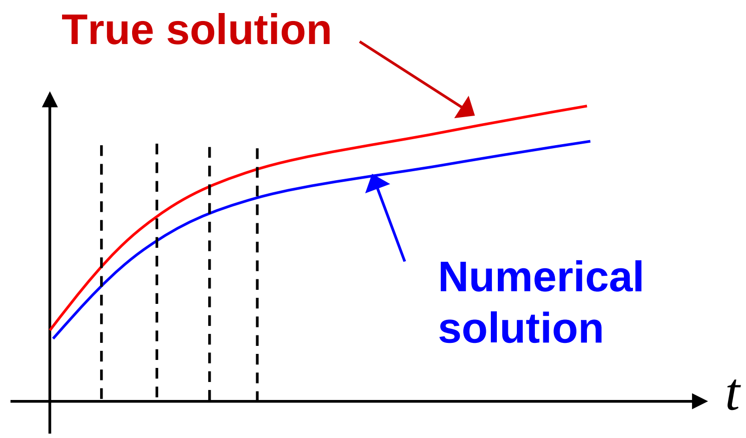
Lecture 12.
Error Control in Transient Simulation

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

Outline

- **Truncation error**
- **Local Truncation Error (LTE)**
  - **for FE/BE/TR**
- **Local error of LMS methods**
- **Time step control**
  - **Finite difference calculation**

Error Analysis



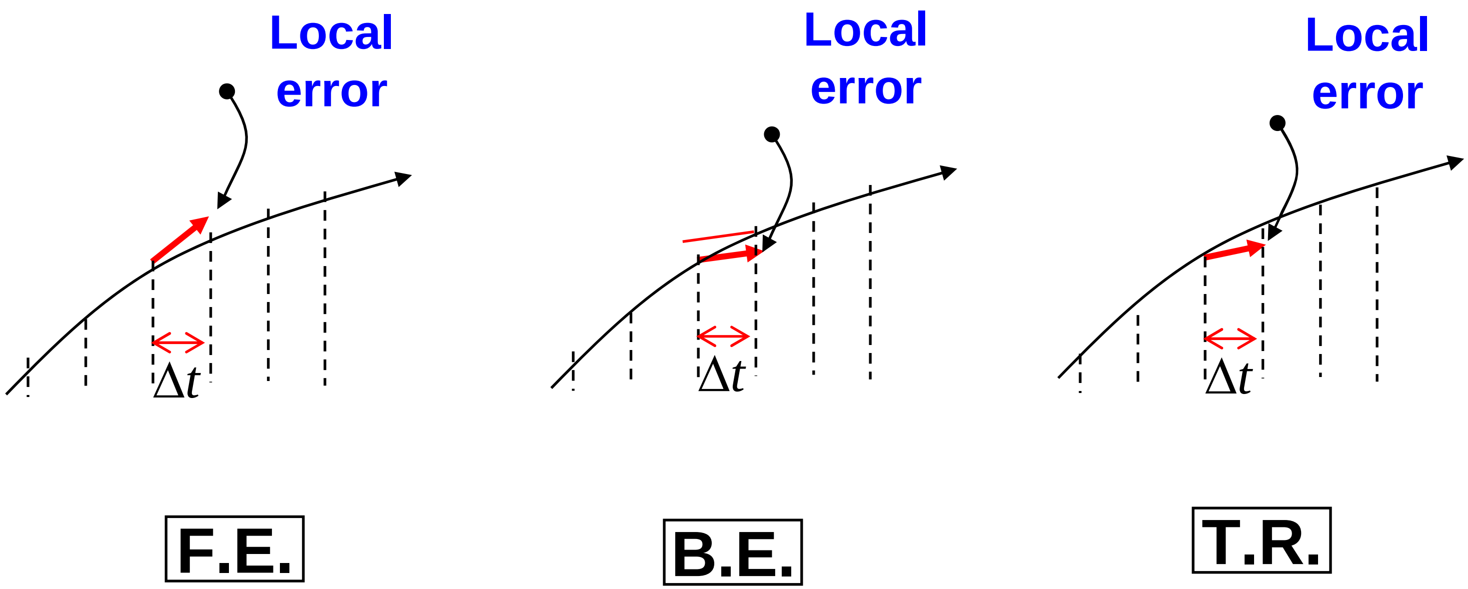
Integration Method

Round-off errors

Error of numerical solution

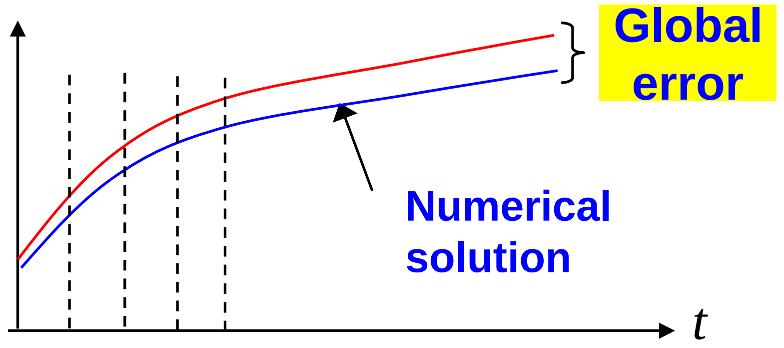
Local Error

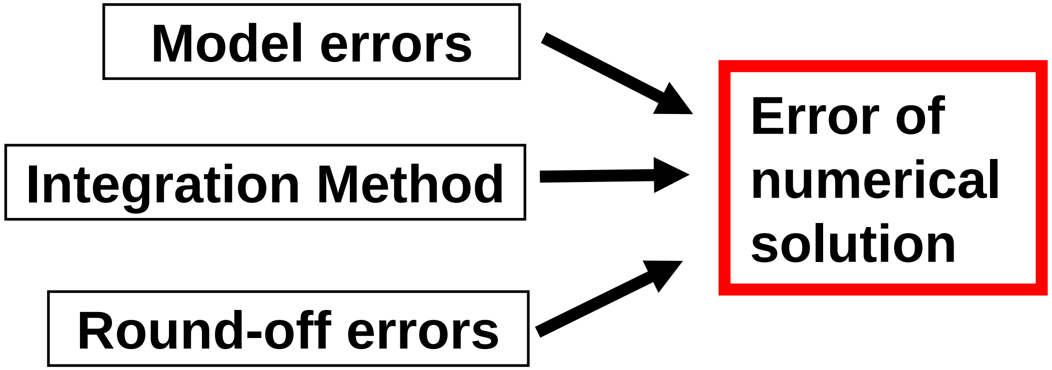
- **Local error** is the error that could happen **in one time step Δt**, assuming no error at the beginning.



