EECE 552
Numerical Circuit Analysis

Chapter Nine

APPLICATION TO TRANSIENT ANALYSIS OF ELECTRICAL CIRCUITS

I. Hajj

Application to Electrical Circuits

- Method 1: Construct state equations

$$\dot{x} = \mathbf{f}(\mathbf{x}, \mathbf{t})$$

- Method 2: Apply directly to Tableau Equations, which are mixed algebraic-differential equations:

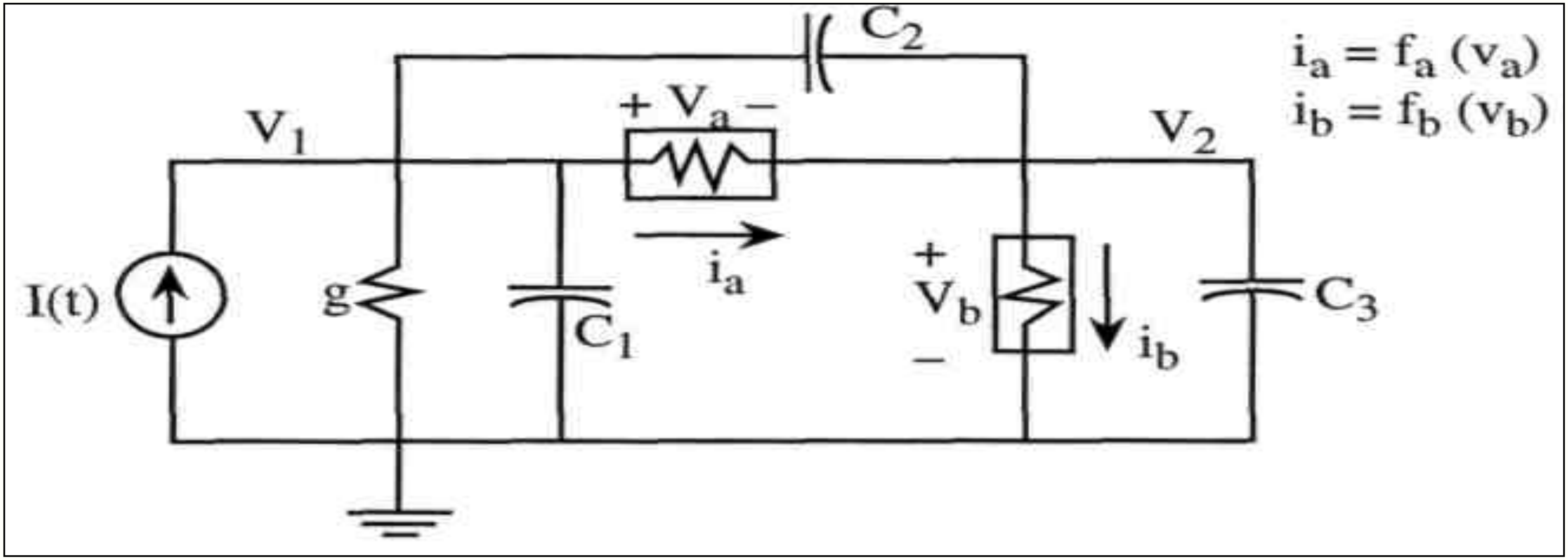
$$\mathbf{f}(\mathbf{x}, \dot{x}, \mathbf{t}) = \mathbf{0}$$

Special Case for Constructing State Equations

For circuits suitable for nodal analysis (no inductors, no current variables; if voltage sources exist, transform them into current sources using Norton equivalent circuit), **and** if there is a capacitive path from every node to ground, the nodal equations generate state equations.

$$\mathbf{C}\frac{d\mathbf{V}_n}{dt} = \mathbf{f}(\mathbf{V}_n, t) \Rightarrow \boxed{\dot{\mathbf{V}}_n = \mathbf{C}^{-1}\mathbf{f}(\mathbf{V}_n, t)}$$

Example



$$C_1 \frac{dV_1}{dt} + g_1 V_1 + C_2 \frac{d}{dt}(V_1 - V_2) + f_a(V_1 - V_2) = I(t)$$

$$C_3 \frac{dV_2}{dt} + C_2 \frac{d}{dt}(V_2 - V_1) + f_b(V_2) - f_a(V_1 - V_2) = 0$$

$$\begin{bmatrix} C_1 + C_2 & -C_2 \\ -C_2 & C_2 + C_3 \end{bmatrix} \begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \end{bmatrix} = \begin{bmatrix} -g_1 V_1 - f_a(V_1 - V_2) \\ -f_b(V_2) + f_a(V_1 - V_2) \end{bmatrix} + \begin{bmatrix} I(t) \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} C_1 + C_2 & -C_2 \\ -C_2 & C_2 + C_3 \end{bmatrix}^{-1} \begin{bmatrix} -g_1 V_1 - f_a(V_1 - V_2) \\ -f_b(V_2) + f_a(V_1 - V_2) \end{bmatrix} + \begin{bmatrix} C_1 + C_2 & -C_2 \\ -C_2 & C_2 + C_3 \end{bmatrix}^{-1} \begin{bmatrix} I(t) \\ 0 \end{bmatrix}$$

Linear Circuit $i_a = g_a v_a$, $i_b = g_b v_b$

$$\boxed{\begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \end{bmatrix} = \begin{bmatrix} C_1 + C_2 & -C_2 \\ -C_2 & C_2 + C_3 \end{bmatrix}^{-1} \begin{bmatrix} -g_1 - g_a & +g_a \\ +g_a & -g_b - g_a \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} C_1 + C_2 & -C_2 \\ -C_2 & C_2 + C_3 \end{bmatrix}^{-1} \begin{bmatrix} I(t) \\ 0 \end{bmatrix}}$$

State Equations

More generally, the MNA equations of a circuit containing linear capacitors and inductors and linear and nonlinear resistors can be written in the form

$$\mathbf{f}(\mathbf{x}(t)) + \mathbf{C}\dot{\mathbf{x}}(t) = \mathbf{y}(t)$$

If **C** is nonsingular,

$$\dot{\mathbf{x}}(t) = -\mathbf{C}^{-1}\mathbf{f}(\mathbf{x}(t)) + \mathbf{C}^{-1}\mathbf{y}(t)$$

General Case: $f(x, \dot{x}, t) = 0$

apply $$\boxed{\text{lmf:}\ \sum_{i=0}^{k} (\alpha_i x_{n-i} + h\beta_i\, \dot{x}_{n-i}) = 0}$$

or $$\beta_o \neq 0:\ \ \dot{x}_n = \frac{1}{h\beta_o}\left(\alpha_o x_n + \underbrace{\sum_{i=1}^{k} (\alpha_i x_{n-1} + h\beta_i \dot{x}_{n-i})}_{\text{known}}\right) = g(x_n, t_n)$$

get $$\boxed{\mathbf{f}(x_n, \mathbf{g}(\mathbf{x}_n, t_n)) = 0} \ \Leftarrow \text{ nonlinear algebraic equation}$$

Tableau Equations of General Linear and Nonlinear Circuits

$$\left.\begin{array}{ll}\text{KCL}: & \mathbf{Ai}=\mathbf{0} \\ \\ \text{KVL}: & \mathbf{v}=\mathbf{A}^{\mathbf{T}}\mathbf{V}_{\mathbf{n}}\end{array}\right\} \begin{array}{l}\text{Linear} \\ \text{algebraic} \\ \text{equations}\end{array}$$

Element Characteristics

(a) Resistors (linear and nonlinear, including independent sources)

$$\mathrm{f_R(i_R, v_R) = s_R}$$

Capacitors

(b) *Capacitors:*

Voltage-controlled capacitors:

$$\mathbf{q_c} = \mathrm{f}_c(\mathbf{v_c}), \quad \mathbf{i_c} = \frac{d\mathbf{q_c}}{dt}$$

Charge-controlled capacitors:

$$\mathbf{v_c} = \mathrm{f}_c(\mathbf{q_c}), \quad \mathbf{i_c} = \frac{d\mathbf{q_c}}{dt}$$

Charge-and-voltage-controlled capacitors:

$$\mathrm{f_c}(\mathbf{v_c}, \mathbf{q_c}) = \mathbf{0}, \ \mathbf{i_c} = \frac{d\mathbf{q_c}}{dt}$$

Inductors

(c) *Inductors:*

Current-controlled inductors:

$$\boldsymbol{\phi}_{\mathbf{L}} = \mathbf{f}_L(\mathbf{i}_{\mathbf{L}}),\ \mathbf{v}_{\mathbf{L}} = \frac{\mathrm{d}\boldsymbol{\phi}_{\mathrm{L}}}{dt}$$

Flux-controlled inductors:

$$\mathbf{i}_{\mathbf{L}} = \mathbf{f}_L(\boldsymbol{\phi}_{\mathbf{L}}),\ \mathbf{v}_{\mathbf{L}} = \frac{d\boldsymbol{\phi}_{\mathbf{L}}}{dt}$$

Flux-and-current-controlled inductors:

$$\mathbf{f}_{\mathrm{L}}(\mathbf{i}_{\mathrm{L}}, \boldsymbol{\phi}_{\mathrm{L}}) = \mathbf{0},\ \mathbf{v}_{\mathrm{L}} = \frac{d\boldsymbol{\phi}_{\mathrm{L}}}{dt}$$

memrisitve systems

(d) *Mem-Systems:*

Current-controlled memristive systems:

$$v_M(t) = f_1(\mathbf{x}, i_M, t) i_M(t)$$
$$\frac{d\mathbf{x}}{dt} = f_2(\mathbf{x}, i_M, t)$$

Voltage-controlled memristive systems:

$$i_M(t) = f_1(\mathbf{x}, v_M, t) v_M(t)$$
$$\frac{d\mathbf{x}}{dt} = f_2(\mathbf{x}, v_M, t)$$

memcapacitive systems

Voltage-controlled memcapacitive systems:

$$q_M(t) = f_1(\mathbf{x}, v_M, t) v_M(t), \; i_M(t) = \frac{dq_M}{dt}$$

$$\frac{d\mathbf{x}}{dt} = f_2(\mathbf{x}, v_M, t)$$

Charge-controlled memcapacitive systems:

$$v_M(t) = f_1(\mathbf{x}, q_M, t) q_M(t), \; i_M(t) = \frac{dq_M}{dt}$$

$$\frac{d\mathbf{x}}{dt} = f_2(\mathbf{x}, q_M, t)$$

meminductive systems

Current-controlled meminductive systems:

$$\phi_M(t) = f_1(\mathbf{x}, i_M, t) i_M(t),\ v_M(t) = \frac{d\phi_M}{dt}$$

Flux-controlled meminductive systems:

$$i_M(t) = f_1(\mathbf{x}, \phi_M, t) \phi_M(t),\ v_M(t) = \frac{d\phi_M}{dt}$$

$$\frac{d\mathbf{x}}{dt} = f_2(\mathbf{x}, \phi_M, t)$$

Mem Devices: Memristors

Charge-controlled memristors:

$$\phi_M(t) = f_M(q_M),\ i_M(t) = \frac{dq_M}{dt},\ v_M(t) = \frac{d\phi_M}{dt}$$

Flux-controlled memristors:

$$q_M(t) = f_M(\phi_M),\ i_M(t) = \frac{dq_M}{dt},\ v_M(t) = \frac{d\phi_M}{dt}$$

Memcapacitors

Flux-controlled memcapacitors:

$$\sigma_M(t) = f_M(\phi_M),\ q_M(t) = \frac{d\sigma_M}{dt},\ i_M(t) = \frac{dq_M}{dt},\ v_M(t) = \frac{d\phi_M}{dt}$$

σ*-controlled memcapacitors:*

$$\phi_M(t) = f_M(\sigma_M),\ q_M(t) = \frac{d\sigma_M}{dt},\ i_M(t) = \frac{dq_M}{dt},\ v_M(t) = \frac{d\phi_M}{dt}$$

Meminductors

Charge-controlled meminductors:

$$\rho_M(t) = f_M(q_M),\ \phi_M(t) = \frac{d\rho_M}{dt},\ i_M(t) = \frac{dq_M}{dt},\ v_M(t) = \frac{d\phi_M}{dt}$$

ρ-controlled meminductors:

$$q_M(t) = f_M(\rho_M),\ \phi_M(t) = \frac{d\rho_M}{dt},\ i_M(t) = \frac{dq_M}{dt},\ v_M(t) = \frac{d\phi_M}{dt}$$

Companion Models and Stamps

- No need to construct state equations.
- Apply integration formula to differential operator and derive a companion model.
- Derive a stamp to incorporate elements into the linearized circuit equations

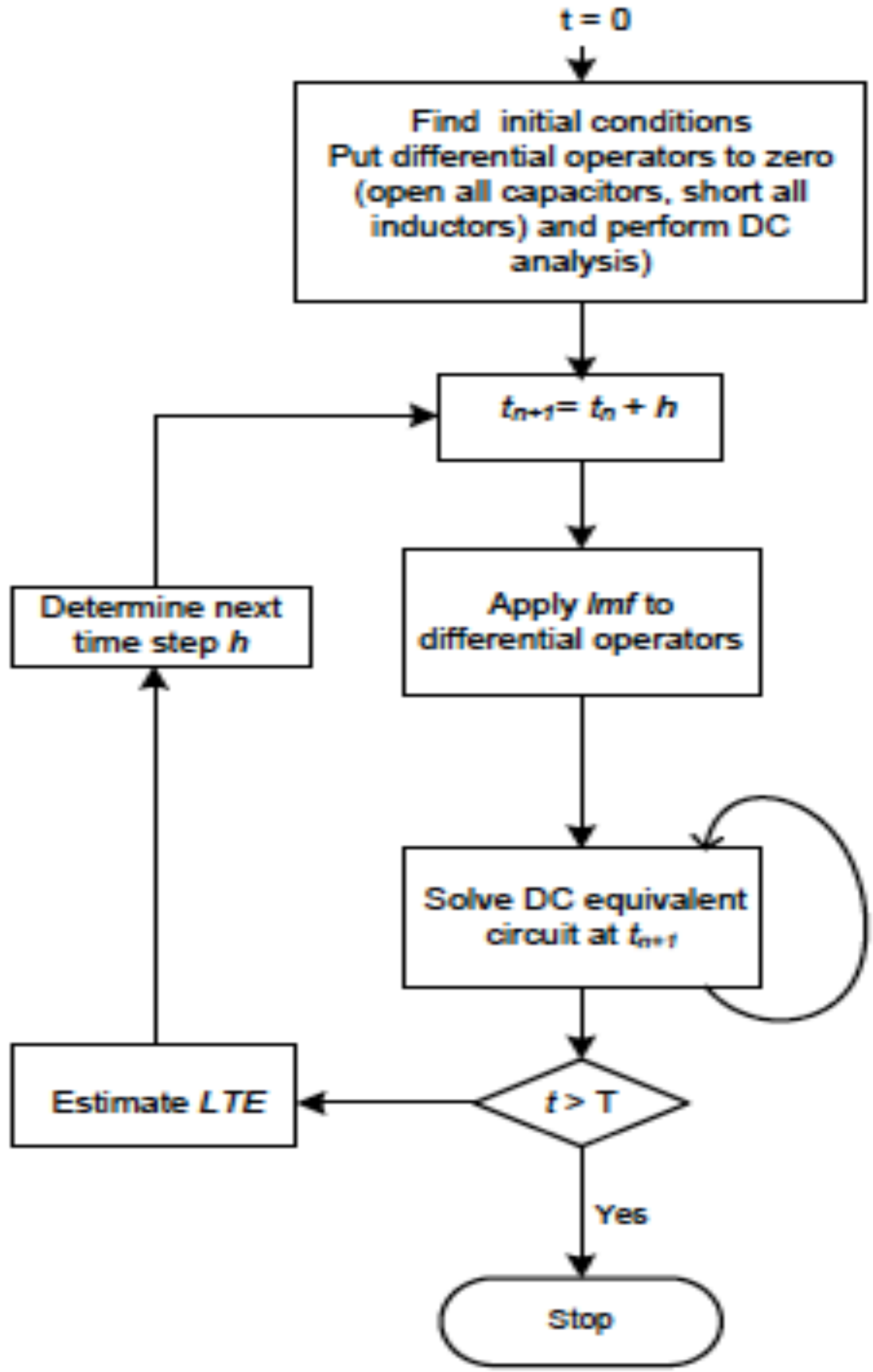


Figure 9.2.1: A flowchart of a typical transient analysis program.

Capacitor 'companion' models using lmf.

