# Math 151B – Applied Numerical Methods II

University of California, Los Angeles

Duc Vu

Winter 2022

This is math 151B – Applied Numerical Methods taught by Professor Jeong. We meet weekly on MWF from 1:00 pm to 1:50 pm for lecture. The recommended textbook for the class is *Numerical Analysis* $10^{th}$ by *Burden*, *Faires* and *Burden*. Other course notes can be found at my blog site. Please let me know through my email if you spot any typos in the note.

## Contents

# List of Theorems

# List of Definitions

# §1 Lec 1: Jan 3, 2022

## §1.1 Review of 151A

From math 151A, we learned

1. how to solve equations numerically
2. interpolating of fitting data points
3. how to numerically integrate
4. solving a system of linear equations

In 151B, we will focus on

1. Numerically methods for solving ordinary differential equations (ODEs) with either initial conditions or boundary conditions
2. Iterative methods for solving linear systems
3. Least square approximation
4. Approximating eigenvalues

**Question 1.1.** Why are we studying numerical methods?

We are interested solving equations such as $4x + 3 = 5$, $x^2 - 5x + 2 = 0$, $e^{x^2+x} = 10 \sin x$, $\mathbf{A}\vec{x} = \vec{b}$. But only few of them can be solved exactly (the first two for example). Even for the class of polynomial equations when the degree of polynomial greater than or equal to 5, we cannot solve the equations algebraically in general but only numerically. Also for $\mathbf{A}\vec{x} = \vec{b}$, we can solve it by hand in principle, but if $\vec{x}$ is high-dimensional, we have to solve it numerically.

**Example 1.1**

Consider $\frac{dy}{dt} = t^2y - 5t^2$, $y(0) = 6$

$$\begin{aligned} \frac{dy}{dt} &= t^2(y-5) \\ \frac{1}{y-5} dy &= t^2 dt \\ \int \frac{1}{y-5} dy &= \int t^2 dt + C \\ \ln(y-5) &= \frac{1}{3}t^3 + C \\ y(t) &= e^{\frac{1}{3}t^3} + 5 \quad \leftarrow \quad y(0) = 6 \end{aligned}$$

On the other hand, consider

$$\frac{dx}{dt} = \cos(x + t^2) + 3x^2 + e^{-2t}$$

This is probably difficult or impossible to solve analytically.

# §2 Lec 2: Jan 5, 2022

## §2.1 Review (Cont'd)

**Example 2.1**

Consider

$$\begin{cases} \frac{dx_1}{dt} = 3x_1 + x_2 \\ \frac{dx_2}{dt} = -x_1 + x_2 \end{cases}$$

Let $\vec{u} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Then

$$\begin{aligned} \frac{d\vec{u}}{dt} := \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} &= \begin{bmatrix} 3x_1 + x_2 \\ -x_1 + x_2 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} 3 & 1 \\ -1 & 1 \end{bmatrix} \vec{u} \end{aligned}$$

This can be solved analytically but it has only two variables $x_1$, $x_2$. But what if we have more than thousands? Only numerical methods are feasible? We'll see the numerical methods based on discretization to find some approximation to ODEs with initial conditions.

First order DE with initial conditions

$$\begin{cases} \frac{dy}{dt} = f(t, y) \\ y(t_0) = \alpha, \quad t \in [t_0, T] \end{cases} \tag{*}$$

We will approximate the analytic solution $y$ to (*) using a finite discrete set of points as follows. We approximate $y(t)$ at the grid (mesh) points $t_0 < t_1 < t_2 < \ldots < t_N = T$ by $y(t_0), y(t_1), \ldots, y(t_N)$. Here the mesh points $t_0, t_1, \ldots, t_N$ are obtained by dividing $[t_0, T]$ into $N$ subintervals with endpoints $t_{i-1}, t_i$ for the interval $[t_{i-1}, t_i]$. Note that the number of subintervals is $N$ whereas the number of mesh points is $N + 1$ not $N$. One natural way to mesh points is to use the uniform mesh points $h = \frac{T - t_0}{N} \implies t_i = t_0 + ih$ for $i = 0, 1, \ldots, N$. We will assume uniform mesh points unless stated otherwise.

Goal: Construct $y_0, y_1, y_2, \ldots, y_N$ by numerical methods to approximate $y(t_0), y(t_1), \ldots, y(t_N)$, i.e., we want to have $y_0 \approx y(t_0), y_1 \approx y(t_1), \ldots, y_N \approx y(t_N)$

## §2.2 Euler's Method for IVP

**Definition 2.2** (Euler's Method) — Consider

$$\begin{cases} \frac{dy}{dt} = f(t, y) \\ y(t_0) = \alpha, \quad t \in [t_0, T] \end{cases} \tag{*}$$

The formula for Euler's method is

$$\begin{cases} y_{i+1} = y_i + hf(t_i, y_i) \text{ for } i = 0, 1, \ldots, N - 1 \\ y_0 = y(t_0) = \alpha \end{cases}$$

# §3 Lec 3: Jan 7, 2022

## §3.1 Euler's Method (Cont'd)

**Example 3.1**

Consider the IVP

$$\begin{cases} \frac{dy}{dt} = y - t^2 + 1, \quad t \in [0,2] \\ y(0) = 0.5 \end{cases} \tag{*}$$

First, let's find the analytic solution to (*)

i) First we compute the homogeneous equation of the ODE in (*)

$$\begin{aligned} \frac{dy}{dt} = y \iff \frac{dy}{y} &= dt \\ \implies \int \frac{dy}{y} &= \int dt + C \\ \implies \ln y &= t + C \\ \implies y &= \tilde{C}e^t \end{aligned}$$

ii) Now we go back to the original ODE in (*), $\frac{dy}{dt} = y - t^2 + 1 \implies$ By the variation of constant (parameter) the solution is of the form of

$$y(t) = \tilde{C}(t)e^t$$

Then,

$$\tilde{C}'(t)e^t + \tilde{C}(t)e^t = \tilde{C}(t)e^t - t^2 + 1$$

So

$$\begin{aligned} \tilde{C}(t) &= \int (-t^2+1)e^{-t}dt + C_1 \\ &= \int (t^2-1)(e^{-t})'dt + C_1 \\ &= (t^2-1)e^{-t} - \int 2te^{-t}\,dt + C_1 \\ &= \ldots \\ &= (t^2-1)e^{-t} + 2te^{-t} + 2e^{-t} + C_1 \\ &= (t+1)^2e^{-t} + C_1 \end{aligned}$$

Thus, $y(t) = \left[(t+1)^2e^{-t} + C_1\right]e^t$. From the initial condition, $y(0) = 0.5$, we get

$$y(t) = (t+1)^2 - 0.5e^{-t}$$

# §4 Lec 4: Jan 10, 2022

## §4.1 Euler's Method (Cont'd)

**Example 4.1** (Cont'd from Lec 3)

Now, let's apply the Euler's method for $N = 2$, i.e., $h = \frac{T-t_0}{N} = \frac{2-0}{2} = 1$. So

$$\begin{aligned} t_0 &= 0 \\ t_1 &= t_0 + 1 \cdot h = 1 \\ t_2 &= 2 \end{aligned}$$

From the initial condition, $y_0 = y(t_0) = 0.5$

$$\begin{aligned} y_1 &= y_0 + hf(t_0, y_0) \\ &= y_0 + h(y_0 - t_0^2 + 1) \\ &= 0.5 + 1 \cdot (0.5 - 0^2 + 1) \\ &= 2 \\ y_2 &= y_1 + hf(t_1, y_1) \\ &= 4 \end{aligned}$$

The values of analytic solution of mesh points $t_0, t_1, t_2$ are

$$\begin{aligned} y(t_0) &= 0.5 \\ y(t_1) &= 4 - 0.5e \\ y(t_2) &= 9 - 0.5e^2 \end{aligned}$$

So, the absolute error is

$$\begin{aligned} e_i &= |y_i - y(t_i)| \\ e_0 &= 0 \\ e_1 &= 0.641 \\ e_2 &= 1.306 \end{aligned}$$

When $N = 4$, $h = \frac{1}{2}$,

$$\begin{aligned} e_0 &= 0 \\ e_1 &= 0.176 \\ e_2 &= 0.391 \\ e_3 &= 0.634 \\ e_4 &= 0.868 \end{aligned}$$

When $N = 6$, we have $h = \frac{1}{3}$, we have

$$\begin{aligned} e_0 &= 0 \\ e_1 &= 0.080 \\ e_2 &= 0.194 \\ e_3 &= 0.283 \\ e_4 &= 0.404 \\ e_5 &= 0.531 \\ e_6 &= 654 \end{aligned}$$

From the errors in the example above, as $h$ decreases, the errors seem to decrease as well.

**Question 4.1.** What is the dependence of the error in terms of the step size $h$?

The error we saw $|y_i - y(t_i)|$ is called the **global truncation error** and it turn out that we need some theory of ODE to discuss this. There is another type of error, called **local truncation error**.

1. This is easy to compute

2. The global truncation error can be bounded in the local truncation error under a certain condition.

## §4.2 Local Truncation Error Analysis

Local truncation error (LTE) measures the accuracy of the method at the specific step by assuming that the values of the variables are exact.

$$\begin{aligned} |y_{i+1} - y(t_{i+1})| &= |y_i + hf(t_i, y_i) - y(t_{i+1})| \\ &= |y(t_i) + hf\left(t_i, y(t_i)\right) - y(t_{i+1})| \end{aligned}$$

# §5 Lec 5: Jan 12, 2022

## §5.1 Local Truncation Error Analysis(Cont'd)

We denote the LTE for Euler's method is

$$\begin{aligned} e_{i+1} &= y(t_{i+1}) - y_{i+1} \\ &= y(t_{i+1}) - (y(t_i) + hf(t_i, y(t_i))) \end{aligned} \tag{*}$$

We assume the analytic solution $y(t)$ has $(n+1)-$th continuous derivatives to analyze $e_{i+1}$. Thus, $y(t)$ has the Taylor series expansion (with remainder term) at $t = t_i$

$$\begin{aligned} y(t_{i+1}) = y(t_i) &+ \frac{dy}{dt}\Big|_{t_i}(t_{i+1}-t_i) + \frac{1}{2!}\frac{d^2y}{dt^2}\Big|_{t_i}(t_{i+1}-t_i)^2 + \ldots + \frac{1}{n!}\frac{d^ny}{dt^n}\Big|_{t_i}(t_{i+1}-t_i)^n \\ &+ \frac{1}{(n+1)!}\frac{d^{n+1}y}{dt^{n+1}}\Big|_{\xi\in[t_i,t_{i+1}]}(t_{i+1}-t_i)^{n+1} \\ = y(t_i) &+ \frac{dy}{dt}\Big|_{t_i} h + \frac{1}{2!}\frac{d^2y}{dt^2}\Big|_{t_i} h^2 + \ldots + \frac{1}{n!}\frac{d^ny}{dt^n}\Big|_{t_i} h^n + \frac{1}{(n+1)!}\frac{d^{n+1}y}{dt^{n+1}}\Big|_{\xi\in[t_i,t_{i+1}]} h^{n+1} \end{aligned}$$

We replace $y(t_{i+1})$ in (*) by the Taylor series expansion above.

$$\begin{aligned} e_{i+1} &= y(t_{i+1}) - (y(t_i) + hf(t_i, y(t_i))) \\ &= \frac{1}{2!}\frac{d^2y}{dt^2}\Big|_{t_i} h^2 + \ldots + \frac{1}{n!}\frac{d^ny}{dt^n}\Big|_{t_i} h^n + \frac{1}{(n+1)!}\frac{d^{n+1}y}{dt^{n+1}}\Big|_{\xi\in[t_i,t_{i+1}]} h^{n+1} \end{aligned} \tag{**}$$

Recall our assumption is that $y(t)$ has $(n+1)-$th continuous derivatives. Thus, $\frac{d^2y}{dt^2}, \ldots, \frac{d^{n+1}y}{dt^{n+1}}$ are all bounded on $[t_0, T]$. Using this fact to (**)

$$\begin{aligned} |e_{i+1}| &\le C_1h^2 + C_2h^3 + \ldots + C_nh^{n+1} \\ &\le Ch^2 \quad \text{(since } h \ll 1) \end{aligned}$$

We only need to assume that $y(t)$ is smooth so that $y''$ is continuous on $[t_0, T]$ where $C = \max_{t\in[t_0,T]} |y''(t)|/2$.

> **Remark 5.1.** Big-O notation: If $|e_{i+1}| \le Ch^n$ where $C$ is some constant independent with $h$, then $e_{i+1} = \mathcal{O}(h^n)$. So the LTE of Euler's method, $e_{i+1}$ satisfies $e_{i+1} = \mathcal{O}(h^2)$.

