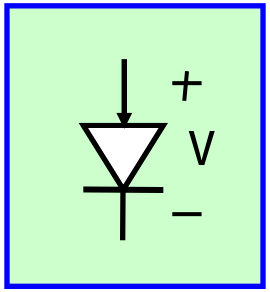
$$i^{(k)} = g\left(v^{(k)}\right)$$

G_0 and I_0 are updated during iteration !

Stamp for Diode

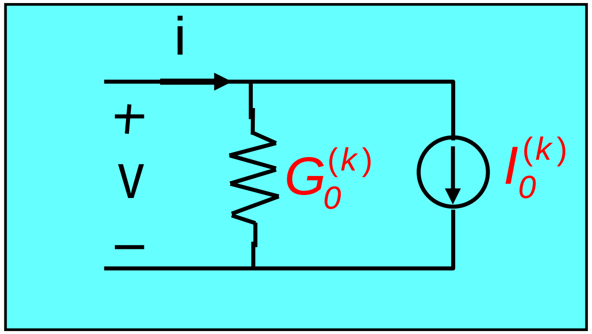
The companion network is used for stamping.

$$i^{(n+1)} = G_0^{(n)}\ v^{(n+1)} + I_0^{(n)}$$



The diode stamp:

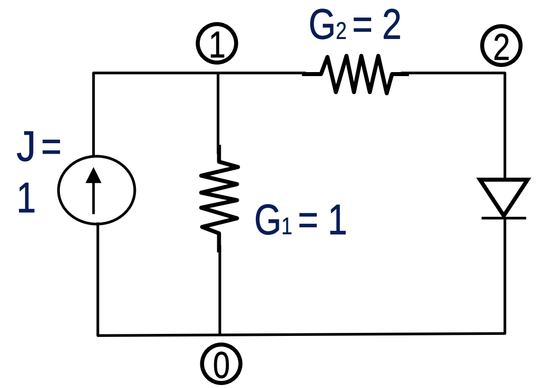
	N^+	N^-	RHS
N^+	$G_0^{(n)}$	$-G_0^{(n)}$	$-I_0^{(n)}$
N^-	$-G_0^{(n)}$	$G_0^{(n)}$	$+I_0^{(n)}$



$$G_0^{(n)} = \frac{\partial g(v^{(n)})}{\partial v}, \quad I_0^{(n)} = i^{(n)} - G_0^{(n)} v^{(n)}$$

Diode Stamping – Look Closer

Solving nonlinear circuit by stamping



$$i_D^{(n+1)} = i_D^{(n)} + \frac{\partial i_D}{\partial u_2}\left(u_2^{(n+1)} - u_2^{(n)}\right)$$

$$= \left[40\,e^{(40u_2^{(n)})}\right] u_2^{(n+1)}$$

$$- \left[40\,e^{(40u_2^{(n)})}\,u_2^{(n)} - \left(e^{(40u_2^{(n)})} - 1\right)\right]$$

$$i_D = e^{40V_D} - 1$$

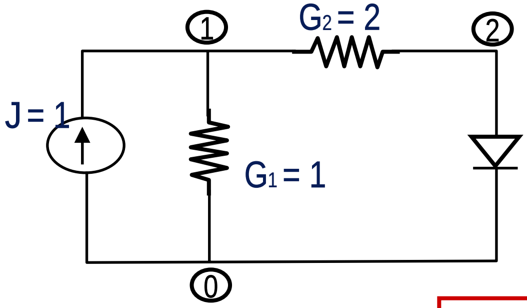
Matrix stamping:

RHS

$$\begin{bmatrix} 1+2 & -2 \\ -2 & 2+40\,e^{40u_2^{(n)}} \end{bmatrix} \begin{bmatrix} u_1^{(n+1)} \\ u_2^{(n+1)} \end{bmatrix} = \begin{bmatrix} 1 \\ 40\,e^{40u_2^{(n)}}\,u_2^{(n)} - \left(e^{40u_2^{(n)}} - 1\right) \end{bmatrix}$$

The diode "load" function does these updates at each iteration.