Consider an **implicit** *lmf*:

$$\dot{x}_n = \frac{1}{h\beta_o} x_n - \underbrace{\frac{1}{h\beta_o}\left[\sum_{i=1}^{k}(\alpha_i x_{n-i} + h\beta_i\, \dot{x}_{n-i})\right]}_{\text{known}}$$

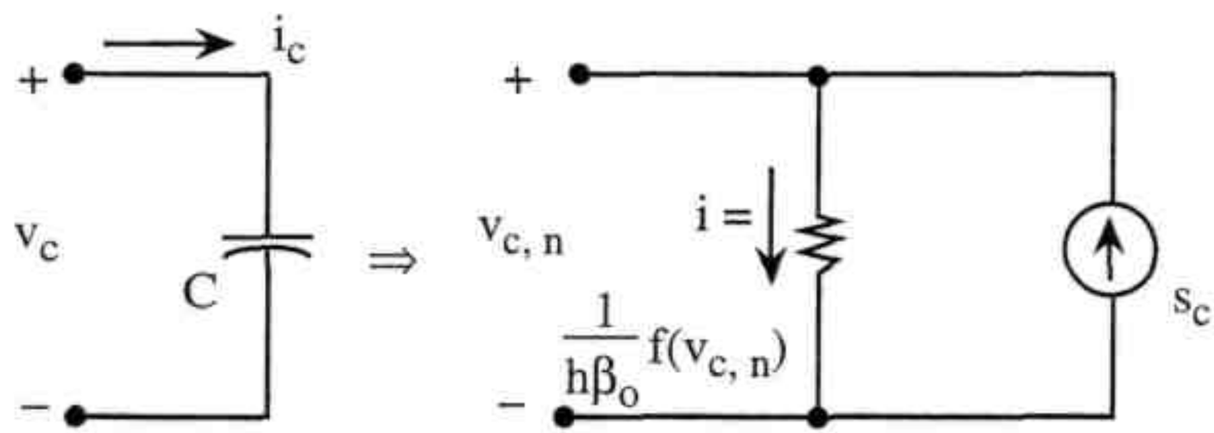
Apply to capacitor characteristic equations

$$i_c = \frac{dq_c}{dt},\; q_c = f(v_c)$$

$$i_{c,\,n} = \dot{q}_{c,\,n} = \frac{1}{h\beta_o} q_{c,\,n} - \underbrace{\frac{1}{h\beta_o}\left[\sum_{i=1}^{k}(\alpha_i q_{c,\,n-i} + h\beta_i\, \dot{q}_{c,\,n-i})\right]}_{\text{known} \equiv s_c},\ \text{where } \dot{q}_{c,\,n-i} = i_{c,\,n-i}$$

Or

$$i_{c,n} = \underbrace{\frac{1}{h\beta_o}}_{\substack{\text{equiv. resistor} \\ \text{depends on } h}} f_c(v_{c,n}) - s_c;$$ s_c is an equivalent current source which depends on h, $v_{c,\,n-i}$, and $i_{c,\,n-i}$



Capacitor Companion Model

Nonlinear Capacitor Stamp – Implicit *lmf*

$$
\begin{array}{c}
\begin{array}{ccccc} & i & & j & \end{array}\\
\begin{bmatrix}
 & \vdots & & \vdots & \\
\cdots & +\frac{\mathrm{a}_{\mathrm{c,n}}^{(k)}}{h\beta_0} & \cdots & -\frac{\mathrm{a}_{\mathrm{c,n}}^{(k)}}{h\beta_0} & \cdots \\
 & \vdots & & \vdots & \\
\cdots & -\frac{\mathrm{a}_{\mathrm{c,n}}^{(k)}}{h\beta_0} & \cdots & +\frac{\mathrm{a}_{\mathrm{c,n}}^{(k)}}{h\beta_0} & \cdots \\
 & \vdots & & \vdots &
\end{bmatrix}
\end{array}
\begin{bmatrix} v_i \\ \\ v_j \end{bmatrix},
\begin{array}{c}
rhs \\
\begin{bmatrix}
-\frac{b_{s,n}^{(k)}}{h\beta_0} + s_{c,n} \\
\\
+\frac{b_{s,n}^{(k)}}{h\beta_0} - s_{c,n}
\end{bmatrix}
\end{array}
$$

Capacitor 'companion' model using B.E.

$$x_n = x_{n-1} + h\dot{x}_n$$

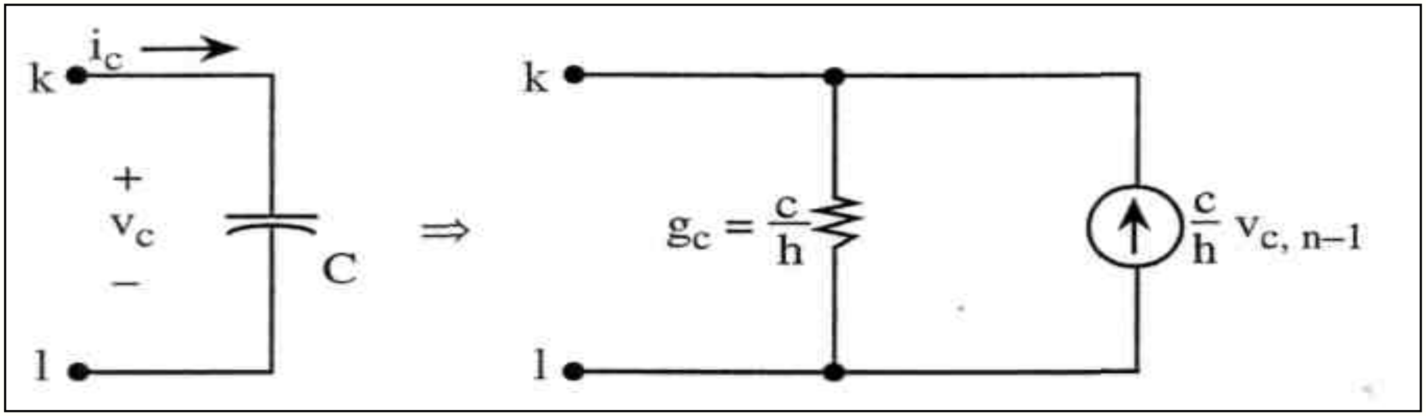
$$\dot{x}_n = (x_n - x_{n-1})/h$$

Linear Capacitor: $q = Cv \Rightarrow i = dq/dt = Cdv/dt$

$$i_n = \dot{q}_n = \frac{1}{h}(q_n - q_{n-1})$$

$$i_n = \frac{C}{h}v_n - \frac{C}{h}v_{n-1}$$

Linear Capacitor (B.E)



MNA Stamp:

$$
\begin{array}{c}
\begin{array}{ccccc} & i & & j & \end{array} \\
\begin{bmatrix}
 & \vdots & & \vdots & \\
\cdots & +\frac{C}{h} & \cdots & -\frac{C}{h} & \cdots \\
 & \vdots & & \vdots & \\
\cdots & -\frac{C}{h} & \cdots & +\frac{C}{h} & \cdots \\
 & \vdots & & \vdots &
\end{bmatrix}
\end{array}
\begin{bmatrix} v_i \\ v_j \end{bmatrix},
\begin{array}{c} rhs \\ \begin{bmatrix} +\frac{Cv_{n-1}}{h} \\ -\frac{Cv_{n-1}}{h} \end{bmatrix} \end{array}
$$

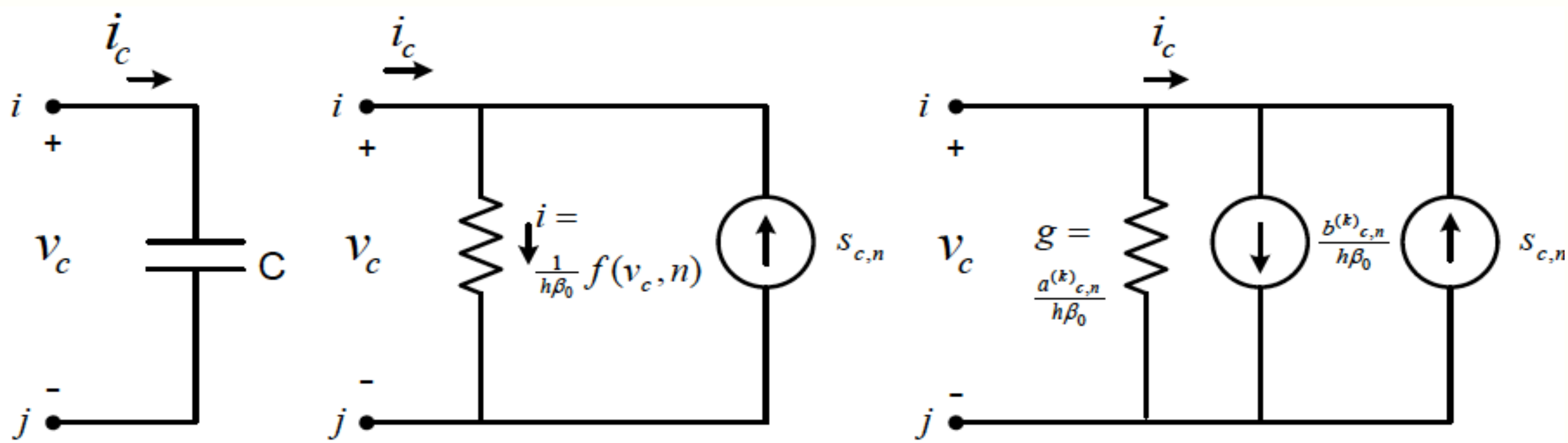
$v_n = v_i - v_j$

Nonlinear Capacitor (voltage-controlled): B.E.

$$q_c = f_c(v_c); \quad i_c = \frac{dq_c}{dt}$$

$$\dot{q}_{c,\,n} = \frac{1}{h} q_{c,\,n} - \frac{1}{h} q_{c,\,n-1} \qquad \text{(B.E)}$$

$$i_{c,\,n} = \frac{1}{h} f_c(v_{c,\,n}) - \frac{1}{h} f_c(v_{c,\,n-1})$$



$\beta_0 = 1$, $s_{c,n} = f_c(v_{c,n-1})/h$

Nonlinear Capacitor (voltage-controlled): B.E.

$$
\begin{array}{ccccccc}
 & & i & & j & & \\
\end{array}
$$

$$
\begin{bmatrix}
 & \vdots & & \vdots & \\
\cdots & +\frac{\mathrm{a}_{c,n}^{(k)}}{h\beta_0} & \cdots & -\frac{\mathrm{a}_{c,n}^{(k)}}{h\beta_0} & \cdots \\
 & \vdots & & \vdots & \\
\cdots & -\frac{\mathrm{a}_{c,n}^{(k)}}{h\beta_0} & \cdots & +\frac{\mathrm{a}_{c,n}^{(k)}}{h\beta_0} & \cdots \\
 & \vdots & & \vdots &
\end{bmatrix}
\begin{bmatrix}
v_i \\ \\ v_j
\end{bmatrix},
\begin{array}{c} rhs \\ \begin{bmatrix}
-\frac{b_{c,n}^{(k)}}{h\beta_0} + s_{c,n} \\ \\
+\frac{b_{c,n}^{(k)}}{h\beta_0} - s_{c,n}
\end{bmatrix}\end{array}
$$

$\beta_0 = 1$, $s_{c,n} = f_c(v_{c,n-1})/h$ (changes at every time point);
$a_{c,n}$ and $b_{c,n}$ change at every iteration point at time point t_n

Trapezoidal Rule

$$\boxed{x_n = x_{n-1} + \frac{h}{2}\left(\dot{x}_n + \dot{x}_{n-1}\right)}$$

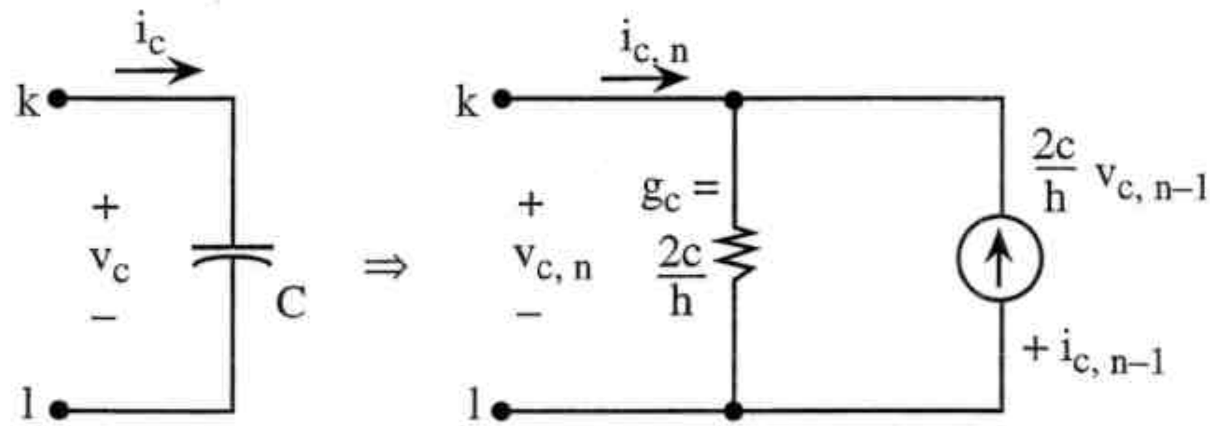
$$\boxed{\dot{x}_n = \frac{2}{h} x_n - \left(\frac{2}{h} x_{n-1} + \dot{x}_{n-1}\right)}$$

Linear Capacitor: T.R.

$$q_c = C v_c,\ i_c = \frac{dq_c}{dt}$$

$$\dot{q}_{c,\,n} = \frac{2}{h} q_{c,\,n} - \left(\frac{2}{h} q_{c,\,n-1} + \dot{q}_{c,\,n-1}\right)$$

$$i_{c,\,n} = \frac{2c}{h} v_{c,\,n} - \left(\frac{2c}{h} v_{c,\,n-1} + i_{c,\,n-1}\right)$$



Linear Capacitor Stamp – T.R.

$$\begin{array}{c} \begin{array}{ccccc} & i & & j & \end{array} \\ \begin{bmatrix} & \vdots & & \vdots & \\ \cdots & +\frac{2C}{h} & \cdots & -\frac{2C}{h} & \cdots \\ & \vdots & & \vdots & \\ \cdots & -\frac{2C}{h} & \cdots & +\frac{2C}{h} & \cdots \\ & \vdots & & \vdots & \end{bmatrix} \end{array} \begin{bmatrix} v_i \\ \\ v_j \end{bmatrix}, \quad \begin{array}{c} rhs \\ \begin{bmatrix} +\frac{2Cv_{n-1}}{h} + i_{n-1} \\ -\frac{2Cv_{n-1}}{h} - i_{n-1} \end{bmatrix} \end{array}$$

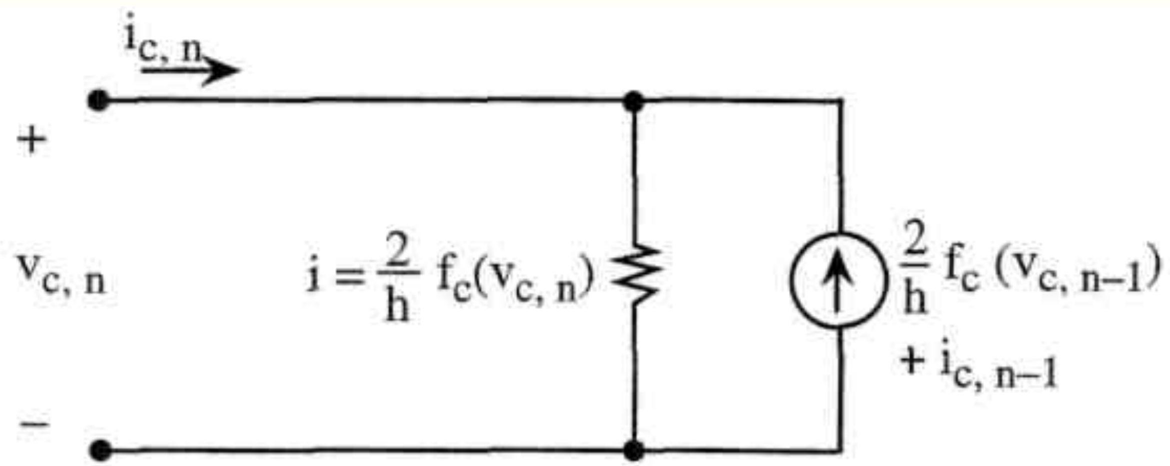
$$i_n = \frac{2C}{h} v_n - \frac{2C}{h} v_{n-1} - i_{n-1} \; ; \; i_0 = 0$$

Nonlinear Capacitor: T.R.

$$q_c = f_c(v_c); \;\; i_c = \frac{dq_c}{dt}$$

$$\dot{q}_{c,n} = \frac{2}{h} q_{c,n} - \left(\frac{2}{h} q_{c,n-1} + \dot{q}_{c,n-1}\right)$$

$$i_{c,n} = \frac{2}{h} f_c(v_{c,n}) - \left(\frac{2}{h} f_c(v_{c,n-1}) + i_{c,n-1}\right)$$



Nonlinear Capacitor: T.R.

$$
\begin{array}{c}
\begin{array}{ccccccc} & i & & j & \end{array} \\
\begin{bmatrix}
 & \vdots & & \vdots & \\
\cdots & +\frac{\mathrm{a}_{c,n}^{(k)}}{h\beta_0} & \cdots & -\frac{\mathrm{a}_{c,n}^{(k)}}{h\beta_0} & \cdots \\
 & \vdots & & \vdots & \\
\cdots & -\frac{\mathrm{a}_{c,n}^{(k)}}{h\beta_0} & \cdots & +\frac{\mathrm{a}_{c,n}^{(k)}}{h\beta_0} & \cdots \\
 & \vdots & & \vdots &
\end{bmatrix}
\end{array}
\begin{bmatrix} v_i \\ \\ v_j \end{bmatrix},
\begin{array}{c}
rhs \\
\begin{bmatrix}
-\frac{b_{c,n}^{(k)}}{h\beta_0} + s_{c,n} \\
\\
+\frac{b_{c,n}^{(k)}}{h\beta_0} - s_{c,n}
\end{bmatrix}
\end{array}
$$

$\beta_0 = 1/2$, $s_{c,n} = 2f_c(v_{c,n-1})/h + i_{c,n-1}$ (changes at every time point);
$a_{c,n}$ and $b_{c,n}$ change at every iteration point at time point t_n

2nd Order Backward Differentiation Formula BDF

$$x_n = \frac{4}{3}x_{n-1} - \frac{1}{3}x_{n-2} + \frac{2}{3}h\dot{x}$$

$$\dot{x} = \frac{3}{2h}(x_n - \frac{4}{3}x_{n-1} + \frac{1}{3}x_{n-2})$$

Linear Capacitor: $q = Cv, \quad i = dq/dt$

$$i_n = \frac{3C}{2h}v_n - \frac{2C}{h}v_{n-1} + \frac{C}{2h}v_{n-2}$$

Linear Capacitor Stamp: BDF

$$
\begin{array}{c}
\begin{array}{ccccc} & i & & j & \end{array}\\
\begin{bmatrix} & \vdots & & \vdots & \\ \cdots & +\frac{3C}{2h} & \cdots & -\frac{3C}{2h} & \cdots \\ & \vdots & & \vdots & \\ \cdots & -\frac{3C}{2h} & \cdots & +\frac{3C}{2h} & \cdots \\ & \vdots & & \vdots & \end{bmatrix}
\end{array}
\begin{bmatrix} i \\ \\ j \end{bmatrix},
\begin{array}{c} rhs \\ \begin{bmatrix} +\frac{2Cv_{n-1}}{h} - \frac{Cv_{n-2}}{2h} \\ -\frac{2Cv_{n-1}}{h} + \frac{Cv_{n-2}}{2h} \end{bmatrix} \end{array}
$$

