AM213B Numerical Methods for the Solution of Differential Equations

Lecture 08

List of topics in this lecture

- Two-point BVP (continued): finite difference methods, matrix form of finite difference method for linear ODE,
- Fully implicit Runge-Kutta methods, Gauss-Legendre methods (two-point 4th order, three-point 6th order)
- Richardson extrapolation, Romberg integration

Review of DIRK methods and BDF methods

DIRK (Diagonally Implicit RK) methods:

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

k_i depends only on $k_1, k_2, \ldots, k_i$, not on $k_{i+1}, \ldots, k_p$.

We solve k_i sequentially, from k_1 to k_p.

2s-DIRK: 2nd order, L-stable

3s-DIRK: 3rd order, L-stable

BDF (Backward Difference Formula) methods:

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

BDF methods are constructed by differentiating the polynomial interpolation of $u(t)$ based on time levels $\{t_n, t_{n+1}, \ldots, t_{n+r}\}$.

BDF1: 1st order, L-stable (This is just the backward Euler)

BDF2: 2nd order, L-stable

BDF3: 3rd order, almost L-stable.

End of review

We continue the discussion of solving two-point BVP $\begin{cases} u'' = f(u, u', t) \\ u(t_0) = \alpha, \quad u(T) = \beta \end{cases}$

Finite difference method (FDM)

New notation:

Since most BVPs describe equilibrium in space instead of time evolution, we use x to denote the independent variable and rewrite the BVP as.

$$\begin{cases} u'' = f(u, u', x) \\ u(a) = \alpha, \quad u(b) = \beta \end{cases}$$

Numerical grid:

# of subintervals = $(N + 1)$, $\qquad h = \dfrac{b-a}{N+1}$

$x_i = a + i\,h$, $\quad i = 0, 1, 2, \ldots, N+1$

$x_0 = a$, $\quad x_{N+1} = a + (N+1)h = b$, $\qquad$ internal points = $\{x_i,\ i = 1, 2, \ldots, N\}$

u_i = numerical approximation of $u(x_i)$

Note:

- $\{ u_i \}$ is unknown on internal points $\{x_i,\ i = 1, 2, \ldots, N\}$.
- # of unknown $\{u_1, u_2, \ldots, u_N\} = N$.
- In simulations, we specify $N1 = N + 1$, which directly controls h.

Finite difference approximation:

In ODE $u'' = f(u, u', x)$, we use finite differences to replace derivatives

$$u''\big|_{x_i} \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}$$

$$u'\big|_{x_i} \approx \frac{u_{i+1} - u_{i-1}}{2h}$$

which leads to a finite difference discretization of the two-point BVP

$$\begin{cases} \dfrac{u_{i+1} - 2u_i + u_{i-1}}{h^2} = f\left(u_i,\ \dfrac{u_{i+1} - u_{i-1}}{2h},\ x_i\right), \qquad 1 \le i \le N \\ u_0 = \alpha, \qquad u_{N+1} = \beta \end{cases}$$

Finite difference is especially useful for solving linear ODEs.

Second order linear ODE:

$$u'' + p(x)u' + q(x)u = g(x)$$

Finite difference discretization:

$$\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}+p_i\frac{u_{i+1}-u_{i-1}}{2h}+q_i u_i=g_i\,,\qquad 1\le i\le N$$

where

$p_i = p(x_i)$, $\quad q_i = q(x_i)$, $\quad g_i = g(x_i)$,

This is a <u>linear system</u> of $\{u_1, u_2, \ldots, u_N\}$:

$$\left(\frac{1}{h^2}+\frac{p_i}{2h}\right)u_{i+1}+\left(-\frac{2}{h^2}+q_i\right)u_i+\left(\frac{1}{h^2}-\frac{p_i}{2h}\right)u_{i-1}=g_i\,,\quad 1\le i\le N \tag{E01}$$

$$u_0=\alpha\,,\qquad u_{N+1}=\beta$$

We write it in the matrix-vector form

$$Au=b$$

where

u = a column vector of size N

b = a column vector of size N

$$\vec{u}=\begin{pmatrix}u_1\\u_2\\\vdots\\u_N\end{pmatrix},\qquad \vec{b}=\begin{pmatrix}b_1\\b_2\\\vdots\\b_N\end{pmatrix}$$

