Transient Analysis of Nonlinear Dynamic Circuits

Equations:  
KCL  
KVL  
BE

New:  
BE6:  
\[ q_6 - q_0 \left( \exp\left(\frac{v_6}{V_T}\right) - 1 \right) = 0 \]

\[ \frac{dq_6}{dt} - i_6 = 0 \]

or

\[ i_6 - \frac{q_0}{v_T} \exp\left(\frac{v_6}{V_T}\right) \frac{dv_6}{dt} = 0 \]

think of \( dq_6/dt \) or \( dv_6/dt \) as new variables.

We need one more “branch relation”.  

Linear Multistep Methods

\[ \dot{y} = \frac{dy}{dt} = f(y) \]

\[ \sum_{i=0}^{k} \alpha_i y_{n-i} + h_n \sum_{i=0}^{k} \beta_i \dot{y}_{n-i} = 0 \]

\[ y_{n-i} \equiv y(t_{n-i}) \quad \dot{y}_{n-i} = \frac{dy}{dt}(t_{n-i}) \quad t_n = t_{n-1} + h_n \]

Examples:

1. Forward Euler
   \[ \alpha_0 = 1, \alpha_1 = -1, \beta_1 = -1, \beta_0 = 0 \]
   \[ y_n - y_{n-1} - h_n \dot{y}_{n-1} = 0 \quad \Rightarrow \dot{y}_{n-1} = \frac{y_n - y_{n-1}}{h_n} \]

2. Backward Euler
   \[ \alpha_0 = 1, \alpha_1 = -1, \beta_0 = -1, \beta_1 = 0 \]
   \[ y_n - y_{n-1} - h_n \dot{y}_n = 0 \quad \Rightarrow \dot{y}_n = \frac{y_n - y_{n-1}}{h_n} \]

3. Trapezoidal
   \[ \alpha_0 = 1, \alpha_1 = -1, \beta_0 = -1/2, \beta_1 = -1/2 \]
   \[ y_n - y_{n-1} - \frac{h_n}{2}(\dot{y}_n + \dot{y}_{n-1}) = 0 \quad \Rightarrow \frac{\dot{y}_n + \dot{y}_{n-1}}{2} = \frac{y_n - y_{n-1}}{h_n} \]

Two Classes: Explicit (\( \beta_0 = 0 \))
   Implicit (\( \beta_0 \neq 0 \))
Solving Ordinary Differential Equations

At each point in “state-space” we have a vector \((f(x,y), g(x,y))\) - vector field -

Solution is any curve such that tangent of curve at \((x,y) = vector at point (x,y)\)

**Numerical Solution:** Discrete time steps
Sources of Error

\[ \dot{x} = f(x) \]

Local error is due to finite $\Delta t$ at time $t$

\[ x_{n+1} = x_n + \Delta t f \]

Global error due to flow and round-off and local error

Flow can help or hurt

diverging flow

converging flow
Local Truncation Error

Local Truncation Error (LTE) ≡ Error we make taking one time step, assuming past data is perfect.

\[ \text{LTE} \triangleq x(t_n) - x_n \]

where \( x(t_n) \) is exact solution at \( t_n \) and \( x_n \) is computed result using exact past data. i.e.

\[
x_n + \sum_{i=1}^{k} \alpha_i x(t_{n-i}) + h \left( \beta_0 f(x_n) + \sum_{i=1}^{k} \beta_i \dot{x}(t_{n-i}) \right) = 0
\]

Local Error \((E_k)\)

\[
E_k \triangleq \sum_{i=0}^{k} \alpha_i x(t_{n-i}) + h_n \beta_i \dot{x}(t_{n-i})
\]

= amount by which the linear multi-step formula is wrong

\[
x(t_n) - x_n = \text{LTE} = - \sum_{i=1}^{k} \alpha_i x(t_{n-i}) - h \beta_0 \dot{x}(t_n) - h_n \sum_{i=1}^{k} \beta_i \dot{x}(t_{n-i}) + E_k
\]

\[
= -h \beta_0 (\dot{x}(t_n) - f(t_n)) + E_k
\]

\[
= -h \beta_0 (f(x(t_n)) - f(t_n)) + E_k \quad l \text{ is Lipschitz constant for } f
\]

Thus,

\[
|\text{LTE}| \leq |E_k| + |h \beta_0|/l |x(t_n) - x_n| \leq \frac{|E_k|}{1 - |h \beta_0|/l}
\]