Charge-Controlled Capacitor

$$\mathbf{v_c} = \mathbf{f}_c(\mathbf{q_c}), \quad \mathbf{i_c} = \frac{d\mathbf{q_c}}{dt}$$

$$\mathbf{i_{c,n}} = \dot{\mathbf{q}}_{\mathbf{c,n}} = \frac{1}{h\beta_0}\mathbf{q_{c,n}} - \mathbf{s_{c,n}} \qquad (1)$$

$$\mathbf{s_{c,n}} = \tfrac{1}{h\beta_0}\sum_{i=1}^{k}(\alpha_i \mathbf{q_{c,n-i}} + h\beta_i \mathbf{i}_{c,n-i}).$$

$$\mathbf{v_{c,n}} = \mathbf{f}_c(\mathbf{q}_{c,n})$$

$$\mathbf{v}_{c,n} = \mathbf{B}_{c,n}^{(k)}\mathbf{q}_{c,n} + \mathbf{c}_{c,n}^{(k)} \qquad (2)$$

Stamp of Two-Terminal Charge-Controlled Capacitor: B.E. and ENA

$$i_{c,n} = \frac{1}{h}(q_{c,n} - q_{c,n-1})$$

$$v_{c,n} = b_{c,n}^{(k)} q_{c,n} + c_{c,n}^{(k)}$$

$$\begin{array}{c} \begin{array}{ccc} \quad i \quad & \quad j \quad & \quad \quad \end{array} \\ \begin{bmatrix} \vdots & \vdots & +\frac{1}{h} \\ \vdots & \vdots & -\frac{1}{h} \\ +1 & -1 & -b_{c,n}^{(k)} \end{bmatrix} \end{array} \begin{bmatrix} v_i \\ v_j \\ q_c \end{bmatrix}, \begin{bmatrix} +\frac{q_{c,n-1}}{h} \\ -\frac{q_{c,n-1}}{h} \\ +c_{c,n}^{(k)} \end{bmatrix}$$

Charge-and-Voltage Controlled Capacitor

$$\mathbf{f_c}(\mathbf{v_c}, \mathbf{q_c}) = 0,\ \mathbf{i_c} = \frac{d\mathbf{q_c}}{dt}$$

$$\mathbf{i_{c,n}} = \dot{\mathbf{q}}_{\mathbf{c,n}} = \frac{1}{h\beta_0}\mathbf{q_{c,n}} - \mathbf{s_{c,n}}$$

$$\mathbf{A}_{c,n}^{(k)}\mathbf{v}_{c,n} + \mathbf{B}_{c,n}^{(k)}\mathbf{q}_{c,n} + \mathbf{c}_{c,n}^{(k)} = \mathbf{0}$$

$$\mathbf{c}_{c,n}^{(k)} = \mathbf{f}_c(\mathbf{q}_{c,n}^{(k)}, \mathbf{v}_{c,n}^{(k)}) - \mathbf{A}_{c,n}^{(k)}\mathbf{v}_{c,n}^{(k)} - \mathbf{B}_{c,n}^{(k)}\mathbf{q}_{c,n}^{(k)}$$

Stamp of Two-Terminal Charge-and-Voltage-Controlled Capacitor: B.E. and ENA

$$i_{c,n} = \frac{1}{h}(q_{c,n} - q_{c,n-1})$$

$$a^{(k)}_{c,n)} v_{c,n} + b^{(k)}_{c,n)} q_{c,n} + c^{(k)}_{c,n} = 0$$

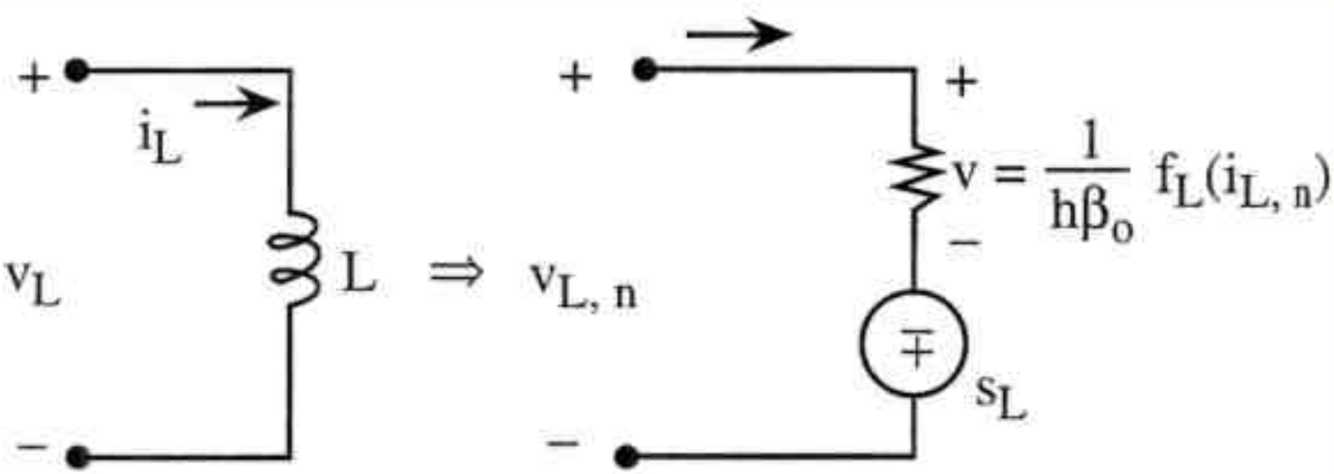
$$\begin{array}{c} \begin{array}{ccc} i & j & \phantom{+b^{(k)}_{c,n}} \end{array} \\ \left[\begin{array}{ccc} \vdots & \vdots & +\frac{1}{h} \\ \vdots & \vdots & -\frac{1}{h} \\ +a^{(k)}_{c,n} & -a^{(k)}_{c,n} & +b^{(k)}_{c,n} \end{array}\right] \end{array} \left[\begin{array}{c} v_i \\ v_j \\ q_c \end{array}\right], \left[\begin{array}{c} +\frac{q_{c,n-1}}{h} \\ -\frac{q_{c,n-1}}{h} \\ -c^{(k)}_{c,n} \end{array}\right]$$

Inductor

$$v_L = \frac{d\phi_L}{dt}, \ \phi_L = f_L(i_L)$$

$$v_{L,n} = \dot{\phi}_{L,n} = \frac{1}{h\beta_o}\phi_{L,n} - \underbrace{\frac{1}{h\beta_o}\left[\sum_{i=1}^{k}(\alpha_i \phi_{L,n-i} + h\beta_i \dot{\phi}_{n-i})\right]}_{\text{known} \equiv s_L}, \ \dot{\phi}_{n-i} = v_{L,n-i}$$

$$v_{L,n} = \frac{1}{h\beta_o} f(i_{L,n}) - s_L$$



Inductor Companion Model

Linear Inductor: B.E.

$$\boxed{\phi_L = L\, i_L; \;\; v_L = \frac{d\phi_L}{dt}}$$

$$\text{B.E.:}\;\; \dot{\phi}_{L,n} = \frac{1}{h}\left(\phi_{L,n} - \phi_{L,n-1}\right)$$

$$v_{L,n} = \frac{L}{h} i_{L,n} - \frac{L}{h} i_{L,n-1}$$

Stamp of Linear Inductor: B.E.

$$
\begin{array}{ccc}
\begin{array}{ccccc} & i & & j & i_L \end{array} & & rhs \\
\begin{bmatrix} & \vdots & & \vdots & \\ \cdots & & \cdots & & +1 \\ \cdots & & \cdots & & -1 \\ & +1 & & -1 & -\frac{L}{h} \end{bmatrix}
\begin{bmatrix} \\ v_i \\ v_j \\ i_L \end{bmatrix}, &
&
\begin{bmatrix} \\ \\ \\ -\frac{L}{h} i_{L,n-1} \end{bmatrix}
\end{array}
$$

Linear Inductor: T.R.

$$\phi_L = L\, i_L; \;\; v_L = \frac{d\phi_L}{dt}$$

T.R. $$\dot{x}_n = \frac{2}{h} x_n - \left(\frac{2}{h} x_{n-1} + \dot{x}_{n-1}\right)$$

$$\dot{\phi}_{L,\,n} = \frac{2}{h} \phi_{L,\,n} - \left(\frac{2}{h} \phi_{L,\,n-1} + \dot{\phi}_{L,\,n-1}\right)$$

$$v_{L,\,n} = \frac{2}{h} L\, i_{L,\,n} - \left(\frac{2}{h} L\, i_{L,\,n-1} + v_{L,\,n-1}\right)$$

Stamp of Linear Inductor: T.R.

$$
\begin{array}{c}
\begin{array}{ccccc} & i & & j & i_L \end{array} \\
\begin{bmatrix}
 & \vdots & & \vdots & \\
\cdots & & \cdots & & +1 \\
 & & & & \\
\cdots & & \cdots & & -1 \\
 & +1 & & -1 & -\frac{2L}{h}
\end{bmatrix}
\end{array}
\begin{bmatrix}
 \\
v_i \\
 \\
v_j \\
i_L
\end{bmatrix},
\begin{array}{c}
rhs \\
\begin{bmatrix}
 \\
 \\
 \\
-\frac{2L}{h} i_{L,n-1} - v_{L,n-1}
\end{bmatrix}
\end{array}
$$

Linear Inductor: 2nd Order BDF

$$\boxed{\phi_L = L\, i_L; \;\; v_L = \frac{d\phi_L}{dt}}$$

$$\dot{\phi}_{L,n} = \frac{3}{2h}\phi_{L,n} - \frac{2}{h}\phi_{L,n-1} + \frac{1}{2h}\phi_{L,n-2}$$

$$v_{L,n} = \frac{3L}{2h}i_{L,n} - \frac{2L}{h}i_{L,n-1} + \frac{L}{2h}i_{L,n-2}$$

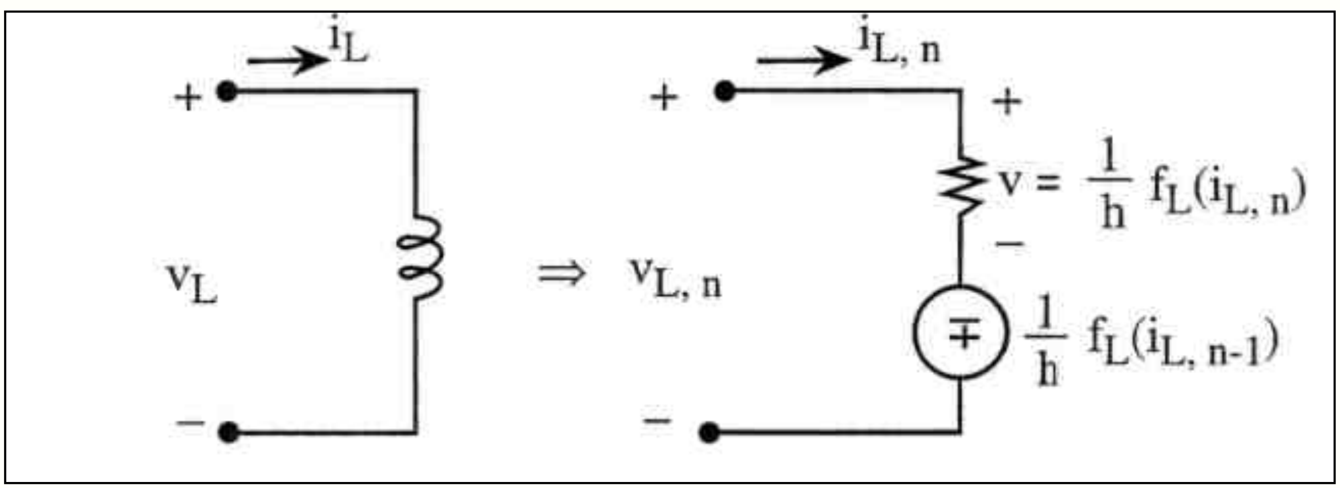
$$\begin{array}{c} \begin{matrix} \quad & i & \quad & j & i_L \end{matrix} \\ \begin{bmatrix} & \vdots & & \vdots & \\ \cdots & & \cdots & & +1 \\ \cdots & & \cdots & & -1 \\ & +1 & & -1 & -\frac{3L}{2h} \end{bmatrix} \end{array} \begin{array}{c} \\ \begin{bmatrix} \\ v_i \\ v_j \\ i_L \end{bmatrix} \end{array}, \quad \begin{array}{c} rhs \\ \begin{bmatrix} \\ \\ \\ -\frac{2L}{h}i_{L,n-1} - \frac{L}{2h}i_{L,n-2} \end{bmatrix} \end{array}$$

Nonlinear Current-Controlled Inductor

$\Phi_L = f_L(i_L)$; $v_L = (d\Phi_L)/(d_t)$

$\Phi_{L,n} = (1/h)\,\Phi_{L,n} - (1/h)\,\Phi_{L,n-1}$ (B.E)

$= (1/h)f_L(i_{L,\,n}) - (1/h)f_L(i_{L,\,n\text{-}l})$



Stamp of Linearized Current-Controlled Inductor

$$
\begin{array}{c}
\begin{array}{cccccc} & i & & j & & i_L \end{array} \\
\left[\begin{array}{cccccc}
 & \vdots & & \vdots & & \\
\cdots & & \cdots & & \cdots & +1 \\
 & \vdots & & \vdots & & \\
\cdots & & \cdots & & & -1 \\
 & +1 & & -1 & & -\frac{\mathrm{a}_{\mathrm{L,n}}^{(k)}}{h\beta_0}
\end{array}\right]
\end{array}
\begin{bmatrix} \\ v_i \\ \\ v_j \\ i_L \end{bmatrix}
,\quad
\begin{array}{c}
rhs \\
\begin{bmatrix} \\ \\ \\ \\ +\frac{\mathrm{b}_{\mathrm{L,n}}^{(k)}}{h\beta_0} - s_{L,n} \end{bmatrix}
\end{array}
$$

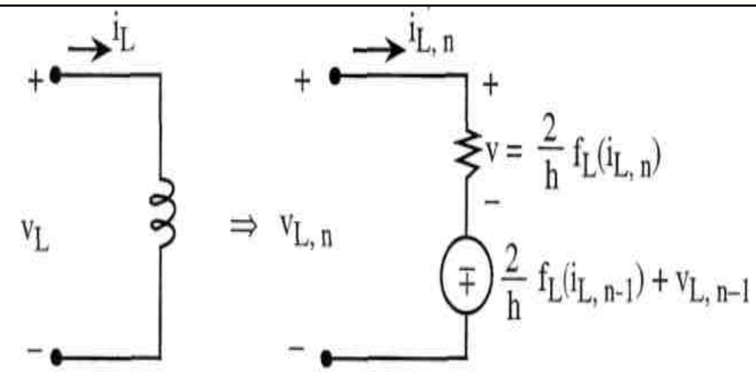
For B.E.: $\beta_0 = 1$, $s_{L,n} = f_L(i_{L,n-1})/h$

Nonlinear Inductor: T.R.

$$\phi_L = f_L(i_L); \quad v_L = \frac{d\phi_L}{dt}$$

$$\dot{\phi}_{L,n} = \frac{2}{h}\phi_{L,n} - \left(\frac{2}{h}\phi_{L,n-1} + \dot{\phi}_{L,n-1}\right)$$

$$v_{L,n} = \frac{2}{h} f_L(i_{L,n}) - \left(\frac{2}{h} f_L(i_{L,n-1}) + v_{L,n-1}\right)$$



Stamp of Linearized Current-Controlled Inductor: T.R.

$$
\begin{array}{cccccc}
 & i & & j & & i_L
\end{array}
\qquad\qquad\qquad\qquad
rhs
$$

$$
\begin{bmatrix}
 & \vdots & & \vdots & & \\
\cdots & & \cdots & & \cdots & +1 \\
 & \vdots & & \vdots & & \\
\cdots & & \cdots & & & -1 \\
 & +1 & & -1 & & -\frac{\mathrm{a}_{\mathrm{L,n}}^{(k)}}{h\beta_0}
\end{bmatrix}
\begin{bmatrix}
 \\
v_i \\
 \\
v_j \\
i_L
\end{bmatrix},
\begin{bmatrix}
 \\
 \\
 \\
 \\
+\frac{\mathrm{b}_{\mathrm{L,n}}^{(k)}}{h\beta_0} - s_{L,n}
\end{bmatrix}
$$

$\beta_0 = 1/2$, $s_{L,n} = 2\, f_L(i_{L,n-1})/h + v_{L,n-1}$

Flux-Controlled Inductor

$$\mathbf{i_L} = \mathrm{f}_L(\phi_\mathbf{L}),\ \mathbf{v_L} = \frac{d\phi_\mathbf{L}}{dt}$$

$$\text{B.E.:}\ \dot{\phi}_{L,n} = \tfrac{1}{h}(\phi_{L,n} - \phi_{L,n-1})$$

$$v_{L,n} = \frac{1}{h}(\phi_{L,n} - \phi_{L,n-1})$$

$$i_{L,n} = b_{L,n}^{(k)}\phi_{L,n} + c_{L,n}^{(k)}$$

Stamp of (Linearized) Flux-Controlled Inductor: B.E. (ENA)

$$\begin{array}{c} \begin{matrix} \quad i & \quad\quad j & \quad\quad\quad\quad \end{matrix} \\ \begin{bmatrix} \vdots & \vdots & +b_{L,n}^{(k)} \\ \vdots & \vdots & -b_{L,n}^{(k)} \\ +1 & -1 & -\frac{1}{h} \end{bmatrix} \end{array} \begin{bmatrix} v_i \\ \\ v_j \\ \phi_L \end{bmatrix}, \begin{bmatrix} -c_{L,n}^{(k)} \\ \\ +c_{L,n}^{(k)} \\ -\frac{\phi_{L,n-1}}{h} \end{bmatrix}$$

Flux-and-Current Controlled Inductor

$$\mathbf{f}_{\mathrm{L}}(\mathbf{i}_{\mathrm{L}}, \phi_{\mathrm{L}}) = \mathbf{0},\ \mathbf{v}_{\mathrm{L}} = \frac{d\phi_{\mathrm{L}}}{dt}$$

$$\text{B.E.:}\ \ \dot{\phi}_{L,n} = \tfrac{1}{h}(\phi_{L,n} - \phi_{L,n-1})$$

$$v_{L,n} = \frac{1}{h}(\phi_{L,n} - \phi_{L,n-1})$$

$$a_{L,n}^{(k)} i_{L,n} + b_{L,n}^{(k)} \phi_{L,n} + c_{L,n}^{(k)} = 0$$

Stamp of (Linearized) Flux-and-Current Controlled Inductor: B.E. (ENA)

$$
\begin{array}{c}
\begin{array}{cccc} \quad i \quad & \quad j \quad & \qquad & \qquad \end{array} \\
\begin{bmatrix}
\vdots & \vdots & +1 & \vdots \\
\vdots & \vdots & -1 & \vdots \\
 & & +a_{L,n}^{(k)} & +b_{L,n}^{(k)} \\
+1 & -1 & 0 & -\frac{1}{h}
\end{bmatrix}
\end{array}
\begin{bmatrix}
v_i \\
v_j \\
i_L \\
\phi_L
\end{bmatrix},
\begin{bmatrix}
 \\
 \\
-c_{L,n}^{(k)} \\
-\frac{\phi_{L,n-1}}{h}
\end{bmatrix}
$$

Current-Controlled Memrisitive System

$$v_M(t) = f_1(\mathbf{x}, i_M, t) i_M(t)$$

$$\frac{d\mathbf{x}}{dt} = f_2(\mathbf{x}, i_M, t)$$

$$\frac{1}{h}(x_n - x_{n-1}) = f_2(x_n, i_{M,n}, t_n)$$

$$v_{M,n} = a_n^{(k)} x_n + b_n^{(k)} i_{M,n} + c_n^{(k)}$$

$$\frac{x_n}{h} - d_n^{(k)} x_n - e_n^{(k)} i_{M,n} = g_n^{(k)} + \frac{x_{n-1}}{h}$$

Stamp of (Linearized) Current Controlled Memristive system: B.E. (ENA)

$$
\begin{array}{c}
\begin{array}{cccc} \;\; i & \;\; j & \qquad & \qquad \end{array} \\
\begin{bmatrix}
\vdots & \vdots & +1 & \vdots \\
\vdots & \vdots & -1 & \vdots \\
+1 & -1 & -b_n^{(k)} & -a_n^{(k)} \\
\vdots & \vdots & -e_n^{(k)} & \left(\frac{1}{h} - d_n^{(k)}\right)
\end{bmatrix}
\end{array}
\begin{bmatrix} v_i \\ v_j \\ i_M \\ x \end{bmatrix},
\begin{bmatrix} \\ \\ +c_n^{(k)} \\ +s_{x,n}^{(k)} \end{bmatrix}
$$

where $s_{x,n}^{(k)} = g_n^{(k)} + \frac{x_{n-1}}{h}$.

