ESS101- Modeling and Simulation
Lecture 17

Paolo Falcone

Department of Signals and Systems

Chalmers University of Technology

Göteborg, Sweden

Today (Chapters 8, 9)

- Implicit methods
  - Fixed-point method
  - Error constants
  - Predictor-correct method
  - Newton-Raphson method

Stiff systems

Linear systems

$$\dot{x}(t) = Ax(t), \quad \mathrm{Re}(\lambda_i) < 0, \quad \forall i$$

may have eigenvalue with the real part very much different in absolute value. I.e., high condition number

$$\left| \frac{\max_i \mathrm{Re}(\lambda_i)}{\min_i \mathrm{Re}(\lambda_i)} \right| \qquad \textit{Stiff systems}$$

In order to avoid over accurate solutions due to the absolute stability requirements set by the fastest eigenvalue, A-stable *(implicit)* methods should be used

A-Stability

Definition (A-Stability). A method is *A-stable* if its stability region is the left half-plane

Theorem (Dahlquist's Second Barrier Theorem).

1. There's no A-stable explicit LMM
2. An A-stable LMM cannot have $p>2$

In conclusion A-stable implicit methods allows efficiently selecting the step size to achieve the desired accuracy, without unnecessary further reduction because of stability

k-step LMMs

For the IVP problem

$$\dot{x}(t) = f(t, x(t)), \quad x(0) = x_0$$

A k-step LMM looks like

$$x_{n+k} + \alpha_{k-1} x_{n+k-1} + \cdots + \alpha_0 x_n = h\left(\beta_k f_{n+k} + \beta_{k-1} f_{n+k-1} + \cdots + \beta_0 f_n\right)$$

Recall that $f_{n+k} = f\left(t_{n+k}, x_{n+k}\right)$

Hence, x_{n+k} is solution of the nonlinear equation

$$x_{n+k} = h\beta_k f\left(t_{n+k}, x_{n+k}\right) + \overbrace{h\left(\beta_{k-1} f_{n+k-1} + \cdots + \beta_0 f_n\right) - \alpha_{k-1} x_{n+k-1} - \cdots - \alpha_0 x_n}^{g_n}$$

Fixed-point method

Calculate x_{n+k} as solution of the iterative process

$$u^{[l+1]} = h\beta_k f\left(t_{n+k}, u^{[l]}\right) + g_n, \quad l = 0,1,2,\ldots$$

1. How do we choose $u^{[0]}$?
   1. $u^{[0]}=x_{k+n-1}$ makes sense

2. Does the sequence $u^{[1]}$, $u^{[2]}$, …converge to x_{n+k}? Yes, but..

Let $u^{[l]} = x_{n+k} + E^{[l]}$ Does $E^{[l]}$ converge to 0?

$$f\left(t_{n+k}, u^{[l]}\right) = f\left(t_{n+k}, x_{n+k} + E^{[l]}\right)$$

$$\approx f\left(t_{n+k}, x_{n+k}\right) + \frac{\partial f}{\partial x}\left(t_{n+k}, x_{n+k}\right) E^{[l]}$$

Fixed-point method

Substitute $f\left(t_{n+k}, u^{[l]}\right) \approx f\left(t_{n+k}, x_{n+k}\right) + \dfrac{\partial f}{\partial x}\left(t_{n+k}, x_{n+k}\right) E^{[l]}$

in $\quad u^{[l+1]} = h\beta_k f\left(t_{n+k}, u^{[l]}\right) + g_n, \quad l = 0, 1, 2, \ldots$

From $\quad u^{[l+1]} = h\beta_k f\left(t_{n+k}, x_{n+k}\right) + h\beta_k \underbrace{\dfrac{\partial f}{\partial x}\left(t_{n+k}, x_{n+k}\right)}_{B} E^{[l]} + g_n$

Subtract $\; x_{n+k} = h\beta_k f\left(t_{n+k}, x_{n+k}\right) + g_n$

$$\underbrace{E^{[l+1]}}_{u^{[l+1]} - x_{n+k}} = h\beta_k B E^{[l]}$$

The error $E^{[l]}$ converge if the eigenvalues of B are within the unitary circle. ***(Again!!). Not for stiff problems***

Error constants

Consider the *consistency equation* written for 2-step LMMs

$$x(t+2h)+\alpha_1 x(t+h)+\alpha_0 x(t)=h\left[\beta_2\dot{x}(t+2h)+\beta_1\dot{x}(t+h)+\beta_0\dot{x}(t)\right]$$

