Lecture 10.
Integration Methods for Transient Simulation
-- Linear Multi-Step Methods

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

Outline

- **Linear Multi-Step (LMS) methods**
- **Coefficients of integration formula**
- **Order of integration method**
- **Implicit method with nonlinear devices**

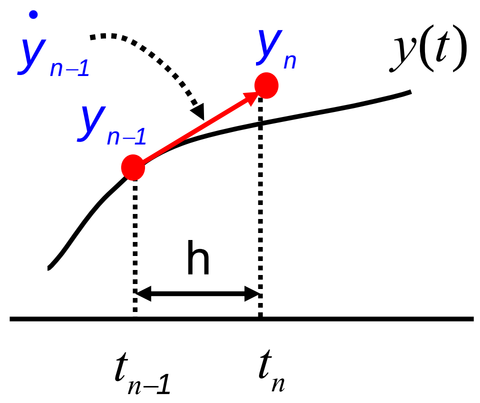
Review of FE and BE

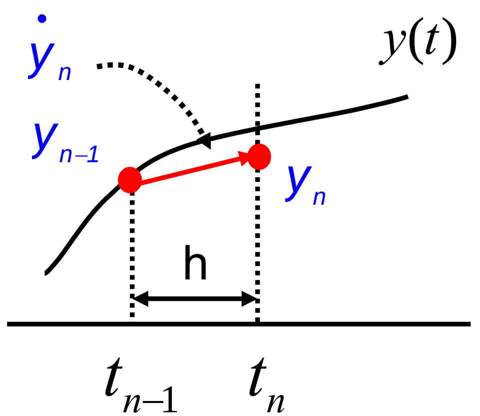
$$\boxed{\dot{y}(t) = \frac{dy(t)}{dt} = f(y(t))}$$

Let y_n be the simulated value of $y(t_n)$:

$$y_n \approx y(t_n)$$

Let $\dot{y}_n := f(y_n) \approx f(y(t_n)) = \dot{y}(t_n)$





$$\boxed{y_n - y_{n-1} - h\,\dot{y}_{n-1} = 0}$$

Forward Euler

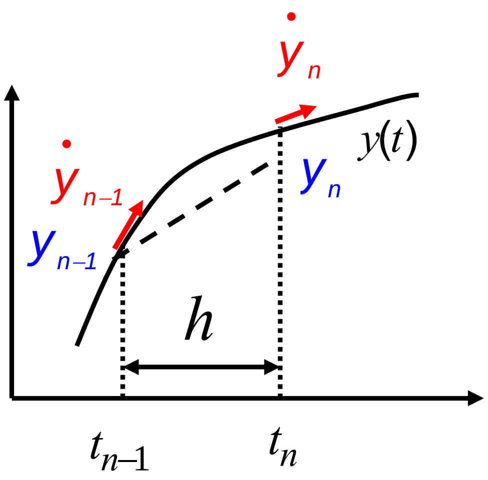
$$\boxed{y_n - y_{n-1} - h\,\dot{y}_n = 0}$$

Backward Euler

Review of TR

$$\boxed{y_n - y_{n-1} - \frac{h}{2}\left(\dot{y}_n + \dot{y}_{n-1}\right) = 0}$$

Trapezoidal Rule



Common Feature

FE $y_n - y_{n-1} - h\,\dot{y}_{n-1} = 0$

BE $y_n - y_{n-1} - h\,\dot{y}_n = 0$

TR $y_n - y_{n-1} - \frac{h}{2}\left(\dot{y}_n + \dot{y}_{n-1}\right) = 0$

$\dot{y}_n = f(y_n)$

Common Feature:

The next step value y_n is solved from a *linear* combination of y_{n-k} and y'_{n-k} at multiple backward time steps.

➔ **We can develop other integration methods based on the same principle.**

Linear Multi-Step (LMS) Method

- Generalized Integration Formula

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

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

$$\dot{y}_{n-i} \approx \frac{dy(t_{n-i})}{dt}$$

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

Coefficients α_i's and β_j's are to be determined.

non-uniform time-step

Solving the Unknowns

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

$$\overbrace{y_n + \sum_{i=1}^{\ell} \alpha_i y_{n-i}} + \overbrace{\left(h_n \beta_0 \dot{y}_n + h_n \sum_{j=1}^{m} \beta_j \dot{y}_{n-j}\right)} = 0$$

$$\dot{y}_n = f(y_n)$$

Unknown

Unknown

Depending on whether β_0 is zero, the methods are divided into "explicit" and "implicit" methods.