A = a <u>tri-diagonal</u> matrix of size $N\times N$

'tri-diagonal' means $a_{ij} = 0$ for $|j-i| > 1$

$$A=\begin{pmatrix} * & * & 0 & & 0\\ * & * & * & \ddots & \\ 0 & * & * & \ddots & 0\\ & \ddots & \ddots & \ddots & *\\ 0 & & 0 & * & * \end{pmatrix}$$

<u>Row i of $Au = b$ corresponds to equation (E01) at index i.</u>

<u>$i = 1$:</u>

$$\left(\frac{1}{h^2}+\frac{p_1}{2h}\right)u_2+\left(-\frac{2}{h^2}+q_1\right)u_1=g_1-\left(\frac{1}{h^2}-\frac{p_1}{2h}\right)\alpha$$

==> $\quad a_{11}=-\dfrac{2}{h^2}+q_1\,,\qquad a_{12}=\dfrac{1}{h^2}+\dfrac{p_1}{2h}$

$$b_1 = g_1 - \left(\frac{1}{h^2} - \frac{p_1}{2h}\right)\alpha$$

<u>$1 < i < N$:</u>

$$\left(\frac{1}{h^2} + \frac{p_i}{2h}\right)u_{i+1} + \left(-\frac{2}{h^2} + q_i\right)u_i + \left(\frac{1}{h^2} - \frac{p_i}{2h}\right)u_{i-1} = g_i$$

==> $\quad a_{i,i-1} = \dfrac{1}{h^2} - \dfrac{p_i}{2h}, \qquad a_{i,i} = -\dfrac{2}{h^2} + q_i, \qquad a_{i,i+1} = \dfrac{1}{h^2} + \dfrac{p_i}{2h}$

$$b_i = g_i$$

<u>$i = N$:</u>

$$\left(-\frac{2}{h^2} + q_N\right)u_N + \left(\frac{1}{h^2} - \frac{p_N}{2h}\right)u_{N-1} = g_N - \left(\frac{1}{h^2} + \frac{p_N}{2h}\right)\beta$$

==> $\quad a_{N,N-1} = \dfrac{1}{h^2} - \dfrac{p_N}{2h}, \qquad a_{N,N} = -\dfrac{2}{h^2} + q_N$

$$b_N = g_N - \left(\frac{1}{h^2} + \frac{p_N}{2h}\right)\beta$$

(See sample Matlab code on how to build matrix A and vector b).

Once we have matrix A and vector b in Matlab form, we solve $Au = b$.

In Matlab, $\quad u$= A\b

Next we discuss a <u>collocation method</u> for the two-point BVP.

Fully implicit Runge-Kutta methods for $u' = f(u, t)$

"Fully implicit" means

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

(k_1, k_2, …, k_p) needs to be solved simultaneously from a joint system.

Recall that for explicit Runge-Kutta methods, with p stages, the highest order of accuracy we can get is p (or less if $p > 4$).

For fully implicit RK methods, with p stages, we can achieve order $2p$.

<u>One-stage high order implicit RK method</u>

The general 1-stage implicit RK ($p = 1$) is

$$k_1 = h\, f\left(u_n + a_{11}k_1, t_n + c_1 h\right)$$
$$u_{n+1} = u_n + b_1 k_1$$

Highest order of accuracy = 2.

The resulting method is the implicit midpoint method.

Its Butcher tableau is

$$\begin{array}{c|c} c^T & A \\ \hline & b \end{array} = \begin{array}{c|c} \frac{1}{2} & \frac{1}{2} \\ \hline & 1 \end{array}$$

Two-stage high order implicit RK method (two-point Gauss-Legendre method)

The general 2-stage implicit RK (p = 2) is

$$k_1 = h\,f\left(u_n + a_{11}k_1 + a_{12}k_2,\ t_n + c_1 h\right)$$
$$k_2 = h\,f\left(u_n + a_{21}k_1 + a_{22}k_2,\ t_n + c_2 h\right)$$
$$u_{n+1} = u_n + b_1 k_1 + b_2 k_2$$

Highest order of accuracy = 4.