## §5.2 Global Truncation Error

Recall that the global truncation error for Euler's method is given by

$$\begin{cases} e_{i+1} = y(t_{i+1}) - y_{i+1} = y(t_{i+1}) - (y_i + f(t_i, y_i)) \\ y_0 = y(t_0) \end{cases}$$

To study the global truncation error further, we need to introduce the definite of Lipschitz condition/constant.

**Definition 5.2** (Lipschitz Condition) — A function $f(t, y)$ is said to satisfy a Lipschitz condition in $y$ on a set $D : [t_0, T] \times (-\infty, \infty)$ with the Lipschitz constant $L$ if for all $y_1, y_2 \in (-\infty, \infty)$ and $t \in [t_0, T]$ we have

$$|f(t, y_1) - f(t, y_2)| \le L|y_1 - y_2|$$

**Theorem 5.3**

Suppose that $f(t, y)$ is continuous and $\frac{\partial f}{\partial y}(t, y)$ is bounded by $L$ on $D : [t_0, T] \times (-\infty, \infty)$. Then $f(t, y)$ satisfies the Lipschitz condition with constant $L$.

# §6 Lec 6: Jan 14, 2022

## §6.1 Global Truncation Error (Cont'd)

**Theorem 6.1**

Suppose that $D = [t_0, T] \times \mathbb{R}$ and the function $f(t, y)$ satisfies

1. $f(t, y)$ is continuous on $D$
2. $f(t, y)$ satisfies the Lipschitz condition on $D$ in the variable $y$

Then the IVP in the form

$$\begin{cases} \frac{dy}{dt} = f(t, y) \\ y(t_0) = y_0 \end{cases}$$

has a unique solution $y(t)$ for $t \in [t_0, T]$.

**Theorem 6.2** (Global Truncation Error Bound for Euler's Method)

If

1. $f(t, y)$ satisfies the Lipschitz condition in $y$ on $D = [t_0, T] \times (-\infty, \infty)$ with Lipschitz constant $L$.
2. $|y''(t)| \leq M < \infty$ for all $t \in [t_0, T]$,

$$\max_{0 \leq i \leq N} |y(t_i) - y_i| \leq e^{L(T-t_0)} |e_0| + \left( \frac{e^{L(T-t_0)} - 1}{L} \cdot \frac{Mh}{2} \right)$$

# §7 Lec 7: Jan 19, 2022

## §7.1 Global Truncation Error (Cont'd)

Note that if $|e_0| = 0$, we have

$$|e_i| \leq \left(\frac{e^{NhL} - 1}{2L}\right) \cdot Mh$$

**Remark 7.1.** The global truncation error for Euler's method is $\mathcal{O}(h)$.

## §7.2 Relation Between LTE and GTE

Next, we will remark the relation between the local truncation and global truncation errors. Let $\tau_i$ be the local truncation error and $e_i$ denote the global truncation error.

**Remark 7.2.** We have

$$\begin{aligned} |\tau_i| &\leq \frac{\max|y''|}{2} \cdot h^2 \\ |e_i| &\leq \frac{e^{(T-t_0)\cdot L}}{2L} \cdot \max|y''|\, h \end{aligned}$$

Although we cannot apply the bound on $\tau_i$ to the bound of $e_i$ above, it turns out that

$$|e_i| \leq C\left(\frac{1}{h}\max|\tau_i|\right)$$

| Local truncation error | Global truncation error |
|---|---|
| $\mathcal{O}(h^2)$, $\frac{Mh^2}{2}$ | $\mathcal{O}(h)$, $\frac{e^{L(T-t_0)}-1}{2} \cdot \frac{Mh}{2}$ |

**Remark 7.3.** An important message from above table that holds for global/local truncation errors of other numerical methods: If we have a numerical method for IVPs whose local truncation error satisfies $|\tau_i| \leq Ch^{p+1}$ or $\mathcal{O}(h^{p+1})$, then this method has a global truncation error $|e_i| \leq \tilde{C} \cdot h^p$ or $\mathcal{O}(h^p)$.

**Question 7.1.** What other methods for IVP has global error of $\mathcal{O}(h^p)$ ($p$th order of accuracy)?

Consider the Taylor method of order $p$ given as follows:

$$\begin{cases} y_{i+1} = y_i + h \cdot T^{(p)}(t_i, y_i), \quad i = 0, \ldots, N-1 \\ y_0 = y(t_0) \end{cases}$$

where

$$T^{(p)}(t_i, y_i) = f(t_i, y_i) + \frac{h}{2} f'(t_i, y_i) + \ldots + \frac{h^{p-1}}{p!} f^{(p-1)}(t_i, y_i)$$

**Definition 7.4** (Order Accuracy) — A numerical method is called $p$th order accuracy method if its global truncation error bound is $|e_i| \leq Ch^p$ or $e_i = \mathcal{O}(h^p)$ where $C$ is some constant independent of $h$.

We saw the Euler's method has the global truncation error $e_i = \mathcal{O}(h)$, so it is 1st order accuracy method.

**Question 7.2.** Suppose we have a numerical method for IVP with a formula of the method, but we don't know its order accuracy. How can we determine it?

There are two cases:

1. The analytic solution $y(t)$ is known.
2. The analytic solution $y(t)$ is unknown.

Case 1: $y(t)$ is given. Note that $\frac{dy}{dt} = f(t, y)$. Let's assume that $f(t, y)$ is smooth enough. Let the formula of the method is given by $y_{i+1} = g(t_i, y_i)$ for some function $g$.

i) Compute the local truncation error $\tau_{i+1}(h) = y(t_{i+1}) - \tilde{y}_{i+1}$, where $\tilde{y}_{i+1} = g(t_i, y(t_i))$.

ii) Use the Taylor series expansion for $y(t_{i+1})$ at $y(t_i)$ along with the fact that $\frac{dy}{dt} = f(t, y)$. Then, apply the similar argument used for the local truncation error for the Euler's method.

iii) If the exponent of $h$ of the leading term of $\tau_{i+1}(h)$ is $p + 1$ (i.e., the smallest exponent of $h$ is $\tau_{i+1}(h)$),

$$|\tau_{i+1}(h)| \leq Ch^{p+1}$$

By the remark about LTE and GTE, the accuracy of the method is $p$.

# §8 Lec 8: Jan 21, 2022

## §8.1 Relation Between LTE and GTE (Cont'd)

Case 2: $y(t)$ is not given. First we assume that the method is at least $p$-th order accuracy method for $p \geq 1$ but the value of $p$ is not known.
Recall the global truncation error for the $p$-th order method

$$\max_{0 \leq i \leq N} |e_i| \leq C_0 |e_0| + C_1 h^p$$

and if $e_0 = 0$, $\max_{0 \leq i \leq N} |e_i| \leq C_1 h^p$

**Definition 8.1** (Convergent Method) — If $\lim_{h \to 0} \max_{0 \leq i \leq N} |e_i| = 0$, then the given method is convergent.

Then, let's observe how the error changes if step size $h$ is decreases by half as below.

| Step size | 1st order | 2nd order | pth order |
|---|---|---|---|
| $h$ | $\mathcal{O}(h)$ | $\mathcal{O}(h^2)$ | $\mathcal{O}(h^p)$ |
| $\frac{h}{2}$ | $\mathcal{O}\left(\frac{h}{2}\right)$ | $\mathcal{O}\left(\frac{h^2}{2^2}\right)$ | $\mathcal{O}\left(\frac{h^p}{2^p}\right)$ |

Note that if the step size $h$ is decreased by a factor of 2, then the ratio of the global truncation error bound is improved by a factor of $2^p$:

- 1st order: $\frac{\mathcal{O}(h)}{\mathcal{O}\left(\frac{h}{2}\right)} = 2$
- 2nd order: $\frac{\mathcal{O}(h^2)}{\mathcal{O}\left(\frac{h^2}{2^2}\right)} = 2^2$
- pth order: $\frac{\mathcal{O}(h^p)}{\mathcal{O}\left(\frac{h^p}{2^p}\right)} = 2^p$

This gives some idea about how to estimate the order of accuracy for a numerical method empirically.

**Theorem 8.2** (Asymptotic Error Estimation – Aitken's Estimation)

For Euler's method: assume that $y(t)$ is 3-times continuously differentiable and $\frac{\partial f}{\partial y}, \frac{\partial^2 f}{\partial y^2}$ are continuous and bounded on $D = [t_0, T] \times \mathbb{R}$. Then there exists $D(t)$ s.t. $y(t_i) - y_i = D(t_i)h + \mathcal{O}(h^2)$. Specifically, consider $t_i = T$, then

$$y(T) - \tilde{y}_h = Dh + \mathcal{O}(h^2)$$

where $\tilde{y}_h \coloneqq y_N$ is the estimate of $y(T)$ by Euler's method with step size $h$. Then

$$\frac{\tilde{y}_h - \tilde{y}_{\frac{h}{2}}}{\tilde{y}_{\frac{h}{2}} - \tilde{y}_{\frac{h}{4}}} \approx 2$$

As $h \to 0$,

$$\frac{\tilde{y}_h - \tilde{y}_{\frac{h}{2}}}{\tilde{y}_{\frac{h}{2}} - \tilde{y}_{\frac{h}{4}}} = \frac{D \cdot \left(\frac{1}{2} - 1\right)}{D\left(\frac{1}{4} - \frac{1}{2}\right)} = 2$$

Note that to identify $p$ in $2^p$, we can take

$$\frac{\log 2^p}{\log 2} = \frac{p \log 2}{\log 2} = p$$

Therefore,

$$\lim_{h\to 0} \frac{\log\left(\frac{\tilde{y}_h - \tilde{y}_{\frac{h}{2}}}{\tilde{y}_{\frac{h}{2}} - \tilde{y}_{\frac{h}{4}}}\right)}{\log 2} = 1$$

Thus, the order of accuracy of Euler's method is 1.
The Aitken's estimation can be generalized to general order $p$ accuracy methods. For a general pth order method, it has the asymptotic error bound in this form

$$y(T) - \tilde{y}_h = Dh^p + \mathcal{O}(h^{p+1})$$

where $D$ is some constant independent of $h$. Thus, we have

$$\lim_{h\to 0} \frac{\tilde{y}_h - \tilde{y}_{\frac{h}{2}}}{\tilde{y}_{\frac{h}{2}} - \tilde{y}_{\frac{h}{4}}} = 2^p$$

# §9 Lec 9: Jan 24, 2022

## §9.1 Relation between LTE and GTE (Cont'd)

Recall

$$p = \frac{\log 2^p}{\log 2} = \frac{\lim_{h\to 0} \log\left(\frac{\tilde{y}_h - \tilde{y}_{\frac{h}{2}}}{\tilde{y}_{\frac{h}{2}} - \tilde{y}_{\frac{h}{4}}}\right)}{\log 2}$$

Note that to obtain the exact value of $p$, we need to let $h \to 0$. In practice, we have estimate with some step size $h > 0$. When $h$ is not small enough, the estimate for $p$ may not be reliable. Typically, the smaller $h$ is the better estimate for $p$ is. For example, let $p_1, p_2, p_3$ are the estimates for $p$ with step sizes $h$, $\frac{h}{2}$, $\frac{h}{4}$.

$$|p - p_3| \le |p - p_2| \le |p - p_1|$$

## §9.2 An Overview

First, let's take a look at the table of classification of numerical methods.

| Explicit methods | Implicit methods |
|---|---|
| $y_{i+1} = \phi\left(y_i, y_{i-1}, \ldots, y_0\right)$<br>∘ Do not need to solve an equation to find $y_{i+1}$<br>∘ Fast<br>∘ Less stable<br>∘ For non-stiff differential equations | $y_{i+1} = \phi\left(y_{i+1}, y_i, y_{i-1}, \ldots, y_0\right)$<br>∘ Need to solve an equation to find $y_{i+1}$<br>∘ Slow<br>∘ More stable<br>∘ For stiff problems |

For example, recall the formula of Euler's method

$$y_{i+1} = y_i + hf\left(t_i, y_i\right)$$

Euler's method is an explicit method. Next, we will take a look at the classification of numerical methods for IVP

| 1-step method | Multi-step methods |
|---|---|
| $y_{i+1}$ is only based on $y_i$ | If $y_{i+1}$ is based on $y_i, y_{i-1}$ (2-step method)<br>or $y_i, y_{i-1}, y_{i-2}$ (3-step method)<br>$\ldots$<br>or $y_i, \ldots, y_{i+1-m}$ (m-step method) |

**Remark 9.1.** Euler's method is 1-step method.

| Explicit | Implicit |
|---|---|
| 1. Euler (1-step)<br>2. Runge-Kutta<br>3. Adam-Bashforth (multi-step) | 1. Backward Euler (1-step)<br>2. Trapezoidal Method (Cronk-Nicolson method)<br>3. Adams-Moulton (multi-step) |

## §9.3 Runge-Kutta Method

Recall the Taylor's method of order $p$

$$\begin{cases} y_{i+1} = y_i + h \cdot T^{(p)}(t_i, y_i), \quad i = 0, \ldots, N-1 \\ y_0 = y(t_0) \end{cases}$$

where

$$T^{(p)}(t_i, y_i) = f(t_i, y_i) + \frac{h^2}{2} f'(t_i, y_i) + \ldots + \frac{h^{p-1}}{(p-1)!} f^{(p-1)}(t_i, y_i)$$

Although the Taylor method provides higher order of accuracy, it is computationally heavy since it requires to compute high order derivatives of $f(t, y)$. Thus, Runge-Kutta method is computationally cheap and also have high order of accuracy.

