Numerical Solutions to Differential Equations

Lecture Notes #6 — Linear Multistep Methods

Peter Blomgren
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Spring 2015

Outline

Quick Recap...

Our favorite problem is

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad t \in [t_0, T].$$

The simplest numerical method is Euler's method:

$$y_{n+1} = y_n + hf(t_n, y_n), \quad t_n = t_0 + nh.$$

Euler's method, is

1. only **first order accurate**, and
2. has a fairly small region of stability.

At this point we are mostly concerned with the first problem — we have looked for higher order accurate methods.

The Search for Higher Order Methods

Taylor Series Methods

If we can Taylor expand $f(t, y)$ further (remember $f(t, y)$ may only be available from experiments, or measurements), we can build higher order methods using more terms from the Taylor series.

Runge-Kutta Methods

When the Taylor expansion of $f(t, y)$ is not available (or expensive to compute) but $f(t, y)$ is cheap/easy to evaluate, Runge-Kutta methods are a good choice. In order to move from time level t_n to t_{n+1} we compute (sample) $f(t, y)$ in (carefully selected) multiple locations and combine the measurements to generate an accurate step.

Linear Multistep Methods — The Preview

The Idea

Use values of y and f on multiple time levels to compute y_{n+1}:

$$e.g. \quad y_{n+1} = y_n + \frac{h}{12}\left[5f_{n+1} + 8f_n - f_{n-1}\right]$$

The Connection

Strongly connected to **polynomial interpolation** ideas introduced in Math 541.

As Usual: Bad and Good News

The Bad News

1. LMMs are not self-starting
   - They depend on a potentially unavailable history.
2. Hard to change the step-size h on-the-fly
   - Complicated formulas (from Taylor expansions) when step sizes change between steps.

The Good News

- Only one (new) evaluation of f needed per step — less computational effort (faster) than RK-methods.

Linear Multistep Methods — The Language

Notation

$$f_{n+j} = f(t_{n+j}, y_{n+j}).$$

LMM in standard form:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}, \quad \alpha_k = 1, \quad |\alpha_0| + |\beta_0| \neq 0.$$

Euler's Method:

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

Linear Multistep Methods — The Language, II

LMM in standard form:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}, \quad \alpha_k = 1, \quad |\alpha_0| + |\beta_0| \neq 0.$$

Characteristic Polynomials: (same α's and β's)

$$\rho(\zeta) = \sum_{j=0}^{k} \alpha_j \zeta^j, \quad \sigma(\zeta) = \sum_{j=0}^{k} \beta_j \zeta^j$$

$\mathbb{E}$ is the Forward Shift Operator (FSO):

$$\mathbb{E} y_n = y_{n+1}$$

LMM using Characteristic Polynomials + FSO:

$$\rho(\mathbb{E}) y_n = h \sigma(\mathbb{E}) f_n$$

Linear Multistep Methods — Explicit and Implicit

LMM in standard form:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}, \quad \alpha_k = 1, \quad |\alpha_0| + |\beta_0| \neq 0.$$

- If $\beta_{\mathbf{k}} = \mathbf{0}$, the method is **explicit** — which means the sequence $\{y_n\}_{n=1}^{N}$ can be computed directly (once we are given additional starting values).
- If $\beta_{\mathbf{k}} \neq \mathbf{0}$, the method is **implicit** — at each step we have to solve the (usually) non-linear system of equations:

$$y_{n+k} = h\beta_k f(t_{n+k}, y_{n+k}) + \underbrace{\left\{\textit{previously computed values of } y_j \textit{ and } f_j\right\}}_{G_{n+k}}$$

Implicit Linear Multistep Methods (technical details)

If f is **Lipschitz continuous** wrt. y, *i.e.*

$$\|f(t,y) - f(t,y^*)\| \leq L\|y - y^*\|$$

and the step size is small enough (usually not very restrictive)

$$h < \frac{1}{|\beta_k| L},$$

then the nonlinear system of equations

$$y_{n+k} = h\beta_k f(t_{n+k}, y_{n+k}) + G_{n+k}$$

can be solved by fixed point iteration

$$y_{n+k}^{[\nu+1]} = h\beta_k f(t_{n+k}, y_{n+k}^{[\nu]}) + G_{n+k}, \quad y_{n+k}^{[0]} \text{ arbitrary}$$

