AM213B Numerical Methods for the Solution of Differential Equations

Lecture 07

List of topics in this lecture

- L-stable Runge-Kutta methods, Diagonally Implicit RK methods, 2S-DIRK
- L-stable LMM, Backward Difference Formula methods (BDF)
- Construction of BDF, satisfying the second requirement of L-stability
- Accuracy of a general LMM, condition for the p-th order accuracy
- Two-point boundary value problem (BVP), shooting method

Review of A-stability, L-stability

We apply the numerical method to solving model ODE $u' = \gamma u$.

Runge-Kutta methods:

$$u_{n+1} = \phi(z) u_n, \qquad z = h\gamma \qquad \text{for ODE } u' = \gamma u$$

$\phi(z)$ is called the stability function

Region of absolute stability:

$$S = \left\{ z \in \mathbb{C} \,\middle|\, |\phi(z)| < 1 \right\}$$

A-stability:

$$|\phi(z)| < 1 \quad \text{for all } \operatorname{Re}(z) < 0$$

L-stability (two requirements)

i) is A-stable; and

ii) $\phi(z) \to 0$ as $z \to \infty$

Necessary condition for A-stability:

An A-stable RK method must be implicit.

LMM (linear multi-step methods):

$$\sum_{j=0}^{r} \alpha_j u_{n+j} = h\gamma \sum_{j=0}^{r} \beta_j u_{n+j}, \qquad \alpha_r = 1 \qquad \text{for ODE } u' = \gamma u$$

Characteristic polynomials:

$$\rho(\xi) \equiv \sum_{j=0}^{r} \alpha_j \xi^j , \qquad \sigma(\xi) \equiv \sum_{j=0}^{r} \beta_j \xi^j$$

If $\{u_k = \xi^k, k=0, 1, 2, \ldots\}$ is a solution, then ξ satisfies $\rho(\xi) - z\sigma(\xi) = 0$.

Stability polynomial:

$\pi(\xi, z) \equiv \rho(\xi) - z\sigma(\xi)$ $\quad$ ξ: independent variable; $\quad$ z: parameter

Region of absolute stability:

$$S = \left\{ z \in \mathbb{C} \,\middle|\, \text{All roots of } \pi(\xi, z) \text{ satisfy } \left|\xi_j(z)\right| < 1 \right\}$$

A-stability:

All roots of $\pi(\xi, z)$ satisfy $\left|\xi_j(z)\right| < 1$ for all $\mathrm{Re}(z) < 0$

L-stability (two requirements)

i) is A-stable; and

ii) all roots of $\pi(\xi, z)$ satisfy $\xi_j(z) \to 0$ as $z \to \infty$

Necessary condition for A-stability:

An A-stable LMM must be implicit.

End of review

So far, the only L-stable Runge-Kutta method we know is the backward Euler method, which has only the first-order accuracy.

We now introduce two L-stable Runge-Kutt methods that have higher orders.

L-stable Runge-Kutta methods

Diagonally **I**mplicit **R**unge-**K**utta methods (abbreviated as DIRK methods):

- Recall that "explicit RK" means:

  $a_{ij} = 0$ for $j \geq i$

  That is, k_i depends only on $k_1, \ldots, k_{i-1}$, but not on $k_i, \ldots, k_p$.

  $\{k_i\}$ is calculated sequentially, from k_1 to k_p. Each k_i is calculated explicitly without solving any equation.

- "Diagonally implicit RK" means:

  $a_{ij} = 0$ for $j > i$

  That is, k_i depends on $k_1, \ldots, k_{i-1}$ and k_i, but not on $k_{i+1}, \ldots, k_p$.

$\{k_i\}$ is calculated sequentially, from k_1 to k_p. The calculation of each k_i does require solving an equation. But the equation involves only k_i as the unknown. In particular, we don't need to solve a joint system involving $k_1, k_2, \ldots, k_p$.

- "Fully implicit RK" means

  a_{ij} may be non-zero for any i and j

  $\{k_1, k_2, \ldots, k_p\}$ needs to be solved simultaneously from a joint system.