# §10 Lec 10: Jan 26, 2022

## §10.1 Runge-Kutta Method (Cont'd)

2-stage Runge-Kutta method: Recall the Euler's method

$$\begin{cases} y_{k+1} = y_k + hf\left(t_k, y_k\right) \\ y_0 = y(t_0) \end{cases}$$

Here, we have used $f(t_k, y_k)$ as the estimation of the slope $\frac{dy}{dt}\big|_{t_k}$. For Runge-Kutta method,

1. We use a better estimate of slope rather than using $f(t_k, y_k)$
2. Then use it to compute $y_{k+1}$

There are several versions of 2-stage Runge-Kutta method

i) Midpoint method

$$\begin{cases} y^* = y_k + \frac{h}{2} f\left(t_k, y_k\right) \\ y_{k+1} = y_k + hf\left(t_k + \frac{h}{2}, y^*\right) \\ y_0 = y(t_0) \end{cases}$$

First, note that $y^* = y_k + \frac{h}{2} f\left(t_k, y_k\right)$ is the formula of Euler's method at $t_k$ but with step size $\frac{h}{2}$. So, for the midpoint method at $(t_k, y_k)$ with step size $h$, we use the Euler's method to find the estimate $y^*$ for $y\left(t_k + \frac{h}{2}\right) = y\left(\frac{t_k + t_{k+1}}{2}\right)$. Then, using the estimate $y^*$ to compute $f$ at $\frac{t_k + t_{k+1}}{2}$, which is used to estimate $y_{k+1}$.

ii) Modified Euler's method

$$\begin{cases} S_1 = f(t_k, y_k) \\ S_2 = f\left(t_{k+1}, y_k + hS_1\right) \\ y_{k+1} = y_k + h \cdot \frac{S_1 + S_2}{2} \\ y_0 = y(t_0) \end{cases}$$

iii) Generalized Runge-Kutta method

$$\begin{cases} S_1 = f(t_k, y_k) \\ S_2 = f\left(t_k + \alpha h, y_k + \alpha h S_1\right) \\ y_{k+1} = y_k + h\left(\beta_1 S_1 + \beta_2 S_2\right), \quad k = 0, 1, \dots, N-1 \\ y_0 = y(t_0) \end{cases}$$

with $\alpha \in [0, 1]$ and $\beta_1 + \beta_2 = 1$.

It is easy to see that the midpoint method is a special case of the generalized Runge-Kutta method with $\alpha = \frac{1}{2}, \beta_1 = 0, \beta_2 = 1$. Similarly, the modified Euler's method is a special case of the generalized Runge-Kutta method with $\alpha = 1$, $\beta_1 = \beta_2 = \frac{1}{2}$.
Now, we will consider a simpler type of IVP with autonomous ODE

$$\begin{cases} \frac{dy}{dt} = f(y) \\ y_0 = y(t_0), \quad t \in [0, T] \end{cases}$$

- The local truncation error analysis is much simpler for autonomous case but it still contains of the main idea of analysis of LTE for nonautonomous case.
- The relation between LTE and GTE still holds. If $|\tau_i| = \mathcal{O}(h^{p+1})$, then $|e_i| = \mathcal{O}(h^p)$.

We will start with LTE analysis for the midpoint method. Recall

$$\begin{cases} y^* = y_k + \frac{h}{2} f\left(t_k, y_k\right) \\ y_{k+1} = y_k + hf\left(t_k + \frac{h}{2}, y^*\right) \\ y_0 = y(t_0) \end{cases}$$

So, $y_{k+1} = y_k + hf\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} f(t_k, y_k)\right)$.

1. Since we assumed the autonomous ODE,

$$y_{i+1} = y_i + hf\left(y_i + \frac{h}{2} f(t_i, y_i)\right)$$

Then, the LTE

$$\begin{aligned} \tau_{i+1} &= y(t_{i+1}) - y_{i+1} \\ &= y(t_{i+1}) - \left[y(t_i) + hf\left(y(t_i) + \frac{h}{2} f\left(t_i, y(t_i)\right)\right)\right] \end{aligned}$$

2. Next, we apply the Taylor series expansion on $y(t_{i+1})$ at $t = t_i$.

$$y(t_{i+1}) = y(t_i) + y'(t_i)h + \frac{1}{2!} y''(t_i)h^2 + \frac{1}{3!} y^{(3)}(t_i)h^3 + \dots$$

# §11 Lec 11: Jan 28, 2022

## §11.1 Runge-Kutta Method (Cont'd)

3. We will consider the Taylor series expansion for $f\left(y(t_i)+\frac{h}{2}f(y(t_i))\right)$. Note that the Taylor series expansion for $f(a+b)$ at $a$ is given by

$$f(a+b)=f(a)+\frac{df}{dt}\Big|_a b+\frac{1}{2!}\frac{d^2f}{dt^2}\Big|_a b^2+\frac{1}{3!}\frac{d^3f}{dt^3}\Big|_a b^3+\dots$$

So

$$f\left(y(t_i)+\frac{h}{2}f(y(t_i))\right)=f\left(y(t_i)\right)+\frac{df}{dy}\Big|_{y(t_i)}\frac{h}{2}f\left(y(t_i)\right)+\frac{1}{2!}\frac{d^2f}{dy^2}\Big|_{y(t_i)}\left(\frac{h}{2}f(y(t_i))\right)^2 \\ +\frac{1}{3!}\frac{d^3f}{dy^3}\Big|_{y(t_i)}\left(\frac{h}{2}f(y(t_i))\right)^3+\dots$$

4. Using the results 1), 2), and 3)

$$\tau_{i+1}=y'(t_i)h+\frac{1}{2!}y''(t_i)h^2+\frac{1}{3!}y^{(3)}(t_i)h^3+\dots \\ -h\left[f(y(t_i))+\frac{df}{dy}\Big|_{y(t_i)}\frac{h}{2}f(y(t_i))+\frac{1}{2!}\frac{d^2f}{dy^2}\Big|_{y(t_i)}\left(\frac{h}{2}f(y(t_i))\right)^2+\dots\right]\dots$$

After some manipulation, we obtain

$$\tau_{i+1}=\left[\frac{1}{24}f_{yy}\left(y(t_i)\right)f^2\left(y(t_i)\right)+\frac{1}{6}f_y^2\left(y(t_i)\right)f\left(y(t_i)\right)\right]h^3+\mathcal{O}(h^4)$$

5. By our assumption, since $f$ is smooth enough so that $f, f_y, f_{yy}, f_{yyy}$ are continuous on $[t_0,T]$ and $[t_0,T]$ is bounded and closed. So,

$$|\tau_{i+1}|\le\tilde{C}\cdot h^3$$

i.e., $\tau_{i+1}=\mathcal{O}(h^3)$. Therefore, $e_{i+1}=\mathcal{O}(h^2)$

$$\implies \max_{0\le i\le N}|e_i|\le C_2h^2$$

and the global truncation error of the midpoint method is $\mathcal{O}(h^2)$.

**Exercise 11.1.** Analyze the modified Euler's method and the generalized 2-stage Runge-Kutta methods (which are similar to the analysis of the midpoint method above).

# §12 Lec 12: Jan 31, 2022

## §12.1 Runge-Kutta Method (Cont'd)

Nonautonomous case:

$$\begin{cases} \frac{dy}{dt} = f(t,y) \\ y_0 = y(t_0), \quad t \in [t_0, T] \end{cases}$$

From the formula of the midpoint method

$$y_{i+1} = y_i + hf\left(t_i + \frac{h}{2}, y_i + \frac{h}{2} f(t_i, y_i)\right)$$

So the LTE is

$$\begin{aligned} \tau_{i+1} &= y(t_{i+1}) - y_{i+1} \\ &= y(t_{i+1}) - \left[y(t_i) + hf\left(t_i + \frac{h}{2}, y(t_i) + \frac{h}{2} f\left(t_i, y(t_i)\right)\right)\right] \end{aligned}$$

Again, the LTE analysis for the nonautonomous case is similar to the autonomous case, except we need to apply the Taylor series expansion with 2 variables. Then, $\tau_{i+1} = \mathcal{O}(h^3)$, so the global truncation error is $\mathcal{O}(h^2)$.

## §12.2 High-order Runge-Kutta Method

The order of accuracy can be improved by constructing a better estimate $g(t_i, y_i)$ for the slope $f(t,y)$ in the iteration step $y_{i+1} = y_i + hg(t_i, y_i)$. This can be done by considering more careful linear combinations of $f\left(t_i + ch_1\tilde{y}\right)$.

The most popular high-order RK method is classical RK4 (4-stage RK). For nonautonomous case DE, the formula is given by

$$\begin{cases} S_1 = f(t_i, y_i) \\ S_2 = f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2} S_1\right) \\ S_3 = f\left(t_i + \frac{h}{2}, y_i + \frac{h}{2} S_2\right) \\ S_4 = f\left(t_{i+1}, y_i + hS_3\right) \\ y_{i+1} = y_i + \frac{h}{6}\left(S_1 + 2S_2 + 2S_3 + S_4\right) \\ y_0 = y(t_0), \quad i = 0, \dots, N-1 \end{cases}$$

For autonomous DE, the formula is

$$\begin{cases} S_1 = f(y_i) \\ S_2 = f\left(y_i + \frac{h}{2} S_1\right) \\ S_3 = f\left(y_i + \frac{h}{2} S_2\right) \\ S_4 = f\left(y_i + hS_3\right) \\ y_{i+1} = y_i + \frac{h}{6}\left(S_1 + 2S_2 + 2S_3 + S_4\right) \\ y_0 = y(t_0), \quad i = 0, \dots, N-1 \end{cases}$$

The local truncation error $\tau_i(h)$ of RK4 is $\mathcal{O}(h^5)$, so RK4 is 4th order method (when $y(t)$ is 5-times differentiable), i.e.,

$$\max_{0 \le i \le N} |e_i| \le C_1 \underbrace{|e_0|}_{=0} + C_2 h^4$$

General s-stage explicit RK method for nonautonomous DE is

$$\begin{cases} S_1 = f(t_i, y_i) \\ S_2 = f\left(t_i + c_2 h, y_i + \alpha_{21} h S_1\right) \\ S_3 = f\left(t_i + c_3 h, y_i + \alpha_{31} h S_1 + \alpha_{32} h S_2\right) \\ \vdots \\ S_s = f\left(t_i + c_s h, y_i + h \sum_{j=1}^{s-1} \alpha_{sj} S_j\right) \\ y_{i+1} = y_i + h\left(\sum_{j=1}^{s} \beta_j S_j\right), \quad i = 0, \dots, N-1 \\ y_0 = y(t_0) \end{cases}$$

where $c_k = \sum_{j=1}^{s} \alpha_{kj}$ and $\sum_{j=1}^{s} \beta_j = 1$.
These parameters can be represented in "Butcher table".

$$\begin{array}{c|cccc} c_1 & \alpha_{11} & \alpha_{12} & \dots & \alpha_{1s} \\ c_2 & \alpha_{21} & \alpha_{22} & \dots & \alpha_{2s} \\ & \vdots & \vdots & \vdots & \\ c_s & \alpha_{s1} & \alpha_{s2} & \dots & \alpha_{ss} \\ \hline 1 & \beta_1 & \beta_2 & \dots & \beta_s \end{array}$$

# §13 Lec 13: Feb 1, 2022

## §13.1 High-order Runge-Kutta Method (Cont'd)

Recall the formula of the midpoint method (RK2)

$$\begin{cases} y^* = y_k + \frac{h}{2} f(t_k, y_k) \\ y_{k+1} = y_k + hf\left(t_k + \frac{h}{2}, y^*\right) \\ y_0 = y(t_0) \end{cases} \iff \begin{cases} S_1 = f(t_k, y_k) \\ S_2 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} S_1\right) \\ y_{k+1} = y_k + hS_2 \end{cases}$$

So, the corresponding Butcher's table

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \hline \frac{1}{2} & \frac{1}{2} & 0 \\ \hline 1 & 0 & 1 \end{array}$$

The Butcher's table for modified Euler's method

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline 1 & \frac{1}{2} & \frac{1}{2} \end{array}$$

The Butcher's table for classical RK4

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline 1 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

**Question 13.1.** How do we choose the parameters in a general s-stage method?

One is typically interested in achieving high-order of accuracy. To do that we can choose the parameters $\alpha_{kj}, \beta_j$ so that it makes the order of LTE as high as possible by making the exponent of the leading term in LTE as high as possible.
The classical RK4 method has become one of the most popular numerical methods.