A closer look at Diode Stamping



Stamp for G_1 and Diode

$$\begin{bmatrix} 1 & \\ & 40\,e^{(40u_2^{(n)})} \end{bmatrix}$$

RHS

$$\begin{pmatrix} J \\ 40\,e^{40u_2^{(n)}}\,u_2^{(n)} - \left(e^{40u_2^{(n)}} - 1\right) \end{pmatrix}$$

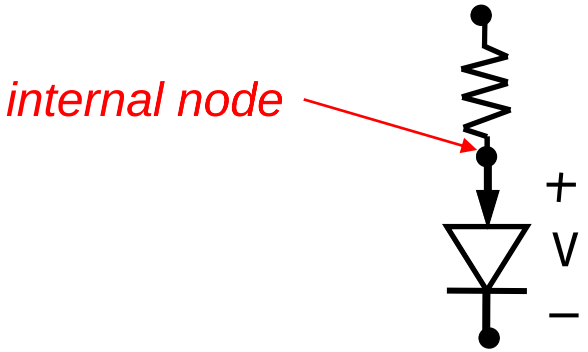
Stamp for G_2

$$\begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$$

putting together

$$\begin{bmatrix} 1+2 & -2 \\ -2 & 2+40\,e^{(40u_2^{(n)})} \end{bmatrix}$$

Resistive Diode



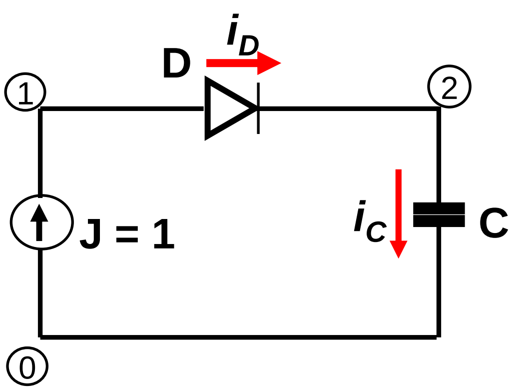
Exercise: Write down the stamp for this resistive diode.

Spice has such a diode model.

Nonlinear Transient Simulation

- **How to do a transient simulation of a circuit with nonlinear elements?**

$i_D = e^{\alpha V_D} - 1$



Backward Euler + Newton-Raphson

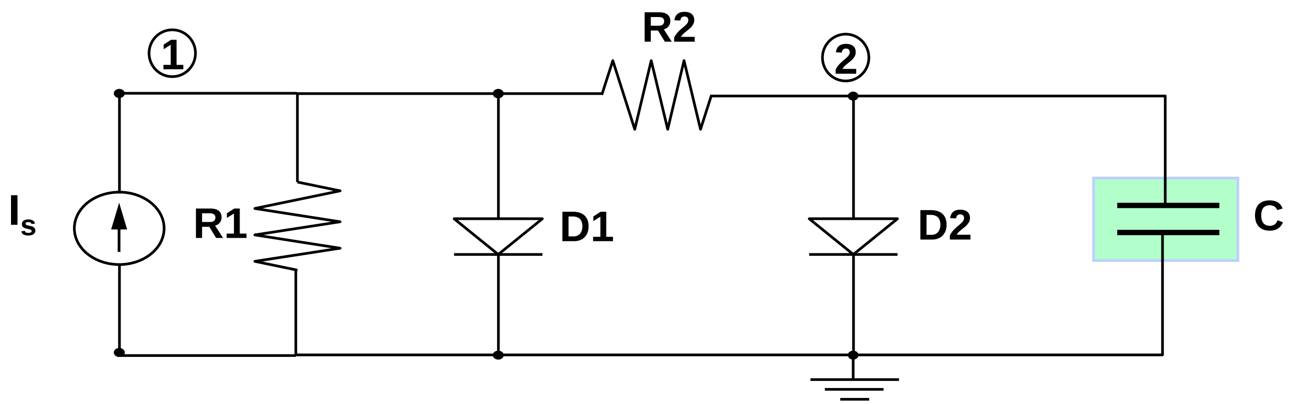
	0	1	2	RHS
0	$\frac{C}{h}$	0	$-\frac{C}{h}$	$-\frac{C}{h} v_c(t-h)$
1	0	$\alpha\, e^{\alpha v_D^{(n)}}$	$-\alpha\, e^{\alpha v_D^{(n)}}$	$J - I_D^{(n)}$
2	$-\frac{C}{h}$	$-\alpha\, e^{\alpha v_D^{(n)}}$	$\left(\alpha\, e^{\alpha v_D^{(n)}} + \frac{C}{h}\right)$	$\frac{C}{h} v_C^{(n)}(t-h) + I_D^{(n)}$