<u>A two-stage diagonally implicit Runge-Kutta (2S-DIRK):</u>

The method is specified by the Butcher Tableau.

$$\begin{array}{c|c} c^T & A \\ \hline & b \end{array} = \begin{array}{c|cc} \alpha & \alpha & 0 \\ 1 & 1-\alpha & \alpha \\ \hline & 1-\alpha & \alpha \end{array}$$

where $0 < \alpha < 1$. It is a two-stage method. When $\alpha = 1 - \dfrac{1}{\sqrt{2}}$, it is second-order, A-stable, and L-stable (proof is in your homework).

<u>A three-stage diagonally implicit Runge-Kutta (3S-DIRK):</u>

The method is specified by the Butcher Tableau.

$$\begin{array}{c|c} c^T & A \\ \hline & b \end{array} = \begin{array}{c|ccc} \alpha & \alpha & 0 & 0 \\ \frac{1}{2}(1+\alpha) & \frac{1}{2}(1-\alpha) & \alpha & 0 \\ 1 & \frac{1}{4}(-6\alpha^2+16\alpha-1) & \frac{1}{4}(6\alpha^2-20\alpha+5) & \alpha \\ \hline & \frac{1}{4}(-6\alpha^2+16\alpha-1) & \frac{1}{4}(6\alpha^2-20\alpha+5) & \alpha \end{array}$$

where $0 < \alpha < 1$. It is a three-stage method. When α is the root of $\alpha^3 - 3\alpha^2 + \dfrac{3}{2}\alpha - \dfrac{1}{6} = 0$ near $\alpha = 0.435867$, it is third-order, A-stable, and L-stable (proof not presented).

In the above, we got 2nd-order accuracy for using 2 stages and 3rd-order for 3 stages. If we want a 4th-order L-stable DIRK method, we need to use at least 5 stages.

Detailed necessary condition for L-stability of LMM

$$\sum_{j=0}^{r} \alpha_j u_{n+j} = h \sum_{j=0}^{r} \beta_j f(u_{n+j}, t_{n+j}), \qquad \alpha_r = 1$$

First of all, an L-stable method must be implicit. ==> $\beta_r \neq 0$.

We formulate a more detailed necessary condition for L-stability.

The stability polynomial is

$$\pi(\xi,z) \equiv \rho(\xi) - z\sigma(\xi) = (1-z\beta_r)\xi^r + \sum_{j=0}^{r-1}(\alpha_j - z\beta_j)\xi^j , \qquad \beta_r \neq 0$$

We divide the stability polynomial by $(1-z\beta_r)$ and write it as

$$\frac{\pi(\xi,z)}{1-z\beta_r} = \xi^r + \frac{\alpha_{r-1}-z\beta_{r-1}}{1-z\beta_r}\xi^{r-1} + \frac{\alpha_{r-2}-z\beta_{r-2}}{1-z\beta_r}\xi^{r-2} + \cdots + \frac{\alpha_0 - z\beta_0}{1-z\beta_r} \tag{E01}$$

On the other hand, the normalized polynomial is expressed in terms of its roots

$$\begin{aligned}\frac{\pi(\xi,z)}{1-z\beta_r} &= \prod_{k=1}^{r}\left(\xi - \xi_k(z)\right) \\ &= \xi^r - \left(\sum_k \xi_k(z)\right)\xi^{r-1} + \left(\sum_{k_1,k_2}\xi_{k_1}(z)\xi_{k_2}(z)\right)\xi^{r-2} + \cdots + \left(-1\right)^r \prod_{k=1}^{r}\xi_k(z)\end{aligned} \tag{E02}$$

Comparing the corresponding coefficients in (E01) and (E02), we obtain

$$\begin{aligned}\frac{\alpha_{r-1}-z\beta_{r-1}}{1-z\beta_r} &= -\left(\sum_k \xi_k(z)\right) \\ \frac{\alpha_{r-2}-z\beta_{r-2}}{1-z\beta_r} &= \left(\sum_{k_1,k_2}\xi_{k_1}(z)\xi_{k_2}(z)\right)\end{aligned} \tag{E03}$$

…

Since the LMM is L-stable, all roots must satisfy

$$\xi_k(z) \to 0 \quad \text{as } z \to \infty$$

It follows that the LHS of (E03) must converge to 0 as $z \to \infty$.

$$\frac{\alpha_{r-j}-z\beta_{r-j}}{1-z\beta_r} \to 0 \quad \text{as } z\to\infty \quad \text{for } j \geq 1$$

==> $\beta_{r-j} = 0$ for $j \geq 1$ and $\beta_r \neq 0$

Therefore, β_r is the only non-zero coefficient in $\{ \beta_{r-j} \}$.