The resulting method is the two-point Gauss-Legendre method.

Its Butcher tableau is

$$\begin{array}{c|c} c^T & A \\ \hline & b \end{array} = \begin{array}{c|cc} \frac{1}{2}-\frac{1}{6}\sqrt{3} & \frac{1}{4} & \frac{1}{4}-\frac{1}{6}\sqrt{3} \\ \frac{1}{2}+\frac{1}{6}\sqrt{3} & \frac{1}{4}+\frac{1}{6}\sqrt{3} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

Remarks:

- The two-point Gauss-Legendre method is A-stable (see below).
- The Gauss-Legendre quadrature for approximating integral $\int_0^h f(x)dx$ is a special case of the Gauss-Legendre method.

Gauss-Legendre quadrature:

$$\underbrace{h\cdot\left(b_1 f(c_1 h) + b_2 f(c_2 h)\right)}_{\text{Numerical approximation}} = \underbrace{\int_0^h f(x)dx}_{\text{Exact}} + \underbrace{h\cdot O(h^4)}_{\text{Error}}$$

where $c_1 = \frac{1}{2}-\frac{1}{6}\sqrt{3}$, $c_2 = \frac{1}{2}+\frac{1}{6}\sqrt{3}$

A-stability of the two-point Gauss-Legendre method

Applying the method to model ODE $u' = \gamma\, u$, we have

$$\begin{cases} k_1 = z\left(u_n + a_{11}k_1 + a_{12}k_2\right), & z = h\gamma \\ k_2 = z\left(u_n + a_{21}k_1 + a_{22}k_2\right) \end{cases}$$

Using the values of { a_{ij} } from the Butcher tableau, we get

$$\begin{cases} \left(1-\tfrac{1}{4}z\right)k_1 - \left(\tfrac{1}{4}-\tfrac{1}{6}\sqrt{3}\right)z k_2 = z u_n \\ -\left(\tfrac{1}{4}+\tfrac{1}{6}\sqrt{3}\right)z k_1 + \left(1-\tfrac{1}{4}z\right)k_2 = z u_n \end{cases}$$

$$\Longrightarrow \quad \begin{cases} k_1 = \dfrac{\left(1-\tfrac{1}{6}\sqrt{3}\,z\right)z}{1-\tfrac{1}{2}z+\tfrac{1}{12}z^2}u_n \\ k_2 = \dfrac{\left(1+\tfrac{1}{6}\sqrt{3}\,z\right)z}{1-\tfrac{1}{2}z+\tfrac{1}{12}z^2}u_n \end{cases}$$

Substituting into $u_{n+1} = u_n + \tfrac{1}{2}k_1 + \tfrac{1}{2}k_2$, we obtain the stability function $\phi(z)$

$$u_{n+1} = \phi(z)u_n, \qquad \phi(z) = \frac{1+\dfrac{1}{2}z+\dfrac{1}{12}z^2}{1-\dfrac{1}{2}z+\dfrac{1}{12}z^2}$$

For Re(z) < 0, we write $z = -a + ib$ ($a > 0$). The stability function $\phi(z)$ becomes

$$\phi(z) = \frac{1+\dfrac{1}{2}\left(-a+ib\right)+\dfrac{1}{12}\left(a^2-b^2-i2ab\right)}{1-\dfrac{1}{2}\left(-a+ib\right)+\dfrac{1}{12}\left(a^2-b^2-i2ab\right)} = \frac{\left(1-\dfrac{1}{2}a+\dfrac{1}{12}(a^2-b^2)\right)+i\dfrac{1}{2}b\left(1-\dfrac{1}{3}a\right)}{\left(1+\dfrac{1}{2}a+\dfrac{1}{12}(a^2-b^2)\right)-i\dfrac{1}{2}b\left(1+\dfrac{1}{3}a\right)}$$

We can verify that

$$\left|1-\frac{1}{2}a+\frac{1}{12}(a^2-b^2)\right| < \left|1+\frac{1}{2}a+\frac{1}{12}(a^2-b^2)\right|$$

$$\text{and} \quad \left|\frac{1}{2}b\left(1-\frac{1}{3}a\right)\right| < \left|\frac{1}{2}b\left(1+\frac{1}{3}a\right)\right| \qquad \text{for } a < 0$$

which leads to $|\phi(z)| < 1$ for Re(z) < 0.