Adams Methods

We are mainly going to look at a sub-class of the Linear Multistep Methods known as **Adams Methods**. They are characterized by

$$\rho(\zeta) = \zeta^k - \zeta^{k-1}.$$

- **Adams-Bashforth methods** are explicit: *e.g.* 1-step Adams-Bashforth method (Euler's Method):

$$y_{n+1} - y_n = hf_n.$$

- **Adams-Moulton methods** are implicit: *e.g.* 1-step Adams-Moulton method (Trapezoidal Rule):

$$y_{n+1} - y_n = \frac{h}{2}\left[f_{n+1} + f_n\right].$$

Other LMM Subclasses

- **Nyström Methods** are explicit methods characterized by
$$\rho(\zeta) = \zeta^k - \zeta^{k-2}.$$

- **Generalized Milne-Simpson Methods** are implicit methods characterized by
$$\rho(\zeta) = \zeta^k - \zeta^{k-2}.$$

- **Backward Differentiation Formulas (BDF)** are implicit methods with
$$\sigma(\zeta) = \beta_k \zeta^k.$$

We will revisit some of these methods later, especially when we start worrying about the size of the stability region.

Examples of LMMs (end of introduction)

- **Mid-point rule** (Nyström method)

$$y_{n+2} - y_n = 2hf_{n+1}.$$

- **Simpson's Rule** (Generalized Milne-Simpson method)

$$y_{n+2} - y_n = \frac{h}{3}\left[f_{n+2} + 4f_{n+1} + f_n\right].$$

- **3rd order Adams-Bashforth**

$$y_{n+1} - y_n = \frac{h}{12}\left[23f_n - 16f_{n-1} + 5f_{n-2}\right] + \mathcal{O}\left(h^4\right).$$

- **4th order Adams-Moulton**

$$y_{n+1} - y_n = \frac{h}{24}\left[9f_{n+1} + 19f_n - 5_{n-1} + f_{n-2}\right] + \mathcal{O}\left(h^5\right).$$

Finite Difference Approximations (Tools from Math 541)

We frequently use

$$\frac{y_{n+1} - y_n}{h} = y'(t_n) + \mathcal{O}(h).$$

We are going to need finite difference approximations for higher derivatives, and also higher-order-accurate approximations.

Finding these formulas is an exercise in Taylor expansions, and is strongly connected to polynomial interpolation...

Here we will boldly state some common finite difference formulas, and sweep all the Taylor expansions under the rug!

Finite Difference Formulas — Error $\sim \mathcal{O}(h)$

Forward Differences, truncation error $\mathcal{O}(h)$

$$
\begin{aligned}
y_n' &\approx [y_{n+1} - y_n]/h \\
y_n'' &\approx [y_{n+2} - 2y_{n+1} + y_n]/h^2 \\
y_n''' &\approx [y_{n+3} - 3y_{n+2} + 3y_{n+1} - y_n]/h^3 \\
y_n^{(4)} &\approx [y_{n+4} - 4y_{n+3} + 6y_{n+2} - 4y_{n+1} + y_n]/h^4
\end{aligned}
$$

Backward Differences, truncation error $\mathcal{O}(h)$

$$
\begin{aligned}
y_n' &\approx [y_n - y_{n-1}]/h \\
y_n'' &\approx [y_n - 2y_{n-1} + y_{n-2}]/h^2 \\
y_n''' &\approx [y_n - 3y_{n-1} + 3y_{n-2} - y_{n-3}]/h^3 \\
y_n^{(4)} &\approx [y_n - 4y_{n-1} + 6y_{n-2} - 4y_{n-3} + y_{n-4}]/h^4
\end{aligned}
$$