- Reasonably high accuracy: 4th order
- Still easy to implement
- No 5-stage RK method exists that provides 5th order of accuracy. There are RK methods with order of accuracy higher than 4, but they require more than 5-stages (more complicated and expensive to compute)

Summary of RK methods:

1. Any s-stage explicit RK method cannot have order greater than $s$

$$|e_i(h)| \sim \mathcal{O}(h^p)$$

for some $p \leq s$.

2. There exists no 5-stage explicit RK method with order 5. More precisely, see the following table

| Number of stage | $1 \leq s \leq 4$ | $5 \leq s \leq 4$ | $8 \leq s \leq 9$ | $s \geq 10$ |
|---|---|---|---|---|
| Order of accuracy | $s$ | $s-1$ | $s-2$ | $s-3$ |

## §13.2 Stability of Numerical Methods

We have seen that the error bounds of a numerical method depends on the step size $h$. In practice, we typically have a desired error and choose the step size $h$ accordingly. The order of accuracy, say $\mathcal{O}(h^p)$, gives a rough idea about how to choose the step size $h$, but usually this is not enough: In $\mathcal{O}(h^p)$, there are hidden dependence on other parameters such as Lipschitz constant $L$ or $T$. Absolute stability provides more precise information to determine the step size $h$.
First, let's start with recalling the error bound of Euler's method

$$\max_{0\le i\le N} |y(t_i) - y_i| \le e^{L(T-t_0)}|e_0| + \frac{e^{L(T-t_0)}}{L} \cdot \frac{Mh}{2}$$

Let's motivate the importance of choosing the step size $h$.

**Example 13.1**
Consider

$$\begin{cases} \frac{dy}{dt} = \lambda y \\ y(t_0) = y_0 \end{cases}$$

We will see the step size requirements to solve above IVP for Euler's method, the midpoint method, and Trapezoidal method (Cronk-Nicholson)

- Trapezoidal method: Let an IVP be given by

  $$\begin{cases} \frac{dy}{dt} = f(t,y) \\ y(t_0) = y_0, \quad t \in [t_0, T] \end{cases}$$

  The formula is given by

  $$y_{i+1} = y_i + \frac{h}{2}\left(f(t_i, y_i) + f(t_{i+1}, y_{i+1})\right)$$

  So, for the ODE in the example $\frac{dy}{dt} = \lambda y$, we have

  $$y_{i+1} = y_i + \frac{h}{2}\left(\lambda y_i + \lambda y_{i+1}\right)$$

  We need to solve for $y_{i+1}$, which is easy in this case

  $$\left(1 - \frac{\lambda h}{2}\right) y_{i+1} = \left(1 + \frac{\lambda h}{2}\right) y_i$$
  $$y_{i+1} = \left(\frac{1 + \frac{\lambda h}{2}}{1 - \frac{\lambda h}{2}}\right) y_i$$

# §14 Lec 14: Feb 5, 2022

## §14.1 Stability of Numerical Methods (Cont'd)

We can deduce that the proper step size $h$ to get a desired error not only depends on the problem (such as L, T) but the numerical method.

**Definition 14.1** (Interval of Absolute Stability) — The interval for step size of absolute stability of a numerical method is the set

$$S = \{h\lambda : |y_i| \to 0 \text{ as } i \to \infty \,\forall \text{ initial data } y_0\}$$

when the method is applied to the problem

$$\begin{cases} \frac{dy}{dt} = \lambda y \\ y(t_0) = y_0 \end{cases}$$

Let's consider the IVP

$$\begin{cases} \frac{dy}{dt} = \lambda y \\ y(t_0) = y_0, \quad t \in [t_0, T] \end{cases}$$

Then, we have

$$\begin{aligned} \frac{dy}{dt} &= \lambda y \\ \implies y(t) &= \tilde{C}e^{\lambda t} \\ \implies y(t) &= y_0 e^{\lambda(t-t_0)} \end{aligned}$$

Thus, $|y(t)| \to \infty$ if $\lambda > 0$ and $|y(t)| \to 0$ if $\lambda < 0$.
Goal: To find $S$ when $\lambda < 0$ for a numerical method. By the definition of $S$, we need to find the condition on $h$ s.t. $y_i \to 0$ as $i \to \infty$ for $\lambda < 0$ for a numerical method.

1. Euler's method

$$\begin{cases} \frac{dy}{dt} = \lambda y \\ y(t_0) = t_0, \quad t \in [t_0, T] \end{cases}$$

Recall the formula of Euler's method

$$\begin{aligned} y_{i+1} &= y_i + hf(t_i, y_i) \\ &= y_i + h\lambda y_i \\ &= (1 + h\lambda)y_i \\ &= (1 + h\lambda)^2 y_{i-1} \\ &= \dots \\ &= (1 + h\lambda)^{i+1} y_0 \end{aligned}$$

So, to have $|y_{i+1}| \to 0$ as $i \to \infty$,

$$\left|(1 + h\lambda)^{i+1} y_0\right| = |1 + h\lambda|^{i+1} |y_0| \to 0$$

as $i \to \infty$.

$$\begin{aligned} \iff &|1 + h\lambda| < 1 \\ \iff &-1 < 1 + h\lambda < 1 \\ \iff &-2 < h\lambda < 0 \end{aligned}$$

Thus, the interval of absolute stability of Euler's method is $S = (-2, 0)$ and the corresponding step size $h$ satisfies $0 < h < -\frac{2}{\lambda}$ (since $\lambda < 0$)

2. For the midpoint method, the formula is given by

$$\begin{cases} y^* = y_i + \frac{h}{2} f(t_i, y_i) \\ y_{i+1} = y_i + hf\left(t_i + \frac{h}{2}, y^*\right) \end{cases}$$

Since the IVP is

$$\begin{cases} \frac{dy}{dt} = \lambda y \\ y(t_0) = y_0, \quad t \in [t_0, T] \end{cases}$$

Then,

$$\begin{cases} y^* = y_i + \frac{h}{2} \lambda y_i \\ y_{i+1} = y_i + h\lambda y^* \end{cases}$$

$\implies y_{i+1} = y_i + h\lambda\left(y_i + \frac{h\lambda}{2} y_i\right)$. So,

$$y_{i+1} = \left(1 + h\lambda + \frac{(h\lambda)^2}{2}\right) y_i$$

In order to have $|y_{i+1}| \to 0$ as $i \to \infty$,

$$\begin{aligned} \left|1 + h\lambda + \frac{(h\lambda)^2}{2}\right| &< 1 \\ -2 < h\lambda + \frac{(h\lambda)^2}{2} &< 0 \\ -2 < h\lambda &< 0 \end{aligned}$$

The interval of absolute stability for the midpoint method is $S = (-2, 0)$ and the corresponding $h$ satisfies $h \in \left(0, -\frac{2}{\lambda}\right)$.

3. Trapezoidal method: From the last lecture, we have

$$y_{i+1} = \left(\frac{1 + \frac{\lambda h}{2}}{1 - \frac{\lambda h}{2}}\right) y_i$$

Therefore to have $|y_{i+1}| \to 0$ as $i \to \infty$,

$$\left|\frac{1 + \frac{\lambda h}{2}}{1 - \frac{\lambda h}{2}}\right| < 1$$

Check that the interval of stability is $(-\infty, o)$ for the trapezoidal method so the corresponding $h$ satisfies $h \in (0, \infty)$. The similar argument can be applied to other numerical methods to find its interval of stability for the IVP $\frac{dy}{dt} = \lambda y$ $(\lambda < 0)$.

**Question 14.1.** How do we find the interval of stability of a numerical method for general IVPs, $\frac{dy}{dt} = f(t, y)$?

Let $y(t)$ be analytic solution to $\frac{dy}{dt} = f(t, y)$ and $\tilde{y}(t)$ be the numerical solution to $\frac{dy}{dt} = f(t, y)$. Let $v(t)$ be the difference of $\tilde{y}(t)$ and $y(t)$, i.e., $v(t) = \tilde{y}(t) - y(t)$. Then,

$$\begin{aligned} \frac{dv}{dt} &= \frac{d}{dt}\left(\tilde{y}(t) - y(t)\right) \\ &= \frac{d}{dt}\tilde{y}(t) - \frac{d}{dt}y(t) \\ &\approx f\left(t, \tilde{y}(t)\right) - f(t, y) \end{aligned}$$

Thus, $\frac{dv}{dt} = f(t,\tilde{y}) - f(t,y) \approx \frac{\partial f}{\partial y} v(t)$. This is similar to the previous IVP, $\frac{dz}{dt} = \lambda z$ for $\lambda < 0$ case

$$\implies |z(t)| \to 0 \quad \text{as} \quad t \to \infty$$

If $\frac{\partial f}{\partial y} < 0$, this implies that $|v(t)| \to 0$ as $t \to \infty$, above finding implies that $h \cdot \frac{\partial f}{\partial y} \in S$ gives the step size requirement for the IVP $\frac{dy}{dt} = f(t,y)$.

# §15 Lec 15: Feb 11, 2022

## §15.1 Stability of Numerical Methods (Cont'd)

**Fact 15.1.** To estimate the step size requirement for a good numerical solution for the IVP $\frac{dy}{dt} = f(t,y)$. If $\frac{\partial f}{\partial y} < 0$, we need to find the stability restriction on $h$ s.t. $h \cdot \frac{\partial f}{\partial y} \in S$.

**Example 15.1**

Use the Euler's to solve following IVPs

$$\frac{dy}{dt} = \cos(y) + 1 \quad \text{with} \quad y(0) = 0 \quad \text{and} \quad t \in [0,5]$$

For this IVP, $f(t,y) = \cos(y)$, then $\frac{\partial f}{\partial y} = -\sin(y)$. It turns out that $y(t) \in [0, \pi]$

$$\begin{aligned} \frac{dy}{1+\cos y} &= dt \\ \frac{1}{2}\sec^2\left(\frac{y}{2}\right) dy &= dt \\ t &= \tan\left(\frac{y}{2}\right) \\ y &= 2\tan^{-1}(t) \end{aligned}$$

We want to find $h$ s.t. $h \cdot \frac{\partial f}{\partial y} \in S = (-2,0)$. By considering the points to find worst possible case which happens when $\frac{\partial f}{\partial y} = -1$, $-h \in (-2,0) \implies h \in (0,2)$.

A summary for the $\tau_i, e_i$ the interval of stability for Euler's method RK2, RK4, and trapezoidal

| Method | Order of $\tau_i$ | Order of $e_i$ | Interval of AS |
|---|---|---|---|
| Euler | $\mathcal{O}(h^2)$ | $\mathcal{O}(h)$ | $(-2,0)$ |
| RK2 | $\mathcal{O}(h^3)$ | $\mathcal{O}(h^2)$ | $(-2,0)$ |
| RK4 | $\mathcal{O}(h^5)$ | $\mathcal{O}(h^4)$ | $(-2.78,0)$ |
| Trapezoidal | $\mathcal{O}(h^3)$ | $\mathcal{O}(h^2)$ | $(-\infty,0)$ |

Note that from this table, the interval of absolute stability of trapezoidal method is $(-\infty, 0)$. This implies that when $\frac{\partial f}{\partial y} < 0$, $h \cdot \frac{\partial f}{\partial y} \in (-\infty, 0)$ for any positive step size $h$. In other words, the trapezoidal method is unconditionally stable (no restriction on the step size). Typically, implicit methods are more stable than the implicit methods.

## §15.2 Stiff Problems

**Definition 15.2** (Stiff Problem) —

- A problem is stiff if $\frac{\partial f}{\partial y} < 0$ and $\left|\frac{\partial f}{\partial y}\right|$ is large.
- A problem is stiff if explicit methods don't work or work only with very small step size $h$.
- A problem is stiff if some components of solution decay much faster than others, for example $y(t) = e^{-2t} + e^{-500t}$

Implicit methods are typically a good choice for solving the stiff problems. For example, the trapezoidal method is unconditionally stable, so it has no restrictions on the step size $h$ to be stable. Summary of choosing numerical methods for solving on IVP

- If the IVP is non-stiff, we usually prefer to use explicit methods (e.g., te classical RK4) and the step size $h$ is determined by the accuracy.
- If the IVP is stiff, we choose the implicit methods so that $h$ doesn't need to be extremely small.