We summarize this necessary condition for L-stability in the theorem below.

Theorem: (a detailed necessary condition for L-stability)

An L-stable LMM must have the form:

$$\sum_{j=0}^{r} \alpha_j u_{n+j} = h\beta_r f(u_{n+r}, t_{n+r}), \qquad \beta_r \neq 0$$

Backward Difference Formula methods (BDF)

When β_r is the only non-zero coefficient on the RHS of an LMM, we set $\beta_r = 1$ (and rescind the constraint $\alpha_r = 1$). The LMM becomes

$$\sum_{j=0}^{r} \alpha_j u_{n+j} = h f(u_{n+r}, t_{n+r}), \qquad \beta_r = 1$$

where coefficients $\{\alpha_j\}$ have been scaled to make $\beta_r = 1$, and in general, $\alpha_r \neq 1$.

LMMs of this form are called **B**ackward **D**ifference **F**ormula methods (BDF).

Basic idea of constructing BDFs

Notation:

Let $p(t \,|\, \{g(t): t_n, t_{n+1}, \ldots, t_{n+s}\})$ denote the polynomial interpolation of function $g(t)$ based on points $\{t_n, t_{n+1}, \ldots, t_{n+s}\}$. With this concise notation, we can write out the construction of Adams methods in a simple way.

Recall that Adams-Bashforth and Adams-Moulton methods are based on integrating polynomial interpolation of $f(u(t), t)$.

r-step Adams-Bashforth:

$$u_{n+r} = u_{n+r-1} + \int_{t_{n+r-1}}^{t_{n+r}} p\Big(t \,\Big|\, \big\{ f(u(t),t):\ t_n, t_{n+1}, \ldots, t_{n+r-1} \big\}\Big) dt$$

The interpolation is based on r points: $\{t_n, t_{n+1}, \ldots, t_{n+r-1}\}$.

r-step Adams-Moulton: interpolation using $(r+1)$ points

$$u_{n+r} = u_{n+r-1} + \int_{t_{n+r-1}}^{t_{n+r}} p\Big(t \,\Big|\, \big\{ f(u(t),t):\ t_n, t_{n+1}, \ldots, t_{n+r} \big\}\Big) dt$$

The interpolation is based on $(r+1)$ points: $\{t_n, t_{n+1}, \ldots, t_{n+r}\}$.

BDF methods are based on differentiating polynomial interpolation of $u(t)$.

Formulation of r-step BDF (simply denoted by BDF-r or BDFr)

$$h \cdot \frac{d}{dt} p\Big(t \,\Big|\, \big\{ u(t):\ t_n, t_{n+1}, \ldots, t_{n+r} \big\}\Big)\Bigg|_{t=t_{n+r}} = h f(u_{n+r}, t_{n+r})$$

Remarks:

- While Adams-Bashforth and Adams-Moulton methods are based on integrating the polynomial interpolation of $\underline{f(u(t), t)}$, BDF methods are based on differentiating the polynomial interpolation of $\underline{u(t)}$.
- The multiplier h is to make the coefficients on the LHS independent of h.

Polynomial interpolation over $\{x_1, x_2, \ldots, x_m\}$

$$p(x)=\sum_{j=1}^{m} y_j p_j(x), \qquad p_j(x)\equiv\prod_{\substack{k=1\\k\neq j}}^{m}\left(\frac{x-x_k}{x_j-x_k}\right)$$

Numerator of $p_j(x)$ = product of $(x - x_k)$ over index k, excluding $k = j$.

Denominator = (Numerator | $x=x_j$).