Global Error

- Global error is the accumulated error over the simulation period.





Global Truncation Errors

Global Truncation Error (GTE)

Assume initial condition known: $x(t_0) = x_0$

exact

The error at time point t_{k+1}: $e_k := \left\| x(t_{k+1}) - \hat{x}_{k+1} \right\|$

No available because the exact solution $x(t_{k+1})$ is unknown.

Local Truncation Errors

Local Truncation Error (LTE) **(one-step error)**

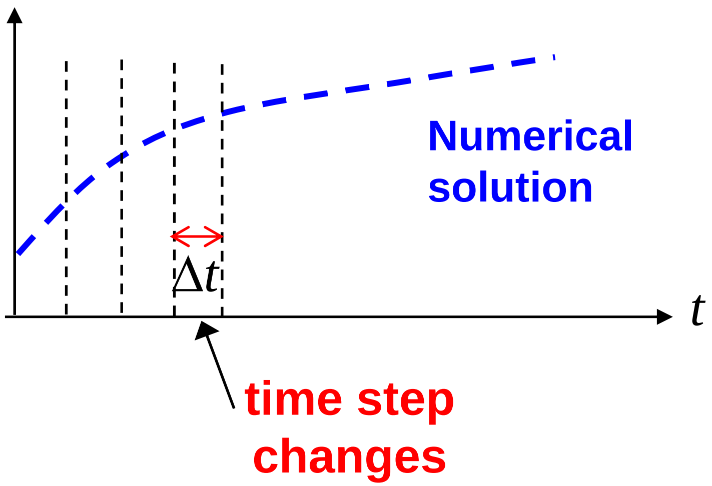
Assume x_k is known exactly: $\boxed{x(t_k) = \hat{x}_k}$ **(for convenience)**

Estimate the error at the next time point t_{k+1}:

$$e_k := \left\| x(t_{k+1}) - \hat{x}_{k+1} \right\|$$

- **Can be calculated approximately;**
- **Used to determine the next time step size (Δt) in SPICE**

Dynamic Time Step Control



- **Dynamic time step control – increase or decrease time step (Δt) for accuracy control.**
- **Adopted by most SPICE simulators.**
- **Typically use Local Truncation Error (LTE).**

Local Truncation Error (LTE)

LTE ≡ One-step error, assuming perfect past data.

$$\boxed{LTE := y(t_n) - y_n}$$

exact solution simulated

If using a linear multi-step (LMS) method:

perfect past data

$$\boxed{y_n} + \sum_{i=1}^{k} \alpha_i y(t_{n-i}) + h_n \left(\beta_0 f(y_n) + \sum_{i=1}^{m} \beta_i y'(t_{n-i}) \right) = 0$$

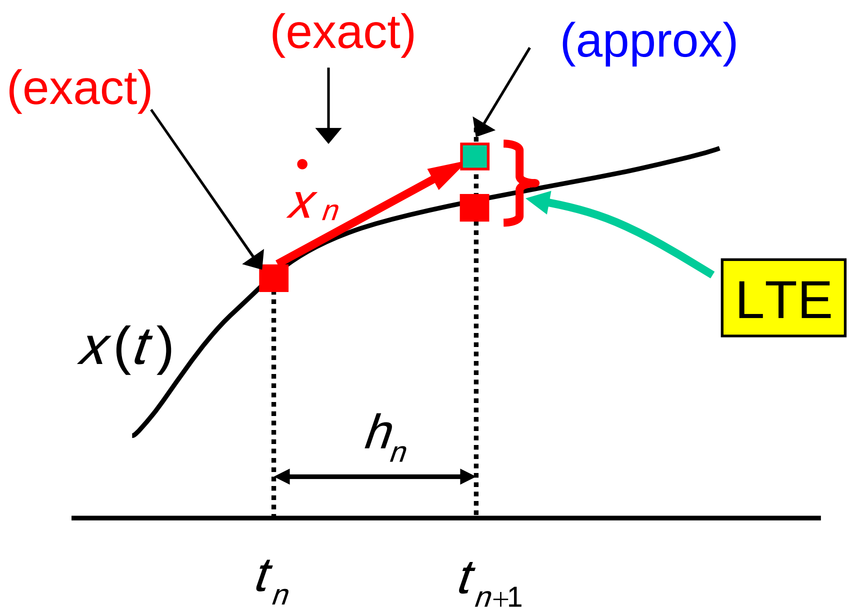
Approximation of $y(t_n)$

LTE for Typical Methods

- **We shall derive the LTE's of three basic methods**
  - **Forward Euler (FE)**
  - **Backward Euler (BE)**
  - **Trapezoidal Rule (TR)**

- **The key technique used – Taylor expansion**

Forward Euler (FE)



$x_n, \dot{x}_n$ **are assumed known exactly**

LTE of Forward Euler

$x_n, \dot{x}_n$ **are assumed known exactly**

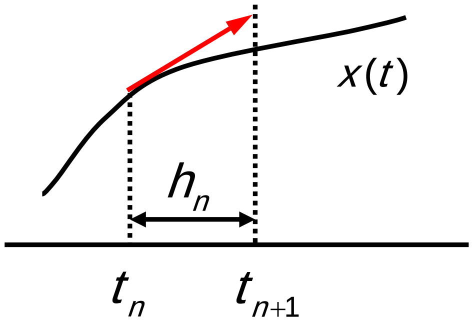
(assume exact)

$$\boxed{\begin{aligned} x_n &= x(t_n) \\ \dot{x}_n &= \dot{x}(t_n) \end{aligned}}$$

Forward Euler: $\boxed{x_{n+1} = x_n + h_n \dot{x}_n}$

$$\Updownarrow$$

$$\boxed{x(t_n) + h_n \dot{x}(t_n) = x_{n+1} \approx x(t_{n+1})}$$



The error at t_{n+1} is the LTE:

$$\boxed{LTE := x(t_{n+1}) - x_{n+1} = ?}$$

LTE of Forward Euler

Taylor expanding *x(t)* at *t = t*$_{n+1}$:

(assume exact)

$$x(t_{n+1}) = x(t_n) + h_n \dot{x}(t_n) + \frac{h_n^2}{2}\ddot{x}(t_n) + O(h_n^3)$$

$$x_n = x(t_n)$$
$$\dot{x}_n = \dot{x}(t_n)$$

$$\Rightarrow \quad x(t_{n+1}) = \left[x_n + h_n \dot{x}_n \right] + \frac{h_n^2}{2}\ddot{x}_n + O(h_n^3)$$

$$x_{n+1} = x_n + h_n \dot{x}_n \qquad = x_{n+1} + \frac{h_n^2}{2}\ddot{x}_n + O(h_n^3)$$

Forward Euler

formula

LTE of Forward Euler (cont'd)

Ignore the high-order terms.