## §15.3 Multi-step Methods

Multi-step methods have become popular in machine learning. We will briefly take a look at some multi-step methods:

1. Adam-Bashforth (AB method) which is on explicit method
2. Adam-Moulton (AM method) which is an implicit method.

**Definition 15.3** (k-step multi-step method) — A k-step multi-step method for solving the IVP

$$\begin{cases} \frac{dy}{dt} = f(t,y) \\ y(t_0) = y_0, \quad t \in [t_0, T] \end{cases}$$

has a difference equation for finding the approximation $y_{i+1}$ at the mesh point $t_{i+1}$ represented by

$$y_{i+1} = \sum_{j=1}^{k} \alpha_j y_{i+1-j} + h \sum_{j=0}^{k} \beta_j f\left(t_{i+1-j}, y_{i+1-j}\right) \tag{*}$$

- From the formula, we notice that we need to solve previous k solution values $(y_i, \ldots, y_{i+1-k})$ at each step for the computation of next step.
- Also from the formula (*) of the k-step method, we need to know the first k values: $y_0, y_1, \ldots, y_{k-1}$ in advance for the method to work. However, we only know $y_0$ from the initial condition. To compute $y_1, y_2, \ldots, y_{k-1}$, one often use the other 1-step or multi-step methods such as RK2.
- Note that the meaning of the word "multi-step" differs from "multi-stage". For example, the s-stage RK methods are all 1-step methods since at each step, we only need to know $(y_i, t_i)$ to compute $y_{i+1}$.

**Definition 15.4** (Adam-Bashforth Method (AB Method)) — The form of the formula of AB method is

$$y_{i+1} = y_i + h \sum_{j=1}^{k} \beta_j f\left(t_{i+1-j}, y_{i+1-j}\right)$$

with $\sum_{j=1}^{k} \beta_j = 1$ where $\beta_j \in \mathbb{R}$.

For each $k$, we will choose $\beta_1, \beta_2, \ldots, \beta_k$ appropriately to maximize the order of accuracy. One way is to use Taylor series expansion for LTE analysis and choose $\beta_j$'s to eliminate as many terms as possible. Another way to determine $\beta_1, \ldots, \beta_k$ is based on polynomial interpolation (fitting) which is the route we take.

For each subinterval $[t_i, t_{i+1}]$ we have $\frac{dy}{dt} = f(t, y)$ on $[t_i, t_{i+1}]$. Then we have

$$\int_{t_i}^{t_{i+1}} \frac{dy}{dt} dt = \int_{t_i}^{t_{i+1}} f(t, y) dt$$
$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(t, y) dt$$

We replace $f(t, y(t))$ with its interpolation polynomial by using the data obtained previously $(t_0, y_0), (t_1, y_1), \ldots, (t_i, y_i)$ where $y_i$ is the estimate for $y(t_i)$. We expect to have estimates of

$$f\left(t_j, y(t_j)\right) \approx f(t_j, y_j) \coloneqq f_j$$

More precisely, to get the k-step AB formula, we use the Lagrange interpolation polynomial with degree k.

# §16 Lec 16: Feb 14, 2022

## §16.1 Multi-step Methods (Cont'd)

Let

$$P_j(t) = \frac{\prod_{i-k+1\le l\ne j\le i}(t-t_l)}{\prod_{i-k+1\le l\ne j\le i}(t_j-t_l)}$$

So $P_j$ is a polynomial with roots $t_i, t_{i-1}, \ldots, t_{j+1}, t_{j-1}, t_{j-2}, \ldots, t_{i-k+1}$ and $P_j(t_j) = 1$. Let $P(t) = \sum_{j=1}^k f_{i+1-j}P_{i+1-j}(t)$. By the properties of $P_j$'s, $P(t_{i+1-j}) = f_{i+1-j}$. Hence,

$$\begin{aligned}\int_{t_i}^{t_{i+1}} f(t,y)\,dt &\approx \int_{t_i}^{t_{i+1}} p(t)dt\\ &= \int_{t_i}^{t_{i+1}} \sum_{j=1}^k f_{i+1-j}P_{i+1-j}(t)\,dt\\ &= h\cdot\sum_{j=1}^k \beta_j f_{i+1-j}\end{aligned}$$

where $\beta_j = \frac{1}{h}\int_{t_i}^{t_{i+1}} P_{i+1-j}(t)dt$. Note that

$$\sum_{j=1}^k \beta_j = 1$$

Also,

$$\sum_{j=1}^k P_{i+1-j}(t) = 1$$

and the polynomial $\sum_{j=1}^k P_{i+1-j}(t) - 1$ must be the zero polynomial.

## §16.2 Special Cases of AB Method

1. $k = 1$ (AB1)

   The AB method reduces to the Euler's method in this case. From the form of AB formula,

   $$\begin{aligned}y_{i+1} &= y_i + \beta_1 hf(t_i,y_i)\\ &= y_i + hf(t_i,y_i)\end{aligned}$$

   since $\sum_{j=1}^k \beta_j = \beta_1 = 1$. Order of accuracy of AB1 is $\mathcal{O}(h)$.

2. $k = 2$ (AB2)

   $$y_{i+1} = y_i + h\left(\beta_1 f(t_i,y_i) + \beta_2 f(t_{i-1},y_{i-1})\right)$$

$\beta_1, \beta_2$ can be computed as follow

$$\begin{aligned}
\beta_1 &= \frac{1}{h}\int_{t_i}^{t_{i+1}} \frac{t-t_{i-1}}{t_i - t_{i-1}} dt \\
&= \dots \\
&= \frac{3}{2} \\
\beta_2 &= \frac{1}{h}\int_{t_i}^{t_{i+1}} \frac{t-t_i}{t_{i-1} - t_i} dt \\
&= \dots \\
&= -\frac{1}{2}
\end{aligned}$$

So the formula for AB2 is

$$y_{i+1} = y_i + h\left(\frac{3}{2}f(t_i, y_i) - \frac{1}{2}f(t_{i-1}, y_{i-1})\right)$$

This method has $\mathcal{O}(h^2)$ accuracy.

3. $k = 3$ (AB3)

   The formula can be obtained in a similar way

$$y_{i+1} = y_i + h\left(\frac{23}{12}f(t_i, y_i) - \frac{16}{12}f(t_{i-1}, y_{i-1}) + \frac{5}{12}f(t_{i-2}, y_{i-2})\right)$$

   This method is $\mathcal{O}(h^3)$ method.

4. Typically, the k -step AB method is $\mathcal{O}(h^k)$ method.

# §17 Lec 17: Feb 16, 2022

## §17.1 Adam-Moulton Method (AM Method)

The form of the formula of $k$-step AM method is given by

$$y_{i+1} = y_i + h \sum_{j=0}^{k} \beta_j f\left(t_{i+1-j}, y_{i+1-j}\right)$$

with $\sum_{j=0}^{k} \beta_j = 1$ and $\beta_j \in \mathbb{R}$.
Let

$$P_j(t) = \frac{\prod_{i-k+1 \le l \ne i-j+1 \le i+1} (t - t_l)}{\prod_{i-k+1 \le l \ne i-j+1 \le i+1} (t_{i-j+1} - t_l)}$$

The roots of polynomial $P_j(t)$ are $t_l$ with $i-k+1 \le l \ne i-j+1 \le i+1$ and $P_j(t_{i-j+1}) = 1$ and $\beta_j = \frac{1}{h}\int_{t_i}^{t_{i+1}} P_j(t)dt$.

1. $k = 0$ (AM0)

   The formula of AM0 i nthis case is

   $$\begin{aligned} y_{i+1} &= y_i + h \cdot \beta_0 f(t_{i+1}, y_{i+1}) \\ &= y_i + hf\left(t_{i+1}, y_{i+1}\right) \end{aligned}$$

   AM0 is the backward Euler, which is $\mathcal{O}(h)$ method.

2. $k = 1$ (AM1)

   It turns out that AM1 is the trapezoidal method with formula we've seen before

   $$y_{i+1} = y_i + \frac{h}{2}\left(f(t_i, y_i) + f(t_{i+1}, y_{i+1})\right)$$

   which is $\mathcal{O}(h^2)$ method. Note that $\beta_0 = \beta_1 = \frac{1}{2}$ as we did before

   $$\begin{aligned} \beta_0 &= \frac{1}{h}\int_{t_i}^{t_{i+1}} \frac{t - t_i}{t_{i+1} - t_i} dt = \ldots = \frac{1}{2} \\ \beta_1 &= \frac{1}{h}\int_{t_i}^{t_{i+1}} \frac{t - t_{i+1}}{t_i - t_{i+1}} dt = \ldots = \frac{1}{2} \end{aligned}$$

3. $k = 2$ (AM2)

   The formula of AM2 can be obtained by the similar process, which is given by

   $$y_{i+1} = y_i + \frac{h}{12}\left(5f(t_{i+1}, y_{i+1}) + 8f(t_i, y_i) - f(t_{i-1}, y_{i-1})\right)$$

   This is a $\mathcal{O}(h^3)$ method. For example, we can compute $\beta_0$ as follows

   $$\begin{aligned} \beta_0 &= \frac{1}{h}\int_{t_i}^{t_{i+1}} \frac{(t - t_i)(t - t_{i-1})}{(t_{i+1} - t_i)(t_{i+1} - t_{i-1})} dt \\ &= \frac{5}{12} \end{aligned}$$

4. For general $k$, $k$-step AM has accuracy of order $k+1$ ($e_i = \mathcal{O}(h^{k+1})$.

# §18 Lec 18: Feb 23, 2022

## §18.1 AM Method (Cont'd)

Since AM method is implicit, we need to solve the equation to compute $y_{i+1}$ at each step. Recall the AM formula

$$\begin{aligned} y_{i+1} &= y_i + h\sum_{j=0}^{k} \beta_j f\left(t_{i+1-j}, y_{i+1-j}\right) \\ &= y_i + \beta_0 h f(t_{i+1}, y_{i+1}) + h\sum_{j=1}^{k} \beta_j f\left(t_{i+1-j}, y_{i+1-j}\right) \end{aligned}$$

If the function $f$ is linear, the we can solve for $y_{i+1}$ in the formula above. However, $f$ is usually nonlinear, so it is typically not easy to solve for $y_{i+1}$ exactly. We usually solve this by some numerical methods such as Newton's method, bisection methods, etc. to find $y_{i+1}$.

## §18.2 Interval of Absolute Stability

We will extend the notion of interval of absolute stability to complex numbers

- Interval of stability: $\lambda h \in \mathbb{R}$
- Region of stability: $\lambda h \in \mathbb{C}$

**Example 18.1**

Find the region of AS for Euler's method. The formula of Euler's method for the IVP

$$\begin{cases} \frac{dy}{dt} = \lambda y \\ y(t_0) = y_0 \end{cases}$$

is

$$\begin{aligned} y_{i+1} &= y_i + hf(t_i, y_i) \\ &= y_i + h\lambda y_i \\ &= (1 + h\lambda)y_i \\ &= \ldots = (1 + h\lambda)^{i+1} y_0 \end{aligned}$$

when $h\lambda \in \mathbb{C}$, $y_{i+1} \to 0$ iff $|1 + h\lambda| < 1$.

**Example 18.2**

The region of AS for the midpoint method for the same IVP. From the formula of the midpoint method,

$$\begin{cases} y^* = y + \frac{h}{2} f(t_i, y_i) \\ y_{i+1} = y + hf\left(t_i + \frac{h}{2}, y^*\right) \\ y_0 = y(t_0) \end{cases}$$

Then,

$$\begin{aligned} y_{k+1} &= y_i + hf\left(t_i + \frac{h}{2}, y_k + \frac{h}{2} f(t_i, y_i)\right) \\ \implies y_{i+1} &= y_i + h\lambda\left(y_i + \frac{h}{2} f(t_i, y_i)\right) \\ &= \left(1 + h\lambda \frac{(h\lambda)^2}{2}\right) y_i \end{aligned}$$

To make sure $y_{i+1} \to 0$ as $i \to \infty$, $\left|1 + h\lambda + \frac{(h\lambda)^2}{2}\right| < 1$.

**Definition 18.3** (Uncondiitonally Stable Method) — If the region of AS for a method includes the left half complex plane then the method is called unconditionally stable.

For example, trapezoidal method and the backward Euler's method are unconditionally stable. From the formula of trapezoidal method,

$$y_{i+1} = \left(\frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}}\right) y_i$$

To make sure $y_{i+1} \to 0$ as $i \to \infty$

$$\left|\frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}}\right| < 1$$

Let $h\lambda = a + ib$ where $a, b \in \mathbb{R}$ since $h\lambda \in \mathbb{C}$. Note that $|\alpha + i\beta|^2 = \alpha^2 + \beta^2 \quad \forall \alpha, \beta \in \mathbb{R}$. So after some manipulation, we get $a < 0$. The region of AS for the trapezoidal method is the left half of complex plane.