Examples of polynomial interpolations

Two points: $\{x_1 = -1, x_2 = 0\}$

$$p_1(x)=-x, \qquad p_2(x)=x+1$$

Three points: $\{x_2 = -2, x_1 = -1, x_2 = 0\}$

$$p_1(x)=\frac{x(x+1)}{2}, \qquad p_2(x)=-x(x+2), \qquad p_3(x)=\frac{(x+1)(x+2)}{2}$$

1-step BDF:

We map $\{t_n, t_{n+1}\}$ to $\{-1, 0\}$: $\quad x=(t-t_{n+1})/h$.

Let $p(x)$ be the polynomial interpolation of $y(x)$ on $\{-1, 0\}$. We have

$$p'(x)\Big|_{x=0}=y_1 p_1'(x)\Big|_{x=0}+y_2 p_2'(x)\Big|_{x=0}=y_1(-1)+y_2$$

We use the chain rule to differentiate the interpolation of $u(t)$ on $\{t_n, t_{n+1}\}$

$$\frac{d}{dt}p\Big(t\Big|\big\{u(t):\, t_n, t_{n+1}\big\}\Big)\Bigg|_{t=t_{n+1}}=p'(x)\Big|_{x=0}\frac{dx}{dt}=\frac{1}{h}\big(-u_n+u_{n+1}\big)$$

BDF1 (1-step BDF) is

$$h\cdot\frac{1}{h}\big(-u_n+u_{n+1}\big)=hf(u_{n+1},t_{n+1})$$

$$\boxed{\text{BDF1:}\quad u_{n+1}-u_n=hf(u_{n+1},t_{n+1})}$$

This is the backward Euler method.

It is first order, A-stable, and L-stable.

2-step BDF:

We map $\{t_n, t_{n+1}, t_{n+2}\}$ to $\{-2, -1, 0\}$: $\quad x = (t - t_{n+2})/h$.

Let $p(x)$ be the polynomial interpolation of $y(x)$ on $\{-2, -1, 0\}$. We have

$$p'(x)\Big|_{x=0} = y_1 p_1'(x)\Big|_{x=0} + y_2 p_2'(x)\Big|_{x=0} + y_3 p_3'(x)\Big|_{x=0}$$

$$= y_1 \frac{1}{2} + y_2(-2) + y_3 \frac{3}{2}$$

We use the chain rule to differentiate the interpolation of $u(t)$ on $\{t_n, t_{n+1}, t_{n+2}\}$

$$\frac{d}{dt} p\Big(t \Big| \big\{u(t):\, t_n, t_{n+1}, t_{n+2}\big\}\Big)\Bigg|_{t=t_{n+2}} = p'(x)\Big|_{x=0} \frac{dx}{dt} = \frac{1}{h}\left(\frac{1}{2}u_n - 2u_{n+1} + \frac{3}{2}u_{n+2}\right)$$

BDF2 (2-step BDF) is

$$h \cdot \frac{1}{h}\left(\frac{3}{2}u_{n+2} - 2u_{n+1} + \frac{1}{2}u_n\right) = h f(u_{n+2}, t_{n+2})$$

$$\boxed{\text{BDF2:} \quad \frac{3}{2}u_{n+2} - 2u_{n+1} + \frac{1}{2}u_n = h f(u_{n+2}, t_{n+2})}$$

Claim:

BDF2 is 2nd order, A-stable and *L*-stable.

Proof:

- The accuracy of a general LMM is discussed below.
- The region of absolute stability of BDF2 will be studied computationally in your homework. The analytical proof is beyond the scope of this course.
- The second requirement of L-stability is addressed in the theorem below.

Theorem:

All BDF methods satisfy the second requirement of L-stability.

Proof is presented in Appendix A.

Accuracy of LMM

For a general LMM (not just BDF methods), we use Taylor expansion around t_n to find the condition on coefficients for achieving the *p*-th order accuracy.

Condition for the *p*-th order accuracy:

$$\sum_{j=0}^{r} \alpha_j = 0$$

$$\sum_{j=0}^{r} \alpha_j j^k = k \sum_{j=0}^{r} \beta_j j^{(k-1)}, \qquad k = 1, 2, \ldots, p$$

Derivation is presented in Appendix B.

Checking this condition on BDF2, we find that BDF2 is second order.

Below we list BDF1 through BDF4

BDF1:

$$u_{n+1} - u_n = hf(u_{n+1}, t_{n+1})$$

1st order, L-stable.