Explicit and Implicit LMS

- **(EXPLICIT)** $\beta_0 = 0$: **y_n is obtained directly**

$$\boxed{y_n = -\sum_{i=1}^{\ell} \alpha_i y_{n-i} - h_n \sum_{i=1}^{m} \beta_i f(y_{n-i})}$$

- **(IMPLICIT)** $\beta_0 \neq 0$: **y_n has to be solved because of $f(y_n)$**

$$\boxed{y_n + h_n \beta_0 f(y_n) = -\sum_{i=1}^{\ell} \alpha_i y_{n-i} - h_n \sum_{i=1}^{m} \beta_i f(y_{n-i})}$$

(By Newton-Raphson iteration if $f(y_n)$ is nonlinear)

Special Cases of LMS

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

Forward Euler :

$$y_n - y_{n-1} - h\,\dot{y}_{n-1} = 0$$

$$\alpha_0 = 1,\ \alpha_1 = -1; \quad \beta_0 = 0,\ \beta_1 = -1$$

(Explicit)

Backward Euler :

$$y_n - y_{n-1} - h\,\dot{y}_n = 0$$

$$\alpha_0 = 1,\ \alpha_1 = -1; \quad \beta_0 = -1,\ \beta_1 = 0$$

(Implicit)

Trapezoidal Rule :

$$y_n - y_{n-1} - \frac{h}{2}\left(\dot{y}_n + \dot{y}_{n-1}\right) = 0$$

$$\alpha_0 = 1,\ \alpha_1 = -1; \quad \beta_0 = -1/2,\ \beta_1 = -1/2$$

(Implicit)

How to Determine Coefficients ?

Basic Principle –

- **Assume uniform timestep (h): $t_i = ih$.**
- **Assume the LMS method is "exact" for a set of polynomials.**

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

$$\boxed{h_n = h}$$

Test Polynomials

An LMS is required to be **exact for a set of test polynomials**.

Derive α_i and β_i .

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

An n^{th} order polynomial

$$p_n(t) = a_n t^n + a_{n-1} t^{n-1} + \cdots + a_1 t + a_0$$

is a linear combination of basis polynomials: *$1, t, t^2, t^3, \ldots, t^n$*.

$$y(t) = t^k \quad \Rightarrow \quad y(t_{n-i}) = (t_{n-i})^q = [(n-i)h]^q$$

(not convenient)

Why use polynomials?

- **Taylor expansion – Any smooth function can be approximated by a polynomial.**

$$x(t) = \sum_{i=0}^{k} \frac{x^{(i)}(t_n)}{i\,!}(t - t_n)^i + O((t - t_n)^{k+1})$$

- **Any *p*th order polynomial can be expressed in a linear combination of some basis polynomials $b_q(t)$ for q = 0, …, p.**

- **Let the *k*th order LMS be exact for a set of basis polynomials of order $\leq k$ ➔ the coefficients.**

Easier Test Polynomials

Choose another set of polynomials:

$$y(t) = \left(\frac{t_n - t}{h}\right)^q \quad \Rightarrow \quad y(t_{n-i}) = \left(\frac{t_n - t_{n-i}}{h}\right)^q = \left(\frac{nh - (n-i)h}{h}\right)^q$$

$$= \left(\frac{ih}{h}\right)^q = i^q$$

(easier)

$$\dot{y}(t) = -\frac{k}{h}\left(\frac{t_n - t}{h}\right)^{q-1} \quad \Rightarrow \quad \dot{y}(t_{n-i}) = -\frac{k}{h}\left(\frac{t_n - t_{n-i}}{h}\right)^{q-1} = -\frac{k}{h}(i)^{q-1}$$

Test Polynomials

$$\sum_{i=0}^{q}\alpha_i y_{n-i} + h\sum_{i=0}^{m}\beta_i \dot{y}_{n-i} = 0, \quad (\alpha_0 = 1)$$

1) Choose a set of test polynomials out of convenience:

$$\boxed{y(t) = p_q(t) = \left(\frac{t_n - t}{h}\right)^q,} \qquad q = 0, \ldots, p \quad \Rightarrow \quad \begin{cases} y_{n-i} = p_q(t_{n-i}) = i^q \\[2ex] \dot{y}_{n-i} = \dot{p}_q(t_{n-i}) = -\dfrac{q}{h}\, i^{q-1} \end{cases}$$

2) Substitute into the formula and equate to 0:

$$\sum_{i=0}^{k}\alpha_i\, i^q - \sum_{i=0}^{m}\beta_i\, q \cdot i^{q-1} = 0; \qquad q = 0, \cdots, p$$

Determine the Coefficients

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

There are (k+m+1) coefficients:

$$\alpha_i \quad (i = 1, \cdots, \ell) \qquad \beta_i \quad (i = 1, \cdots, m)$$

3) Create *(k+m+1)* equations to solve the coefficients.

