Math 104A - Intro to Numerical Analysis

Numerical Solution of ODE

Jea-Hyun Park

University of California Santa Barbara

Fall 2022

Numerical solution of ODE

Numerical solution of ODE

Intro

Problem of interest

Given $\vec{f} : \mathbb{R}^{1+d} \to \mathbb{R}^d$, and $\vec{x}_0 \in \mathbb{R}^d$, find $\vec{x} : I \to \mathbb{R}^d$, where $t_0 \in I \subset \mathbb{R}$ (often $I = [0, T]$) satisfying

$$\dot{\vec{x}}(t) = \vec{f}(t, \vec{x}(t)) \ \ (t \in I), \quad \vec{x}(t_0) = \vec{x}_0$$

Example: (Lorenz equation; $d = 3$)

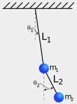
$$\vec{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} \text{ and } f(t, x, y, z) = \begin{bmatrix} \sigma(y - x) \\ x(\rho - z) - y \\ xy - \beta z \end{bmatrix}$$

If we set $\sigma = 1, \rho = \frac{1}{9}, \beta = 2$.

$$\begin{cases} x_t = y - x, \\ y_t = -xz + \frac{1}{9}x - y, \\ z_t = xy - 2z, \end{cases} \qquad \begin{bmatrix} x(0) \\ y(0) \\ z(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{1}$$

- $\dot{(\ )}$ denotes time derivative $\frac{\mathrm{d}}{\mathrm{dt}}(\ )$.
- $\vec{f}$ is called the **slope function**.
- The first piece is called ordinary differential equation (**ODE**) while the second **initial condition**, and altogether an initial value problem (**IVP**).
- f is independent of t in this example, but may depend on time in general.

Problem of interest



Plan

- We mainly focus on one dimensional case ($d = 1$). However, most of the important concepts and intuition are readily extended to higher dimensions (assuming proficiency in vector calculus).

Problem of interest (IVP)

$$\begin{cases} \dot{x} = f(t, x) \\ x(t_0) = x_0 \end{cases}$$

- ODE (more or less synonymous to dynamical system) is a rather general model for physics, biology, etc, anything that depends on time smoothly.
- Since the solution is a function of t (time), it is often called a **trajectory**.

Before we begin

- Some answers to "Would you think of a more intuitive, informal description of Lipschitz function?"
  - It's very similar to the contraction mapping theorem however the L is not restricted in a closed interval.
  - Not sure
  - no i would not come up with something more intuitive. if i dont fully understand it
  - I think the rate of change of the function cannot exceed the rate of change of the domain, you will get a infinite L in that way. (L looks like a magnifying function that takes the difference between two inputs of f)
  - bounding slope through outputs
  - lipschitz function isn't infinitely steep at one point
  - The output values of a function do not have any spikes or radical changes.
  - bounded derivative!

Before we begin

- Some answers to "How would you choose a method? What would you consider?"
  - I may be go with the most stable one with is implicit Euler.
  - Midpoint rule, it seems the most reliable.
  - i would choose implicit Euler method just since it looks easier. right now it's hard to tell the difference between all of them
  - i like to choose the explicit Euler method because i think unknown is unknown, i don't want to use xn+1 when i am actually finding it
  - consider the function itself and which method would be simplest to use
  - i would say depending on how big the domain is for choosing which method to use?
  - My choice of method would depend on the accuracy, the data i already have, and the cost in terms of time and $ when it comes to computation.

Existence and uniqueness of exact solution

$$\begin{cases} \dot{x} = f(t, x) \\ x(t_0) = x_0 \end{cases} \tag{IVP}$$

Theorem (Existence and uniqueness 1)

If f is continuous on a rectangle centered at (t_0, x_0), $D = \{(t, x) \ : \ |t - t_0| \leq \alpha, |x - x_0| \leq \beta\}$, then (IVP) has a solution on $(t_0 - r, t_0 + r)$, where $r = \min(\alpha, \beta/M)$ and $M = \max_{(t,x)\in D} |f(t, x)|$. If, in addition, $\partial f/\partial x$ is continuous on D, then the solution is unique.

Example

Verify that an IVP $x'(t) = x^{2/3}$ subject to $x(0) = 0$ has a solution around $t = 0$, but it is not unique.

- Are you trying to find something that exists?
- If so, does it stay the same every time you find it?
- We don't prove existence theorem
- Don't get overwhelmed by the theorem, in particular, by its details. Focus on the big picture to begin with.
- In words, "if slope function is nice, the system evolves deterministically at least for a short time."