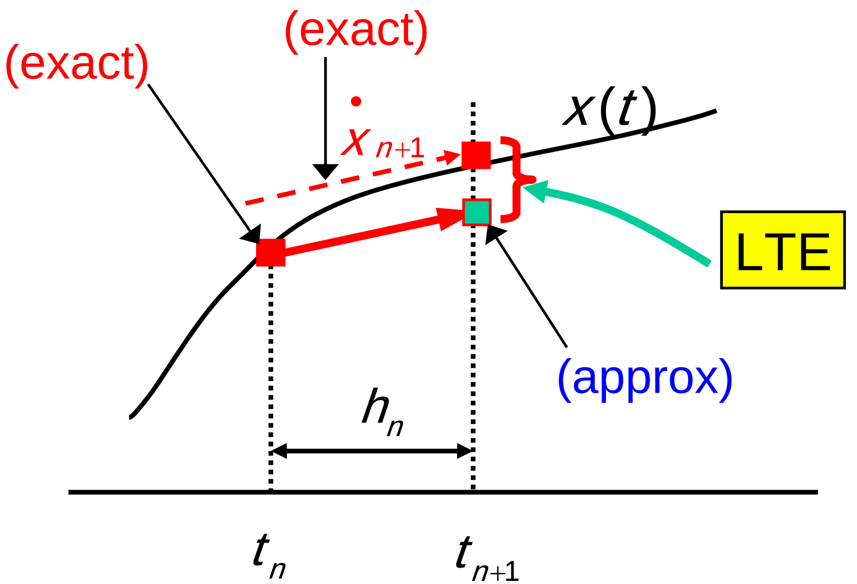
$$LTE := x(t_{n+1}) - x_{n+1} = \underbrace{\frac{h_n^2}{2}\ddot{x}_n + \cancel{O(h_n^3)}}_{\text{LTE}}$$

0

LTE

$$\Rightarrow \boxed{LTE_{FE} = \frac{h_n^2}{2}\ddot{x}_n}$$

Backward Euler (BE)



x_n, $\dot{x}_{n+1}$ **are assumed known exactly**

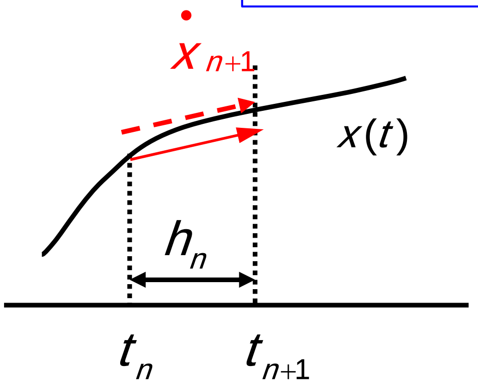
LTE of Backward Euler

(assuming)

$$x_n = x(t_n)$$

$$\dot{x}_{n+1} = \dot{x}(t_{n+1})$$

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



approx.

perfect

$$\boxed{x_{n+1} = x_n + h_n \dot{x}_{n+1}}$$

$$\Updownarrow$$

$$\boxed{x(t_n) + h_n \dot{x}(t_{n+1}) = x_{n+1} \approx x(t_{n+1})}$$

The error between x_{n+1} and $x(t_{n+1})$

$$LTE := x(t_{n+1}) - x_{n+1} = ?$$

Backward Euler

$$x_n = x(t_n)$$
$$\dot{x}_{n+1} = \dot{x}(t_{n+1})$$

Taylor expand $x(t)$ at $t = t_{n+1}$

$$x(t) = x(t_{n+1}) + \dot{x}(t_{n+1})(t - t_{n+1}) + \frac{\ddot{x}(t_{n+1})}{2}(t - t_{n+1})^2 + O((t - t_{n+1})^3)$$

Evaluate at $t = t_n$:

$$x(t_n) = x(t_{n+1}) - h_n \dot{x}(t_{n+1}) + \frac{h_n^2}{2}\ddot{x}(t_{n+1}) + O(h_n^3)$$

$$\Rightarrow \quad x(t_{n+1}) = \left[x_n + h_n \dot{x}_{n+1} \right] - \frac{h_n^2}{2}\ddot{x}(t_{n+1}) + O(h_n^3)$$

$$= x_{n+1} - \frac{h_n^2}{2}\ddot{x}(t_{n+1}) + O(h_n^3)$$

$$\boxed{x_{n+1} = x_n + h_n \dot{x}_{n+1}}$$

by B.E.

Backward Euler (cont'd)

$$x_n = x(t_n)$$
$$\dot{x}_{n+1} = \dot{x}(t_{n+1})$$

- $$x(t_{n+1}) = \left[x_n + h_n \dot{x}_{n+1} \right] - \frac{h_n^2}{2}\ddot{x}(t_{n+1}) + O(h_n^3)$$
$$= x_{n+1} - \frac{h_n^2}{2}\ddot{x}(t_{n+1}) + O(h_n^3)$$

$$x_{n+1} = x_n + h_n \dot{x}_{n+1}$$

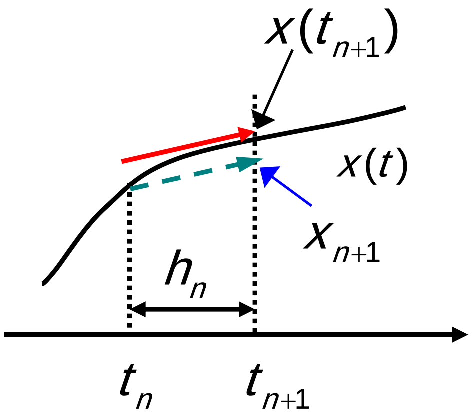
by B.E.