By expanding in Taylor series

$$x(t+2h)+\alpha_1 x(t+h)+\alpha_0 x(t)-h\left[\beta_2\dot{x}(t+2h)+\beta_1\dot{x}(t+h)+\beta_0\dot{x}(t)\right]$$
$$=C_0 x(t)+C_1 h\dot{x}(t)+\cdots+C_p h^p x^{(p)}(t)+O(h^{p+1})$$

we conclude that $C_0=C_1=\cdots=C_p=0$ for a p-th order method

$$x(t+2h)+\alpha_1 x(t+h)+\alpha_0 x(t)-h\left[\beta_2\dot{x}(t+2h)+\beta_1\dot{x}(t+h)+\beta_0\dot{x}(t)\right]$$
$$=C_{p+1}h^{p+1}x^{(p+1)}(t)+O(h^{p+2})$$

$C_{p+1}\neq 0$ ***Error constant***

They will be used to estimate the local errors

Example. Calculation of the error constants

Consider the 1-step LMM

$$x_{n+1} + \alpha_0 x_n = h\left(\beta_1 f_{n+1} + \beta_0 f_n\right)$$

Calculate the coefficients α_0, β_0, β_1 such that the method has order 1. Calculate the error constant.

The corresponding consistency equation is

$$x(t+h) + \alpha_0 x(t) = h\left[\beta_1 \dot{x}(t+h) + \beta_0 \dot{x}(t)\right]$$

Since order must be 1, by expanding $x(t+h),\ \dot{x}(t+h)$ up to the first order term and collecting terms

$$x(t+h) + \alpha_0 x(t) - h\left[\beta_1 \dot{x}(t+h) + \beta_0 \dot{x}(t)\right]$$

$$= \left(1+\alpha_0\right)x(t) + \left[1 - \left(\beta_1 + \beta_0\right)\right]h\dot{x}(t) + O(h^2)$$

Example. Calculation of the error constants

In order to have order 1

$$1+\alpha_0 = 0 \Rightarrow \alpha_0 = -1$$

$$1-\beta_1-\beta_0 = 0 \Rightarrow \beta_1 = \theta, \quad \beta_0 = 1-\theta$$

We have to calculate the error constant C_2

$$x(t+h)+\alpha_0 x(t) - h\left[\beta_1 \dot{x}(t+h)+\beta_0 \dot{x}(t)\right]$$
$$= C_2 h^2 \ddot{x}(t) + O(h^3)$$

It's enough expanding $x(t+h)$, $\dot{x}(t+h)$ up to the second order term and collecting terms

$$x(t+h)+\alpha_0 x(t) - h\left[\beta_1 \dot{x}(t+h)+\beta_0 \dot{x}(t)\right] = \underbrace{\left(\frac{1}{2}-\theta\right)}_{C_2} h^2 \ddot{x}(t) + O(h^3)$$

Predictor-corrector methods

Idea. *Evaluate* f_{n+k} at x_{n+k} *predicted* with an explicit method

Consider an explicit and implicit LMM methods of the ***same order***

$$x_{n+k} + \alpha^e_{k-1} x_{n+k-1} + \cdots + \alpha^e_0 x_n = h\left(\beta^e_{k-1} f_{n+k-1} + \cdots + \beta^e_0 f_n\right)$$

$$x_{n+k} + \alpha^i_{k-1} x_{n+k-1} + \cdots + \alpha^i_0 x_n = h\left(\beta^i_k f_{n+k} + \beta^i_{k-1} f_{n+k-1} + \cdots + \beta^i_0 f_n\right)$$

E.g, Forward+Backward Euler for 1st order, AB(2)+trapezoidal for 2nd order, Adam-Bashforth+Adam-Moulton

Predictor-corrector methods

(P) Use the explicit LMM to calculate $x_{n+k}^{[0]}$

$$x_{n+k}^{[0]} = h\left(\beta_{k-1}^{e} f_{n+k-1} + \cdots + \beta_{0}^{e} f_{n}\right) - \alpha_{k-1}^{e} x_{n+k-1} - \cdots - \alpha_{0}^{e} x_{n}$$

(E) Evaluate f_{n+k} by using the system state update function

$$f_{n+k}^{[0]} = f\left(t_{n+k}, x_{n+k}^{[0]}\right)$$

(C) Calculate x_{n+k} as

$$x_{n+k} = h\left(\beta_{k}^{i} f_{n+k}^{[0]} + \beta_{k-1}^{i} f_{n+k-1} + \cdots + \beta_{0}^{i} f_{n}\right) - \alpha_{k-1}^{i} x_{n+k-1} - \cdots - \alpha_{0}^{i} x_{n}$$

Predictor-corrector methods

Error constants can be used to increase the accuracy

It can be shown that

$$\underbrace{e_n}_{\text{local error}} \approx \frac{C^i_{p+1}}{C^e_{p+1} - C^i_{p+1}} \left(x_{n+k} - x^{[0]}_{n+k} \right)$$