Theorem (Existence and uniqueness 2)

If f is continuous on $[a, b] \times \mathbb{R}$ satisfies the Lipschitz condition in the second variable, x, i.e., there is $L > 0$ such that for all $t \in [a, b]$,

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

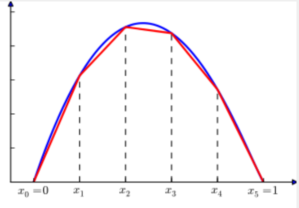
the (IVP) has a unique solution on $[a, b]$.

Remark (Continuous, Lipschitz continuous, continosuly differentiable functions of one variable)

Note that the following inclusions, where UC (nonstandard notation) means uniformly continuous functions,

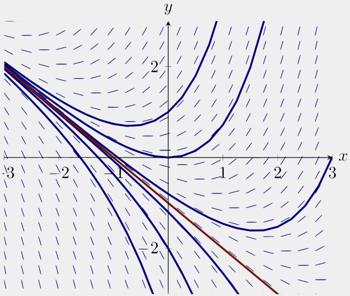
$$C^1[a, b] \subset \mathrm{Lip}[a, b] \subset UC[a, b] = C[a, b].$$

- To make the statement true, we end up needing to classify functions finer and finer.
- **Subjective question**: Lipschitz functions are very important class. Would you come up with a more intuitive, informal description?

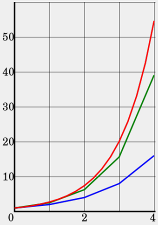


What does a numerical solution look like?

t_0	t_1	t_2	t_3	$\cdots$
x_0	x_1	x_2	x_3	$\cdots$



(A) Slope field



(B) Solutions of $x' = x$, $x(0) = x_0$. Euler (blue, bottom), Midpoint (green, middle), True (red, top)

- A numerical solution is a list of point values.
- (A) Each curve is a solution to IVP with a different initial value.
- (B) For each IVP, you have different numerical solutions depending on the method used.

Numerical solution of ODE

Taylor-series method

Taylor-series method

Setting/Notation

- Final time: T
- Uniform time steps: $h = (T - t_0)/N$ (N is #time steps), $t_n = t_0 + nh$ $(n = 0, 1, \cdots, N)$
- x_n: numerical solution at t_n. We hope/expect $x_n \approx x(t_n)$.

How to approximate the next step computed? $\rightarrow$ Taylor series
To compute $x(t+h)$, take a few terms from

$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \frac{h^3}{3!}x'''(t) + \frac{h^4}{4!}x^{(4)}(t) + \cdots$$

Example: 4th order Taylor method

$$\begin{cases} x'(t) = f(t,x) = \cos t - \sin x + t^2 \\ x(-1) = 3 \end{cases}$$

Numerical example desired.

- Problem of interest

$$\begin{cases} \dot{x} = f(t,x) \\ x(t_0) = x_0 \end{cases}$$

- Note carefully $x_n \neq x(t_n)$ in general.
- Taylor-series method is hard to summarize as a neat formula.

Error of Taylor-series method

For example, if the method include up to 3rd order term, the error is of 4th order.

$$\underbrace{x(t+h)}_{\text{target}} - \underbrace{x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \frac{h^3}{3!}x'''(t) + \frac{h^4}{4!}x^{(4)}(t)}_{\text{approximation}} = \frac{h^5}{5!}x^{(5)}(\xi)$$

Some standard one-step method of non-Taylor type

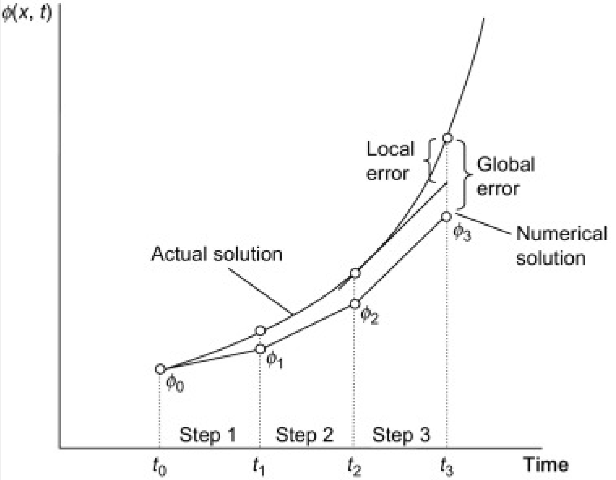
- **Explicit Euler method**: (take first two terms from Taylor series.) $x_{n+1} = x_n + hf(t_n, x_n)$
- **Implicit Euler**: $x_{n+1} = x_n + hf(t_{n+1}, x_{n+1})$
- **Midpoint rule**: $x_{n+1} = x_n + hf\left(t_n + \frac{h}{2}, \frac{1}{2}(x_n + x_{n+1})\right)$
- **Trapezoidal rule**: $x_{n+1} = x_n + \frac{h}{2}\left(f(t_n, x_n) + f(t_{n+1}, x_{n+1})\right)$