- $$\boxed{LTE := x(t_{n+1}) - x_{n+1} = \underbrace{-\frac{h_n^2}{2}\ddot{x}(t_{n+1}) + \cancel{O(h_n^3)}}} \Rightarrow \boxed{LTE_{BE} = -\frac{h_n^2}{2}\ddot{x}(t_{n+1})}$$

Clarification of the Assumption

$$\dot{x}(t_{n+1}) = f\left(x(t_{n+1})\right)$$

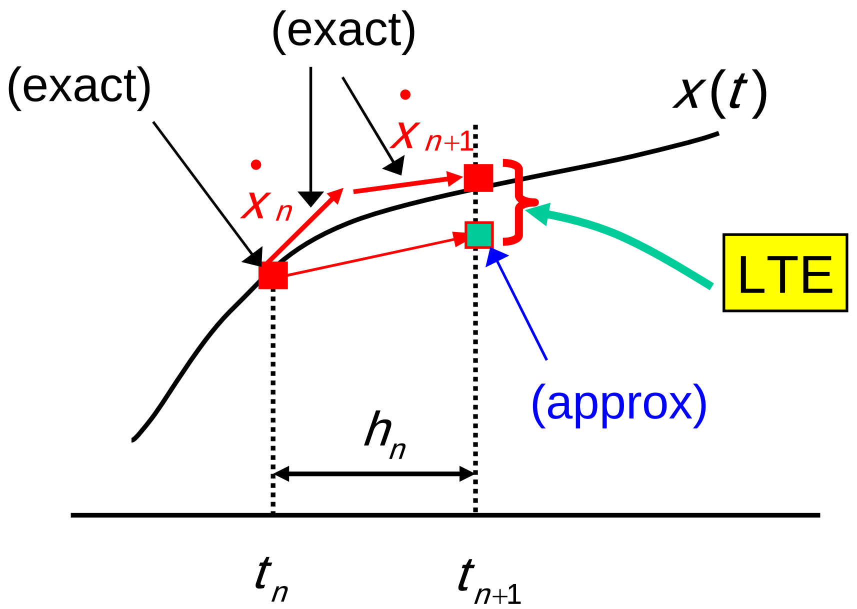
assumed known — unknown



$x(t_{n+1})$ will be solved from BE as x_{n+1}.

$$x_{n+1} = x(t_n) + h_n \dot{x}(t_{n+1})$$

Trapezoidal Rule (TR)

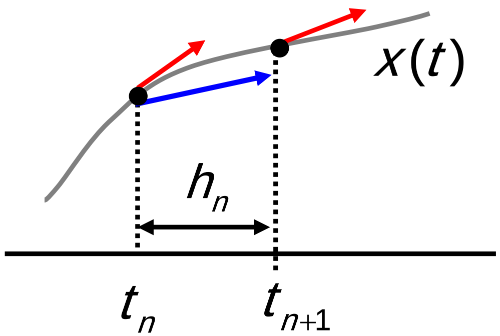


$x_n, \dot{x}_n, \dot{x}_{n+1}$ **are assumed known exactly**

LTE of Trapezoidal Rule (TR)

Assume exactly:

$$x_n = x(t_n) \qquad \dot{x}_n = \dot{x}(t_n) \qquad \dot{x}_{n+1} = \dot{x}(t_{n+1})$$



$$\dot{x}_{n+1} := f(x_{n+1})$$

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

$$x(t_n) + \frac{h_n}{2}\left(\dot{x}(t_n) + \dot{x}(t_{n+1})\right) = x_{n+1} \approx x(t_{n+1})$$

error?

The error between x_{n+1} and $x(t_{n+1})$?

$$LTE := x(t_{n+1}) - x_{n+1} = ?$$

LTE of T.R.

Taylor expand *x(t)* at *t* = t_n and evaluate at *t* = t_{n+1} :

$$x(t_{n+1}) = x_n + h_n \dot{x}_n + \frac{h_n^2}{2}\ddot{x}_n + \frac{h_n^3}{6}\dddot{x}_n + O(h_n^4) \qquad h_n = t_{n+1} - t_n$$

(to be cancelled)

Taylor expand *x'(t)* at *t* = t_n and evaluate at *t* = t_{n+1} :

$$\dot{x}(t_{n+1}) = \dot{x}_n + h_n \ddot{x}_n + \frac{h_n^2}{2}\dddot{x}_n + O(h_n^3)$$

LTE of Trapezoidal Rule

$$\begin{cases} x(t_{n+1}) = x_n + h_n \dot{x}_n + \cancel{\frac{h_n^2}{2}\ddot{x}_n} + \frac{h_n^3}{6}\dddot{x}_n + O(h_n^4) \\ \dot{x}(t_{n+1}) = \dot{x}_n + \cancel{h_n \ddot{x}_n} + \frac{h_n^2}{2}\dddot{x}_n + O(h_n^3) \end{cases} \longleftarrow \boxed{\times \frac{h_n}{2}}$$

$$h_n = t_{n+1} - t_n$$

$$\dot{x}(t_{n+1}) = f(x(t_{n+1}), t_{n+1})$$

$$\dot{x}_{n+1} := f(x_{n+1}, t_{n+1})$$

- **Eliminate** $\ddot{x}_n$ **to get**

$$x(t_{n+1}) = x_n + \frac{h_n}{2}\left[\dot{x}_n + \dot{x}(t_{n+1})\right] - \frac{h_n^3}{12}\dddot{x}_n + O(h_n^4)$$

$$= x_{n+1} - \frac{h_n^3}{12}\dddot{x}_n + O(h_n^4)$$

by T.R.

Assuming $\dot{x}_{n+1} = \dot{x}(t_{n+1})$

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

TR LTE (cont'd)

- $$x(t_{n+1}) = x_n + \frac{h_n}{2}\left[\dot{x}_n + \dot{x}(t_{n+1})\right] - \frac{h_n^3}{12}\dddot{x}_n + O(h_n^4)$$
  $$= x_{n+1} - \frac{h_n^3}{12}\dddot{x}_n + O(h_n^4)$$

$$\Rightarrow \quad x(t_{n+1}) - x_{n+1} = \underbrace{-\frac{h_n^3}{12}\dddot{x}_n + O(h_n^4)}_{\text{LTE}} \quad \Rightarrow \quad \boxed{LTE_{TR} = -\frac{h_n^3}{12}\dddot{x}_n}$$

LMS Error Analysis

Local error of LMS methods

Error expression of pth order LMS

Error Analysis of LMS Methods

- **We shall compare "local error" and "local truncation error":**
  - **Local Error – also known as the Error of the formula (or called "residue")**
  - **LTE – One-step Error of the solution**

General LMS formula:

$$\boxed{\sum_{i=0}^{k} \alpha_i y_{n-i} + h_n \sum_{j=0}^{m} \beta_j \dot{y}_{n-j} = 0, \quad (\alpha_0 = 1)}$$

LMS Error

General LMS :

$$\sum_{i=0}^{k} \alpha_i y_{n-i} + h_n \sum_{j=0}^{m} \beta_j \dot{y}_{n-j} = 0, \quad (\alpha_0 = 1)$$

simulated values

If the **simulated** values are replaced by the **exact** values, the LMS formula might result in "errors", **because the exact solution might not satisfy the LMS formula.**