Charge-Controlled Memristor

$$\phi_M(t) = f_M(q_M),\ i_M(t) = \frac{dq_M}{dt},\ v_M(t) = \frac{d\phi_M}{dt}$$

$$i_{M,n} = \frac{1}{h}(q_{M,n} - q_{M,n-1})$$

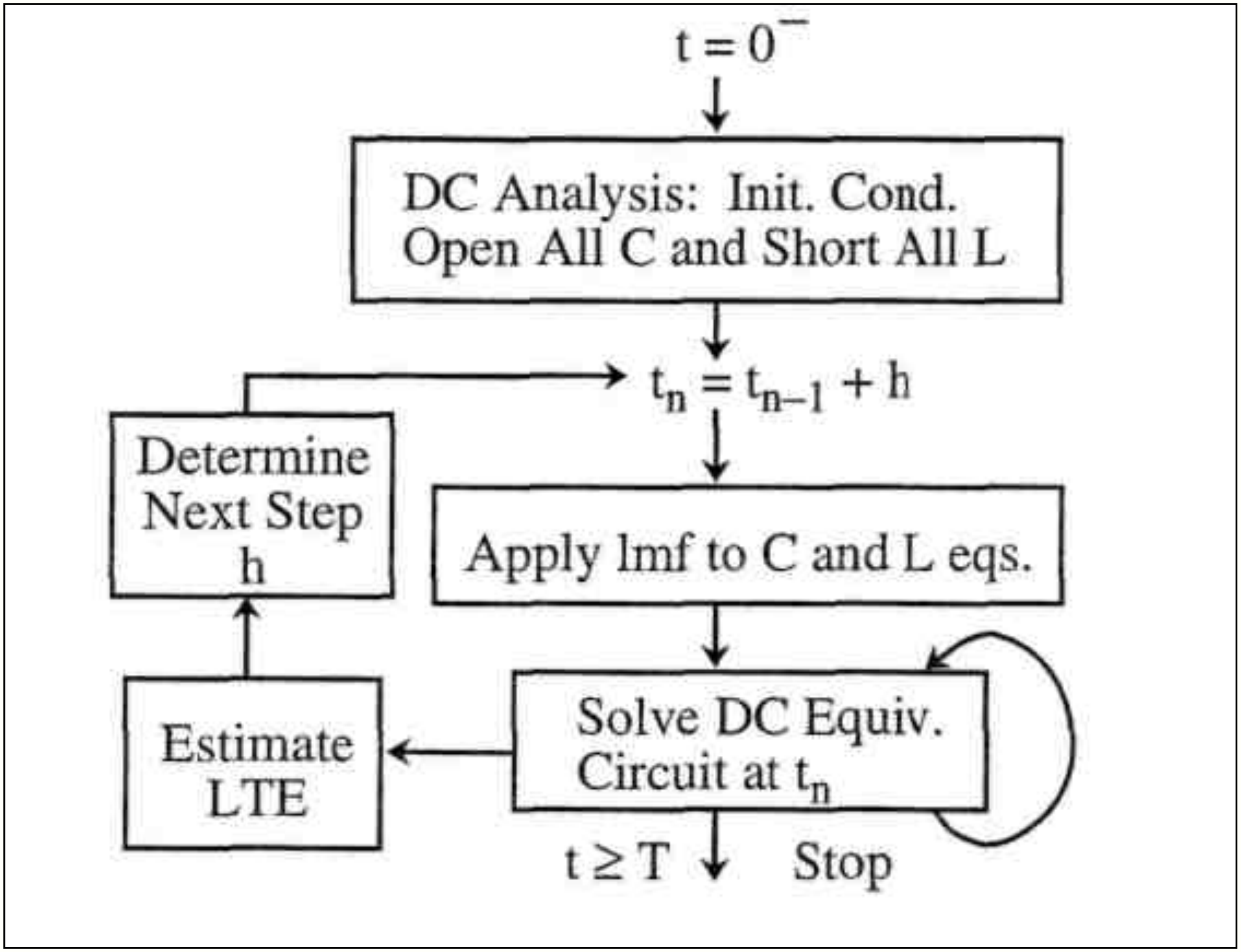
$$v_{M(n)} = \frac{1}{h}(\phi_{M,n} - \phi_{M,n-1})$$

$$v_{M(n)} = \frac{1}{h}(a_n^{(k)} q_{M,n} + b_n^{(k)} - \phi_{M,n-1})$$

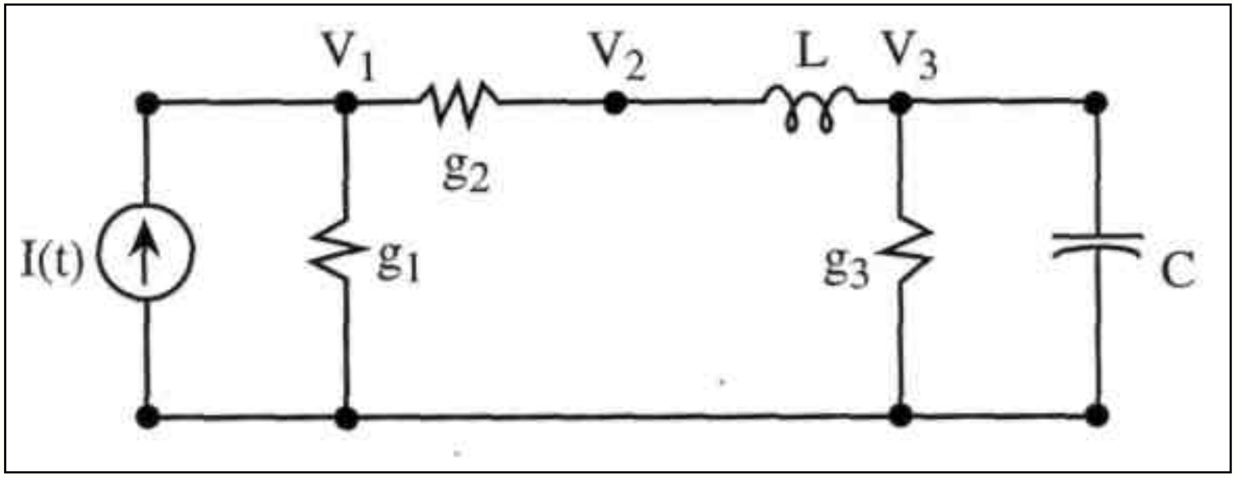
Stamp of (Linearized) Current Controlled Memristive system: B.E. (ENA)

$$\begin{array}{c} \begin{array}{ccc} i \quad & \quad j \quad & \quad \end{array} \\ \begin{bmatrix} \vdots & \vdots & +\frac{1}{h} \\ \vdots & \vdots & -\frac{1}{h} \\ +1 & -1 & -\frac{a_n^{(k)}}{h} \end{bmatrix} \end{array} \begin{bmatrix} v_i \\ v_j \\ q_M \end{bmatrix}, \begin{bmatrix} +\frac{q_{M,n-1}}{h} \\ -\frac{q_{M,n-1}}{h} \\ \frac{b_n^{(k)} - \phi_{M,n-1}}{h} \end{bmatrix}$$

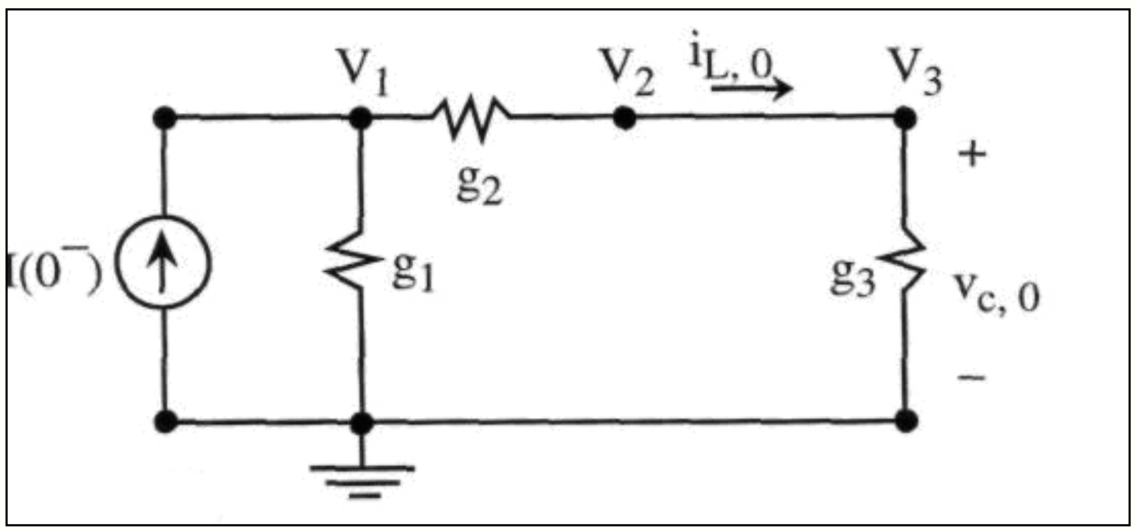
Transient Analysis Flowchart



Example



t = 0



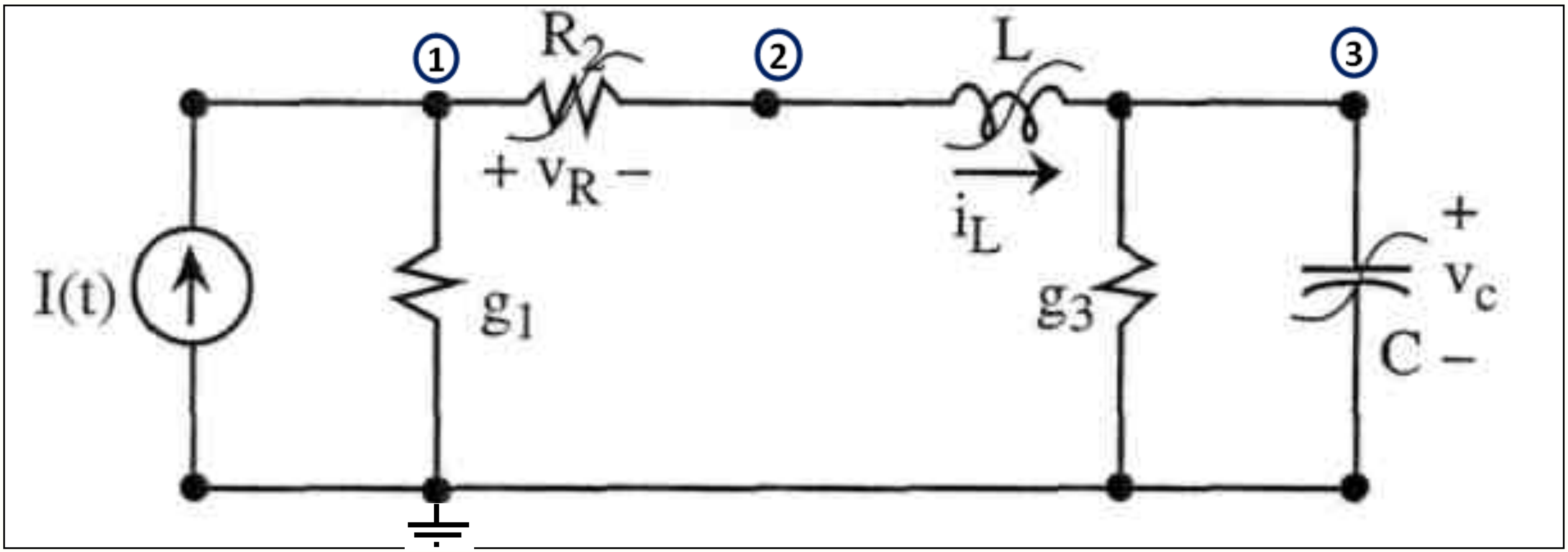
$$\underline{\text{B.E.}} \qquad C:\ i_{c,n} = \frac{C}{h} v_{c,n} - \frac{C}{h} v_{c,n-1}$$

$$L:\ v_{L,n} = \frac{L}{h} i_{L,n} - \frac{L}{h} i_{L,n-1}$$

$$\left[\begin{array}{ccc|c} g_1+g_2 & -g_2 & & \\ -g_2 & +g_2 & & +1 \\ & & \frac{C}{h}+g_3 & -1 \\ \hline & +1 & -1 & -\frac{L}{h} \end{array}\right] \left[\begin{array}{c} V_1 \\ V_2 \\ V_3 \\ \hline i_L \end{array}\right] = \left[\begin{array}{c} I(t_n) \\ 0 \\ \frac{C}{h} v_{c,n-1} \\ \hline -\frac{L}{h} i_{L,n-1} \end{array}\right]$$

At $t = 0^-$ => put $C = 0$, $L = 0$

Example (nonlinear circuit)



- Given:

$$i_R = v_R^3 + v_R$$

$$\phi_L = i_L^3$$

$$q_c = v_c^3$$

- Assume B.E. is used:

$$\dot{q}_{c,n} = \frac{1}{h} q_{c,n} - \frac{1}{h} q_{c,n-1}$$

$$\text{C:}\quad i_{c,n} = \frac{1}{h} v_{c,n}^3 - \frac{1}{h} v_{c,n-1}^3$$

$$\text{L:}\quad v_{L,n} = \frac{1}{h} i_{L,n}^3 - \frac{1}{h} i_{L,n-1}^3$$

Suppose Newton's method is applied to find the solution at time tn; and Taylor series is used to linearize all nonlinear elements at Newton iteration point at tn.

$$V_{1,n} = V^*_{1,n};\quad V_{2,n} = V^*_{2,n};\quad V_{3,n} = V^*_{3,n};\quad i_{L,n} = i^*_{L,n}$$

Linearization

R: $i_R = v_R^3 + v_R$, where $V_R = V_1 - V_2$

$$i_{R,n} = \left(v_R^{*3} + v_R^*\right) + \left(3v_R^{*2} + 1\right)\left(v_{R,n} - v_R^*\right) = \left(3v_R^* + 1\right) v_{R,n} - 2v_R^{*3}$$

C: $i_{c,n} = \frac{1}{h} v_{c,n}^3 - \underbrace{\frac{1}{h} v_{c,n-1}^3}_{\text{known}}$

$$i_{c,n} = \frac{1}{h}\left(v_{c,n}^{*3} + 3v_{c,n}^{*2}\left(v_{c,n} - v_{c,n}^*\right)\right) - \frac{1}{h} v_{c,n-1}^3$$

$$= \underbrace{\frac{3v_{c,n}^{*2}}{h} v_{c,n} - \frac{2}{h} v_{c,n}^{*3}}_{\text{linearization}} - \underbrace{\frac{1}{h} v_{c,n-1}^3}_{\text{initial condition}}$$

Vc = V3

$$\text{L:}\quad v_{L,n} = \frac{1}{h} i_{L,n}^{3} - \frac{1}{h} i_{L,n-1}^{3}$$

$$v_{L,n} = \frac{1}{h}\left(i_{L,n}^{*3} + 3 i_{L,n}^{*2}\left(i_{L,n} - i_{L,n}^{*}\right)\right) - \frac{1}{h} i_{L,n-1}^{3}$$

$$= \underbrace{\frac{3 i_{L,n}^{*2}}{h} i_{L,n} - \frac{2}{h} i_{L,n}^{*3}}_{\text{Linearization}} - \underbrace{\frac{1}{h} i_{L,n-1}^{3}}_{\text{initial condition}}$$

$$\begin{bmatrix} g_1\left(3v_R^{*2}+1\right) & -\left(3v_R^{*2}+1\right) & 0 & 0 \\ -\left(3v_R^{*2}+1\right) & -\left(3v_R^{*2}+1\right) & 0 & +1 \\ 0 & 0 & g_3+\frac{3}{h}v_{c,n}^{*2} & -1 \\ 0 & +1 & -1 & -\frac{3}{h}i_{L,n}^{*2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ i_L \end{bmatrix} = \begin{bmatrix} 2v_R^{*3}+I(t_n) \\ -2v_R^{*3} \\ \frac{1}{h}\left(2v_{c,n}^{*3}+v_{c,n-1}^{3}\right) \\ -\frac{1}{h}\left(2i_{L,n}^{*3}+i_{L,n-1}^{3}\right) \end{bmatrix}$$