Therefore, the two-point Gauss-Legendre method is A-stable.

<u>Three-stage high order implicit RK method</u> (three-point Gauss-Legendre method)

The general 3-stage implicit RK (p = 3) is

$$k_i = h f\left(u_n + \sum_{j=1}^{3} a_{ij} k_j,\ t_n + c_i h\right), \qquad i = 1, 2, 3$$

$$u_{n+1} = u_n + \sum_{i=1}^{3} b_i k_i$$

Highest order of accuracy = 6.

The resulting method is the three-point Gauss-Legendre method.

Its Butcher tableau is

$$\begin{array}{c|c} c^T & A \\ \hline & b \end{array} = \begin{array}{c|ccc} \frac{1}{2}-\frac{1}{10}\sqrt{15} & \frac{5}{36} & \frac{2}{9}-\frac{1}{15}\sqrt{15} & \frac{5}{36}-\frac{1}{30}\sqrt{15} \\ \frac{1}{2} & \frac{5}{36}+\frac{1}{24}\sqrt{15} & \frac{2}{9} & \frac{5}{36}-\frac{1}{24}\sqrt{15} \\ \frac{1}{2}-\frac{1}{10}\sqrt{15} & \frac{5}{36}+\frac{1}{30}\sqrt{15} & \frac{2}{9}+\frac{1}{15}\sqrt{15} & \frac{5}{36} \\ \hline & \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{array}$$

A collocation method for solving the two-point BVP

First, we convert the second order ODE to a first order ODE system.

To make the notation consistent with the discussion of Runge-Kutta methods, we switch back to using t as the independent variable.

<u>The general two-point BVP of a first order ODE system:</u>

$$\begin{cases} \dfrac{d\vec{w}}{dt} = \vec{F}(\vec{w}, t), \\ m_a \text{ conditions at } t = a, \\ (m - m_a) \text{ conditions at } t = b \end{cases}$$

where m = the size of the ODE system. There are m boundary conditions specified.

We use the two-point Gauss-Legendre method (2-stage fully implicit 4th order RK).

<u>Numerical grid:</u>

\# of subintervals = N, $\qquad h = \dfrac{b-a}{N}$

(We specify N in simulations.)

$t_n = a + nh,\quad n = 0, 1, 2, \ldots, N$

$t_0 = a,\quad t_N = a + Nh = b$

<u>Discretization:</u>

The discretization equations are directly from the Gauss-Legendre method.

$$\left.\begin{array}{l}\vec{k}_{n,1}=h\vec{F}\left(\vec{w}_n+a_{11}\vec{k}_{n,1}+a_{12}\vec{k}_{n,2},\ t_n+c_1h\right)\\ \vec{k}_{n,2}=h\vec{F}\left(\vec{w}_n+a_{21}\vec{k}_{n,1}+a_{22}\vec{k}_{n,2},\ t_n+c_2h\right)\\ \vec{w}_{n+1}=\vec{w}_n+b_1\vec{k}_{n,1}+b_2\vec{k}_{n,2}\end{array}\right\},\qquad n=0,\ 2,\ \ldots,\ (N-1)$$

This set of equations is applied to each time interval. There are N time intervals:

$[t_n, t_{n+1}]$, $\quad n = 0, 1, 2, \ldots, (N-1)$

<u>Number of equations:</u>

$$3\times N\times m=3Nm$$

<u>Number of unknowns:</u>

$$\Big(\underbrace{N+N}_{k_{n,1}\text{ and }k_{n,2}}+\underbrace{N+1}_{w}\Big)\times m-\underbrace{m}_{\#\text{ of BCs}}=3Nm$$

Thus, the number of unknowns matches the number of equations. We have a well-posed system. By solving this <u>non-linear</u> system, we obtain a <u>4th-order accurate solution</u>.

The size of the non-linear system is proportional to N. To avoid solving a huge non-linear system, we need to keep N at a moderate level. We need to use a high-order method to achieve a good accuracy at a moderate N.

Richardson extrapolation and Romberg integration

We i) discuss Richardson extrapolation, and then ii) apply the extrapolation repeatedly to an integral to illustrate the Romberg integration technique.