(simulated)

$$y_{n-i} \leftrightarrow y(t_{n-i})$$

$$\dot{y}_{n-j} \leftrightarrow \dot{y}(t_{n-j})$$

(exact)

$$\Rightarrow \quad \sum_{i=0}^{k} \alpha_i y(t_{n-i}) + h_n \sum_{j=0}^{m} \beta_j \dot{y}(t_{n-j}) \neq 0$$

Error

LMS Local Error

$$\sum_{i=0}^{k} \alpha_i y(t_{n-i}) + h_n \sum_{j=0}^{m} \beta_j \dot{y}(t_{n-j}) \neq 0$$

called "Local Error"

Local Error (LE) is defined as the residue of the LMS formula with all y_i and y'_i substituted by their *exact* values, i.e.
$y_i = y(t_i)$, $y'_i = f(y(t_i))$

Define:

$$LE \triangleq \sum_{i=0}^{k} \alpha_i y(t_{n-i}) + h_n \sum_{j=0}^{m} \beta_j \dot{y}(t_{n-j})$$

Local Error (LE) Analysis

Assuming *perfect* past data:

$$y(t_{n-i}) = y_{n-i} \qquad \textit{for} \quad i = 1, 2, \ldots, k$$

Solve the current solution y_n :

$$y_n \approx y(t_n)$$

In general, y_n is only an approximation of $y(t_n)$.

$$\Rightarrow \quad LTE_n := y(t_n) - y_n$$

Question: What is the relation between LE and LTE?

Local Error (LE) Analysis

$$LE_n = \sum_{i=0}^{k} \alpha_i y(t_{n-i}) + h_n \sum_{j=0}^{m} \beta_j \dot{y}(t_{n-j})$$

$$= y(t_n) + h_n \beta_0 \dot{y}(t_n) + \sum_{i=1}^{k} \alpha_i y(t_{n-i}) + h_n \sum_{j=1}^{m} \beta_j \dot{y}(t_{n-j})$$

$$= y(t_n) + h_n \beta_0 \dot{y}(t_n) + \sum_{i=1}^{k} \alpha_i y_{n-i} + h_n \sum_{j=1}^{m} \beta_j \dot{y}_{n-j}$$

by assumption

$$LE_n = \left[y(t_n) - y_n \right] + h_n \beta_0 \left[\dot{y}(t_n) - \dot{y}_n \right] + \sum_{i=0}^{k} \alpha_i y_{n-i} + h_n \sum_{j=0}^{m} \beta_j \dot{y}_{n-j}$$

Note the index change !

Local Error (LE)

$$LE_n = \left[y(t_n) - y_n\right] + h_n\beta_0\left[\dot{y}(t_n) - \dot{y}_n\right] + \underbrace{\sum_{i=0}^{k}\alpha_i y_{n-i} + h_n\sum_{j=0}^{m}\beta_j\,\dot{y}_{n-j}}_{=\,0\text{ by LMS definition}}$$

$$\Rightarrow\quad LE = \left[y(t_n) - y_n\right] + h_n\beta_0\left[\dot{y}(t_n) - \dot{y}_n\right]$$

$$= \underbrace{\left[y(t_n) - y_n\right]}_{=\,LTE} + h_n\beta_0\left[f\left(y(t_n)\right) - \dot{y}_n\right]$$

LTE = 0 ➔ LE = 0 (easy to see)

$$LTE = y(t_n) - y_n = 0 \quad \Rightarrow \quad f(y(t_n)) - \dot{y}_n = f(y(t_n)) - \dot{y}(t_n) = 0$$

What about the converse: LE = 0 ➔ LTE = 0 ?

LTE versus LE

$$\boxed{LTE \triangleq y(t_n) - y_n}$$

$$LE = \left[y(t_n) - y_n \right] + h_n \beta_0 \left[\dot{y}(t_n) - f(y_n) \right]$$

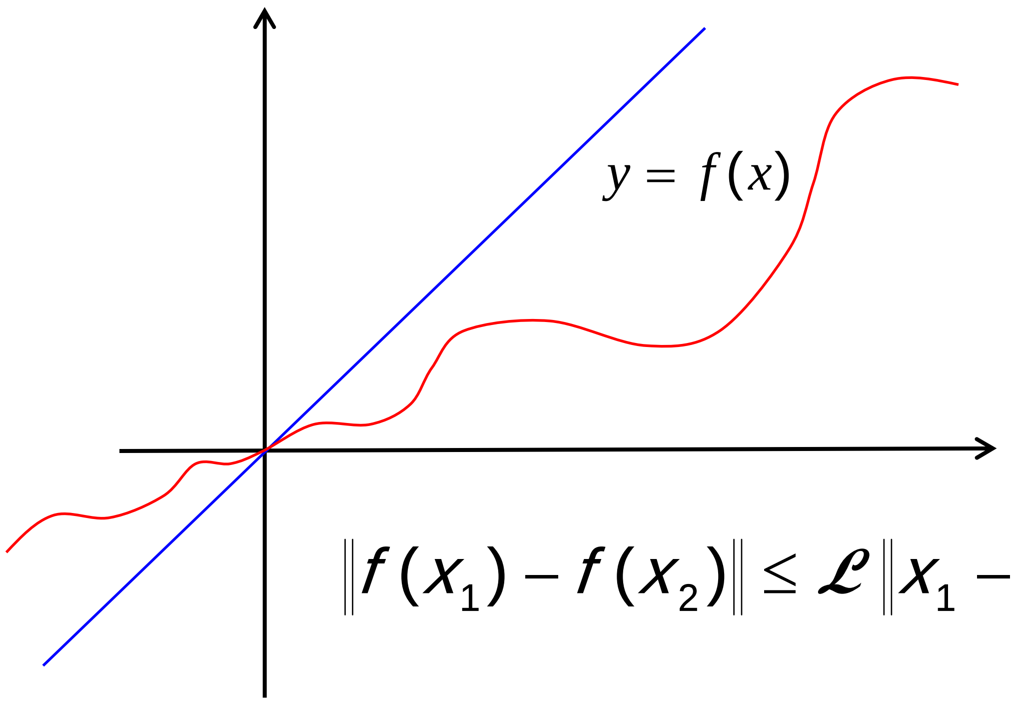
$$LTE = -h_n \beta_0 \left[\dot{y}(t_n) - f(y_n) \right] + LE$$

$$= -h_n \beta_0 \left[f\left(y(t_n) \right) - f(y_n) \right] + LE$$

We have to assume that f(·) is "Lipschitz" in order to derive a bound for the "LTE" in terms of "LE".