Finite Difference Formulas — Error $\sim \mathcal{O}\left(h^2\right)$

Central Differences, truncation error $\mathcal{O}\left(h^2\right)$

$$\begin{aligned} y_n' &\approx [y_{n+1} - y_{n-1}]/2h \\ y_n'' &\approx [y_{n+1} - 2y_n + y_{n-1}]/h^2 \\ y_n''' &\approx [y_{n+2} - 2y_{n+1} + 2y_{n-1} - y_{n-2}]/2h^3 \\ y_n^{(4)} &\approx [y_{n+2} - 4y_{n+1} + 6y_n - 4y_{n-1} + y_{n-2}]/h^4 \end{aligned}$$

Forward Differences, truncation error $\mathcal{O}\left(h^2\right)$

$$\begin{aligned} y_n' &\approx [-y_{n+2} + 4y_{n+1} - 3y_n]/2h \\ y_n'' &\approx [-y_{n+3} + 4y_{n+2} - 5y_{n+1} + 2y_n]/h^2 \\ y_n''' &\approx [-3y_{n+4} + 14y_{n+3} - 24y_{n+2} + 18y_{n+1} - 5y_n]/2h^3 \\ y_n^{(4)} &\approx [-2y_{n+5} + 11y_{n+4} - 24y_{n+3} + 26y_{n+2} - 14y_{n+1} + 3y_n]/h^4 \end{aligned}$$

Finite Difference Formulas — Errors $\sim \mathcal{O}\left(h^2\right)$ or $\mathcal{O}\left(h^4\right)$

Backward Differences, truncation error $\mathcal{O}\left(h^2\right)$

$$
\begin{aligned}
y_n' &\approx [3y_n - 4y_{n-1} + y_{n-2}]/2h \\
y_n'' &\approx [2y_n - 5y_{n-1} + 4y_{n-2} - y_{n-3}]/h^2 \\
y_n''' &\approx [5y_n - 18y_{n-1} + 24y_{n-2} - 14y_{n-3} + 3y_{n-4}]/2h^3 \\
y_n^{(4)} &\approx [3y_n - 14y_{n-1} + 26y_{n-2} - 24y_{n-3} + 11y_{n-4} - 2y_{n-5}]/h^4
\end{aligned}
$$

Central Differences, truncation error $\mathcal{O}\left(h^4\right)$

$$
\begin{aligned}
y_n' &\approx [-y_{n+2} + 8y_{n+1} - 8y_{n-1} + y_{n-2}]/12h \\
y_n'' &\approx [-y_{n+2} + 16y_{n+1} - 30y_n + 16y_{n-1} - y_{n-2}]/12h^2 \\
y_n''' &\approx [-y_{n+3} + 8y_{n+2} - 13y_{n+1} + 13y_{n-1} - 8y_{n-2} + y_{n-3}]/8h^3 \\
y_n^{(4)} &\approx [-y_{n+3} + 12y_{n+2} - 39y_{n+1} + 56y_n - 39y_{n-1} + 12y_{n-2} \\
&\quad -y_{n-3}]/6h^4
\end{aligned}
$$

Building a 3rd Order Adams-Bashforth Method, I/III

We use a Taylor expansion of y around t_n to get an expression for y_{n+1}:

$$y_{n+1} = y_n + \sum_{k=1}^{\infty} \frac{h^k}{k!} y_n^{(k)}$$

Using the equation $y'(t) = f(t, y)$, we get

$$y_{n+1} = y_n + \sum_{k=1}^{\infty} \frac{h^k}{k!} f_n^{(k-1)}$$

If we keep the first four terms:

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2} f_n' + \frac{h^3}{6} f_n'' + \mathcal{O}\left(h^4\right)$$

Building a 3rd Order Adams-Bashforth Method, II/III

We have

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2}\mathbf{f'_n} + \frac{h^3}{6}f''_n + \mathcal{O}\left(h^4\right)$$