- **Question**: Guess the order of accuracy.
- Explicit Euler method is actually a Taylor-series method.
- Input of midpoint rule is the center of rectangle.
- Trapezoidal rule is actually related to trapezoidal (quadrature) rule.
- **Subjective question**: If you have an IVP, how would you choose a method? What would you consider?

Errors in a numerical solution to an IVP

1. **Local truncation error (LTE)** : errors caused by including only finite number of calculations out of an exact procedure assuming the current data is exact.
2. **Local roundoff error**: errors caused by limited precision of computers.
3. **Global truncation error**: accumulation of all LTE. Usually, global error is of one lower order than that of LTE since errors accumulate.
4. **Global roundoff error**: accumulated roundoff errors.
5. **Total error**: sum of the global truncation errors and global roundoff errors.

- 'global error' usually means global truncation error. But people normally say the full name for 'local truncation error.'
- Truncation errors are inherent in the method chosen, and quite independent of the roundoff errors.
- Roundoff errors depend on the computer environment.


$\phi(x, t)$
Local error
Global error
Numerical solution
Actual solution
ϕ_0
ϕ_1
ϕ_2
ϕ_3
Step 1
Step 2
Step 3
t_0
t_1
t_2
t_3
Time

Pros and cons of Taylor-series method

Pros

- Conceptually easy.
- High order methods are obtained easily (just add more terms).
- Inspires other methods.

Cons

- Require a high regularity on the slope function.
- Preliminary analytic work must be done. (During this stage, human-made error can be a disaster.)

Numerical solution of ODE

Runge-Kutta method

Runge-Kutta method

Motivation: In Taylor method, we need to find derivatives prior to coding. Can we reduce the human involvement?

Example: Derive a second order RK method (Board work).
Temporary notation (omitted evaluation) $x = x(t)$ and $f = f(t, x)$ (similarly for $f_t, f_x, \cdots$)

1. Advance one step using Taylor's method.
$$x(t+h) = x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \frac{h^3}{3!}x'''(t) + \frac{h^4}{4!}x^{(4)}(t) + \cdots$$
2. Replace derivatives of x with those (partial derivatives) of f. For this, assume $x(t)$ solves the ODE $x'(t) = f(t, x(t))$.
3. Replace partials of f with only evaluations of f using Taylor series of $f(t+h, x+hf)$ in two variables.
4. Organize it.

- This leads to **Heun's method**.

$$\begin{aligned} &x(t+h) \\ &= x(t) + \frac{1}{2}(F_1 + F_2), \end{aligned}$$

where

$$\begin{cases} F_1 = hf(t, x) \\ F_2 = hf(t+h, x+F_1). \end{cases}$$

Runge-Kutta method

Heun's method is not the only such methods. Every time we choose appropriate numbers for α, β, w_1, w_2 below, we have a method of order 2 (i.e., order 3 for one step):

$$\begin{aligned} x(t+h) &= x + w_1 hf + w_2 hf(t+\alpha h, x+\beta hf) + \mathcal{O}\left(h^3\right) \\ &= x + w_1 hf + w_2 h\left[f + \alpha hf_t + \beta hff_x\right] + \mathcal{O}\left(h^3\right) \end{aligned}$$

Recall Taylor expansion of x requires

$$x(t+h) = x + \frac{1}{2}hf + \frac{1}{2}h\left[f + hf_t + hff_x\right] + O\left(h^3\right).$$

We have a method of order 2 if

$$w_1 + w_2 = 1, \quad w_2\alpha = \frac{1}{2}, \quad w_2\beta = \frac{1}{2}.$$

$w_1 = 0, w_2 = 1, \alpha = \beta = \frac{1}{2}$ yield **modified Euler** method.