Lipschitz Condition

- **A function y = f(x) is Lipschitz if the variation of y can be "linearly bounded" by the variation of x.**



$$\left\| f(x_1) - f(x_2) \right\| \le \mathcal{L} \left\| x_1 - x_2 \right\|$$

$\mathcal{L}$ **is called a *Lipshitz constant*.**

LTE Bound by LE

$$\begin{aligned}\|LTE\| &= \|y(t_n) - y_n\| \\ &\le |h_n\beta_0|\,\|f(y(t_n)) - f(y_n)\| + \|LE\| \\ &\le |h_n\beta_0|\,\mathcal{L}\,\|y(t_n) - y_n\| + \|LE\| \\ &\le |h_n\beta_0|\,\mathcal{L}\,\|LTE\| + \|LE\|\end{aligned}$$

$\mathcal{L}$ is the Lipschitz constant for f(x)

$$\|f(x_1) - f(x_2)\| \le \mathcal{L}\,\|x_1 - x_2\|$$

for $h_n = h$

$$\boxed{\|LTE\| \le \frac{\|LE\|}{1 - |h\beta_0|\,\mathcal{L}}}$$

If $\mathcal{L}$ is large, choose *h* small to get a valid bound.

Taylor Analysis of Local Error

$$LE = \mathcal{E}[y(t),h] := \sum_{i=0}^{k} \alpha_i y(t_{n-i}) + h \sum_{i=0}^{m} \beta_i \dot{y}(t_{n-i})$$

Suppose y(t) is smooth. Its Taylor expansion at $t = t_n$

$$y(t) = y(t_n) + y^{(1)}(t_n)(t - t_n) + \ldots + \frac{y^{(p+1)}(t_n)}{(p+1)!}(t - t_n)^{p+1} + \ldots$$

$$= q_p(t) + r_p(t)$$

remainder

$$q_p(t) = \sum_{i=0}^{p} \frac{y^{(i)}(t_n)}{i!}(t - t_n)^i$$

$$r_p(t) = \frac{y^{(p+1)}(t_n)}{(p+1)!}(t - t_n)^{p+1} + \ldots$$

$$= O\left((t - t_n)^{p+1}\right)$$

(polynomial part)

for the remainder

Analytical Expression for LE

linear

$$LE = \mathcal{E}[y(t),h] := \sum_{i=0}^{k} \alpha_i y(t_{n-i}) + h \sum_{i=0}^{m} \beta_i \dot{y}(t_{n-i})$$

plug in

$$y(t) = q_p(t) + r_p(t)$$

pth order polynomial: $q_p(t) = \sum_{i=0}^{p} \frac{y^{(i)}(t_n)}{i!}(t - t_n)^i$

$\Rightarrow$

remainder: $r_p(t) = \frac{y^{(p+1)}(t_n)}{(p+1)!}(t - t_n)^{p+1} + O\left((t - t_n)^{p+2}\right)$

Analytical Expression for LE

Assume the LMS is a pth order method; namely, it is exact for any pth order polynomials.

$$\cancel{LE} = \mathcal{E}\left[q_p(t), h\right] + \mathcal{E}\left[r_p(t), h\right]$$

with $\mathcal{E}\left[q_p(t), h\right] \to 0$

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

$$+ O(h^{p+2})$$

Terms from the remainder of y(t)

$$y(t) = y(t_n) + y^{(1)}(t_n)(t - t_n) + \ldots + \left[\frac{y^{(p+1)}(t_n)}{(p+1)!}(t - t_n)^{p+1}\right] + \ldots$$

Analytical Expression for Local Error

$$
\begin{aligned}
LE &= \mathcal{E}\left[r_p(t), h \right] \\
&= \sum_{i=0}^{k}\left[\frac{\alpha_i y^{(p+1)}(t_n)}{(p+1)!}(t_{n-i}-t_n)^{p+1} \right] + \sum_{i=1}^{m}\left(\frac{h\beta_i y^{(p+1)}(t_n)}{p!}(t_{n-i}-t_n)^{p} \right) + O(h^{p+2}) \\
&= \sum_{i=0}^{k}\left[\alpha_i \left(\frac{t_{n-i}-t_n}{h} \right)^{p+1} \right] + (p+1)\sum_{i=1}^{m}\left[\beta_i \left(\frac{t_{n-i}-t_n}{h} \right)^{p} \right] \frac{y^{(p+1)}(t_n)}{(p+1)!} h^{p+1} + O(h^{p+2}) \\
&= \varepsilon_{p+1} y^{(p+1)}(t_n) h^{p+1} + \cdots
\end{aligned}
$$

High Order Terms

$O(h^{p+1})$

coefficient

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

for any method of order p

Expression for Local Error

$$\varepsilon_{p+1} \triangleq \frac{1}{(p+1)!}\left\{\sum_{i=1}^{k}\alpha_i\left(\frac{t_{n-i}-t_n}{h}\right)^{p+1} + (p+1)\sum_{i=1}^{m}\beta_i\left(\frac{t_{n-i}-t_n}{h}\right)^{p}\right\}$$

$$h = t_n - t_{n-1}$$

time stepsize

$$\boxed{\varepsilon_{p+1} \triangleq \frac{1}{(p+1)!}\sum_{\mathrm{i}=1}^{\mathrm{k}}(-i)^{p+1}\alpha_i + \frac{1}{p!}\sum_{\mathrm{i}=1}^{\mathrm{m}}(-i)^{p}\beta_i}$$

Coefficient of the local error (LE)

$$LE = \mathcal{E}\left[r_p(t), h\right] = \varepsilon_{p+1}\, y^{(p+1)}(t_n)h^{p+1} + \cdots$$

(for a pth order LMS)

LE Examples

Applying the analytical LE expression to some basic integration methods:

- **Forward Euler**
- **Backward Euler**
- **Trapezoidal Rule**
- **Gear Integration Formulas**

LE of Forward Euler

$$\boxed{\varepsilon_{p+1} \triangleq \frac{1}{(p+1)!}\sum_{i=1}^{k}(-i)^{p+1}\alpha_i + \frac{1}{p!}\sum_{i=1}^{m}(-i)^{p}\beta_i}$$

F.E. : $\alpha_0 = 1, \alpha_1 = -1, \beta_0 = 0, \beta_1 = -1$ → Order of method p = 1

$\varepsilon_2 = [\,(-1)(-1)^2 - 2(-1)\,]\,/2 =$ ***1/2*** ← **Coefficient of LE**

$$LE = \mathcal{E}\left[r_1(t),\, h\right] = \varepsilon_2\, \ddot{y}(t_n)h^2 = \frac{1}{2}\ddot{y}(t_n)h^2$$

(Same result as the LTE analysis)

LE of Backward Euler

$$\boxed{\varepsilon_{p+1} \triangleq \frac{1}{(p+1)!}\sum_{i=1}^{k}(-i)^{p+1}\alpha_i + \frac{1}{p!}\sum_{i=1}^{m}(-i)^{p}\beta_i}$$