<u>Richardson extrapolation:</u>

Consider $T(h)$, a numerical approximation of quantity I, obtained using a p-th order numerical method with step size h.

$$\underbrace{T(h)}_{\substack{\text{Numerical}\\\text{approximation}}}=\underbrace{I}_{\text{Exact}}+\underbrace{E(h)}_{\text{Error}}$$

$$E(h)=C\,h^p+o(h^p)$$

<u>Definition</u> (small o notation)

If $G(h)$ satisfies $\lim\limits_{h\to 0}\dfrac{G(h)}{h^p}$, then we say $G(h) = o(h^p)$.

<u>Question:</u>

How to obtain a higher order approximation for quantity I?

<u>Strategy:</u>

Calculate both $T(h)$ and $T(h/2)$.

$$T(h) = I + Ch^p + o(h^p)$$

$$T\left(\frac{h}{2}\right) = I + \frac{1}{2^p}Ch^p + o(h^p)$$

Recall that in numerical error estimation, we get rid of unknown quantity *I* and estimate Ch^p, the leading term of error.

Here the goal is to construct a more accurate approximation of *I*. We need to get rid of Ch^p, the leading term of error.

To get rid of Ch^p, we multiply the second equation by 2^p

$$2^p T\left(\frac{h}{2}\right) = 2^p I + Ch^p + o(h^p)$$

Subtract the first equation from it, we get

$$2^p T\left(\frac{h}{2}\right) - T(h) = (2^p - 1)I + o(h^p)$$

==> $$\underbrace{\frac{1}{2^p - 1}\cdot\left[2^p T\left(\frac{h}{2}\right) - T(h)\right]}_{\text{A higher order approximation}} = \underbrace{I}_{\text{Exact}} + \underbrace{o(h^p)}_{\text{Error}}$$

This gives us a higher order approximation for quantity I.

This procedure is called <u>Richardson extrapolation</u>.

<u>Relation between Richardson extrapolation and numerical error estimation:</u>

We write quantity *I* as

$$I = T(h) - E(h)$$

We don't know the exact error $E(h)$. We use the estimated error.

$$E(h) \approx \frac{1}{1 - \left(\frac{1}{2}\right)^p}\cdot\left[T(h) - T\left(\frac{h}{2}\right)\right]$$

$$I = T(h) - E(h) \approx T(h) - \underbrace{\frac{1}{1 - \left(\frac{1}{2}\right)^p}\cdot\left[T(h) - T\left(\frac{h}{2}\right)\right]}_{\text{Estimated error}} = \underbrace{\frac{1}{2^p - 1}\cdot\left[2^p T\left(\frac{h}{2}\right) - T(h)\right]}_{\text{Richardson extrapolation}}$$

<u>Setup of Romberg integration:</u>

Consider the <u>composite trapezoidal</u> rule in the approximation-error framework

$$T(h) = I + E(h)$$

Claim:

Error $E(h)$ has the expansion

$$E(h) = C_2 h^2 + C_4 h^4 + C_6 h^6 + \cdots$$

Proof:

Consider the trapezoidal rule for $\int_{-h/2}^{h/2} f(x)dx$.

Any interval of size h can be converted to this by a shifting. We write the trapezoidal rule in the approximation-error framework

$$\frac{h}{2}\left[f\left(\frac{-h}{2}\right) + f\left(\frac{h}{2}\right) \right] = \int_{-h/2}^{h/2} f(x)dx + e(h)$$

Expand everything around $x = 0$, we obtain

$$\frac{1}{2}\left[f\left(\frac{-h}{2}\right) + f\left(\frac{h}{2}\right) \right] = f(0) + f''(0)\frac{1}{2!}\left(\frac{h}{2}\right)^2 + f^{(4)}(0)\frac{1}{4!}\left(\frac{h}{2}\right)^4 + \cdots$$

$$\int_{-h/2}^{h/2} f(x)dx = \int_{-h/2}^{h/2} \left[f(0) + f''(0)\frac{x^2}{2!} + f^{(4)}(0)\frac{x^4}{4!} + \cdots \right] dx$$

$$= h\left[f(0) + f''(0)\frac{1}{3!}\left(\frac{h}{2}\right)^2 + f^{(4)}(0)\frac{1}{5!}\left(\frac{h}{2}\right)^4 + \cdots \right]$$