Typically we use \( E_k \) instead of LTE to estimate error (same if \( \beta_0 = 0 \))
\[ E_k = E[x(t), h] \triangleq \sum_{i=0}^{k} \alpha_i x(t_{n-i}) + h_n \beta_i \dot{x}(t_{n-i}) \]

Expand as a Taylor series in h

\[ E[x(t), h] \triangleq E[x,0] + E^{(1)}[x,0]h + E^{(2)}[x,0] \frac{h^2}{2} \]
\[ + \ldots + E^{(k+1)}[x,0] \frac{h^{k+1}}{(k+1)!} + O(h^{k+2}) \]

**Definition:** A multi-step formula is said to be a \( p^{th} \) order method if

\[ E^{(i)}[x,0] = 0 \]

for \( 0 \leq i \leq p \) and for all \( x(t) \) with at least \( p+1 \) derivatives.

This is the same as saying that \( E[q(t), h] = 0 \) for any polynomial \( q(t) \) of degree \( p \) or less.

\[ x(t) = q(t) + r(t) \]
\[ E[x, h] = E[q, h] + E[r, h] \]
\[
q(t) = \left(\frac{t_n - t}{h}\right)^l \quad l = 0, ..., p \quad \text{Basis}
\]

\[
E[q(t),h] = \sum_{i=0}^{k} \alpha_i q(t_{n-i}) + h\beta_i \dot{q}(t_{n-i})
\]

\[
= \sum_{i=0}^{k} \alpha_i \left(\frac{t_n - t_{n-i}}{h}\right)^l + h\beta_i \left(\frac{-l}{h}\right)\left(\frac{t_n - t_{n-i}}{h}\right)^{l-1}
\]

or

\[
0 = \sum_{i=0}^{k} \alpha_i \left(\frac{t_n - t_{n-i}}{h}\right)^l - l\beta_i \left(\frac{t_n - t_{n-i}}{h}\right)^{l-1}
\]

for \(l = 0, ..., p\)

**Note:** if \(t_i = ih\) (uniform time step) then

\[
\sum_{i=0}^{k} \alpha_i (i)^l - l\beta_i (i)^{l-1} = 0
\]

\[
\sum_{i=0}^{k} \alpha_i = 0 \quad l = 0
\]

\[
\sum_{i=0}^{k} \alpha_i i - \beta_i = 0 \quad l = 1
\]

\[
\sum_{i=0}^{k} [(\alpha_i i - p\beta_i) i^{p-1}] = 0 \quad l = p > 1
\]

**Exactness equations (uniform time step)**
**Examples:**

1. **Forward Euler:** \( x_n = x_{n-1} + h\dot{x}_{n-1} \)
   \[
   \begin{align*}
   \alpha_0 &= 1, \quad \alpha_1 = -1, \quad \beta_1 = -1, \quad \beta_0 = 0 \\
   \sum \alpha_i &= 0 \quad \checkmark \quad l = 0 \\
   \sum \alpha_i - \beta_i &= 0 \Rightarrow \alpha_1 - \beta_0 - \beta_1 = 0 \quad \checkmark \quad l = 1 \\
   \sum \alpha_i^2 - 2\beta_i &= 0 \Rightarrow \alpha_1 - 2\beta_1 = 0 \quad \times \quad l = 2 
   \end{align*}
   \]

2. **Backward Euler**
   \( x_n = x_{n-1} + h\dot{x}_n \)
   \[
   \begin{align*}
   \alpha_0 &= 1, \quad \alpha_1 = -1, \quad \beta_0 = -1, \quad \beta_1 = 0 \\
   \sum \alpha_i &= 0 \quad \checkmark \quad l = 0 \\
   \alpha_1 - \beta_0 - \beta_1 &= 0 \quad \checkmark \quad l = 1 \\
   \alpha_1 - 2\beta_1 &= 0 \quad \times \quad l = 2 
   \end{align*}
   \]