Coefficient Equations

$$\boxed{\sum_{i=0}^{k}(i^q)\alpha_i - \sum_{i=0}^{m}(q\cdot i^{q-1})\beta_i = 0;} \qquad q = 0,\cdots,k+m$$

$$\alpha_0 = 1$$

$$\begin{cases} \sum_{i=1}^{k}\alpha_i = -\alpha_0 = -1; & \boxed{q=0} \quad \sum_{i=0}^{k}\alpha_i = 0 \\ \sum_{i=1}^{k} i\alpha_i - \sum_{i=0}^{m}\beta_i = 0; & \boxed{q=1} \\ \sum_{i=1}^{k}(i^k)\alpha_i - \sum_{i=1}^{m}(q\, i^{q-1})\beta_i = 0; & \boxed{q=2,3,\cdots,(k+m)} \end{cases}$$

$$\Rightarrow \quad \alpha_i \ \ (i=1,\cdots,\ell); \qquad \beta_i \ \ (i=1,\cdots,m);$$

Order of Integration Method

Definition:

An LMS formula is a p^{th} order method if

1) the LMS formula holds *exactly* for all polynomials p(t) of up to degree p;
2) but *not* for some polynomial p(t) of degree p+1.

The highest order of polynomials for which the LMS formula holds exactly.

Order of Forward Euler

1. Forward Euler: $y_n = y_{n-1} + h\,\dot{y}_{n-1}$ ➡ 1st order method

$\alpha_0 = 1,\ \alpha_1 = -1;\quad \beta_0 = 0,\ \beta_1 = -1;$

$$\sum_{i=0}^{1} \alpha_i = 0 \qquad ✓ \qquad q = 0$$

$$\sum_{i=0}^{1} i\alpha_i - \beta_i = \alpha_1 - \beta_0 - \beta_1 = 0 \qquad ✓ \qquad q = 1$$

$$\sum_{i=0}^{1} i^2\alpha_i - 2i\beta_i = \alpha_1 - 2\beta_1 \neq 0 \qquad ✗ \qquad q = 2$$

Exact for linear functions.

Order of Backward Euler

2. Backward Euler: $y_n = y_{n-1} + h\dot{y}_n$ ➡ 1^{st} order method

$$\alpha_0 = 1,\ \alpha_1 = -1;\quad \beta_0 = -1,\ \beta_1 = 0;$$

$$\sum_{i=0}^{1} \alpha_i = 0 \qquad ✓ \qquad k = 0$$

$$\sum_{i=0}^{1} i\alpha_i - \beta_i = \alpha_1 - \beta_0 - \beta_1 = 0 \qquad ✓ \qquad k = 1$$

$$\sum_{i=0}^{1} i^2\alpha_i - 2i\beta_i = \alpha_1 - 2\beta_1 \neq 0 \qquad ✗ \qquad k = 2$$

Exact for linear functions.

Order of Method

3. Trapezoidal Rule:

$$y_n = y_{n-1} + \frac{h}{2}\left(\dot{y}_n + \dot{y}_{n-1}\right)$$

$\alpha_0 = 1,\ \alpha_1 = -1;\ \ \beta_0 = -1/2,\ \beta_1 = -1/2;$

$$\sum_{i=0}^{1} \alpha_i = 0 \quad ✓ \quad q = 0$$

$$\sum_{i=0}^{1} i\alpha_i - \beta_i = \alpha_1 - \beta_0 - \beta_1 = 0 \quad ✓ \quad q = 1$$

$$\sum_{i=0}^{1} i^2\alpha_i - 2i\beta_i = \alpha_1 - 2\beta_1 = 0 \quad ✓ \quad q = 2$$

$$\sum_{i=0}^{1} i^3\alpha_i - 3i^2\beta_i = \alpha_1 - 3\beta_1 \neq 0 \quad ✗ \quad q = 3$$

➡ 2nd order method

Exact for quadratic functions.

Creating your own LMS

1. **Choose an order of accuracy *p*.**
2. **Choose the number of integration steps, $k \geq p/2$.**
3. **Write down ($p+1$) coefficient equations.**
4. **If $k > p/2$, choose other constraints to determine α_i and β_i .**

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

There are 2k unknowns if $\beta_0 = 0$;

2k+1 unknowns if $\beta_0 \neq 0$.