Now, use a finite difference approximation for f'_n

$$\mathbf{f'_n} = \frac{f_n - f_{n-1}}{h} + \frac{h}{2}f''_n + \mathcal{O}\left(h^2\right)$$

and get

$$\begin{aligned} y_{n+1} &= y_n + hf_n + \frac{h^2}{2}\left[\frac{\mathbf{f_n} - \mathbf{f_{n-1}}}{\mathbf{h}} + \frac{\mathbf{h}}{\mathbf{2}}\mathbf{f''_n} + \mathcal{O}\left(\mathbf{h^2}\right)\right] + \frac{h^3}{6}f''_n + \mathcal{O}\left(h^4\right) \\ &= y_n + \frac{h}{2}\left[3f_n - f_{n-1}\right] + \frac{5h^3}{12}f''_n + \mathcal{O}\left(h^4\right) \end{aligned}$$

Building a 3rd Order Adams-Bashforth Method, III/III

We have

$$y_{n+1} = y_n + \frac{h}{2}\left[3f_n - f_{n-1}\right] + \frac{5h^3}{12}\mathbf{f''_n} + \mathcal{O}\left(h^4\right)$$

Now, we use a first order finite difference approximation for f''_n:

$$\mathbf{f''_n} = \frac{f_n - 2f_{n-1} + f_{n-2}}{h^2} + \mathcal{O}\left(h\right)$$

and we get

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}\left[3f_n - f_{n-1}\right] + \frac{5h^3}{12}\left[\frac{\mathbf{f_n - 2f_{n-1} + f_{n-2}}}{\mathbf{h^2}} + \mathcal{O}\left(\mathbf{h}\right)\right] + \mathcal{O}\left(h^4\right) \\ &= y_n + \frac{h}{2}\left[3f_n - f_{n-1}\right] + \frac{5h}{12}\left[f_n - 2f_{n-1} + f_{n-2}\right] + \mathcal{O}\left(h^4\right) \\ &= \mathbf{y_n + h\left[\frac{23}{12}f_n - \frac{4}{3}f_{n-1} + \frac{5}{12}f_{n-2}\right]} + \mathcal{O}\left(h^4\right) \end{aligned}$$

Building a 3rd Order Adams-Bashforth Method — alternative

If we instead use

$$\mathbf{f}_n' = \frac{3f_n - 4f_{n-1} + f_{n-2}}{2h} + \mathcal{O}\left(h^2\right), \quad \mathbf{f}_n'' = \frac{f_n - 2f_{n-1} + f_{n-2}}{h^2} + \mathcal{O}\left(h\right)$$

in

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2}\mathbf{f}_n' + \frac{h^3}{6}\mathbf{f}_n'' + \mathcal{O}\left(h^4\right)$$

we get

$$\begin{aligned} y_{n+1} &= y_n + hf_n + \frac{h}{4}\left[3f_n - 4f_{n-1} + f_{n+2}\right] + \frac{h}{6}\left[f_n - 2f_{n-1} + f_{n+2}\right] \\ &= y_n + h\left[\left(1 + \frac{3}{4} + \frac{1}{6}\right)f_n + \left(-1 - \frac{1}{3}\right)f_{n-1} + \left(\frac{1}{4} + \frac{1}{6}\right)f_{n-2}\right] \\ &= y_n + h\left[\frac{23}{12}f_n - \frac{4}{3}f_{n-1} + \frac{5}{12}f_{n-2}\right] \end{aligned}$$

A Note on the Order

The scheme

$$y_{n+1} = y_n + h \left[\frac{23}{12} f_n - \frac{4}{3} f_{n-1} + \frac{5}{12} f_{n-2} \right] + \mathcal{O}\left(h^4\right)$$

is **3rd Order**, since

$$\frac{y_{n+1} - y_n}{h} = \left[\frac{23}{12} f_n - \frac{4}{3} f_{n-1} + \frac{5}{12} f_{n-2} \right] + \mathcal{O}\left(h^3\right)$$

This form, where the left-hand-side converges to $y'(t)$ as $h \to 0$ is sometimes referred to as the **consistent form**.