Hence, x_{t+k} at step 3 can be updated as

$$\hat{x}_{n+k} \approx x_{n+k} + \frac{C^i_{p+1}}{C^e_{p+1} - C^i_{p+1}} \left(x_{n+k} - x^{[0]}_{n+k} \right)$$

with accuracy of order $O(h^{p+1})$

Example

Simulate the system

$$\dot{x}(t) = (1-2t)x(t), \quad x(0) = 1$$

over the time interval $0 \le t \le 0.9$

Recall the ***exact solution*** $x(t) = e^{\frac{1}{4}-\left(\frac{1}{2}-t\right)^2}$

Let's use the forward and backward Euler methods (order 1) with *h=0.1*

(P) (FE Method) $x_1^{[0]} = x_0 + 0.1 f_0 = 0.232$

(E) (sys dyn) $f_1^{[0]} = f\left(t_1, x_1^{[0]}\right) = 0.35635$

(C) (BE Method) $x_1 = x_0 + 0.1 f_1^{[0]} = 0.23564$

(E) (sys dyn) $f_1^{[0]} = f\left(t_1, x_1^{[0]}\right) = 0.36023$

Example

The error constants are

$$C_2^i = -1/2, \quad C_2^e = 1/2$$

Hence, the accuracy of x_1 can be improved as follows

$$\hat{x}_1 \approx x_1 + \frac{C_2^i}{C_2^e - C_2^i}\left(x_1 - x_1^{[0]}\right)$$

$$= 0.23372$$

while the exact value is

$$x(t_1) = 0.23392$$

Newton-Raphson method

Consider the equation $F(u) = 0$ with

$$F(u) = u - h\beta_k f\left(t_{n+k}, u\right) - g_n$$

with solution $u = x_{t+k}$

Define $E^{[l]} = u^{[l]} - x_{t+k}$ and expand in Taylor series $F(x_{t+k})$

$$\begin{aligned} F(x_{n+k}) &= 0 \\ &= F(u^{[l]} - E^{[l]}) \\ &\approx F(u^{[l]}) - \frac{\partial F}{\partial x}(u^{[l]})E^{[l]} \end{aligned}$$

Assume a non-singular Jacobian and calculate

$$\hat{E}^{[l]} = \left[\frac{\partial F}{\partial x}(u^{[l]})\right]^{-1} F(u^{[l]})$$

Newton-Raphson method

Update $u^{[l]}$ as

$$u^{[l+1]} = u^{[l]} - \hat{E}^{[l]}$$

It can be show that the convergence is quadratic, that is the number of corrected digits double at every iteration

Example

Solve the IVP

$$\dot{x}(t) = -2y(t)^3$$

$$\dot{y}(t) = 2x(t) - y(t)^3$$

$$x(0) = 1, \quad y(0) = 1$$

with the backward Euler method and *h=0.1*

Define $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad F(u) = \begin{bmatrix} u_1 - x_n + 2hu_2^3 \\ u_2 - y_n + h\left(2u_1 - u_2^3\right) \end{bmatrix} = 0$

The Jacobian is $\dfrac{\partial F}{\partial x}(u) = \begin{bmatrix} 1 & 6hu_2^2 \\ -2h & 1 + 3hu_2^2 \end{bmatrix}$

Example

A generic iteration looks like

$$u^{[l+1]} = u^{[l]} - \overbrace{\begin{bmatrix} 1 & 6h\left(u_2^{[l]}\right)^2 \\ -2h & 1+3h\left(u_2^{[l]}\right)^2 \end{bmatrix}}^{\hat{E}^{[l]}}{}^{-1} \times$$

$$\times \underbrace{\begin{bmatrix} u_1^{[l]} - x_n + 2h\left(u_2^{[l]}\right)^3 \\ u_2^{[l]} - y_n + h\left(2u_1^{[l]} - \left(u_2^{[l]}\right)^3\right) \end{bmatrix}}_{F(u^{[l]})}$$

To calculate $x(t_1), y(t_1)$ we can set $u^{[0]} = \begin{bmatrix} x(0) \\ y(0) \end{bmatrix}$

Example

ℓ	0	1	2	3
$\boldsymbol{u}^{[\ell]}$	1.00	$0.774\,647\,887$	$0.773\,901\,924$	$0.773\,901\,807$
	1.00	$1.042\,253\,521$	$1.041\,731\,347$	$1.041\,731\,265$
$\widehat{\boldsymbol{E}}^{[\ell]}$	2.2535×10^{-1}	7.4596×10^{-4}	1.1711×10^{-7}	2.9131×10^{-15}
	-4.2254×10^{-2}	5.2217×10^{-4}	8.1975×10^{-8}	1.9628×10^{-15}
$\boldsymbol{E}^{[\ell]}$	2.2610×10^{-1}	7.4608×10^{-4}	1.1711×10^{-7}	2.8866×10^{-15}
	-4.1731×10^{-2}	5.2226×10^{-4}	8.1975×10^{-8}	1.9984×10^{-15}