BDF2:

$$\frac{3}{2}u_{n+2} - 2u_{n+1} + \frac{1}{2}u_n = hf(u_{n+2}, t_{n+2})$$

2nd order, L-stable.

BDF3:

$$\frac{11}{6}u_{n+3} - 3u_{n+2} + \frac{3}{2}u_{n+1} - \frac{1}{3}u_n = hf(u_{n+3}, t_{n+3}),$$

3rd order, almost A-stable (not exactly A-stable, Dahlquist second barrier); therefore, it is almost L-stable.

BDF4:

$$\frac{25}{12}u_{n+4} - 4u_{n+3} + 3u_{n+2} - \frac{4}{3}u_{n+1} + \frac{1}{4}u_n = hf(u_{n+4}, t_{n+4})$$

4th order, not A-stable (Dahlquist second barrier).

For $r > 6$, r-step BDF is not zero-stable (and thus, is useless in applications!).

Two-point BVP (boundary value problem) of ODEs

Consider a second order ODE

$$u'' = f(u, u', t)$$

To uniquely determine a solution, we need two conditions.

The two conditions may be in the form of both conditions at one end.

<u>IVP with two initial conditions:</u>

$$\begin{cases} u'' = f(u, u', t) \\ u(t_0) = u_0\,, \quad u'(t_0) = v_0 \end{cases}$$

We know how to solve this IVP numerically.

The two conditions may be in the form of one condition at each end.

<u>Two-point BVP:</u>

$$\begin{cases} u'' = f(u, u', t) \\ u(0) = \alpha\,, \quad u(T) = \beta \end{cases}$$

Now we discuss several approaches for solving the two-point BVP.

Shooting method:

<u>The strategy:</u>

i) We solve the IVP below numerically using an RK solver.

$$\begin{cases} u'' = f(u, u', t) \\ u(0) = \alpha \;\leftarrow\; \text{known / given} \\ u'(0) = v \;\leftarrow\; \text{a guess / a trial value} \end{cases}$$

ii) We calculate how well the boundary condition at the right end is satisfied.

$$G(v) \equiv \underbrace{u_N\Big|_{u'(0)=v}}_{\substack{\text{Solution at} \\ T=Nh}} - \beta$$

iii) We adjust v to make $G(v) = 0$.

<u>The task</u>: solving $G(v) = 0$

We notice some features of function $G(v)$:

- Evaluation of function $G(v)$ is computationally expensive.
- $G'(v)$ is not directly available.

<u>Numerical tools for solving $G(v) = 0$</u>

<u>Option 1:</u> we use <u>Newton's method</u> to solve $G(v) = 0$.

v_0 = initial guess

$$v_{n+1} = v_n - \frac{G(v_n)}{G'(v_n)}$$

<u>Question:</u> How to calculate $G'(v_n)$?

<u>Answer:</u> the second order numerical differentiation

$$G'(v_n) \approx \frac{G(v_n + h_v) - G(v_n - h_v)}{2h_v}$$

- It requires two additional evaluations of $G(v)$.
- This approximation is second order.
- Here h_v is <u>not the same</u> as time step h used in solving the IVP.

  h_v is the step size in numerical differentiation.
- To minimize <u>the total error</u>, we should use

  $h_v \sim V \times 10^{-5}$

  where V is a typical value of v (magnitude of v).

  We will discuss later the finite precision number representation system, machine precision, round-off error, and the total error.

If we want <u>no additional evaluation</u> of $G(v)$, we can recycle the function value from the previous iteration step and use the approximation

$$G'(v_n) \approx \frac{G(v_n) - G(v_{n-1})}{v_n - v_{n-1}}$$

The resulting iterative method is the <u>secant method</u>

<u>Option 2:</u> <u>The secant method for solving $G(v) = 0$:</u>

v_0, v_1 = two initial guesses

$$v_{n+1} = v_n - \frac{G(v_n)}{G(v_n) - G(v_{n-1})}(v_n - v_{n-1})$$

Appendix A: All BDF methods satisfy the second requirement of L-stability

Proof:

The characteristic polynomials are

$$\rho(\xi) = \alpha_r \xi^r + \alpha_{r-1}\xi^{r-1} + \cdots + \alpha_1 \xi + \alpha_0 , \qquad \sigma(\xi) = \xi^r$$