(1) How to estimate LTE in practice

Use **Finite Difference Interpolation:**

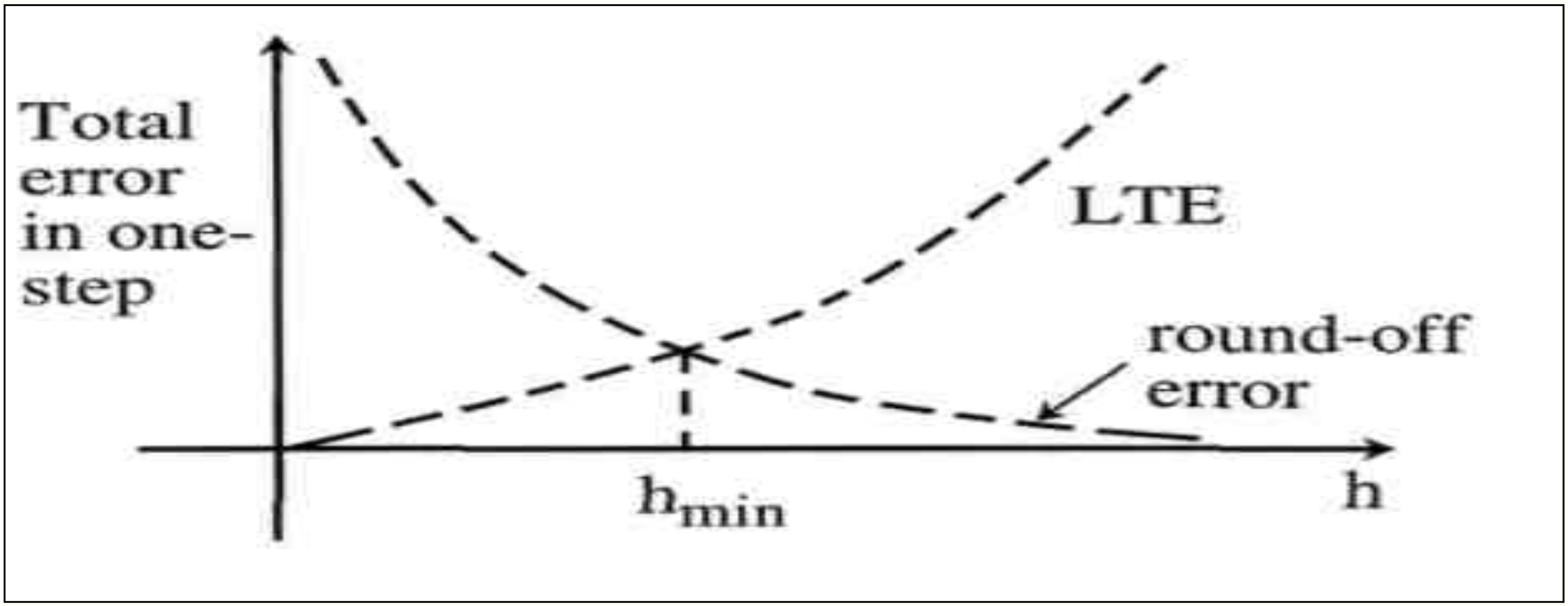
$$\frac{dx_n}{dt} = \frac{x_n - x_{n-1}}{t_n - t_{n-1}} \triangleq x[t_n, t_{n-1}]$$

$$\frac{d^2x_n}{dt^2} \cong 2\,\frac{x[t_n, t_{n-1}] - x[t_{n-1}, t_{n-2}]}{t_n - t_{n-2}}$$

$$\cong 2\,\frac{\dfrac{x_n - x_{n-1}}{t_n - t_{n-1}} - \dfrac{x_{n-1} - x_{n-2}}{t_{n-1} - t_{n-2}}}{t_n - t_{n-2}}$$

$$\frac{d^kx_n}{dt^k} = k!\,\frac{x[t_n, t_{n-1}, \ldots t_{n-k+1}] - x[t_{n-1}, \ldots t_{n-k}]}{t_n - t_{n-k}}$$

Note: LTE decreases with h, but when h decreases, round-off error increases.



(2) The timestep h is determined from the LTE estimate

$$\boxed{LTE = \left| C_{p+1}\, h^{p+1} \frac{d^{p+1}x}{dt^{p+1}} \right| \leq B}$$

$$\boxed{h = \sqrt[p+1]{\frac{B}{C_{p+1} \frac{d^{p+1}x}{dt^{p+1}}}}}$$

(3) Specify two bounds:

Upper bound B_u

Lower bound B_l

(i) If $B_l < \text{LTE} < B_u$, keep same *h*

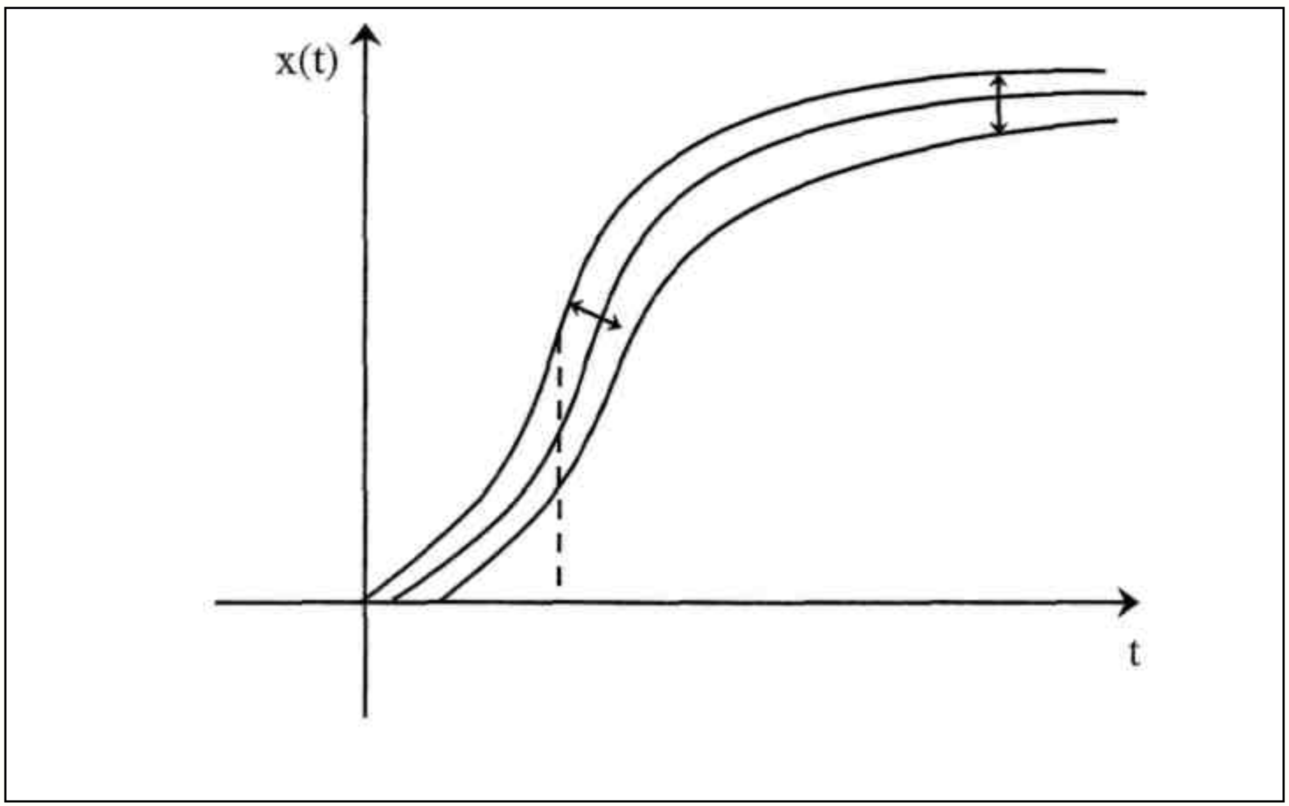
(ii) If $\text{LTE} > B_u$ —> reject x_n, decrease *h* and re-compute x_n

(iii) If $\text{LTE} < B_1$, accept x_n, increase next *h*

(4) Decrease or increase h so that

$$h = \sqrt[p+1]{\frac{B}{C_{p+1}\frac{d^{p+1}x}{dt^{p+1}}}}$$

$$B = \frac{B_u + B_l}{2}$$



(5) B_u and B_1 can be either constants, or variables as functions of $\dot{x}(t)$

$B = \varepsilon_r |\dot{x}| + \varepsilon_a$ (for both B_u, B_l)

$\Rightarrow$ Error bound is more relaxed when $|\dot{x}|$ is changing rapidly.

(6) Use voltage across each capacitor and current in each inductor to estimate LTE. Choose the largest one to determine h.

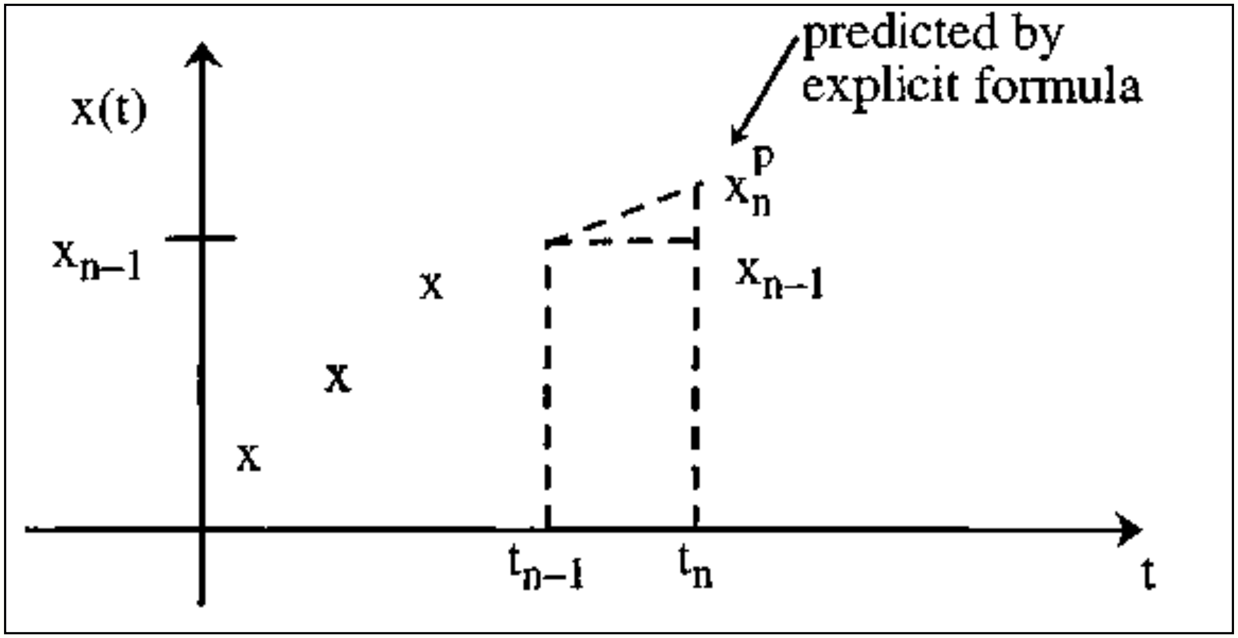
(7) Specify minimum time step h_{min}, based on computer word length — do not choose $h < h_{min}$ even if LTE is still too large.

(8) At time t = 0, after initial conditions are found, start with one-step formula (B.E. or T.R.) and with h_{min} for at least two steps before computing LTE and increasing h. After that, h can change and multistep formulas can be used.

(9) If the time response at any point in the circuit becomes discontinuous in t or its time derivative becomes discontinuous, restart with one-step formula and h_{min} after that time point.

(10) In solving for x_n at t_n using Newton's method, x_{n-1} is usually chosen as an initial guess. However, another initial guess can be used, such as obtained (or predicted) by an explicit formula: x_n^p

Companion models using explicit formulas



Explicit Formulas:

$$x_n = \sum_{i=1}^{k} (\alpha_i \, x_{n-i} + h \, \beta_i \, \dot{x}_{n-i})$$

e.g., F.E. $x_n = x_{n-1} + h \, \dot{x}_{n-1}$

If $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ is available, then $\mathbf{x}_n = \mathbf{x}_{n-1} + h\,\mathbf{f}(\mathbf{x}_{n-1}, t_{n-1})$ can be computed explicitly at time t_n in terms of previous solutions and time step h.

If state equations are not available, apply formula to capacitor and inductor equations to derive "explicit" companion models as follows.

Linear Cap

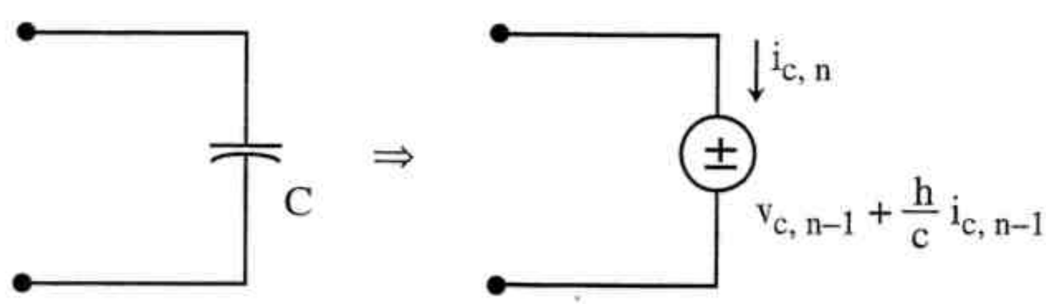
$$i_c = \frac{dq_c}{dt}\,,\; q_c = C\, v_c$$

F.E. $\quad x_n = x_{n-1} + h\,\dot{x}_{n-1}$

$$q_{c,n} = q_{c,n-1} + h\dot{q}_{c,n-1}$$

$$Cv_{c,n} = Cv_{c,n-1} + hi_{c,n-1}$$

$$v_{c,\,n} = \underbrace{v_{c,\,n-1} + \frac{h}{c}\, i_{c,\,n-1}}_{\text{known}}$$



at $t = 0$, $i_{c,\,0} = 0$

Nonlinear Cap

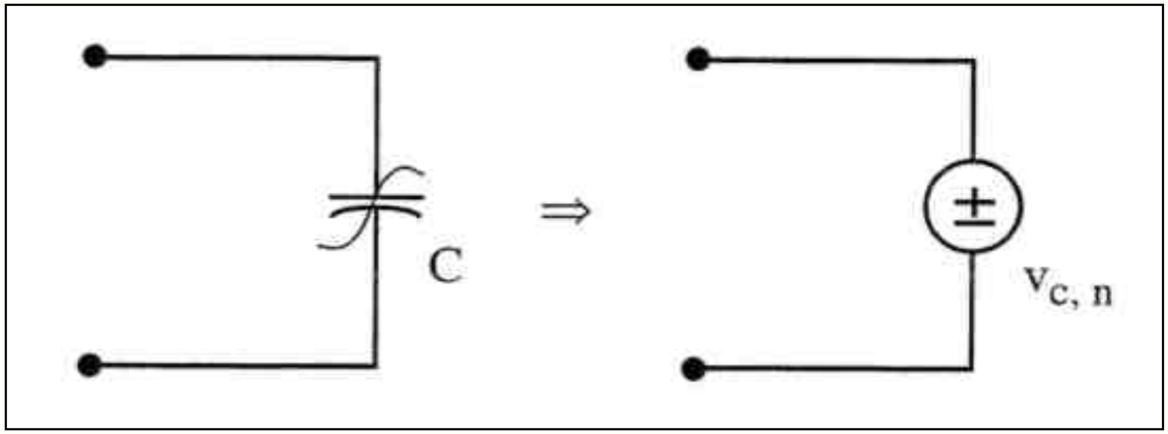
$$i_c = \frac{dq_c}{dt}, \; q_c = f_c(v_c)$$

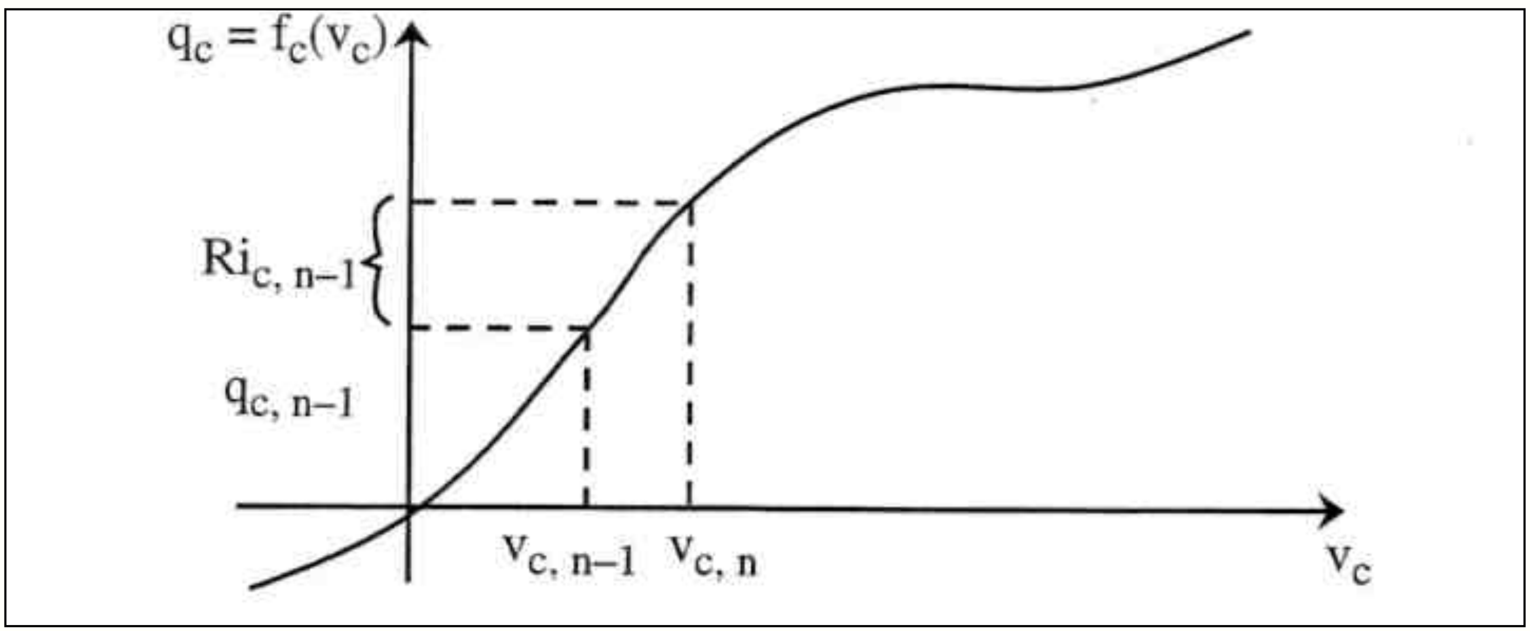
F.E. $$q_{c,\,n} = q_{c,\,n-1} + h\,\dot{q}_{c,\,n-1}$$

$$f_c(v_{c,\,n}) = q_{c,\,n-1} + h\,i_{c,\,n-1}$$

$$v_{c,\,n} = f_c^{-1}\,(q_{c,\,n-1} + h\,i_{c,\,n-1})$$

Nonlinear Cap (cont.)