3. **Trapezoidal**
   \( x_n = x_{n-1} + \frac{h}{2}(\dot{x}_n + \dot{x}_{n-1}) \)
   \[
   \begin{align*}
   \alpha_0 &= 1, \quad \alpha_1 = -1, \quad \beta_0 = -1/2, \quad \beta_1 = -1/2 \\
   \sum \alpha_i &= 0 \quad \checkmark \quad l = 0 \\
   \alpha_1 - \beta_0 - \beta_1 &= 0 \quad \checkmark \quad l = 1 \\
   \alpha_1 - 2\beta_1 &= 0 \quad \checkmark \quad l = 2 \\
   \sum \alpha_i^3 - 3\beta_i &= 0 \Rightarrow \alpha_1 - 3\beta_1 = 0 \quad \times \quad l = 3 
   \end{align*}
   \]

Thus: Backward and Forward Euler are 1st order methods and Trapezoidal Rule is a 2nd order method.
**Note:** In

\[ 0 = \sum_{i=0}^{k} \alpha_i \left( \frac{t_n - t_{n-i}}{h} \right) - l\beta_i \left( \frac{t_n - t_{n-i}}{h} \right)^{l-1} \quad 0 \leq l \leq p \]

there are \(2(k+1)-1\) unknowns \((\alpha_0 = 1)\) and \(p+1\) equations. Thus we need

\[ 2(k + 1) - 1 \geq p + 1 \quad \text{or} \quad k \geq \frac{p}{2} \]

**Algorithm for choosing Linear Multistep Method:**

1. Choose order of accuracy \(p\).
2. Choose step length \((k)\) of the method where \(k \geq \frac{p}{2}\).
3. Write down the \(p+1\) exactness equations.
4. If \(k > \frac{p}{2}\), choose \(2k+1-(p+1) = 2k-p\) other constraints.

**Example:**

with \(p = k\), the additional \(k\) constraints might be

\[ \beta_1 = \beta_2 = \ldots = \beta_k = 0 \]

\[ \sum_{i=0}^{k} \alpha_i x_{n-i} + h\beta_0 \dot{x}_n = 0 \]

\[ \dot{x}_n = \frac{1}{\beta_0 h} \sum_{i=0}^{k} \alpha_i x_{n-i} \]

\(^{\text{kth}}\text{order}\)

**backward differentiation method**
Determination of Local Error

\[ E_k = E[x(t), h] = \sum_{i=0}^{k} \alpha_i x(t_{n-i}) + h_n \beta_i \dot{x}(t_{n-i}) \]

**smooth solution**

\[ x(t) = x(t_n) + x^{(1)}(t_n)(t - t_n) + \ldots + \frac{x^{(p+1)}(t_n)}{p+1!}(t - t_n)^{p+1} + \ldots = q_p(t) + r(t) \]

where

\[ r(t) = \frac{x^{(p+1)}(t_n)}{p+1!}(t - t_n)^{p+1} + \ldots \]

\[ E_k = E[q_p(t), h] + E[r, h] \]

\[ = 0 + E[r(t), h] \]

\[ = \sum \frac{\alpha_i x^{(p+1)}(t_n)}{p+1!}(t_{n-i} - t_n)^{p+1} + \frac{h \beta_i x^{(p+1)}(t_n)}{p!}(t_{n-i} - t_n)^p + O(h^{p+2}) \]

\[ = [\sum \alpha_i (t_n - t_{n-i})^{p+1} - (p+1) h \beta_i (t_n - t_{n-i})^p] \frac{x^{(p+1)}(t_n)}{p+1!} + O(h^{p+2}) \]

\[ = \left[ \sum_{i=1}^{k} \alpha_i \left( \frac{t_n - t_{n-i}}{h} \right)^{p+1} - (p+1) \beta_i \left( \frac{t_n - t_{n-i}}{h} \right)^p \right] \frac{x^{(p+1)}(t_n)}{p+1!} h^{p+1} + O(h^{p+2}) \]

\[ \equiv \varepsilon_{p+1} x^{(p+1)}(t_n) h^{p+1} + \ldots \]

**Examples:**

FE: \( \alpha_0 = 1, \alpha_1 = -1, \beta_0 = 0, \beta_1 = -1: \varepsilon_2 = ((-1)(1)^2 - 2(-1))/2 = 1/2 \)

BE: \( \alpha_0 = 1, \alpha_1 = -1, \beta_0 = -1, \beta_1 = 0: \varepsilon_2 = \text{same!} = -1/2 \)

TR: \( \alpha_0 = 1, \alpha_1 = -1, \beta_0 = -1/2, \beta_1 = -1/2: \varepsilon_3: [-1-3(-1/2)]/3! = 1/12 \)
Summary