The stability polynomial is

$$\pi(\xi, z) = \rho(\xi) - z\sigma(\xi) = (\alpha_r - z)\xi^r + \alpha_{r-1}\xi^{r-1} + \cdots + \alpha_1 \xi + \alpha_0$$

Dividing by ξ^r, we have

$$\frac{\pi(\xi, z)}{\xi^r} = (\alpha_r - z) + \alpha_{r-1}\frac{1}{\xi} + \cdots + \alpha_1 \frac{1}{\xi^{(r-1)}} + \alpha_0 \frac{1}{\xi^r}$$

$$\Longrightarrow \quad \left|\frac{\pi(\xi, z)}{\xi^r}\right| \geq |z| - |\alpha_r| - \left(|\alpha_{r-1}| \frac{1}{|\xi|} + \cdots + |\alpha_1| \frac{1}{|\xi|^{(r-1)}} + |\alpha_0| \frac{1}{|\xi|^r} \right)$$

For any $\varepsilon > 0$, there exists $R > 0$ (depending on ε) such that $|z| > R$ implies

$$\left|\frac{\pi(\xi, z)}{\xi^r}\right| > 0 \qquad \text{for } |\xi| > \varepsilon$$

In other words, when $|z| > R$, all roots of $\pi(\xi, z)$ are inside the circle $|\xi| \leq \varepsilon$.

Therefore, we conclude that all roots of $\pi(\xi, z)$ satisfy

$$\lim_{z\to\infty} \xi_j(z) = 0$$

which is the second requirement of L-stability.

Appendix B: Condition for the p-th order accuracy of LMM

The general r-step LMM has the form

$$\sum_{j=0}^{r} \alpha_j u_{n+j} = h\sum_{j=0}^{r} \beta_j f(u_{n+j}, t_{n+j})$$

The local truncation error is

$$e_n(h) = \sum_{j=0}^{r} \alpha_j u(t_n + jh) - h\sum_{j=0}^{r} \beta_j f\left(u(t_n + jh), t_n + jh\right)$$

We expand every term around t_n.

$$u(t_n + jh) = u(t_n) + \sum_{k=1}^{p} \frac{u^{(k)}(t_n)}{k!} j^k h^k + O(h^{p+1})$$

$$hf\left(u(t_n + jh), t_n + jh\right) = hu'(t_n + jh) = h\left(\sum_{k=0}^{p-1} \frac{u^{(k+1)}(t_n)}{k!} j^k h^k + O(h^p)\right)$$

$$= \sum_{k=1}^{p} k \frac{u^{(k)}(t_n)}{k!} j^{(k-1)} h^k + O(h^{p+1})$$

Substituting these expansions into $e_n(h)$, leads to

$$e_n(h) = \sum_{j=0}^{r}\left[\alpha_j u(t_n + jh) - \beta_j h f\left(u(t_n + jh), t_n + jh\right)\right]$$

$$= \sum_{j=0}^{r}\left[\alpha_j \left(u(t_n) + \sum_{k=1}^{p} \frac{u^{(k)}(t_n)}{k!} j^k h^k + \right) - \beta_j \sum_{k=1}^{p} k \frac{u^{(k)}(t_n)}{k!} j^{(k-1)} h^k + O(h^{p+1})\right]$$

$$= u(t_n) \sum_{j=0}^{r} \alpha_j + \sum_{k=1}^{p} \frac{u^{(k)}(t_n)}{k!} h^k \left(\sum_{j=0}^{r} \alpha_j j^k - k \sum_{j=0}^{r} \beta_j j^{(k-1)} \right) + O(h^{p+1})$$

To achieve the p-th order, we need $e_n(h) = O(h^{p+1})$. In the expansion of $e_n(h)$, setting coefficients of h^k to zero for $k = 0, 1, \ldots, p$, we arrive at

$$\sum_{j=0}^{r} \alpha_j = 0$$

$$\sum_{j=0}^{r} \alpha_j j^k = k \sum_{j=0}^{r} \beta_j j^{(k-1)}, \qquad k = 1, 2, \ldots, p$$

This is the condition on coefficients for achieving the p-th order accuracy.