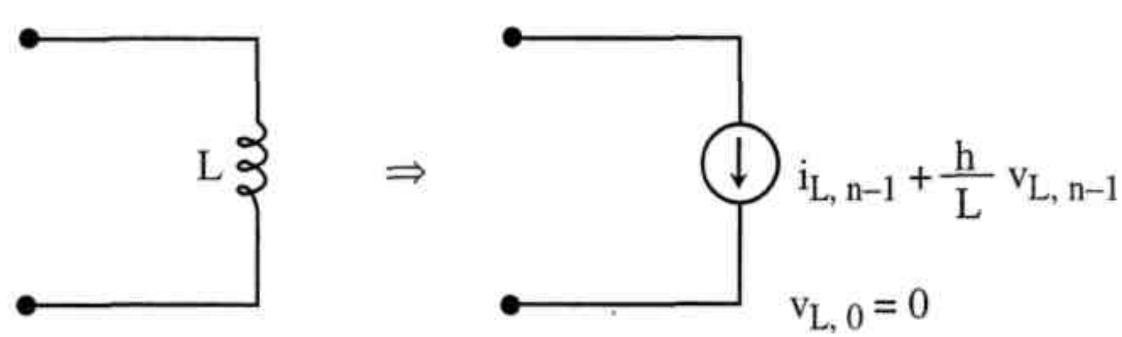
Linear Inductor

$$\phi_L = L\, i_L,\ v_L = \frac{d\phi}{dt}$$

F.E. $$\phi_{L,\,n} = \phi_{L,\,n-1} + h\,\dot{\phi}_{L,\,n-1}$$

$$L i_{L,\,n} = L i_{L,\,n-1} + h\, v_{L,\,n-1}$$

$$i_{L,\,n} = i_{L,\,n-1} + \frac{h}{L} v_{L,\,n-1}$$



Nonlinear Inductor

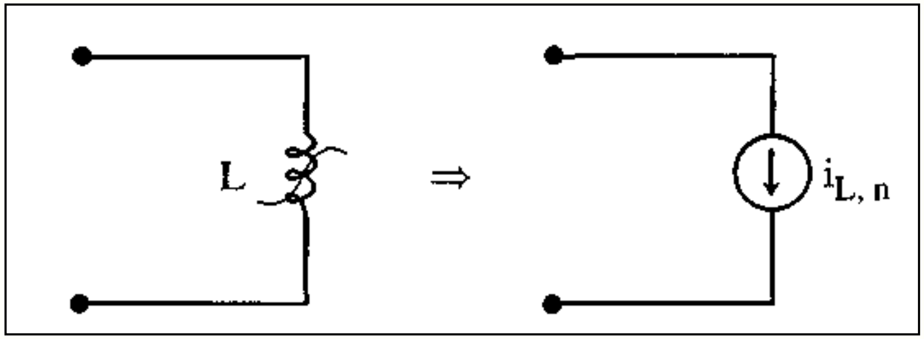
$$\phi_L = f_L(i_L),\ v_L = \frac{d\phi_L}{dt}$$

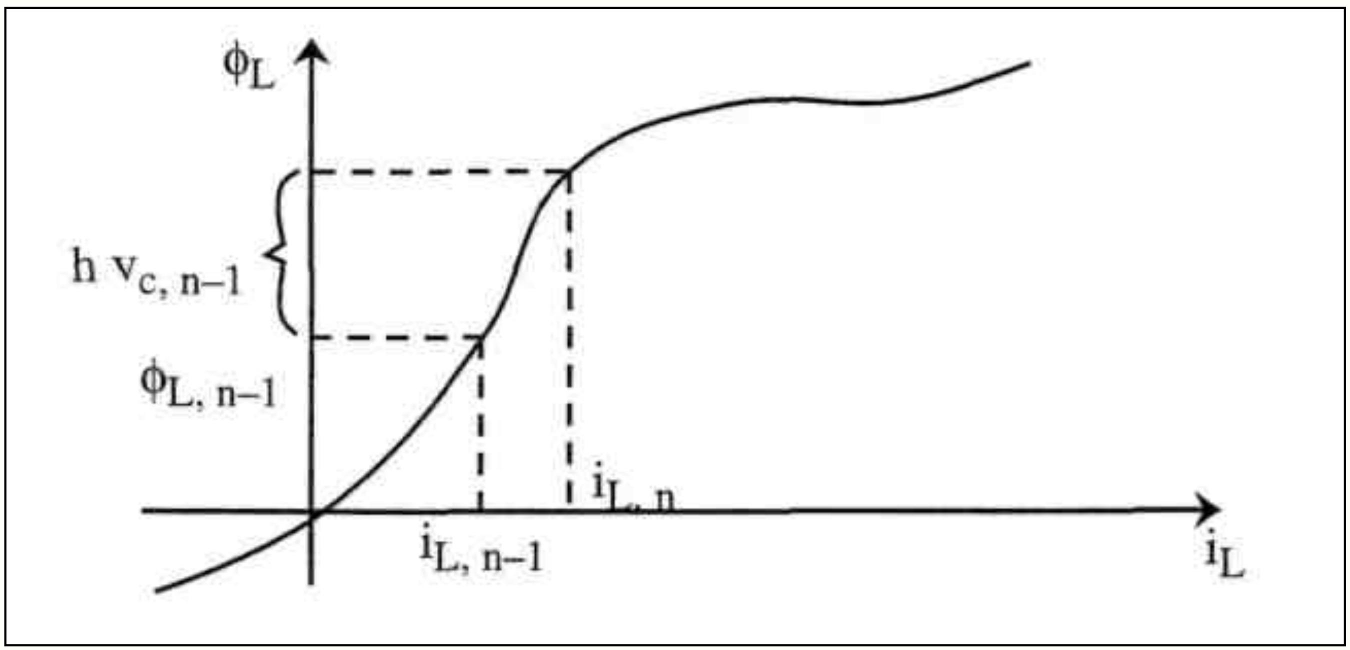
$$\phi_{L,\,n} = \phi_{L,\,n-1} + h\,\dot{\phi}_{L,\,n-1}$$

$$f_L(i_{L,\,n}) = \phi_{L,\,n-1} + h\,v_{L,\,n-1}$$

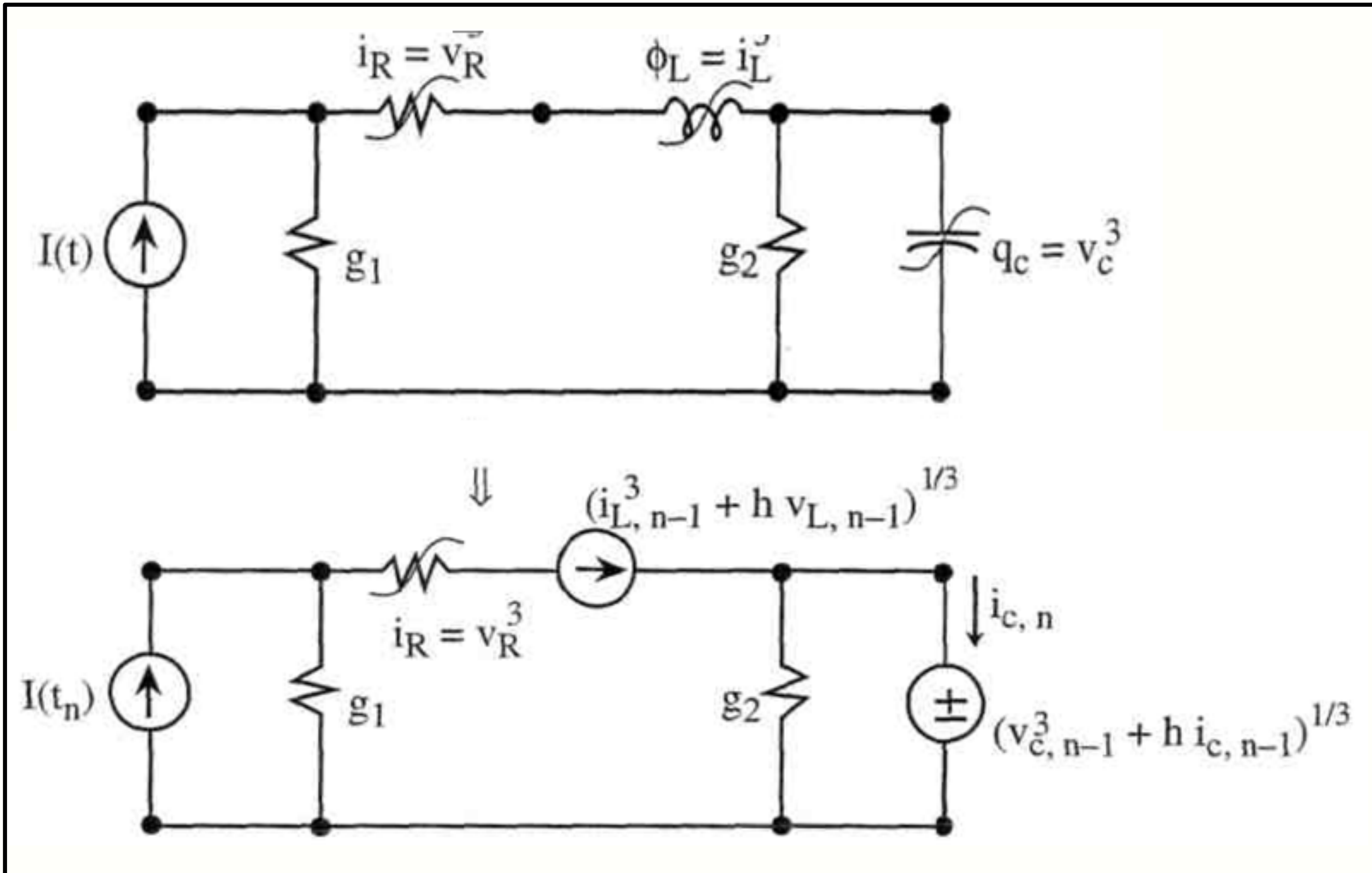
$$i_{L,\,n} = f^{-1}\,(\phi_{L,\,n-1} + h\,v_{L,\,n-1})$$

Nonlinear Inductor (cont.)



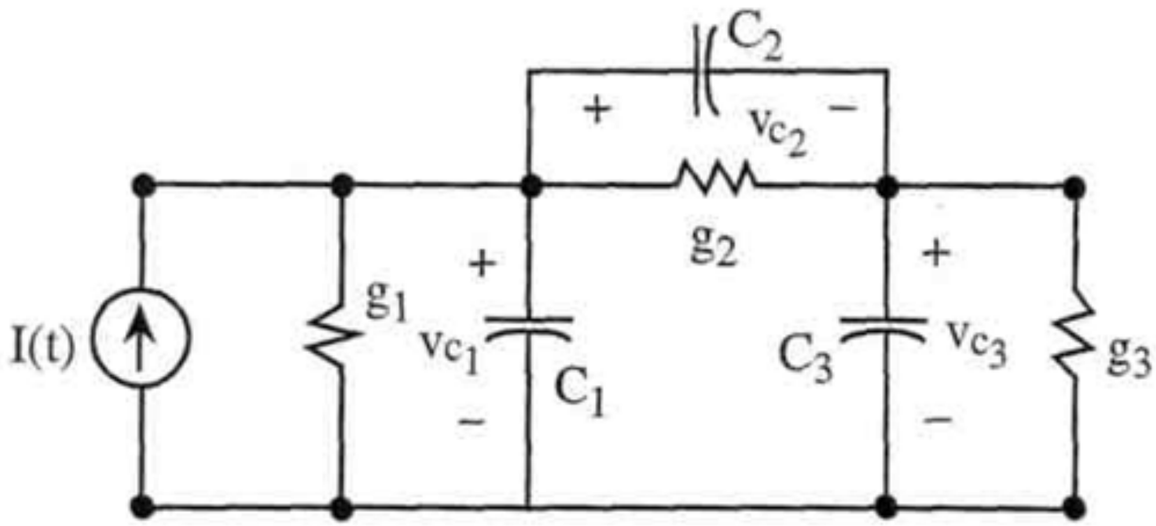


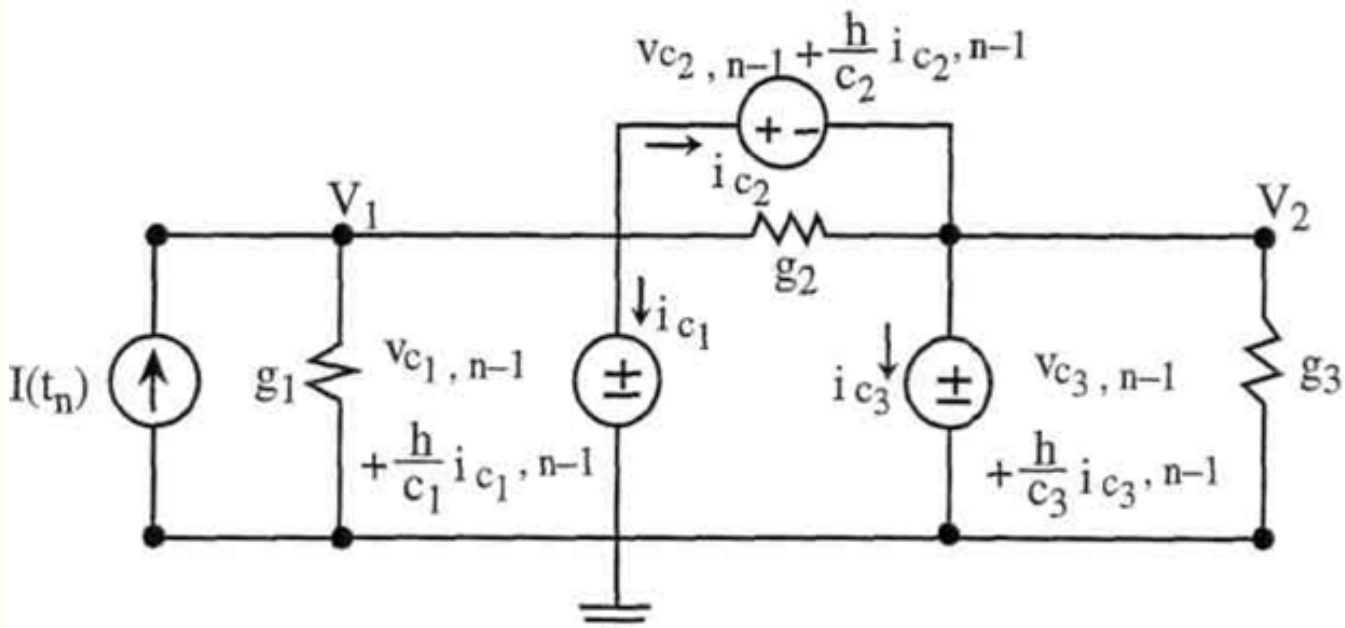
Example



$i_R = v_R^3$
$\phi_L = i_L^3$
$I(t)$
g_1
g_2
$q_c = v_c^3$
$\Downarrow$
$(i_{L,\,n-1}^3 + h\, v_{L,\,n-1})^{1/3}$
$i_R = v_R^3$
$i_{c,\,n}$
$I(t_n)$
g_1
g_2
$\pm$
$(v_{c,\,n-1}^3 + h\, i_{c,\,n-1})^{1/3}$

Loops of Capacitors (and voltage sources)