- **Linear Multi-Step Method (LMS)**

\[
\sum_{i=0}^{k} \left[ \alpha_i x_{n-i} + h \beta_i \dot{x}_{n-i} \right] = 0
\]

- We have been examining accuracy of the method
  - main analysis technique so far: Taylor Series Expansion

- Exactness conditions (in general for variable time steps)
  - make exact on all polynomials of degree \( p \) or less

- Error Analysis

\[
E_k = \left[ \sum_{i=1}^{k} \alpha_i \left( \frac{t_n - t_{n-i}}{h} \right)^{p+1} - (p+1) \beta_i \left( \frac{t_n - t_{n-i}}{h} \right)^p \right] \frac{x^{(p+1)}(t_n)}{p+1!} h^{p+1}
\]

\[\Delta \equiv \varepsilon_{p+1} x^{(p+1)}(t_n) h^{p+1}\]
Implicit Methods

\[ \sum_{i=0}^{k} [\alpha_i x_{n-i} + h\beta_i \dot{x}_{n-i}] = 0 \quad (\beta_0 \neq 0) \]
\[ \dot{x} = f(x) \]

\[ = x_n + h\beta_0 f(x_n, t_n) + \sum_{i=1}^{k} [\alpha_i x_{n-i} + h\beta_i \dot{x}_{n-i}] \]

Already computed \( x_i \) and \( \dot{x}_i \) for \( i \leq n - 1 \)
Now compute \( x_n \)

**Note:**

\[ x_n + h\beta_0 f(x_n, t_n) = b = \text{known} \]
\[ F(x_n) \triangleq x_n + h\beta_0 f(x_n, t_n) - b = 0 \]

How do we solve for \( x_n \)?

Use Newton-Raphson \( \rightarrow \)

\[ F'(x_n^{(k)})(x_n^{(k+1)} - x_n^{(k)}) + F(x_n^{(k)}) = 0 \]

\[ (I + h\beta_0 f')(x_n^{(k+1)} - x_n^{(k)}) + x_n^{(k)} + h\beta_0 f(x_n^{(k)}, t_n) - b = 0 \]

\[ x_n^{(k+1)} = x_n^{(k)} - (I + h\beta_0 f'(x_n^{(k)}))^{-1} [x_n^{(k)} + h\beta_0 f(x_n^{(k)}, t_n) - b] \]

\[ = \Delta \tilde{F}(x_n^{(k)}) \]

In particular \( \{ x_n^{(k)} \} \) converges if \( \tilde{F} \) is contractive

(a little difficult)
**Convergence**

\[ F(x_n) = x_n + h\beta_0 f(x_n, t_n) - b = 0 \]

Assume \( \frac{\partial F}{\partial x} \) is Lipschitz continuous

\[
J(x) = \frac{\partial F}{\partial x} = I + h\beta_0 \frac{\partial f}{\partial x}
\]

and

\[
\|J(x) - J(x')\| = \left\| I + h\beta_0 \frac{\partial f}{\partial x}(x) - \left( I + h\beta_0 \frac{\partial f}{\partial x}(x') \right) \right\|
\]

\[
= |h\beta_0| \left\| \frac{\partial f}{\partial x}(x) - \frac{\partial f}{\partial x}(x') \right\|
\]

\[
\leq |h\beta_0| L \|x - x'\|
\]

Lipschitz constant for \( \frac{\partial f}{\partial x} \)

Thus \( J \) is Lipschitz continuous

By Newton-Raphson convergence theorem \{ x^{(k)}_{n} \} will converge provided \( x^{(0)}_{n} \) is close enough to \( x^*_{n} \)

Since we can use \( x^{(0)}_{n} = x_{n-1} \), then for small enough \( h \), \( x^{(0)}_{n} \) will be close to \( x^*_{n} \)
Can also solve

\[ F(x_n) \Delta x_n + h\beta_0 f(x_n, t_n) - b = 0 \]

by fixed point iteration

\[ x_n^{(k+1)} = -h\beta_0 f(x_n^{(k)}, t_n) + b \]

**Convergence by Fixed Point Argument**

Assume \( \|f(x,t) - f'(x,t)\| \leq L\|x - x'\| \)

(L is independent of t)

\[ x_n^{(k+1)} = -h\beta_0 f(x_n^{(k)}, t_n) + b \]