⇩

B.E. : $\alpha_0=1$, $\alpha_1=-1$, $\beta_0=-1$, $\beta_1=0$ ➔ Order of method $p = 1$

$\varepsilon_2 = [\,(-1)(-1)^2\,]\,/2 =$ ***-1/2*** ⬅ **Coefficient of LE**

LE of Trapezoidal Rule

$$\boxed{\varepsilon_{p+1} \triangleq \frac{1}{(p+1)!}\sum_{i=1}^{k}(-i)^{p+1}\alpha_i + \frac{1}{p!}\sum_{i=1}^{m}(-i)^{p}\beta_i}$$

$\Downarrow$

T.R. : α_0=1, α_1=-1, β_0=-1/2, β_1=-1/2 ➔ Order of method p = 2

ε_3 = [(-1)(-1)^3 - (3/2) (-1)^2] /3! = ***-1/12*** ← **Coefficient of LE**

LE of Gear's Formula

Gear's formula (kth order):

$$\boxed{\dot{y}_n = -\frac{1}{\beta_0 h}\sum_{i=0}^{k}\alpha_i y_{n-i}}$$

The coefficient equations:

$$\boxed{\sum_{i=0}^{k}(i^q)\alpha_i - (q \bullet 0^{q-1})\beta_0 = 0;} \qquad q = 0,1,\cdots,k$$

Truncation error coefficient:

$$\boxed{\varepsilon_{k+1} \triangleq \frac{1}{(k+1)!}\sum_{i=1}^{k}(-i)^{k+1}\alpha_i}$$

Examples (Gear Coefficients)

1st order Gear (Backward Euler) ***k = 1***

$$\alpha_0 = 1 \qquad \begin{cases} \alpha_0 + \alpha_1 = 0 \\ \alpha_1 = \beta_0 \end{cases} \Rightarrow \boxed{\alpha_1 = -1,\ \beta_0 = -1,}$$

$$\varepsilon_2 = \frac{\alpha_1}{2!} = -\frac{1}{2}$$

2nd order Gear ***k = 2***

$$\alpha_0 = 1 \qquad \begin{cases} \alpha_0 + \alpha_1 + \alpha_2 = 0, \\ \alpha_1 + 2\alpha_2 = \beta_0, \\ \alpha_1 + 4\alpha_2 = 0. \end{cases} \Rightarrow \boxed{\alpha_1 = -\frac{4}{3},\ \alpha_2 = \frac{1}{3},\ \beta_0 = -\frac{2}{3},}$$

$$\varepsilon_3 = \frac{\alpha_1 + 2^3\alpha_2}{3!} = \frac{2}{9} = 0.222222$$

Gear Coefficients

3rd order Gear

***k* = 3**

$$\sum_{i=0}^{3} \alpha_i y_{n-i} + h\beta_0 \dot{y}_n = 0$$

Coef. eqn.:

$$\alpha_0 = 1$$

$$\begin{cases} \alpha_1 + \alpha_2 + \alpha_3 = -1, \\ \alpha_1 + 2\alpha_2 + 2\alpha_3 = \beta_0, \\ \alpha_1 + 4\alpha_2 + 9\alpha_3 = 0, \\ \alpha_1 + 8\alpha_2 + 27\alpha_3 = 0. \end{cases} \Rightarrow \boxed{\begin{array}{l} \alpha_1 = -\dfrac{18}{11}, \alpha_2 = \dfrac{9}{11}, \ \alpha_3 = -\dfrac{2}{11}, \\ \beta_0 = -\dfrac{6}{11}, \end{array}}$$

$$\boxed{\varepsilon_{k+1} \triangleq \frac{1}{(k+1)!} \sum_{i=1}^{k} (-i)^{k+1} \alpha_i} \Rightarrow$$

$$\varepsilon_4 = \frac{\alpha_1 + 2^4 \alpha_2 + 3^4 \alpha_3}{4!} = -\frac{3}{22} = -0.1363636$$

Time Step Control

Calculation of Finite Difference

Time Step Control

- **Let *E* be the allowed LTE bound at any time point.**

$$\left| LTE(t_n) \right| \le E$$

- **Then the time step bound for the TR is calculated as**

$$\left| \frac{h_n^3}{12} \dddot{x}_n \right| \le E \quad \Rightarrow \quad h_n \le \left({12E} \Big/ {\left| \dddot{x}_n \right|} \right)^{1/3}$$

$$\boxed{LTE_{TR}(t_n) = -\frac{h_n^3}{12} \dddot{x}_n}$$

We need to calculate the high-order derivative approximately by finite differences.

Finite Difference Calculation

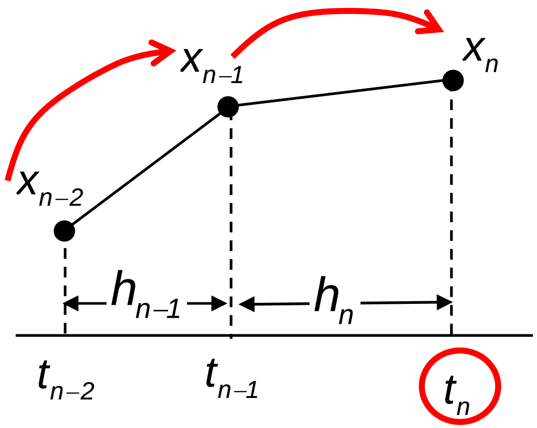
- Let $\Delta^k(t_n)$ be the **kth order finite difference (FD) at t_n**.
- It can be calculated by recursive formulas using the past data.

1st order FD:

$$\Delta x(t_n) = \frac{x_n - x_{n-1}}{h_n} \qquad h_n = t_n - t_{n-1}$$

2nd order FD:

$$\Delta^2 x(t_n) \approx \frac{\Delta x(t_n) - \Delta x(t_{n-1})}{h_n}$$



Higher-order FD computation will have to use polynomial interpolation.

FD Calculation (cont'd)

High-order finite-differences with varying time-steps:

$$\Delta^p x(t_n) = \frac{\Delta^{p-1}x(t_n) - \Delta^{p-1}x(t_{n-1})}{(t_n - t_{n-p})(p-1)/p}$$

$p \geq 2$

Derived from Lagrange interpolation (details omitted).

(approximated by)

$$\Delta^p x(t_n) \approx \frac{\Delta^{p-1}x(t_n) - \Delta^{p-1}x(t_{n-1})}{(h_n + h_{n-1} + \cdots + h_{n-p-2})}$$

OR

$$\Delta^p x(t_n) \approx \frac{\Delta^{p-1}x(t_n) - \Delta^{p-1}x(t_{n-1})}{(p-1)h_n}$$

Implemented in Spice3f5.