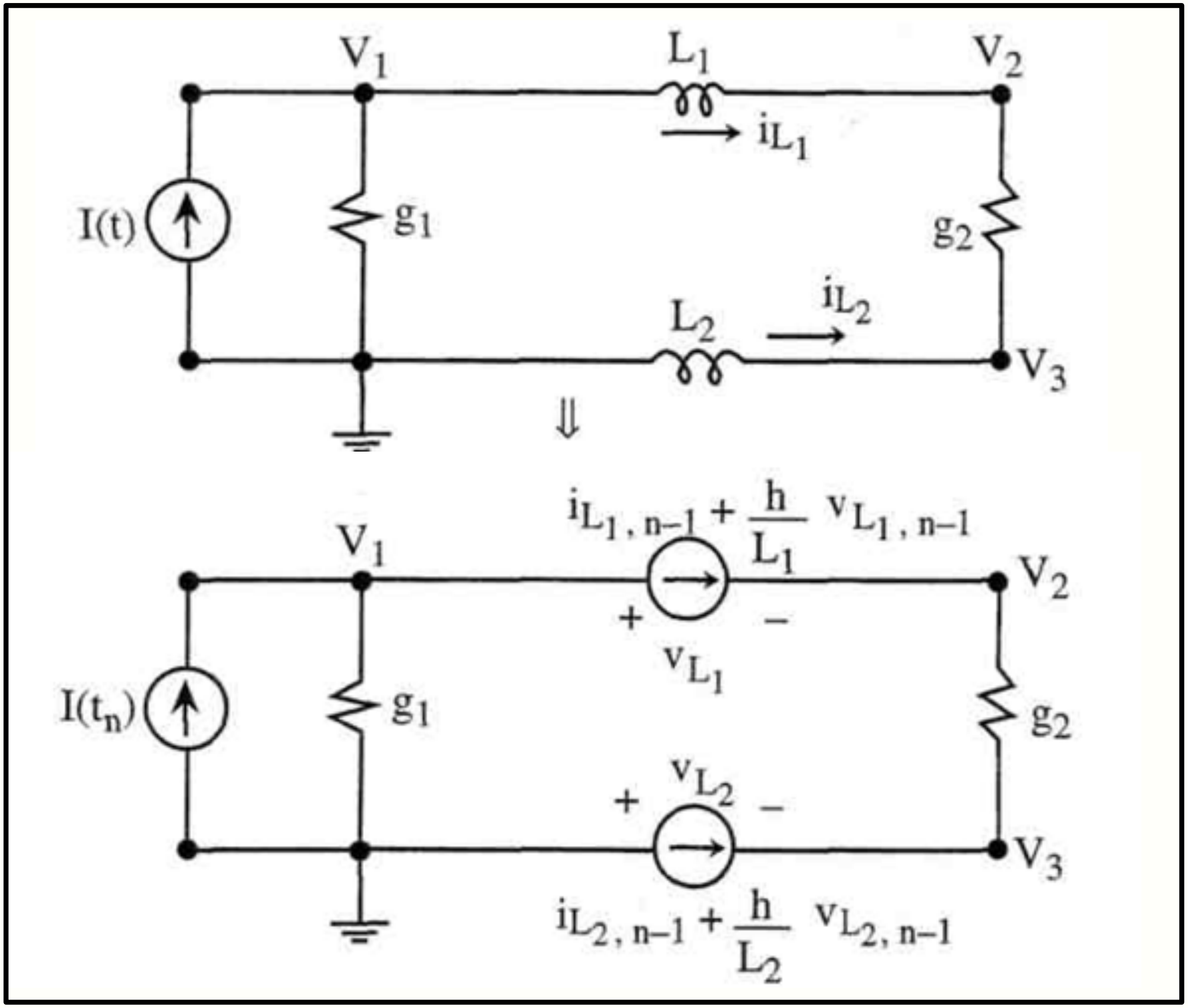




$$
\begin{bmatrix}
(g_1+g_2) & -g_2 & +1 & 0 & +1 \\
-g_2 & (g_1+g_2+g_3) & 0 & +1 & -1 \\
+1 & 0 & 0 & 0 & 0 \\
0 & +1 & 0 & 0 & 0 \\
+1 & -1 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
V_1 \\ V_2 \\ i_{C1} \\ i_{C3} \\ i_{C2}
\end{bmatrix}
=
\begin{bmatrix}
I(t_n) \\ 0 \\ v_{C1,n} \\ v_{C3,n} \\ v_{C2,n}
\end{bmatrix}
$$

Last three rows are dependent.

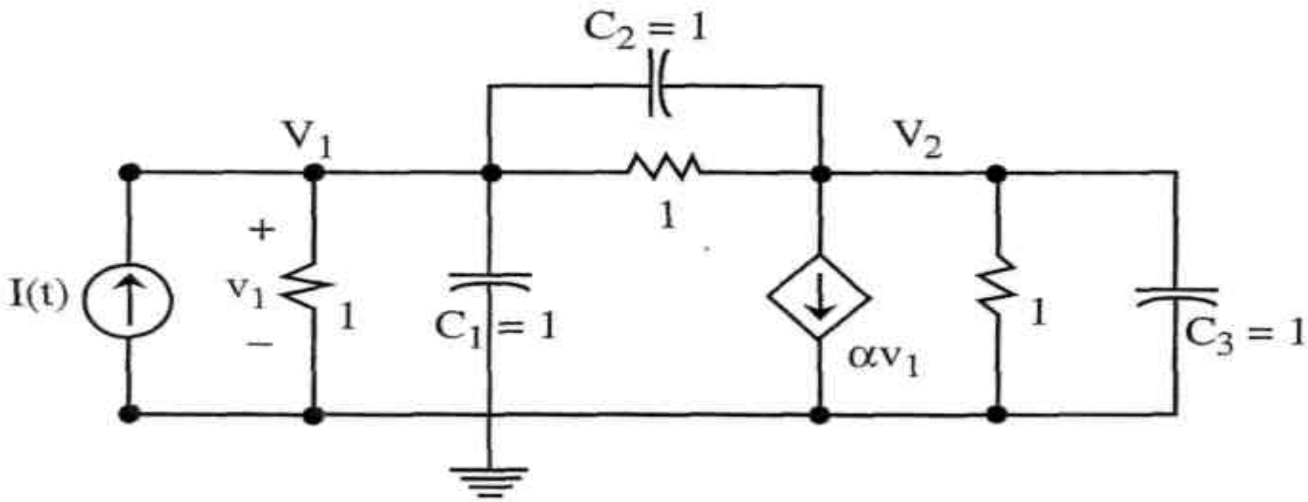
Cutset of Inductors (and current sources)



$$\begin{bmatrix} g_1 & 0 & 0 \\ 0 & g_2 & -g_2 \\ 0 & -g_2 & g_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} I(t_n) \\ i_{L1,n} \\ i_{L2,n} \end{bmatrix}$$

The last two rows are dependent.

Exercise 5:



$$C_1\dot{V}_1 + C_2(\dot{V}_1 - \dot{V}_2) + V_1 + (V_1 - V_2) = I(t)$$

$$C_3\dot{V}_3 + C_2(\dot{V}_2 - \dot{V}_1) + V_2 + \alpha V_1 = 0$$

$$2\dot{V}_1 - \dot{V}_2 = -2V_1 + V_2 + I(t)$$

$$2\dot{V}_2 - \dot{V}_1 = -V_2 + \alpha V_1$$

TIME DOMAIN SENSITIVITY

Linear Case

$$\mathbf{A}(\mathbf{p})\dot{\mathbf{x}}(t) + \mathbf{B}(\mathbf{p})\mathbf{x}(t) = \mathbf{y}(t) \qquad (9.237)$$

where some elements of $\mathbf{A}$ and $\mathbf{B}$ are functions of a set $\mathbf{p}$ of k parameters. For RLC linear circuits, $\mathbf{A}$ contains the capacitors and inductors, and $\mathbf{B}$ the resistors. If MNA formulation is used, $\mathbf{x}$ contains node voltages and current variables, such as currents through inductors; $\mathbf{y}(t)$ contains the independent sources. Note that the matrix $\mathbf{A}$ could be singular.

$$\mathbf{A}(\mathbf{p})\dot{\mathbf{x}}(t) + \mathbf{B}(\mathbf{p})\mathbf{x}(t) = \mathbf{y}(t)$$

Let $\mathbf{x_s}(t)$ be the solution trajectory of (9.237) when $\mathbf{p} = \mathbf{p^o}$, the nominal values of the parameters in $\mathbf{p}$, and initial conditions $\mathbf{x}(0)$. The aim is to compute the sensitivity of $\mathbf{x_s}(t)$ with respect to $\mathbf{p}$. Differentiating (9.237) along the solution trajectory, and assuming $\mathbf{y}(t)$ to be independent of $\mathbf{p}$, one gets

$$\frac{\partial \mathbf{A}}{\partial \mathbf{p}}\dot{\mathbf{x}} + \mathbf{A}\frac{\mathrm{d}}{\mathrm{dt}}\frac{\partial \mathbf{x}}{\partial \mathbf{p}} + \frac{\partial \mathbf{B}}{\partial \mathbf{p}}\mathbf{x} + \mathbf{B}\frac{\partial \mathbf{x}}{\partial \mathbf{p}} = \mathbf{0} \tag{9.238}$$

or

$$\mathbf{A}\frac{\mathrm{d}}{\mathrm{dt}}\left[\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}\right] + \mathbf{B}\left[\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}\right] = -\left[\frac{\partial \mathbf{A}}{\partial \mathbf{p}}\dot{\mathbf{x}} + \frac{\partial \mathbf{B}}{\partial \mathbf{p}}\mathbf{x}\right] \triangleq -\mathbf{G}(t) \tag{9.239}$$

$$\mathbf{A}\frac{\mathrm{d}}{\mathrm{dt}}\left[\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}\right] + \mathbf{B}\left[\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}\right] = -\left[\frac{\partial \mathbf{A}}{\partial \mathbf{p}}\dot{\mathbf{x}} + \frac{\partial \mathbf{B}}{\partial \mathbf{p}}\mathbf{x}\right] \triangleq -\mathbf{G}(t)$$

where $\left[\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}\right] \triangleq \left[\frac{\partial \mathbf{x}}{\partial \mathbf{p}}\right]$, the sensitivity matrix; $\left[\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}\right]$ and $\mathbf{G}(t)$ are n-by-k matrices whose entries are functions of time. The matrices $\mathbf{A}$ and $\mathbf{B}$ are evaluated at the nominal values of $\mathbf{p} = \mathbf{p}^{\mathrm{o}}$. Note that the sensitivity equation (9.239) is a set on k differential equations that can be solved together with system differential equation (9.237). Recall that column i of $\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}$, $\frac{\partial \mathbf{x}}{\partial p_i} \triangleq \mathbf{s}_i$, is the sensitivity of $\mathbf{x}$ with respect to parameter p_i; while row j of $\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}$, $\frac{\partial x_j}{\partial \mathbf{p}}$, is the sensitivity of x_j with respect to all parameters $\mathbf{p}$.

$$\mathbf{A}\frac{\mathrm{d}}{\mathrm{dt}}\left[\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}\right] + \mathbf{B}\left[\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}\right] = -\left[\frac{\partial \mathbf{A}}{\partial \mathbf{p}}\dot{\mathbf{x}} + \frac{\partial \mathbf{B}}{\partial \mathbf{p}}\mathbf{x}\right] \triangleq -\mathbf{G}(t)$$

i-th equation:

$$\mathbf{A}\frac{\mathrm{d}}{\mathrm{dt}}\mathbf{s}_{ci} + \mathbf{B}\mathbf{s}_{ci} = -\left[\frac{\partial \mathbf{A}}{\partial p_i}\dot{\mathbf{x}} + \frac{\partial \mathbf{B}}{\partial p_i}\mathbf{x}\right] = -\mathbf{g}_i(t)$$

Applying the B.E. formula

$$[\frac{1}{h}\mathbf{A} + \mathbf{B}]\mathbf{x}_n = \mathbf{y}(\mathrm{t_n}) + \frac{1}{h}\mathbf{A}\mathbf{x}_{n-1}$$

$$(\frac{1}{h}\mathbf{A} + \mathbf{B})\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(n) = -\mathbf{G}(t_{\mathrm{n}}) + \frac{1}{h}\mathbf{A}\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(n-1)$$

Algorithm 9.3.1 Computation of Time-Domain Sensitivity Matrix $\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(t)$ of a Linear System $\mathbf{A}(\mathbf{p})\dot{\mathbf{x}}(t) + \mathbf{B}(\mathbf{p})\mathbf{x}(t) = \mathbf{y}(t)$

{$\mathbf{A}$ and $\mathbf{B}$ are evaluated at the nominal values of $\mathbf{p} = \mathbf{p}^{\mathrm{o}}$}
1. At time t = 0, solve $\mathbf{B}\mathbf{x}(0) = \mathbf{y}(0)$ to find the initial condition $\mathbf{x}(0)$.
2. For $i = 1 : k$,
3. Construct $\mathbf{A}_{p_i} = \frac{\partial \mathbf{A}}{\partial p_i}$ and $\mathbf{B}_{p_i} = \frac{\partial \mathbf{B}}{\partial p_i}$ at $\mathbf{p} = \mathbf{p}^{\mathrm{o}}$.
4. Construct $\mathbf{g}_i(0) = \mathbf{B}_{p_i}\mathbf{x}(0)$ {$\mathbf{g}_i$ is the i-th column of $\mathbf{G}_i$}
5. Solve $\mathbf{B}\mathbf{s}_i = -\mathbf{g}_i(0)$ to find the sensitivity vector $\mathbf{s}_i(0)$ at the initial condition $\mathbf{x}(0)$.
6. End

At time t_n,

7. Solve $[\frac{1}{h}\mathbf{A} + \mathbf{B}]\mathbf{x}_n = \mathbf{y}(\mathrm{t_n}) + \frac{1}{h}\mathbf{A}\mathbf{x}_{n-1}$
8. For $i = 1 : k$
9. Construct $\mathbf{g}_{i,n} = \mathbf{A}_{p_i}\dot{\mathbf{x}}_n + \mathbf{B}_{p_i}\mathbf{x}_n$, where $\dot{\mathbf{x}}_n \simeq \frac{1}{h}(\mathbf{x}_n - \mathbf{x}_{n-1})$
10. Solve $[\frac{1}{h}\mathbf{A} + \mathbf{B}]\mathbf{s}_{i,n} = -\mathbf{g}_{i,n} + \frac{1}{h}\mathbf{A}\mathbf{s}_{i,n-1}$
11. End

$$x_{out} = \mathbf{e}^{\mathrm{T}}\mathbf{x}(t) \qquad \frac{\partial x_{out}(t)}{\partial \mathbf{p}} = \mathbf{e}^{\mathrm{T}}\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(t)$$

$$\mathrm{A}\frac{\mathrm{d}}{\mathrm{dt}}\left[\mathrm{S}_{\mathrm{p}}^{\mathrm{x}}\right] + \mathrm{B}\left[\mathrm{S}_{\mathrm{p}}^{\mathrm{x}}\right] = -\left[\frac{\partial \mathrm{A}}{\partial \mathbf{p}}\dot{\mathrm{x}} + \frac{\partial \mathrm{B}}{\partial \mathbf{p}}\mathrm{x}\right] \triangleq -\mathbf{G}(t)$$

Applying Laplace transform

$$\mathbf{A}\left[s\mathbf{S}_L - \mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(0^-)\right] + \mathbf{B}[\mathbf{S}_L] = -\mathbf{G}(s)$$

$$[s\mathbf{A} + \mathbf{B}]\mathbf{S}_L = -\mathbf{G}(s) + \mathbf{A}\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(0^-)$$

$$\mathbf{S}_L = [s\mathbf{A} + \mathbf{B}]^{-1}[-\mathbf{G}(s) + \mathbf{A}\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(0^-)]$$

$\mathbf{S}_L$ is the Laplace transform of $\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}$

$$\frac{\partial x_{out}(s)}{\partial \mathbf{p}} = \mathbf{e}^{\mathrm{T}}\mathbf{S}_L = \mathbf{e}^{\mathrm{T}}[s\mathbf{A}+\mathbf{B}]^{-1}[-\mathbf{G}(s)+\mathbf{A}\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(0^-)]$$

$$\mathbf{u}^{\mathrm{T}}(s) = \mathbf{e}^{\mathrm{T}}[s\mathbf{A}+\mathbf{B}]^{-1}$$

$$[s\mathbf{A}+\mathbf{B}]^{\mathrm{T}}\mathbf{u}(s) = \mathbf{e}$$

$$\frac{\partial x_{out}(s)}{\partial \mathbf{p}} = \mathbf{u}(s)^{\mathrm{T}}[-\mathbf{G}(s)+\mathbf{A}\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(0^-)]$$

In the time domain,

$$\frac{\partial x_{out}(\mathrm{t})}{\partial \mathbf{p}} = -\mathbf{u}^{\mathrm{T}}(t) * \mathbf{G}(\mathrm{t}) + \mathbf{u}^{\mathrm{T}}(t)\mathbf{A}\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(0^-)$$

$\mathbf{u}^{\mathrm{T}}(t) * \mathbf{G}(\mathrm{t})$ $\qquad$ $n \times k$ convolution operations

Given two functions $f_1(t)$ and $f_2(t)$, where $f_1(t) = 0$ and $f_2(t) = 0$ for $t < 0$. Suppose $f_1(t)$ and $f_2(t)$ are approximated by staircase functions with equal time intervals h, with values a_{i+1} and b_{i+1} in the interval $t_i < t < t_{i+1}$, respectively, where $t_{i+1} = t_{i+h}$. The convolution of $f_1(t)$ with $f_2(t)$, $y(t) = f_1(t) * f_2(t)$ at time t_n, $y(t_n)$, where t_n is a boundary point, is given by

$$y(t_n) = h \sum_{i=1}^{n} a_i b_{n-i+1}$$

Algorithm 9.3.2 Computation of Time-domain Sensitivity Vector $\frac{\partial x_{out}(\mathrm{t})}{\partial \mathbf{p}}$ of a Linear System $\mathbf{A}(\mathbf{p})\dot{\mathbf{x}}(t) + \mathbf{B}(\mathbf{p})\mathbf{x}(t) = \mathbf{y}(t)$

$\{\mathbf{A}$ and $\mathbf{B}$ are evaluated at the nominal values of $\mathbf{p} = \mathbf{p}^{\mathrm{o}}\}$.

1. At time t = 0, solve $\mathbf{B}\mathbf{x}(0) = \mathbf{y}(0)$ to find the initial condition $\mathbf{x}(0)$.
2. Solve $[\frac{1}{h_{min}}\mathbf{A} + \mathbf{B}]^{\mathrm{T}}\mathbf{u}_1 = \frac{1}{h_{min}}\mathbf{e}$ {to find initial conditions at $t = 0^+$ of the transpose or adjoint circuit, assuming $\mathbf{u}(0^-) = \mathbf{0}$; $\frac{1}{h_{min}}\mathbf{e}$ is a pulse of height $\frac{1}{h_{min}}\mathbf{e}$ and width h_{min}, and is an approximation of a delta function $\delta(t)\mathbf{e}$}

Algorithm 9.3.2 Computation of Time-domain Sensitivity Vector $\frac{\partial x_{out}(\mathrm{t})}{\partial \mathbf{p}}$ of a Linear System $\mathbf{A}(\mathbf{p})\dot{\mathbf{x}}(t) + \mathbf{B}(\mathbf{p})\mathbf{x}(t) = \mathbf{y}(t)$

{$\mathbf{A}$ and $\mathbf{B}$ are evaluated at the nominal values of $\mathbf{p} = \mathbf{p}^{\mathrm{o}}$}.