Need G contractive, i.e.

\[ \|G(x) - G(x')\| \leq \alpha\|x - x'\| \quad 0 \leq \alpha < 1 \]

\[ \| -h\beta_0 f(x,t) + b - (-h\beta_0 f(x',t) + b) \| = \| -h\beta_0 (f(x,t) - f(x',t)) \| \]

\[ \leq h|\beta_0|L\|x - x'\| \]

Hence:

\[ \text{if } h < \frac{1}{|\beta_0|L} \text{ then convergence} \]

Note: Here we have a requirement on h for convergence, whereas for NR we had convergence provided initial guess \( x_n^{(0)} \) was close enough to \( x^*_{n} \). Thus no limitations on h *per se.*
better idea: Instead of using $x^{(0)}_n = x_{n-1}$, use extrapolation to predict $x^{(0)}_n$, i.e. fit past data with a $p$ degree polynomial, through the points $x_{n-1}, x_{n-2}, \ldots, x_{n-k-1}$.

Predicted value is $P(t_n) \to x^{(0)}_n$

Example:

Backward Euler: $p = 1 \Rightarrow$ use $1^{st}$ degree polynomial

$$P(t) = x_{n-1} + \dot{x}_{n-1}(t - t_{n-1})$$

$$= x_{n-1} + \frac{(x_{n-1} - x_{n-2})}{(t_{n-1} - t_{n-2})}(t - t_{n-1})$$

Trapezoidal:

$$P(t) = x_{n-1} + \dot{x}_{n-1}(t - t_{n-1}) + \frac{(\dot{x}_{n-1} - \dot{x}_{n-2})(t - t_{n-1})^2}{2(t_{n-1} - t_{n-2})}$$
Other Methods

1. Adams-Bashforth (explicit $\beta_0 = 0$)

2. Adams-Moulton (Implicit $\beta_0 \neq 0$)

**AB:** $p = k$, $\beta_0 = 0$, $\alpha_2, \ldots, \alpha_k = 0$

$p = 1$  
$x_n = x_{n-1} + hx_{n-1}$

$p = 2$  
$x_n = x_{n-1} + h\left(\frac{3}{2}\dot{x}_{n-1} - \frac{1}{2}\dot{x}_{n-2}\right)$

**AM:** $p = k+1$ (for $p>1$), $\beta_0 \neq 0$, $\alpha_2, \ldots, \alpha_k = 0$

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

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

$p = 3$  
$x_n = x_{n-1} + h(-\beta_0\dot{x}_n - \beta_1\dot{x}_{n-1} - \beta_2\dot{x}_{n-2})$

Question: How do we start a multistep method? 

$X_0 = ?$
**Consistency**

A method is said to be consistent if \( p \geq 1 \). Thus,

\[
\lim_{h \to 0} \frac{LTE_i}{h} \to 0 \quad \text{for all } i
\]

**Motivation**

(Global) Error \( \approx \sum_{i=1}^{M} LTE_i \) where \( M = \frac{T}{h} \)

If we want (Global) Error \( \to 0 \) then since,

\[
\text{Error} \approx \sum_{i=1}^{M} LTE_i \approx M \cdot LTE \approx \left( \frac{T}{h} \right) \cdot LTE
\]

thus,

\[
\frac{LTE}{h} \to 0 \quad \text{necessarily}
\]

Since consistent \( \Rightarrow p \geq 1 \), then \( \sum_{i=0}^{k} \alpha_i = 0 \) is consistent.

**Convergence**

A method is said to be convergent if

\[
\lim_{h \to 0} \max_{0 \leq m \leq M} \| \hat{x}(t_m) - x(t_m) \| \to 0
\]

where \( \hat{x}(t_m) \) is the computed solution and \( x(t_m) \) is the true solution, \( t_m = mh, M = T/h \)

**Motivation**

convergent means that the (Global) Error \( \to 0 \) as \( h \to 0 \)
**Stability**

A method is stable if \( \exists h_0 \) and \( k < \infty \) such that for any two different initial conditions \( x_0, x'_0 \) and \( h = T/M < h_0 \), then

\[
\| \hat{x}(t_m) - \hat{x}'(t_m) \| \leq k \| x_0 - x'_0 \|
\]

**Classical Theorem**

Consistency + Stability \( \iff \) Convergence