## §18.3 Numerical Methods for Systems of ODEs

**Definition 18.4** (ODE system) — An m-th order system of 1st order IVP has the following form

$$\begin{cases} \frac{dy_1}{dt} = f_1(t, y_1, y_2, \ldots, y_m) \\ \frac{dy_2}{dt} = f_2(t, y_1, y_2, \ldots, y_m) \\ \vdots \\ \frac{dy_m}{dt} = f_m(t, y_1, y_2, \ldots, y_m) \end{cases}$$

with initial condition

$$\begin{cases} y_1(t_0) = y_{1,0} \\ y_2(t_0) = y_{2,0} \\ \vdots \\ y_m(t_0) = y_{m,0} \end{cases}$$

We can write this IVP in the vector form. The IVP is then written as

$$\begin{cases} \frac{d\vec{y}}{dt} = \vec{F}\,(t, \vec{y}) \\ \vec{y}(t_0) = \vec{y}_0 \end{cases}$$

**Example 18.5**

For $m = 2$,

$$\begin{cases} \frac{dy_1}{dt} = f_1(t, y_1, y_2) \\ \frac{dy_2}{dt} = f_2(t, y_1, y_2) \end{cases} \quad \text{with} \quad \begin{cases} y_1(t_0) = y_{1,0} \\ y_2(t_0) = y_{2,0} \end{cases}$$

Using

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad F = \begin{bmatrix} f_1(t, y_1, y_2) \\ f_2(t, y_1, y_2) \end{bmatrix}, \quad \vec{y}_0 = \begin{bmatrix} y_{1,0} \\ y_{2,0} \end{bmatrix}$$

Goal: Use some numerical methods to compute the estimates $\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_N$ for $\vec{y}(t_0), \vec{y}(t_1), \ldots, \vec{y}(t_N)$.

**Remark 18.6.**

1. If $f_1, f_2, \ldots, f_m$ are linear functions w.r.t. $y_1, y_2, \ldots, y_m$, then $\frac{d\vec{y}}{dt} = \vec{F}(t, \vec{y})$ is a linear system.

$$\vec{F}\,(t, \vec{y}) = \mathbf{A}(t)\vec{y} + \vec{b}(t)$$

**Example 18.7**

Consider

$$\begin{cases} \frac{dy_1}{dt} = \sqrt{t}y_1 - 3y_2 + \sin t \\ \frac{dy_2}{dt} = y_1 + e^{-t} \end{cases}$$

with

$$\begin{cases} y_1(0) = -1 \\ y_2(0) = 2 \end{cases}$$

This system of ODEs is linear.

$$\begin{aligned} \vec{F}(t) &= \begin{bmatrix} \sqrt{t}y_1 - 3y_2 + \sin t \\ y_1 + e^{-t} \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{t} & -3 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} \sin t \\ e^{-t} \end{bmatrix} \\ &= \mathbf{A}(t)\vec{y} + \vec{b}(t) \end{aligned}$$

**Remark 18.8.** 2. Let a system of ODEs be given by

$$\frac{d\vec{y}}{dt} = \vec{F}(t, y) = \mathbf{A}(t)\vec{y} + \vec{b}(t)$$

If $\vec{b}(t) = 0$, then the ODE system $\frac{d\vec{y}}{dt} = \mathbf{A}(t)\vec{y}$ is homogeneous. Otherwise, it is inhomogeneous.

3. If one of the functions $f_1, f_2, \ldots, f_m$ is nonlinear, then $\frac{d\vec{y}}{dt} = \vec{F}(t, \vec{y})$ is nonlinear. In other words, if one of the ODE equations is nonlinear, then the ODE system is nonlinear.

**Example 18.9**

Consider

$$\begin{cases} \frac{dy_1}{dt} = y_1 + 2y_2^2 \\ \frac{dy_2}{dt} = 2y_1 - 3y_2 \end{cases}$$

with

$$\begin{cases} y_1(0) = 2 \\ y_2(0) = 4 \end{cases}$$

Since $y_1 + 2y_2^2$ is nonlinear with respect to $y_2$, this ODE is nonlinear.

Let's get to numerical methods to solve IVP systems.

- Euler's method for one ODE. Recall

  $$\begin{cases} y_{i+1} = y_i + hf(t_i, y_i) \\ y_0 = y(t_0) \end{cases}$$

  So we can extend the method to a system of ODEs as follows

  $$\begin{cases} \frac{d\vec{y}}{dt} = \vec{F}(t, \vec{y}) \\ \vec{y}(t_0) = y_0 \end{cases}$$

  So the formula is

  $$\begin{cases} \vec{y}_{i+1} = \vec{y}_i + h\vec{F}(t_i, \vec{y}_i) \\ \vec{y}_0 = \vec{y}(t_0) \end{cases}$$

where

$$\vec{y}_{i+1} = \begin{bmatrix} y_{1,i+1} \\ y_{2,i+1} \\ \vdots \\ y_{m,i+1} \end{bmatrix}, \quad \vec{y}_i = \begin{bmatrix} y_{1,i} \\ y_{2,i} \\ \vdots \\ y_{m,i} \end{bmatrix}$$

$$\vec{F}\left(t_i, \vec{y}_i\right) = \begin{bmatrix} f_1\left(t_i, y_{1,i}, y_{2,i}, \ldots, y_{m,i}\right) \\ f_2\left(t_i, y_{1,i}, y_{2,i}, \ldots, y_{m,i}\right) \\ \vdots \\ f_m\left(t_i, y_{1,i}, y_{2,i}, \ldots, y_{m,i}\right) \end{bmatrix}$$

In particular, when the ODE system is linear, then $\vec{F}(t_i, \vec{y}_i)$ can be expressed in the matrix form.

- Modified Euler's method: Recall that for ODE with $m = 1$, the formula is

$$\begin{cases} y^* = y_i + hf(t_i, y_i) \\ y_{i+1} = y_i + \frac{h}{2}\left(f(t_i, y_i) + f(t_{i+1}, y^*)\right) \end{cases}$$

or

$$\begin{cases} S_1 = f(t_i, y_i) \\ S_2 = f\left(t_{i+1}, y_i + hf(t_i, y_i)\right) \\ y_{i+1} = y_i + \frac{h}{2}(S_1 + S_2) \end{cases}$$

For the general system of ODEs, the formula of the modified Euler's method is

$$\begin{aligned} \vec{y}^* &= \vec{y}_i + h\vec{F}\left(t_i, \vec{y}_i\right) \\ \vec{y}_{i+1} &= \vec{y}_i + \frac{h}{2}\left(\vec{F}(t_i, \vec{y}_i) + \vec{F}(t_{i+1}, \vec{y}^*)\right) \end{aligned}$$

with the initial conditions $\vec{y}_0 = \vec{y}(t_0)$.

# §19 Lec 19: Feb 25, 2022

## §19.1 Numerical Methods for Systems of ODEs (Cont'd)

**Example 19.1**

Consider

$$\begin{cases} \frac{dy_1}{dt} = 2y_1 - 3y_2 + \sin t + 2e^{-t} \\ \frac{dy_2}{dt} = y_1 + 2y_2 + 3\cos t \end{cases}$$

with initial conditions

$$\begin{cases} y_1(0) = 1 \\ y_2(0) = -2 \end{cases}$$

1. First, let's write down this IVP in a vector/matrix form.

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \quad \mathbf{A}(t) = \begin{bmatrix} 2 & -3 \\ 1 & 2 \end{bmatrix}$$
$$\vec{b}(t) = \begin{bmatrix} \sin t + 2e^{-t} \\ 3\cos t \end{bmatrix}$$

Then, this IVP is

$$\begin{cases} \frac{d\vec{y}}{dt} = \mathbf{A}(t)\vec{y} + \vec{b}(t) \\ \vec{y}(0) = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \end{cases}$$

Next, we apply numerical methods to solve this problem

- Euler's method

$$\begin{cases} \vec{y}_{i+1} = \vec{y}_i + h\left(\begin{bmatrix} 2 & -3 \\ 1 & 2 \end{bmatrix}\vec{y}_i + \begin{bmatrix} \sin(t_i) + 2e^{-t_i} \\ 3\cos(t_i) \end{bmatrix}\right) \\ \vec{y}_0 = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \end{cases}$$

with

$$\vec{y}_{i+1} = \begin{bmatrix} y_{1,i+1} \\ y_{2,i+1} \end{bmatrix}, \quad \vec{y}_i = \begin{bmatrix} y_{1,i} \\ y_{2,i} \end{bmatrix}$$

- Modified Euler's method

$$\begin{cases} \vec{y}^* = \vec{y}_i + h\left(\begin{bmatrix} 2 & -3 \\ 1 & 2 \end{bmatrix}\vec{y}_i + \begin{bmatrix} \sin(t_i) + 2e^{-t_i} \\ 3\cos(t_i) \end{bmatrix}\right) \\ \vec{y}_{i+1} = \vec{y}_i + \frac{h}{2}\left(\begin{bmatrix} 2 & -3 \\ 1 & 2 \end{bmatrix}\vec{y}_i + \begin{bmatrix} \sin(t_i) + 2e^{-t_i} \\ 3\cos(t_i) \end{bmatrix}\right) + \frac{h}{2}\left(\begin{bmatrix} 2 & -3 \\ 1 & 2 \end{bmatrix}\vec{y}^* + \begin{bmatrix} \sin(t_{i+1}) + 2e^{-t_{i+1}} \\ 3\cos(t_{i+1}) \end{bmatrix}\right) \\ \vec{y}_0 = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \end{cases}$$

with

$$\vec{y}_{i+1} = \begin{bmatrix} y_{1,i+1} \\ y_{2,i+1} \end{bmatrix}, \quad \vec{y}_i = \begin{bmatrix} y_{1,i} \\ y_{2,i} \end{bmatrix}, \quad \vec{y}^* = \begin{bmatrix} y_1^* \\ y_2^* \end{bmatrix}$$

## §19.2 Reduction of a Higher ODE to a First Order ODE System

Consider pth order ODE of the form of

$$y^{(p)} = \frac{d^p y}{dt^p} = f\left(t, y, y^{(1)}, y^{(2)}, \dots, y^{(p-1)}\right)$$

for $t \in [t_0, T]$ with initial conditions

$$y(t_0) = u_1, \quad y^{(1)}(t_0) = u_2, \quad \dots \quad y^{(p-1)}(t_0) = u_p$$

**Question 19.1.** How can the pth order ODE be transformed into a first order ODE system?

Let $v_1 = y$, $v_2 = y^{(1)}$, $v_3 = y^{(2)}$, $\dots$, $v_p = y^{(p-1)}$

$$\begin{cases} \frac{dv_1}{dt} = \frac{dy}{dt} = y^{(1)} = v_2 \\ \frac{dv_2}{dt} = \frac{dy_1}{dt} = y^{(2)} = v_3 \\ \vdots \\ \frac{dv_{p-1}}{dt} = \frac{dy^{(p-2)}}{dt} = y^{(p-1)} = v_p \\ \frac{dv_p}{dt} = \frac{dy^{(p-1)}}{dt} = y^{(p)} = f(t, v_1, v_2, \dots, v_p) \end{cases}$$

Therefore, the original ODE can be written as a system of first order ODEs

$$\begin{cases} \frac{dv_1}{dt} = v_2 \\ \frac{dv_2}{dt} = v_3 \\ \vdots \\ \frac{dv_{p-1}}{dt} = v_p \\ \frac{dv_p}{dt} = f\left(t, v_1, v_2, \dots, v_p\right) \end{cases} \tag{*}$$

with the initial conditions

$$v_1(t_0) = u_1, \quad v_2(t_0) = u_2, \quad \dots \quad v_p(t_0) = u_p$$

Let

$$\vec{v}(t) = \begin{bmatrix} v_1(t) \\ v_2(t) \\ \vdots \\ v_p(t) \end{bmatrix} \text{ so } \vec{v}(t_0) = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix}$$

But initially we wanted to solve the pth order ODE numerically. How do we find it? Our goal is to use numerical methods to compute a vector sequence $\vec{v}_0, \vec{v}_1, \dots, \vec{v}_N$, a numerical solution to estimate the analytic solution $\vec{v}(t_0), \vec{v}(t_1), \dots, \vec{v}(t_N)$ of the ODE system. Then, $\vec{v}_0(1)$, $\vec{v}_1(1), \dots, \vec{v}_N(1)$, the first components of the numerical solution are the numerical estimates of $y(t_0), y(t_1), \dots, y(t_N)$, the solution of the original pth order ODE.