1. At time t = 0, solve $\mathbf{Bx}(0) = \mathbf{y}(0)$ to find the initial condition $\mathbf{x}(0)$.
2. Solve $[\frac{1}{h_{min}}\mathbf{A} + \mathbf{B}]^{\mathrm{T}}\mathbf{u}(h_{min}) = \frac{1}{h_{min}}\mathbf{e}$ {to find initial conditions at $t = h_{min}$ of the transpose or adjoint circuit, assuming $\mathbf{u}(0^{-}) = \mathbf{0}$; $\frac{1}{h_{min}}\mathbf{e}$ is a pulse of height $\frac{1}{h_{min}}\mathbf{e}$ and width h_{min}, and is an approximation of a delta function $\delta(t)\mathbf{e}$}

3. For $i = 1 : k$

4. Construct $\mathbf{A}_{p_i} = \frac{\partial \mathbf{A}}{\partial p_i}$ and $\mathbf{B}_{p_i} = \frac{\partial \mathbf{B}}{\partial p_i}$ at $\mathbf{p} = \mathbf{p}^{\mathrm{o}}$.

5. Construct $\mathbf{g}_i(0) = \mathbf{B}_{p_i}\mathbf{x}(0)$

6. Solve $\mathbf{B}\mathbf{s}_i = -\mathbf{g}_i(0)$ to find the sensitivity vector $\mathbf{s}_i(0)$ at the initial condition $\mathbf{x}(0)$ $\{\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(0^-) = [\mathbf{s}_1\,\mathbf{s}_2\,\ldots\,\mathbf{s}_k]$ is the sensitivity matrix at $\mathbf{x}(0)\}$.

7. Find $\mathbf{c}_{\mathrm{i}} = \mathbf{A}\mathbf{s}_i$

8. End

At time t_n,

9. Solve $[\frac{1}{h}\mathbf{A}+\mathbf{B}]\mathbf{x}_n = \mathbf{y}(\mathrm{t_n}) + \frac{1}{h}\mathbf{A}\mathbf{x}_{n-1}$
10. Solve $[\frac{1}{h}\mathbf{A}+\mathbf{B}]^{\mathrm{T}}\mathbf{u}_n = \frac{1}{h}\mathbf{A}^{\mathrm{T}}\mathbf{u}_{n-1}$
11. For $i = 1 : k$
12. Construct $\mathbf{g}_{i,n} = \mathbf{A}_{p_i}\dot{\mathbf{x}}_n + \mathbf{B}_{p_i}\mathbf{x}_n$, where $\dot{\mathbf{x}}_n \simeq \frac{1}{h}(\mathbf{x}_n - \mathbf{x}_{n-1})$
13. Find $\frac{\partial x_{out,n}}{\partial p_i} = -\mathbf{u}_n^{\mathrm{T}} * \mathbf{g}_{i,n} + \mathbf{u}_n^{\mathrm{T}}\mathbf{c}_i$
14. End

Nonlinear Case

$$\mathbf{f}(\mathbf{x},\ \dot{\mathbf{x}},\ \mathbf{p},\ t) = \mathbf{0}, \quad \mathbf{x}(t_o) = \mathbf{x}(0)$$

$$\frac{\partial \mathrm{f}}{\partial \dot{\mathrm{x}}} \frac{\mathrm{d}}{\mathrm{dt}} \frac{\partial \mathrm{x}}{\partial \mathbf{p}} + \frac{\partial \mathrm{f}}{\partial \mathrm{x}} \frac{\partial \mathrm{x}}{\partial \mathbf{p}} + \frac{\partial \mathrm{f}}{\partial \mathbf{p}} = \mathbf{0}$$

$$\mathbf{A}(t_n)\frac{\mathrm{d}}{\mathrm{dt}}\left[\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}\right] + \mathbf{B}(t_n)\left[\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}\right] = -\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$$

Algorithm 9.3.3 Computation of Time-Domain Sensitivity Matrix $\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(t)$ of a Nonlinear System $\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{p}, t) = \mathbf{0}$

1. At time t = 0, solve $\mathbf{f}(\mathbf{x}, \mathbf{0}, \mathbf{p}^{o}, 0) = \mathbf{0}$ to find the initial condition $\mathbf{x}(0)$ {all differential operators are put to zero}.
2. For $i = 1 : k$
3. Let $\mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ evaluated at $\mathbf{x}(0)$, which is available at the solution of $\mathbf{f}(\mathbf{x}, \mathbf{0}, \mathbf{p}^{o}, 0) = \mathbf{0}$ obtained in step (1), and compute $\mathbf{B}_{p_i} = \frac{\partial \mathbf{B}}{\partial p_i}$.
4. Construct $\mathbf{g}_i(0) = \mathbf{B}_{p_i}\mathbf{x}(0)$
5. Solve $\mathbf{B}\mathbf{s}_i = -\mathbf{g}_i(0)$ to find the sensitivity vector $\mathbf{s}_i(0)$ at the initial condition $\mathbf{x}(0)$.
6. End

At time t_n,

7. Solve $\mathbf{f}(\mathbf{x}_\mathrm{n},\ \frac{1}{h}\mathbf{x}_\mathrm{n},\ \frac{1}{h}\mathbf{x}_{\mathrm{n}-1}, \mathbf{p}^{\mathrm{o}},\ t_n) = \mathbf{0}$ for $\mathbf{x}_\mathrm{n}$
8. Let $\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{x}}}$ and $\mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ evaluated at $\mathbf{x}_\mathrm{n}$ $\{\mathbf{A}$ and $\mathbf{B}$ are readily available at the solution $\mathbf{x}_\mathrm{n}$ and change at each time point$\}$
9. For $i = 1 : k$
10. Let $\mathbf{A}_{p_i} = \frac{\partial \mathbf{A}}{\partial p_i}$ and $\mathbf{B}_{p_i} = \frac{\partial \mathbf{B}}{\partial p_i}$
11. Construct $\mathbf{g}_{i,n} = \mathbf{A}_{p_i}\dot{\mathbf{x}}_n + \mathbf{B}_{p_i}\mathbf{x}_n$, where $\dot{\mathbf{x}}_n \simeq \frac{1}{h}(\mathbf{x}_n - \mathbf{x}_{n-1})$
12. Solve $[\frac{1}{h}\mathbf{A} + \mathbf{B}]\mathbf{s}_{i,n} = -\mathbf{g}_{i,n} + \frac{1}{h}\mathbf{A}\mathbf{s}_{i,n-1}$
13. End

Algorithm 9.3.4 Computation of Time-Domain Sensitivity Vector $\frac{\partial x_{out}(\mathrm{t})}{\partial \mathbf{p}}$ of a Nonlinear System $\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{p}, t) = \mathbf{0}$

1. At time $\mathrm{t} = 0$, solve $\mathbf{f}(\mathbf{x}, \mathbf{0}, \mathbf{p}^{o}, 0) = \mathbf{0}$ to find the initial condition $\mathbf{x}(0)$
2. Let $\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{x}}}$ and $\mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ evaluated at $\mathbf{x}(0)$ {$\mathbf{A}$ and $\mathbf{B}$ are readily available at the solution $\mathbf{x}(0)$ and change at each time point}
3. For $i = 1 :, k$
4. Construct $\mathbf{A}_{p_i} = \frac{\partial \mathbf{A}}{\partial p_i}$ and $\mathbf{B}_{p_i} = \frac{\partial \mathbf{B}}{\partial p_i}$.
5. Construct $\mathbf{g}_i(0) = \mathbf{B}_{p_i}\mathbf{x}(0)$
6. Solve $\mathbf{B}\mathbf{s}_i = -\mathbf{g}_i(0)$ to find the sensitivity vector $\mathbf{s}_i(0)$ at the initial condition $\mathbf{x}(0)$ {$\mathbf{S}_{\mathbf{p}}^{\mathbf{x}}(0^{-}) = [\mathbf{s}_1\, \mathbf{s}_2 \ldots \mathbf{s}_k]$ is the sensitivity matrix at $\mathbf{x}(0)$}.
7. Find $\mathbf{c}_{\mathrm{i}} = \mathbf{A}\mathbf{s}_i$
8. End

9. Solve $[\frac{1}{h_{min}}\mathbf{A} + \mathbf{B}]^{\mathrm{T}}\mathbf{u}(h_{min}) = \frac{1}{h_{min}}\mathbf{e}$

At time t_n,

10. Solve $\mathbf{f}(\mathbf{x}_{\mathrm{n}},\ \frac{1}{h}\mathbf{x}_{\mathrm{n}},\ \frac{1}{h}\mathbf{x}_{\mathrm{n-1}}, \mathbf{p}^{\mathrm{o}},\ t_n) = \mathbf{0}$ for $\mathbf{x}_{\mathrm{n}}$
11. Evaluate $\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{x}}}$ and $\mathbf{B} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ evaluated at $\mathbf{x}_{\mathrm{n}}$
12. Solve $[\frac{1}{h}\mathbf{A} + \mathbf{B}]^{\mathrm{T}}\mathbf{u}_n = \frac{1}{h}\mathbf{A}^{\mathrm{T}}\mathbf{u}_{n-1}$
13. For $i = 1 : k$
14. Find $\mathbf{A}_{p_i} = \frac{\partial \mathbf{A}}{\partial p_i}$ and $\mathbf{B}_{p_i} = \frac{\partial \mathbf{B}}{\partial p_i}$
15. Construct $\mathbf{g}_{i,n} = \mathbf{A}_{p_i}\dot{\mathbf{x}}_n + \mathbf{B}_{p_i}\mathbf{x}_n$, where $\dot{\mathbf{x}}_n \simeq \frac{1}{h}(\mathbf{x}_n - \mathbf{x}_{n-1})$
16. Find $\frac{\partial x_{out,n}}{\partial p_i} = -\mathbf{u}_n^{\mathrm{T}} * \mathbf{g}_{i,n} + \mathbf{u}_n^{\mathrm{T}}\mathbf{c}_i$
17. End

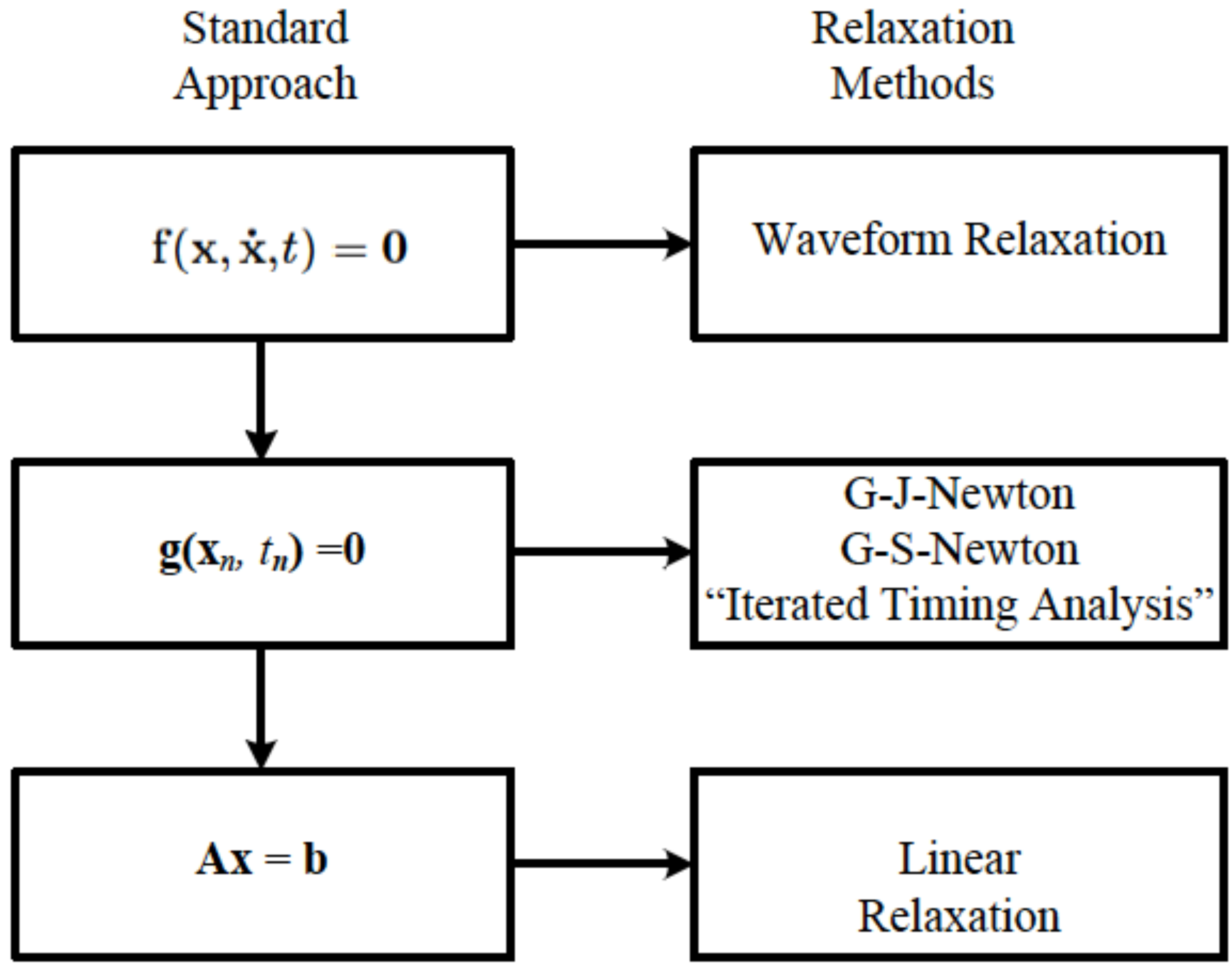
Partitioning and Time-Domain Relaxation

$$\mathbf{f}_1(\mathbf{x}_1, \dot{\mathbf{x}}_1, \mathbf{x}_2, \dot{\mathbf{x}}_2, \cdots, \mathbf{x}_\mathrm{p}, \dot{\mathbf{x}}_\mathrm{p}, t) = \mathbf{0}$$

$$\mathbf{f}_2(\mathbf{x}_1, \dot{\mathbf{x}}_1, \mathbf{x}_2, \dot{\mathbf{x}}_2, \cdots, \mathbf{x}_\mathrm{p}, \dot{\mathbf{x}}_\mathrm{p}, t) = \mathbf{0}$$

$$\vdots$$

$$\mathbf{f}_\mathrm{p}(\mathbf{x}_1, \dot{\mathbf{x}}_1, \mathbf{x}_2, \dot{\mathbf{x}}_2, \cdots, \mathbf{x}_\mathrm{p}, \dot{\mathbf{x}}_\mathrm{p}, t) = \mathbf{0}$$


Standard Approach
Relaxation Methods
$\mathbf{f}(\mathbf{x}, \dot{\mathbf{x}}, t) = \mathbf{0}$
Waveform Relaxation
$\mathbf{g}(\mathbf{x}_n, t_n) = \mathbf{0}$
G-J-Newton
G-S-Newton
"Iterated Timing Analysis"
$\mathbf{Ax} = \mathbf{b}$
Linear Relaxation

Gauss-Jacobi waveform relaxation method

Starting with an initial guess $\mathbf{x}_i(t)$, $\dot{\mathbf{x}}_i(t)$, $i = 1$, ..., p, in the time interval of interest $t\epsilon[t_o,\, t_f]$, solve

$$\mathbf{f}_i(\mathbf{x}_1^{(k))}, \dot{\mathbf{x}}_1^{(k)}, \ldots \mathbf{x}_i, \dot{\mathbf{x}}_i, \ldots, \mathbf{x}_{\mathrm{p}}^{(k)}, \dot{\mathbf{x}}_{\mathrm{p}}^{(k)}, t) = \mathbf{0}$$

to find $\mathbf{x}_i^{(k+1)}(t)$, $t\epsilon[t_o, t_f]$, $i = 1, 2, \ldots, p$, where k is an iteration number. Repeat until all waveforms converge.

Gauss-Seidel waveform relaxation method

Starting with an initial guess $\mathbf{x}_i(t)$, $\dot{\mathbf{x}}_i(t)$, $i = 1$, ..., p, in the time interval of interest $t\epsilon[t_o,\, t_f]$, solve

$$\mathbf{f}_i(\mathbf{x}_1^{(k+1)},\, \dot{\mathbf{x}}_1^{(k+1)},\, \ldots \mathbf{x}_i,\, \dot{\mathbf{x}}_i,\, \ldots,\, \mathbf{x}_\mathrm{p}^{(k)},\, \dot{\mathbf{x}}_\mathrm{p}^{(k)},\, t) = \mathbf{0}$$

to find $\mathbf{x}_i^{(k+1)}(t)$, $t\epsilon[t_o, t_f]$, $i = 1, 2, \ldots, p$, where k is an iteration number. Repeat until all waveforms converge.

Convergence of waveforms:

(a) Two waveforms converge in the time interval $[t_o, t_f]$ if, for all $t \epsilon [t_o, t_f]$, $\| \mathbf{x}^{(k+1)}(t) - \mathbf{x}^{(k)}(t) \| < \varepsilon$. It is possible to have a variable bound at different time points, $B_u = \varepsilon + \varepsilon_r \mid \dot{x}_n \mid$, similar to the bound (9.2.125) used in determining the stepsize in transient analysis.

(b) Convergence check is performed at discrete time points in the interval $t \epsilon [t_o, t_f]$. If a variable timestep strategy is followed, interpolation may be necessary to check convergence at a given time point.

(c) To speed-up convergence and reduce computation, it is sometimes advisable to subdivide the interval $[t_o, t_f]$ into subintervals and apply waveform relaxation in a subinterval before the equations in the next subinterval are solved.

(c) To speed-up convergence and reduce computation, it is sometimes advisable to subdivide the interval $[t_o, t_f]$ into subintervals and apply waveform relaxation in a subinterval before the equations in the next subinterval are solved.

Remarks:

(1) In the Gauss-Jacobi method the solutions of the partitions can be done in parallel

to partitioning based on *strongly-connected components,*

(2) In the Gauss-Seidel method the solutions of the partitions have to be done in a predetermined sequence.

This leads to partitioning based on *strongly-connected components, levelizing, scheduling, active and dormant subcircuits, and event-driven simulation*