*** These formulas are equivalent for the case of uniform time step.**

LTE for C with B.E.

- **With Backward Euler, the LTE's for capacitances (C) is given by**

$$i = \mathbf{C}\frac{dv}{dt} \;\Rightarrow\; \frac{d^2v}{dt^2} = \frac{1}{\mathbf{C}}\frac{di}{dt} \;\Rightarrow\; \ddot{v}(t_n) = \frac{1}{\mathbf{C}}\frac{i(t_{n+1}) - i(t_n)}{h_n}$$

$$\boxed{LTE_{BE} = -\frac{h_n^2}{2}\ddot{x}_n} \;\Rightarrow\; \boxed{\left|LTE_{BE}\right| = \frac{h_n^2}{2}\ddot{v}_n = \frac{h_n}{2C}\left|i(t_{n+1}) - i(t_n)\right| \le E_C}$$

$$\Rightarrow\; \boxed{\Delta t_n \le \frac{2C}{\left|i_C(t_{n+1}) - i_C(t_n)\right|}E_C}$$

E_C is the LTE bound for C

$$\Delta t_n = t_{n+1} - t_n = h_n$$

LTE for L with B.E.

- **By Backward Euler, the LTE for inductances (L) is given by**

$$v = \mathbf{L}\frac{di}{dt} \quad \Rightarrow \quad \frac{d^2 i}{dt^2} = \frac{1}{\mathbf{L}}\frac{dv}{dt} \quad \Rightarrow \quad \ddot{i}(t_n) = \frac{1}{\mathbf{L}}\frac{v(t_{n+1}) - v(t_n)}{h_n}$$

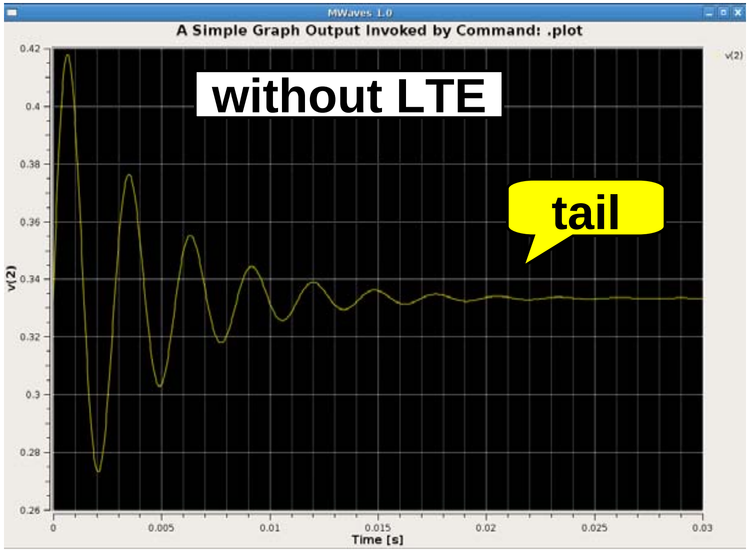
$$\boxed{LTE_{BE} = -\frac{h_n^2}{2}\ddot{x}_n} \quad \Rightarrow \quad \boxed{\left|LTE_{BE}\right| = \frac{h_n^2}{2}\ddot{i}_n = \frac{h_n}{2L}\left|v(t_{n+1}) - v(t_n)\right| \le E_L}$$

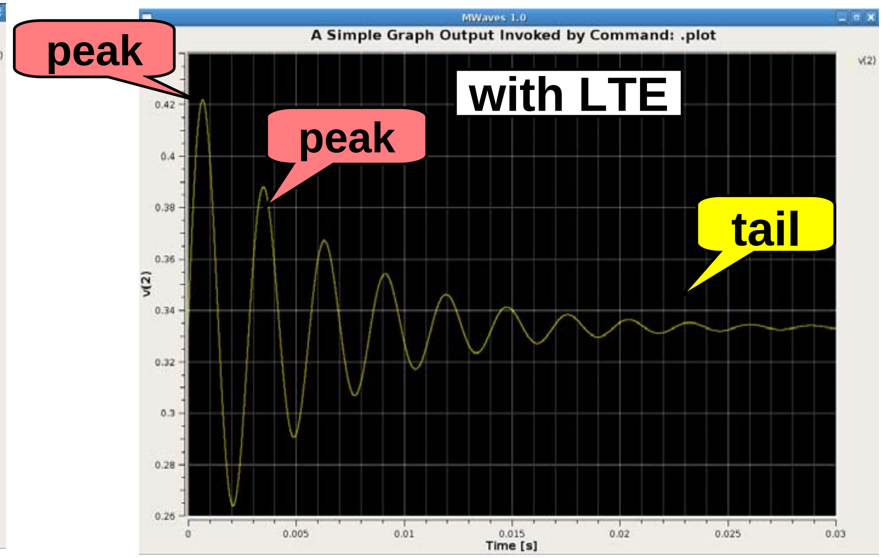
$$\Rightarrow \quad \boxed{\Delta t_n \le \frac{2L}{\left|v_L(t_{n+1}) - v_L(t_n)\right|}E_L}$$

E_L is the LTE bound

$$\Delta t_n = t_{n+1} - t_n = h_n$$

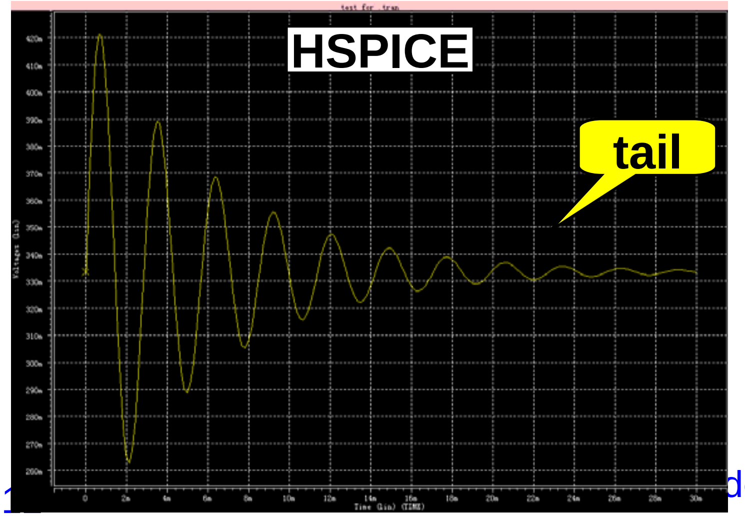
Student Implementations





Implementation by XU Hui

(class 2008)



Acknowledgement

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