# §20 Lec 20: Feb 28, 2022

## §20.1 Reduction of a Higher Order ODE (Cont'd)

Summary: In order to use the numerical methods to solve a p-th order ODE, first transform the p-th order ODE into a first-order ODE system and write it a vector/matrix form.

$$\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_p \end{bmatrix}, \quad \vec{v}_0 = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \end{bmatrix}$$

$$\vec{F}(t, \vec{v}) = \begin{bmatrix} v_2 \\ v_3 \\ \vdots \\ v_p \\ f(t, v_1, v_2, \ldots, v_p) \end{bmatrix}$$

So $\frac{d}{dt}\vec{v} = \vec{F}(t, \vec{v})$ with $\vec{v}(t_0) = \vec{v}_0$. Furthermore, if $\frac{d\vec{v}}{dt} = \vec{F}(t, \vec{v})$ is linear, i.e., $\vec{F}(t, \vec{v})$ is linear then $f(t, v_1, \ldots, v_p)$ is linear and vice versa. In that case, $f(t, v_1, \ldots, v_p) = a_{p,1}v_1 + a_{p,2}v_2 + \ldots + a_{p,p}v_p + b_p(t)$. Thus, the first order ODE system is linear, then, it can be written as

$$\frac{d}{dt}\vec{v} = \begin{bmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & 0 & \ldots & 1 \\ a_{p,1} & a_{p,2} & a_{p,3} & \ldots & a_{p,p} & \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ b_p(t) \end{bmatrix}$$

**Example 20.1**

Consider the following ODE

$$y^{(3)} - 2y' + y + \cos t = 0$$

with $y(0) = 1$, $y'(0) = -1$, $y''(0) = 3$.

Let $v_1 = y_1$, $v_2 = y'$, $v_3 = y''$ and let $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$. Then, we have

$$y^{(3)} = 2y' - y - \cos t$$

So

$$\begin{aligned} \frac{d}{dt}\vec{v} &= \begin{bmatrix} v_2 \\ v_3 \\ 2v_2 - v_1 - \cos t \end{bmatrix} \\ &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 2 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\cos t \end{bmatrix} \end{aligned}$$

with the initial condition $\vec{v}(0) = \begin{bmatrix} 1 \\ -1 \\ 3 \end{bmatrix}$.

The process to solve a higher order IVP

1. Transform the higher-order IVP into a first order ODE system
2. Write the ODE in the vector/matrix form if the ODE system is linear.
3. Use numerical methods to solve the system (Euler's, modified Euler's, RK, etc.)
4. The output of methods are a sequence of vectors $\vec{v}_0, \vec{v}_1, \dots, \vec{v}_N$. If they are saved in the matrix form such as

$$\begin{aligned} v &= \begin{bmatrix} \vec{v}_0 & \vec{v}_1 & \dots & \vec{v}_N \end{bmatrix} \\ &= \begin{bmatrix} u_1 & v_{1,1} & v_{1,2} & \dots & v_{1,N} \\ u_2 & v_{2,1} & v_{2,2} & \dots & v_{2,N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_p & u_{p,1} & u_{p,2} & \dots & u_{p,N} \end{bmatrix} \end{aligned}$$

## §20.2 Boundary Value Problem for ODEs

We will focus on the finite difference method (FDM) for the 2nd order BVP.

**Example 20.2**

The second order ODE with two points boundary value problems has the ODE

$$\frac{d^2y}{dx^2} + f(x, y, y') = 0$$

with the boundary conditions at two boundary points $x = a$ and $x = b$.

**Example 20.3**

Typical boundary conditions

1. Dirichlet boundary conditions

$$y(a) = y_a \text{ and } y(b) = y_b$$

2. Neumann boundary conditions

$$\frac{dy}{dx}(a) = \alpha \text{ and } \frac{dy}{dx}(b) = \beta$$

3. Robin (or mixed) boundary conditions

$$\begin{cases} a_1 y(a) + b_1 \frac{dy}{dx}(a) = g_1(a) \\ a_2 y(b) + b_2 \frac{dy}{dx}(b) = g_2(b) \end{cases}$$

# §21 Lec 21: Mar 3, 2022

## §21.1 Finite Difference Method for BVP

The FDM for a simple 2nd order ODE of the form of $\frac{d^2y}{dx^2} = f(x)$ with the Dirichlet boundary condition

$$y(a) = y_a \text{ and } y(b) = y_b$$

Step 1: We discretize the domain $[a, b]$ into $N$ # of subintervals with endpoints $x_0, x_1, \ldots, x_N$ where $x_i = a + bh$ where $h = \frac{b-a}{N}$. Let $y_i$ be the numerical solution to approximate $y(x_i)$, the exact solution at $x_i$ for $i = 0, 1, \ldots, N$.
Step 2: First, note that $x_0 = a$ and $x_N = b$ and from the boundary conditions. $y(x)$ should satisfy these conditions at the boundary points $x_0 = a$, $x_N = b$. For $x_1, x_2, \ldots, x_{N-1}$, they are the interior points of the domain $[a, b]$ and they need to satisfy the ODE, so $\frac{d^2y}{dx^2}\big|_{x=x_i} = f(x_i)$ for $i = 1, 2, \ldots, N-1$. We want to generate a sequence by numerical methods that satisfies "approximately" the equation $\frac{d^2y}{dx^2}\big|_{x=x_i} = f(x_i)$. Thus, we need to find a numerical quantity approximating $\frac{d^2y}{dx^2}\big|_{x=x_i}$, the second derivative centered difference formula. The idea is as follows

$$\begin{aligned}\frac{dy}{dx}\Big|_{x_{i-\frac{1}{2}}} &\approx \frac{y_i - y_{i-1}}{h}\\ \frac{dy}{dx}\Big|_{x_{i+\frac{1}{2}}} &\approx \frac{y_{i+1} - y_i}{h}\end{aligned}$$

So,

$$\begin{aligned}\frac{d^2y}{dx^2}\Big|_{x=x_i} &\approx \frac{\frac{dy}{dx}\Big|_{x_{i+\frac{1}{2}}} - \frac{dy}{dx}\Big|_{x_{i-\frac{1}{2}}}}{h}\\ &\approx \frac{\frac{y_{i+1}-y_i}{h} - \frac{y_i - y_{i-1}}{h}}{h}\\ &= \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}\end{aligned}$$

Thus, we have

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = f(x_i) \quad \text{for} \quad i = 1, 2, \ldots, N-1$$

Step 3: We will rewrite above equation in a vector/matrix form

$$\begin{aligned} i = 1 &\implies \frac{y_2 - 2y_1 + y_0}{h^2} = f(x_1)\\ &\implies \frac{y_2 - 2y_1}{h^2} = f(x_1) - \frac{y_0}{h^2} = f(x_1) - \frac{y_a}{h^2}\\ i = 2 &\implies \frac{y_3 - 2y_2 + y_1}{h^2} = f(x_2)\\ &\vdots\\ i = N-2 &\implies \frac{y_{N-1} - 2y_{N-2} + y_{N-3}}{h^2} = f(x_{N-2})\\ i = N-1 &\implies \frac{y_N - 2y_{N-1} + y_{N-2}}{h^2} = f(x_{N-1})\\ &\implies \frac{1}{h^2}(-2y_{N-1} + y_{N-2}) = f(x_{N-1}) - \frac{y_N}{h^2} = f(x_{N-1}) - \frac{y_b}{h^2}\end{aligned}$$

Now we are ready to write these equations in the matrix equation form

$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 0 & \\ 1 & -2 & 1 & 0 & \dots & 0 & \\ 0 & 1 & -2 & 1 & \dots & 0 & \\ \vdots & & & & & & \\ 0 & 0 & 0 & \dots & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -2 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-2} \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} f(x_1) - \frac{y_a}{h^2} \\ f(x_2) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) - \frac{y_b}{h^2} \end{bmatrix}$$

Therefore, to solve the Dirichlet problem of the form of $\frac{d^2y}{dx^2} = f(x)$ numerically, solve $\mathbf{A}\vec{y} = \vec{f}_B$. One can check $\mathbf{A}$ is invertible, so $\vec{y} = \mathbf{A}^{-1}\vec{f}_B$. We can use FDM to solve the BVP in a more general form.

$$\begin{cases} \frac{d^2y}{dx^2} + p(x)\frac{dy}{dx} + q(x)y = f(x) \\ y(a) = y_a, \quad y(b) = y_b \end{cases}$$

We can use the centered difference formula to approximate $\frac{d^2y}{dx^2}\Big|_{x_i}$ as before, $\frac{y_{i+1}-y_{i-1}}{2h}$ to approximate $\frac{dy}{dx}\Big|_{x_i}$, i.e.,

$$\begin{cases} \frac{d^2y}{dx^2}\Big|_{x_i} \approx \frac{y_{i+1}-2y_i+y_{i-1}}{h^2} \\ \frac{dy}{dx}\Big|_{x_i} \approx \frac{y_{i+1}-y_{i-1}}{2h} \\ y(x_i) \approx y_i \end{cases}$$

# §22 Lec 22: Mar 7, 2022

## §22.1 Finite Difference Method for BVP (Cont'd)

After plugging them into the ODE, $\frac{d^2y}{dx^2} + p(x)\frac{dy}{dx} + q(x)y = f(x)$, we have

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p(x_i)\frac{y_{i+1} - y_{i-1}}{2h} + q(x_i)y_i = f(x_i)$$

for $i = 1, 2, \dots, N-1$.

$$\begin{aligned}
i = 1 &\implies \frac{y_2 - 2y_1 + y_0}{h^2} + p(x_1)\frac{y_2 - y_0}{2h} + q(x_1)y_1 = f(x_1)\\
&\implies \frac{y_2 - 2y_1}{h^2} + \frac{p(x_1)}{2h}y_2 + q(x_1)y_1 = f(x_1) - \frac{y_a}{h^2} + \frac{p(x_1)}{2h}y_a\\
i = 2 &\implies \frac{y_3 - 2y_2 + y_1}{h^2} + p(x_2)\frac{y_3 - y_1}{2h} + q(x_2)y_2 = f(x_2)\\
&\vdots\\
i = N-2 &\implies \frac{y_{N-1} - 2y_{N-2} + y_{N-3}}{h^2} + p(x_{N-2})\frac{y_{N-1} - y_{N-3}}{2h} + q(x_{N-2})y_{N-2} = f(x_{N-2})\\
i = N-1 &\implies \frac{y_N - 2y_{N-1} + y_{N-2}}{h^2} + p(x_{N-1})\frac{y_N - y_{N-2}}{2h} + q(x_{N-1})y_{N-1} = f(x_{N-1})\\
&\implies \frac{-2y_{N-1} + y_{N-2}}{h^2} - \frac{p(x_{N-1})}{2h}y_{N-2} + q(x_{N-1})y_{N-1} = f(x_{N-1}) - \frac{y_b}{h^2} - \frac{p(x_{N-1})}{2h}y_b
\end{aligned}$$

Matlab code for solving BVP using FDM:

1. Specify boundary conditions $a, b, y_a, y_b$ and the step size $h$
2. Construct $\vec{f}_B$ and $\mathbf{A}$
3. Solve for $\vec{y}$ using $\vec{y} = \mathbf{A} \setminus \vec{f}_B$

## §22.2 Vector Norms

**Definition 22.1** (Vector Norm) — A vector norm for a vector $\vec{v}$ in $\mathbb{R}^m$ will be denoted by $\|\vec{v}\|$ is a function from $\vec{v} \in \mathbb{R}^m \to \mathbb{R}$ s.t.

i) $\|\vec{v}\| \geq 0$ and $\|\vec{v}\| = 0 \iff \vec{v} = \vec{0}$.

ii) $\|\alpha\vec{v}\| = |\alpha|\|\vec{v}\|$ for each $\alpha \in \mathbb{R}$, $\vec{v} \in \mathbb{R}^m$

iii) $\|\vec{u} + \vec{v}\| \leq \|\vec{u}\| + \|\vec{v}\|$ for each $\vec{u}, \vec{v} \in \mathbb{R}^m$.

The norm of $\vec{v}$ is a measure of the "size/generalized length" of $\vec{v}$. Thus, $\|\vec{u} - \vec{v}\|$ is a measure of the distance between $\vec{u}$ and $\vec{v}$. There are many different vector norms

1. 1-norm ($l_1$ norm): $\|\vec{v}\|_1 = \sum_{i=1}^m |v_i|$
2. 2-norm ($l_2$ norm): $\|\vec{v}\|_2 = \sqrt{\sum_{i=1}^m |v_i|^2}$
3. $\infty$-norm ($l_\infty$ norm): $\|\vec{v}\|_\infty = \max_{1\leq i\leq m} |v_i|$

Generalization of $l_1, l_2$ norm

$$l_p \text{ norm: } \|\vec{v}\|_p = \left(\sum_{i=1}^m |v_i|^p\right)^{\frac{1}{p}}$$

Relation between vector norms for $\vec{v} \in \mathbb{R}^m$

1. $\|\vec{v}\|_\infty \le \|\vec{v}\|_1 \le m \cdot \|\vec{v}\|_\infty$
2. $\|\vec{v}\|_\infty \le \|\vec{v}\|_2 \le \sqrt{m}\|\vec{v}\|_\infty$
3. $\frac{1}{\sqrt{m}}\|\vec{v}\|_1 \le \|\vec{v}\|_2 \le \|\vec{v}\|_1$

This means that all three norms $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$ are all equivalent (up to at most constant $m$).