Gear's Integration Formula

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

Choose β_i's = 0 for all i > 0: $\beta_1 = \beta_2 = \ldots = \beta_k = 0;\quad \beta_0 \neq 0$

$$\Rightarrow \quad \boxed{\left(\sum_{i=0}^{k}\alpha_i y_{n-i}\right) + h_n \beta_0 \dot{y}_n = 0}$$

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

called k^th order Backward Differentiation Formula (BDF). Also known as <u>Gear's formula</u>.

Gear's Formulas

kth order Gear formula:

$$\boxed{\left(\sum_{i=0}^{k}\alpha_i y_{n-i}\right)+h_n\beta_0\dot{y}_n=0}$$

The coefficient equations:

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

$$\begin{cases}\sum_{i=0}^{k}\alpha_i=0; & q=0 \qquad \alpha_0=1\\ \sum_{i=1}^{k}i\bullet\alpha_i-\beta_0=0; & q=1\\ \sum_{i=1}^{k}(i^q)\alpha_i-\beta_0=0; & q\geq 2\end{cases}$$

1^{st} Order Gear

k = 1 $\quad \sum_{i=0}^{1} \alpha_i y_{n-i} + h\beta_0 \dot{y}_n = 0 \quad \Rightarrow \quad y_n + \alpha_1 y_{n-1} + h\beta_0 \dot{y}_n = 0$

Coef. eqn.:

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

$$\Rightarrow \quad \boxed{y_n - y_{n-1} = h\dot{y}_n}$$

Same as Backward Euler

2^{nd} Order Gear

k = 2 $$\sum_{i=0}^{2} \alpha_i y_{n-i} + h\beta_0 \dot{y}_n = 0$$

$$\Rightarrow \quad y_n + \alpha_1 y_{n-1} + \alpha_2 y_{n-2} + h\beta_0 \dot{y}_n = 0$$

Coef. eqn.:

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

$$\Rightarrow \quad y_n - \frac{4}{3} y_{n-1} + \frac{1}{3} y_{n-2} = \frac{2h}{3} \dot{y}_n$$

$$\Rightarrow \quad \frac{3y_n - 4y_{n-1} + y_{n-2}}{2h} = \dot{y}_n$$

3^{rd} Order Gear

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

Coef. eqn.:

$$\alpha_0 = 1$$

3

$$\begin{cases} \alpha_1 + \alpha_2 + \alpha_3 = -1, \\ \alpha_1 + 2\alpha_2 + 2\alpha_3 = \beta_0, \\ \alpha_1 + 4\alpha_2 + 9\alpha_3 = 0, \\ \alpha_1 + 8\alpha_2 + 27\alpha_3 = 0. \end{cases}$$

$$\boxed{\begin{array}{l} \alpha_1 = -\frac{18}{11}, \alpha_2 = \frac{9}{11}, \quad \alpha_3 = -\frac{2}{11}, \\ \beta_0 = -\frac{6}{11}, \end{array}}$$

Other LMS Methods

Adams-Bashforth ($\beta_0 = 0$; explicit)

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

$$\beta_0 = 0; \quad \alpha_i = 0, \;\; i = 2, \cdots, k;$$

order = 1 $\quad y_n = y_{n-1} + h \dot{y}_{n-1}$ **Same as F.E.**

order = 2 $\quad y_n = y_{n-1} + h\left(\frac{3}{2} \dot{y}_{n-1} - \frac{1}{2} \dot{y}_{n-2} \right)$

order = k

$$\sum_{i=0}^{k} \alpha_i = 0; \Rightarrow \begin{cases} \alpha_0 = 1 \\ \alpha_1 = -1 \end{cases}$$

Other LMS Methods

Adams-Moulton ($\beta_0 \neq 0$; Implicit)

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

$$\beta_0 \neq 0; \quad \alpha_i = 0, \quad i = 2, \cdots, k;$$

order = 1 $\quad y_n = y_{n-1} + h\dot{y}_n$ **Same as B.E.**

order = 2 $\quad y_n = y_{n-1} + h\left(\frac{1}{2}\dot{y}_n - \frac{1}{2}\dot{y}_{n-1}\right)$ **Same as T.R.**

order = 3 $\quad y_n = y_{n-1} + h(-\beta_0 \dot{y}_n - \beta_1 \dot{y}_{n-1} - \beta_2 \dot{y}_{n-2})$

order = k+1

Summary

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

$$\sum_{i=0}^{k} \alpha_i x_{n-i} + \sum_{i=0}^{m} h\beta_i \dot{x}_{n-i} = 0$$

- **Order of method**
  - **Tested by polynomials of degree p**
  - **If order of method is p, then the Local Truncation Error is $O(h^{p+1})$**