==> $$e(h) = \frac{h}{2}\left[f\left(\frac{-h}{2}\right) + f\left(\frac{h}{2}\right) \right] - \int_{-h/2}^{h/2} f(x)dx = h\left[C_2 h^2 + C_4 h^4 + C_6 h^6 + \cdots \right]$$

==> $$E(h) = C_2 h^2 + C_4 h^4 + C_6 h^6 + \cdots$$

End of proof

Procedure of Romberg integration:

We start with the composite trapezoidal rule and denote it by $T^{(1)}(h)$.

$$T^{(1)}(h) = I + C_2 h^2 + C_4 h^4 + C_6 h^6 + \cdots$$

After one step of extrapolation, the result is denoted by $T^{(2)}(h)$.

$$T^{(2)}(h) = \frac{1}{2^2 - 1}\left[2^2 T^{(1)}\left(\frac{h}{2}\right) - T^{(1)}(h) \right]$$

$T^{(2)}(h)$ has the expansion

$$T^{(2)}(h) = I + \tilde{C}_4 h^4 + \tilde{C}_6 h^6 + \cdots$$

(we shall drop the tilde and recycle the notation C_4, C_6, …)

After two steps of extrapolation, the result is denoted by $T^{(3)}(h)$

$$T^{(3)}(h) = \frac{1}{2^4 - 1}\left[2^4 T^{(2)}\left(\frac{h}{2}\right) - T^{(2)}(h)\right]$$

$T^{(3)}(h)$ has the expansion

$$T^{(3)}(h) = I + C_6 h^6 + \cdots$$

In general, after k steps of extrapolation, the result is denoted by $T^{(k+1)}(h)$

$$T^{(k+1)}(h) = \frac{1}{2^{2k} - 1}\left[2^{2k} T^{(k)}\left(\frac{h}{2}\right) - T^{(k)}(h)\right]$$

$T^{(k+1)}(h)$ has the expansion

$$T^{(k+1)}(h) = I + C_{2(k+1)} h^{2(k+1)} + \cdots$$

Romberg integration is summarized in the diagram below.

$$\begin{array}{lllllllllll}
T^{(1)}(h_0) & & T^{(2)}(h_0) & & T^{(3)}(h_0) & & T^{(4)}(h_0) & & T^{(5)}(h_0) & & T^{(6)}(h_0) \\
T^{(1)}\left(\frac{h_0}{2}\right) & \nearrow & T^{(2)}\left(\frac{h_0}{2}\right) & \nearrow & T^{(3)}\left(\frac{h_0}{2}\right) & \nearrow & T^{(4)}\left(\frac{h_0}{2}\right) & \nearrow & T^{(5)}\left(\frac{h_0}{2}\right) & \nearrow & \\
T^{(1)}\left(\frac{h_0}{2^2}\right) & \nearrow & T^{(2)}\left(\frac{h_0}{2^2}\right) & \nearrow & T^{(3)}\left(\frac{h_0}{2^2}\right) & \nearrow & T^{(4)}\left(\frac{h_0}{2^2}\right) & \nearrow & & & \\
T^{(1)}\left(\frac{h_0}{2^3}\right) & \nearrow & T^{(2)}\left(\frac{h_0}{2^3}\right) & \nearrow & T^{(3)}\left(\frac{h_0}{2^3}\right) & \nearrow & & & & & \\
T^{(1)}\left(\frac{h_0}{2^4}\right) & \nearrow & T^{(2)}\left(\frac{h_0}{2^4}\right) & \nearrow & & & & & & & \\
T^{(1)}\left(\frac{h_0}{2^5}\right) & \nearrow & & & & & & & & &
\end{array}$$

We estimate the error in $T^{(k)}(h)$ as $E^{(k)}(h) = \dfrac{1}{1 - 2^{-2k}}\left(T^{(k)}(h) - T^{(k)}\left(\dfrac{h}{2}\right)\right)$.

If the estimated error is below the specified tolerance, we stop at $T^{(k+1)}(h)$.