$$v_D^{(n)} = v_2^{(n)} - v_1^{(n)}$$

$$I_D^{(n)} = \alpha\, e^{\alpha v_D^{(n)}}\, v_D^{(n)} - \left(e^{\alpha v_D^{(n)}} - 1\right)$$

At time $t = t_k$, do NR-iteration until convergence, then advance time to $t = t_{k+1}$.

Simulate this circuit



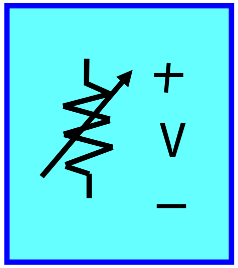
Assume an appropriate model for the two diodes.

Nonlinear Resistor

$$v^{(n+1)} = r\left(i^{(n+1)}\right) \approx r\left(i^{(n)}\right) + \dot{r}\left(i^{(n)}\right)\left(i^{(n+1)} - i^{(n)}\right)$$

$$v^{(n+1)} = \dot{r}\left(i^{(n)}\right) i^{(n+1)} + V_0^{(n)}$$

$v = r(i)$



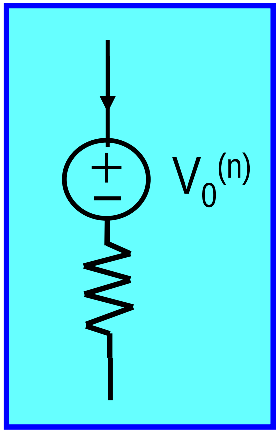
$$V_0^{(n)} = \left[r\left(i^{(n)}\right) - \dot{r}\left(i^{(n)}\right) i^{(n)} \right]$$

MNA stamp:

	N^+	N^-	i	RHS
N^+			+1	
N^-			-1	
	+1	-1	$-\dot{r}\left(i^{(n)}\right)$	$V_0^{(n)}$

current controlled nonlinear resistor

$R^{(n)} = r'\left(i^{(n)}\right)$

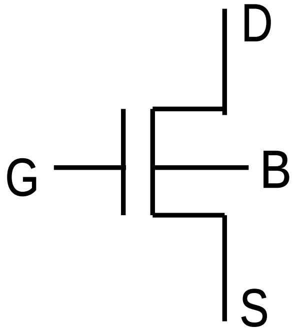


MOSFET Model

Level 1 Spice model (resistive region)

$$V_{gs} > V_t$$

$$V_{ds} < V_{gs} - V_t$$



$$I_{ds} = I_{ds}(V_{gs}, V_{ds})$$
$$= \left(\frac{W}{L}\right) K' \left[(V_{gs} - V_t) V_{ds} - \frac{1}{2} V_{ds}^2 \right]$$

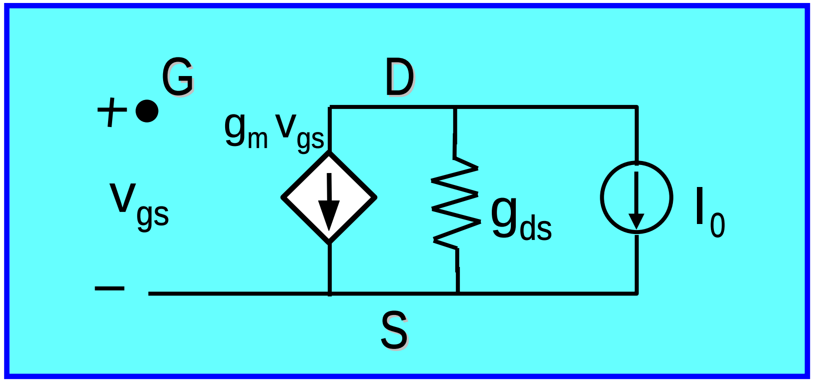
Assume V_t is independent of V_{bs}

Choose source (S) as the reference node.