Adams-Bashforth Methods, with Error Terms

$$y_{n+1} = y_n + hf_n + \frac{h^2}{2} f'(\xi)$$

$$y_{n+1} = y_n + \frac{h}{2}\left[3f_n - f_{n-1}\right] + \frac{5\,h^3}{12} f''(\xi)$$

$$y_{n+1} = y_n + \frac{h}{24}\left[55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}\right] + \frac{251\,h^5}{720} f^{(4)}(\xi)$$

$$\begin{aligned} y_{n+1} = y_n + \frac{h}{720}\big[&1901f_n - 2774f_{n-1} + 2616f_{n-2} - 1274f_{n-3} \\ &+ 251f_{n-4}\,\big] + \frac{475\,h^6}{1440} f^{(5)}(\xi) \end{aligned}$$

$$\begin{aligned} y_{n+1} = y_n + \frac{h}{1440}\big[&4277f_n - 7923f_{n-1} + 9982f_{n-2} - 7298f_{n-3} \\ &+ 2877f_{n-4} - 475f_{n-5}\,\big] + \frac{19087\,h^7}{60480} f^{(6)}(\xi) \end{aligned}$$

Building Adams-Moulton Methods, I/II

We use a Taylor expansion of y around t_{n+1} to get an expression for y_n (with step-length $-h$):

$$y_n = y_{n+1} + \sum_{k=1}^{\infty} \frac{(-1)^k h^k}{k!} y_{n+1}^{(k)}$$

Using the equation $y'(t) = f(t, y)$, we get

$$y_n = y_{n+1} + \sum_{k=1}^{\infty} \frac{(-1)^k h^k}{k!} f_{n+1}^{(k-1)}$$

We re-arrange

$$y_{n+1} = y_n - \sum_{k=1}^{\infty} \frac{(-1)^k h^k}{k!} f_{n+1}^{(k-1)}$$

Building Adams-Moulton Methods, II/II

$$y_{n+1} = y_n - \sum_{k=1}^{\infty} \frac{(-1)^k h^k}{k!} f_{n+1}^{(k-1)},$$

Starting from the re-arranged expression, we: —

1. Keep as many terms as needed to get an m^{th} order scheme. The first error term is

$$\frac{(-1)^{m+1} h^{m+1}}{(m+1)!} f^{(m)}(\xi) = \mathcal{O}\left(h^{m+1}\right)$$

2. Use finite-difference approximations of high enough order for the derivatives — so that the size of the error from each approximation gets absorbed into the $\mathcal{O}\left(h^{m+1}\right)$ term.

Adams-Moulton Methods, with Error Terms

$$
\begin{aligned}
y_{n+1} &= y_n + hf_{n+1} - \frac{h^2}{2} f'(\xi) \\
y_{n+1} &= y_n + \frac{h}{2}\left[f_{n+1} + f_n\right] - \frac{h^3}{12} f''(\xi) \\
y_{n+1} &= y_n + \frac{h}{12}\left[5f_{n+1} + 8f_n - f_{n-1}\right] - \frac{h^4}{24} f'''(\xi) \\
y_{n+1} &= y_n + \frac{h}{24}\left[9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}\right] - \frac{19\,h^5}{720} f^{(4)}(\xi) \\
y_{n+1} &= y_n + \frac{h}{720}\left[251f_{n+1} + 646f_n - 264f_{n-1} + 106f_{n-2}\right. \\
&\qquad \left. - 19f_{n-3}\right] - \frac{27\,h^6}{1440} f^{(5)}(\xi) \\
y_{n+1} &= y_n + \frac{h}{1440}\left[475f_{n+1} + 1427f_n - 798f_{n-1} + 482f_{n-2}\right. \\
&\qquad \left. - 173f_{n-3} + 27f_{n-4}\right] + \frac{863\,h^7}{60480} f^{(6)}(\xi)
\end{aligned}
$$

Issues, Issues, Issues...

Now that we "know" what some of the LMMs look like, we have some issues to iron out.

- LMMs are not self-starting — so how do we start?
- We can (usually) use the fixed point iteration

$$y_{n+k}^{[\nu+1]} = h\beta_k f(t_{n+k}, y_{n+k}^{[\nu]}) + G_{n+k}, \quad y_{n+k}^{[0]} \text{ arbitrary}$$

  for the implicit LMMs — How do we start, and what is the stopping criterion?
- Stability issues — We know we can build high-order accurate schemes, but [when] are they stable?