## §22.3 Matrix Norms

**Definition 22.2** (Matrix Norm) — Suppose that $\mathbf{A}, \mathbf{B}$ are square $m \times m$ matrices. A matrix norm of $\mathbf{A}$, denoted by $\|\mathbf{A}\|$, is a scalar valued function satisfying the following properties:

i) $\|\mathbf{A}\| \ge 0$ and $\|\mathbf{A}\| = 0 \iff \mathbf{A} = \mathbf{0}$

ii) $\|\alpha\mathbf{A}\| = |\alpha|\|\mathbf{A}\|$ for each $\alpha \in \mathbb{R}$, $\mathbf{A} \in \mathbb{R}^{m\times m}$

iii) $\|\mathbf{A}+\mathbf{B}\| \le \|\mathbf{A}\| + \|\mathbf{B}\|$ for each $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times m}$

iv) $\|\mathbf{AB}\| \le \|\mathbf{A}\|\|\mathbf{B}\|$

Given a vector norm, a matrix norm can be obtained by defining

$$\|\mathbf{A}\| = \max_{\|\vec{v}\|=1} \|\mathbf{A}\vec{v}\|$$

This norm is sometimes called an operator norm or induced matrix norm associated with a vector norm. Note that it depends on what vector norm we are in the definition.

Property of an operator norm:

For any $\vec{v}$, matrix $\mathbf{A}$, and an operator norm, we have

$$\|\mathbf{A}\vec{v}\| \le \|\mathbf{A}\|\|\vec{v}\|$$

**Example 22.3**

Consider these examples associated with vector norms $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$ respectively.

1. $l_1$-norm

$$\begin{aligned}\|\mathbf{A}\|_1 &= \max_{\|\vec{v}\|_1=1} \|\mathbf{A}\vec{v}\|_1 \\ &= \max_j \sum_{i=1}^m |a_{ij}| \text{ (maximum column sum)}\end{aligned}$$

Example

$$\mathbf{A} = \begin{bmatrix} 1 & -2 & -5 \\ -3 & 6 & 0 \\ 2 & -3 & 5 \end{bmatrix}$$
$$\|\mathbf{A}\|_1 = \max\{1+3+2, 2+6+3, 5+0+5\} = 11$$

2. $l_\infty$ norm

$$\begin{aligned}\|\mathbf{A}\|_\infty &= \max_{\|\vec{v}\|_\infty=1} \|\mathbf{A}\vec{v}\|_\infty \\ &= \max_i \sum_{j=1}^m |a_{ij}| \ \text{ (maximum row sum)}\end{aligned}$$

Example

$$\mathbf{A} = \begin{bmatrix} 1 & -2 & -5 \\ -3 & 6 & 0 \\ 2 & -3 & 5 \end{bmatrix}$$
$$\|\mathbf{A}\|_\infty = \max\{1+2+5, 3+6+0, 2+3+5\} = 10$$

3. $l_2$-norm

$$\begin{aligned}\|\mathbf{A}\|_2 &= \max_{\|\vec{v}\|_2=1} \|\mathbf{A}\vec{v}\|_2 \\ &= \sqrt{\lambda_{\max}\left(\mathbf{A}^\top\mathbf{A}\right)} \\ &= S_{\max}(A) \text{ (maximum singular value of } \mathbf{A}\text{)}\end{aligned}$$

This is also called the spectral norm.

## §22.4 Error Bound of FDM

Consider the BVP

$$\begin{cases} \frac{d^2y}{dx^2} = f(x) \\ y(a) = y_a, \quad y(b) = y_b \end{cases} \tag{*}$$

Mesh points of FDM are

$$x_i = a + ih \text{ for } i = 1, \dots, N-1, \quad h = \frac{b-a}{N}$$

1. First write down errors at $x_i$, $e_i = y(x_i) - y_i$ for $i = 1, \dots, N-1$ where $y(x_i)$ is the exact solution at $x_i$ and $y_i$ is the numerical estimate for $y(x_i)$. If we write down these in the vector

form,

$$\begin{aligned}\vec{e} &= \vec{y}_{\text{exact}} - \vec{y}_{\text{est}} \\ &= \begin{bmatrix} y(x_1) \\ y(x_2) \\ \vdots \\ y(x_{N-1}) \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{N-1} \end{bmatrix}\end{aligned}$$

2. Our goal is to find an error bound for $\vec{e} : \|\vec{e}\| \le Ch^p$ ($C > 0$ constant, $p > 0$) and identify how large $p$ is (order of accuracy). Recall that the numerical estimate $\vec{y}$ using FDM can be obtained by solving

$$\mathbf{A}\vec{y}_{\text{est}} = \vec{f}_B$$

Let $\vec{\tau} \coloneqq \mathbf{A}\vec{y}_{\text{exact}} - \mathbf{A}\vec{y}_{\text{est}}$. Then

$$\begin{aligned}\vec{\tau} &= \mathbf{A}\left(\vec{y}_{\text{exact}} - \vec{y}_{\text{est}}\right) \\ &= \mathbf{A}\vec{e} \\ \vec{e} &= \mathbf{A}^{-1}\vec{\tau}\end{aligned}$$

So, if $\|\vec{\tau}\| \le C_\tau h^p$ for some $C_\tau > 0$, $p > 0$ then

$$\begin{aligned}\|\vec{e}\| &= \|\mathbf{A}^{-1}\vec{\tau}\| \\ &\le \|\mathbf{A}^{-1}\|\|\vec{\tau}\| \qquad (**) \\ &\le C_\tau \|\mathbf{A}^{-1}\| h^p\end{aligned}$$

We need a bound for $\vec{\tau}$, which will be our next step.

3. Since $\vec{\tau} = \mathbf{A}\vec{y}_{\text{exact}} - \mathbf{A}\vec{y}_{\text{est}} = \begin{bmatrix}\tau_1 & \tau_2 & \dots & \tau_{N-1}\end{bmatrix}^\top$ and

$$\mathbf{A} = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 0 & \\ 1 & -2 & 1 & 0 & \dots & 0 & \\ 0 & 1 & -2 & 1 & \dots & 0 & \\ \vdots & & & & & & \\ 0 & 0 & 0 & \dots & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -2 \end{bmatrix}$$

Then

$$\begin{aligned}\tau_i &= \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} - \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \\ &\approx \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2} - f(x_i) \qquad (+)\end{aligned}$$

Here this approximation step follows from $\frac{d^2y}{dx^2}\big|_{x_i} = f(x_i)$ from the ODE and $\frac{d^2y}{dx^2}\big|_{x_i} \approx \frac{y_{i+1}-2y_i+y_{i-1}}{h^2}$ from the centered difference formula. Now, we use the Taylor series expansions at $y(x_{i+1}) = y(x_i + h)$ and $y(x_{i-1}) = y(x_i - h)$ at $x = x_i$. By plugging these expansion into (+) and using $y''(x_i) = f(x_i)$ we have

$$\tau_i = \frac{h^2}{12}\frac{d^4y(x_i)}{dx^4} + \mathcal{O}(h^4)$$

So if $\frac{d^4y}{dx^4}$ is continuous, then $\frac{d^4y}{dx^4}$ is bounded or $[a, b] \implies |\tau_i| \le C_i h^2 \implies \|\vec{\tau}\| \le C_\tau h^2$ for some $C_\tau$.

# §23 Lec 23: Mar 9, 2022

## §23.1 Error Bound of FDM (Cont'd)

4. Going back to (**) in 2) (from last lecture)

$$\|\vec{e}\| \leq \|\mathbf{A}^{-1}\|\|\vec{\tau}\|$$

Also, $\|\mathbf{A}^{-1}\|$ is bound, i.e., $\|\mathbf{A}^{-1}\| \leq C_{\mathbf{A}}$ for some constant $C_{\mathbf{A}} > 0$. Thus, when we use FDM using the centered approximation formula for $\frac{d^2y}{dx^2}$, we have $\|\vec{\tau}\| \leq C_\tau h^2$

$$\implies \|\vec{e}\| \leq C_{\mathbf{A}} C_\tau h^2 = Ch^2$$

with $C = C_{\mathbf{A}} C_\tau$.

> **Remark 23.1.** If we use a *pth* order difference formula to approximate derivatives,
>
> $$\|\vec{\tau}\| \leq C_\tau h^p \implies \|\vec{e}\| \leq Ch^p$$

## §23.2 Iterative Methods for Solving Linear Systems

Motivation: Recall that we need to solve a linear system $\mathbf{A}\vec{x} = \vec{b}$ in FDM for BVPs. The direct methods based on computing $\mathbf{A}^{-1}$ or apply Gaussian elimination require a lot of computations if the matrix size is big (e.g., for an $n \times n$ matrix $\mathbf{A}$, Gaussian Elimination requires $\Omega(n^3)$ of operations). There are iterative methods for showing large-scale linear systems more efficiently.

- Idea of iterative methods:

  Transform $\mathbf{A}\vec{x} = \vec{b}$ into a simpler systems, for example into $\mathbf{D}\tilde{x} = \tilde{b}$ where $\mathbf{D}$ is diagonal matrix or of special structure so that $\mathbf{D}\tilde{x} = \tilde{b}$ can be easily solved.

- Procedure of iterative methods:

  i) Start with initial guess $\vec{x}^{(0)}$ of the solution

  ii) Use some process depending on $\mathbf{A}, \vec{b}$ and approximates at $k$ steps $\vec{x}^{(k)}, \vec{x}^{(k-1)}, \dots, \vec{x}^{(1)}, \vec{x}^{(0)}$ to create a sequence of approximates $\vec{x}^{(k+1)}$

  iii) Our goal is $\vec{x}^{(k)} \to \vec{x}$ as $k \to \infty$.

## §23.3 Jacobi's Method

Let a linear system, $\mathbf{A}\vec{x} = \vec{b}$ where $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\vec{x}, \vec{b} \in \mathbb{R}^n$ given as follows

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

$$\implies \begin{cases} a_{11}x_1 = b_1 - (a_{12}x_2 + \ldots + a_{1n}x_n) \\ a_{22}x_2 = b_2 - (a_{21}x_1 + a_{23}x_3 + \ldots + a_{2n}x_n) \\ \vdots \\ a_{nn}x_n = b_n - (a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn-1}x_{n-1}) \end{cases}$$

These equations motivate us the following procedure

$$\begin{aligned} x_1^{(k)} &= \frac{1}{a_{11}} \left( b_1 - \left( a_{12}x_2^{(k-1)} + \ldots + a_{1n}x_n^{(k-1)} \right) \right) \\ x_2^{(k)} &= \frac{1}{a_{22}} \left( b_2 - \left( a_{21}x_1^{(k-1)} + a_{23}x_3^{(k-1)} + \ldots + a_{2n}x_n^{(k-1)} \right) \right) \\ &\vdots \\ x_n^{(k)} &= \frac{1}{a_{nn}} \left( b_n - \left( a_{n1}x_1^{(k-1)} + a_{n2}x_2^{(k-2)} + \ldots + a_{nn-1}x_{n-1}^{(k-1)} \right) \right) \end{aligned}$$

Let's set

$$\mathbf{D} = \begin{bmatrix} a_{11} & & & 0 \\ & a_{22} & & \\ & & \ddots & \\ 0 & & & a_{nn} \end{bmatrix}$$

$$\mathbf{L} = \begin{bmatrix} 0 & & & & 0 \\ a_{21} & 0 & & & \\ & & 0 & & \\ \vdots & \ddots & \ddots & & \\ a_{n1} & \dots & & a_{nn-1} & 0 \end{bmatrix}, \mathbf{U} = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ & 0 & \dots & a_{2n} \\ & & \ddots & a_{n-1n} \\ 0 & & & 0 \end{bmatrix}$$

then the iterative formula for the Jacobi's method is

$$\begin{aligned} \mathbf{D}\vec{x}^{(k)} &= \vec{b} - (\mathbf{L} + \mathbf{U})\, \vec{x}^{(k-1)} \\ \vec{x}^{(k)} &= \mathbf{D}^{-1}\vec{b} - \mathbf{D}^{-1}\,(\mathbf{L} + \mathbf{U})\, \vec{x}^{(k-1)} \end{aligned}$$

Note that in order to apply the Jacobi's method, it requires $\mathbf{D}$ to be invertible, so all the diagonal entries of $\mathbf{A}$ need to be nonzero.

## §23.4 Gauss-Seidel Method

Idea: Similar to the Jacobi's method, but when we create $x_i^{(k)}$, we use the updated values $x_1^{(k)}, \ldots, x_{i-1}^{(k)}$ instead of using $x_1^{(k-1)}, \ldots, x_{i-1}^{(k-1)}$. Thus, we need the following equations at time step $k$.

$$\begin{cases} a_{11}x_1^{(k)} + a_{12}x_2^{(k-1)} + \ldots + a_{1n}x_n^{(k-1)} = b_1 \\ a_{21}x_1^{(k)} + a_{22}x_2^{(k)} + \ldots + a_{2n}x_n^{(k-1)} = b_2 \\ \vdots \\ a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + \ldots + a_{nn}x_n^{(k)} = b_n \end{cases}$$