$$\begin{cases} g_m = \dfrac{\partial I_{ds}}{\partial V_{gs}} = \dfrac{W}{L} K' V_{ds} \\ g_{ds} = \dfrac{\partial I_{ds}}{\partial V_{ds}} = \dfrac{W}{L} K' (V_{gs} - V_t - V_{ds}) \end{cases}$$

MOSFET Linearization

$$I_{ds} = I_{ds}(V_{gs}, V_{ds})$$



Linearization

$$I_{ds}^{(n+1)} = I_{ds}\left(V_{gs}^{(n+1)}, V_{ds}^{(n+1)}\right)$$

← (Bi-variate function)

$$= I_{ds}\left(V_{gs}^{(n)}, V_{ds}^{(n)}\right) + g_m^{(n)}\left(V_{gs}^{(n+1)} - V_{gs}^{(n)}\right) + g_{ds}^{(n)}\left(V_{ds}^{(n+1)} - V_{ds}^{(n)}\right)$$

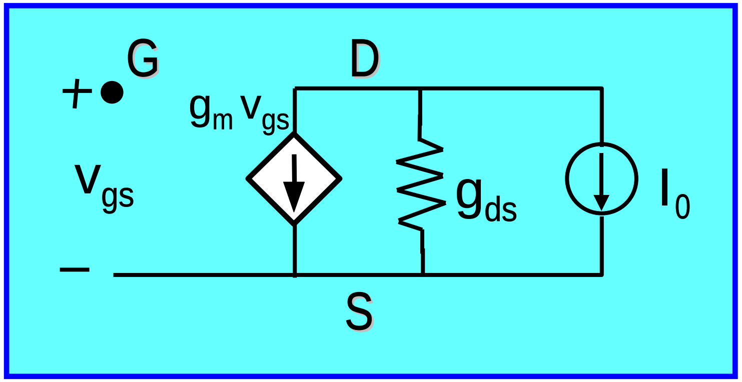
$$= g_m^{(n)} V_{gs}^{(n+1)} + g_{ds}^{(n)} V_{ds}^{(n+1)} + I_0^{(n)}$$

(two partial derivatives)

$$I_0^{(n)} \triangleq I_{ds}\left(V_{gs}^{(n)}, V_{ds}^{(n)}\right) - g_m^{(n)} V_{gs}^{(n)} - g_{ds}^{(n)} V_{ds}^{(n)}$$

MOSFET Stamp

$$I_{ds} = I_{ds}(V_{gs}, V_{ds})$$



$$I_{ds}^{(n+1)} = g_m^{(n)} V_{gs}^{(n+1)} + g_{ds}^{(n)} V_{ds}^{(n+1)} + I_0^{(n)}$$

	N^d	N^s	N^g	RHS
N^d	$g_{ds}^{(n)}$	$-g_{ds}^{(n)} - g_m^{(n)}$	$g_m^{(n)}$	$-I_0^{(n)}$
N^s	$-g_{ds}^{(n)}$	$g_{ds}^{(n)} + g_m^{(n)}$	$-g_m^{(n)}$	$I_0^{(n)}$

$$g_m^{(n)} = \frac{\partial I_{ds}\left(V_{gs}^{(n)}, V_{ds}^{(n)}\right)}{\partial V_{gs}}$$

$$g_{ds}^{(n)} = \frac{\partial I_{ds}\left(V_{gs}^{(n)}, V_{ds}^{(n)}\right)}{\partial V_{ds}}$$

(entries depend on the iteration step)

Acknowledgement

- **Some slides are adapted from**
  - **Prof. Albert Sangiovanni-Vincentelli's lecture at University of California, Berkeley (instructor Alessandra Nardi)**