Issue #1: Starting the LMM

We have seen that a p^{th} order method requires the values f_n, f_{n-1}, $\dots$, f_{n-p+1} in order to compute y_{n+1}.

If we, for instance, want to use a 4th order method, we need the values of f_0, f_1, f_2 and f_3 in order to start the LMM.

We must use **some other method** to compute these values.

Further, the values **must be computed with the same accuracy** as the LMM. — Say we compute the starting values with Euler's Method (1st order), when what's the point of propagating those values using a 4th order method???

Commonly, a more expensive Runge-Kutta method is used as the "starter."

Issue #2: Fixed Point Iteration for Implicit LMMs

In each (implicit) step we must perform the following iteration

$$y_{n+k}^{[\nu+1]} = h\beta_k f(t_{n+k}, y_{n+k}^{[\nu]}) + G_{n+k}, \quad y_{n+k}^{[0]} \text{ arbitrary}$$

It would make sense to start with $y_{n+k}^{[0]} = y_n$, and iterate until

$$\left| \frac{y_{n+k}^{[\nu+1]} - y_{n+k}^{[\nu]}}{y_{n+k}^{[\nu]}} \right| < \epsilon, \quad \text{or} \quad \left| y_{n+k}^{[\nu+1]} - y_{n+k}^{[\nu]} \right| < \epsilon$$

for some tolerance ϵ.

Note: Each stopping criterion comes with its own set of examples where it breaks down.

Introducing Zero-Stability (Issue #3)

Consider the LMM applied to a noise-free problem:

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j}$$
$$y_\mu = \eta_\mu(h), \;\; \mu = 0, 1, \dots, k-1$$

and the same LMM applied to a slightly perturbed system

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = h \sum_{j=0}^{k} \beta_j f_{n+j} + \delta_{n+k}$$
$$y_\mu = \eta_\mu(h) + \delta_\mu, \;\; \mu = 0, 1, \dots, k-1$$

Perturbations are typically due to discretization and round-off.

Defining Zero-Stability

Definition (Zero-stability)

Let $\{\delta_n, n = 0, 1, \ldots, N\}$ and $\{\delta_n^*, n = 0, 1, \ldots, N\}$ be any two perturbations of the LMM, and let $\{y_n, n = 0, 1, \ldots, N\}$ and $\{y_n^*, n = 0, 1, \ldots, N\}$ be the resulting solutions. If there exists constants S and h_0 such that, for all $h \in (0, h_0]$,

$$\|y_n - y_n^*\| \leq S\epsilon, \quad 0 \leq n \leq N$$

whenever

$$\|\delta_n - \delta_n^*\| \leq \epsilon, \quad 0 \leq n \leq N$$

the method is said to be **zero stable**.

Interpreting Zero-Stability

No computer can calculate to infinite precision, so that inevitably round-off errors arise when [the linear combination $\sum_{j=0}^{k} \beta_j f_{n+j}$] is computed. The perturbations $\{\delta_n, n = k, k+1, \ldots, N\}$ could be interpreted as round-off errors.

Likewise, the starting values cannot be represented to infinite precision, the perturbations $\{\delta_n, n = 0, 1, \ldots, k\}$ could be interpreted as round-errors in the starting values.

If the method is not **zero-stable** *then the solutions generated by different rounding procedures (e.g. using different computers) — could result in two numerical solutions being finitely separated, not matter how fine the precision. In other words,* **if the method is not zero-stable, then the solution is essentially not computable.**

Paraphrased from J.D. Lambert (1991).

A Simple Criterion for Zero-Stability

If the roots of the characteristic polynomial

$$\sum_{j=0}^{k} \alpha_j y_{n+j} = 0, \quad \Leftrightarrow \quad \rho(\zeta) = 0$$

satisfies the **root criterion**

$$|r_j| \leq 1, \quad j = 1, 2, \ldots, k$$

then the method is **zero-stable**.

Theorem (Convergence)

The method is **convergent** *if and only if it is consistent and zero-stable.*