Butcher's tableau for Runge-Kutta method

The previous observation motivates Butcher's tableau for RK method. An RK method can be encapsulated by

$$
\begin{array}{c|cccc}
c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\
c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\
\hline
 & b_1 & b_2 & \cdots & b_s
\end{array}
=
\begin{array}{c|c}
\vec{c} & A \\
\hline
 & \vec{b}^T
\end{array}
$$

Previous examples read:

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

Heun's method — modified Euler

Activity: Recover modified Euler from the tableau.

- $\vec{b} \leftrightarrow$ weights of mid-stage slopes for the final advance (w's)
- $\vec{c} \leftrightarrow$ time subgrid for stages (α)
- $A \leftrightarrow$ inner weights (β) for x as an input for mid-stage slopes.
- To yield a meaningful method, $\vec{b}, \vec{c}, A$ must satisfy some requirements.
- We don't pursue detailed investigations on RK methods.

"best" Runge-Kutta method

Irregular accuracy of RK

# function eval.	1	2	3	4	5	6	7	8
Max order of accuracy	1	2	3	4	4	5	6	6

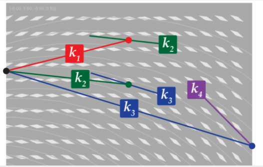
An important example: *The (classical)* RK4

$$\begin{cases} F_1 = hf(t,x) \\ F_2 = hf\left(t+\frac{1}{2}h, x+\frac{1}{2}F_1\right) \\ F_3 = hf\left(t+\frac{1}{2}h, x+\frac{1}{2}F_2\right) \\ F_4 = hf\left(t+h, x+F_3\right) \end{cases}$$

$$x(t+h) = x(t) + \frac{1}{6}\left(F_1 + 2F_2 + 2F_3 + F_4\right)$$

Activity: Construct Butcher's tableau for the RK4.

- Runge-Kutta methods from a slope field angle



- **Subjective question**: How would you summarize Runge-Kutta method in an intuitive language?

Numerical solution of ODE

Multistep Methods

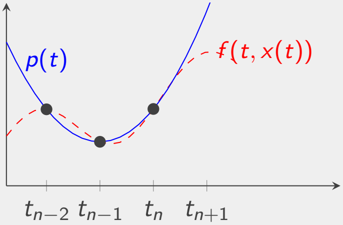
Multistep methods

Single-step methods: Taylor and RK methods use only the data at the most recent time grid: find x_{n+1} given x_n at t_n.

Multistep methods: The methods use more history: find x_{n+1} given $x_n, x_{n-1}, \cdots, x_{n-k+1}$ at $t_n, t_{n-1}, \cdots, t_{n-k+1}$.

- Don't get confused with 'stages.' RK4, for example, uses four different mid-stage slopes which the method *computes* but not *given*.

Multistep methods - Adam-Bashforth



Idea: use interpolation and quadrature ($k = 3$, uniform grid)

1. Suppose x solves the ODE, $x' = f(t, x)$, and integrate

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} x'(t)dt = x(t_n) + \int_{t_n}^{t_{n+1}} f(t, x(t))dt$$

2. Replace $f(t, x(t))$ with its polynomial interpolation $p(t)$ at $(t_{n-2}, f_{n-2}), (t_{n-1}, f_{n-1}), (t_n, f_n)$, where $f_j := f(t_j, x(t_j))$.
3. Obtain a method by labeling $x_j \approx x(t_j)$. It should be clear that

$$x_{n+1} = x_n + Af_n + Bf_{n-1} + Cf_{n-2}$$

- Question: What is the degree of $p(t)$?
- Question: Write out $p(t)$.

Example: Derive 3 step Adam-Bashforth method (AB3)

$$x_{n+1} = x_n + h\left(\frac{23}{12}f_n - \frac{16}{12}f_{n-1} + \frac{5}{12}f_{n-2}\right)$$

- **Subjective question**: What can be a quick sanity check?

Order of Adam-Bashforth methods

Theorem

LTE of k–step AB method is of order $k+1$, that is,
$|x_{n+1} - x(t_{n+1})| = \mathcal{O}\left(h^{k+1}\right)$ *as* $h \to 0$.

Proof.

Board Work. $\square$

$$x_{n+1} = x_n + h\left(\frac{23}{12} f_n - \frac{16}{12} f_{n-1} + \frac{5}{12} f_{n-2}\right)$$

- Recall: Let $x_0, x_1, \cdots, x_n \in [a, b]$ be distinct nodes, $f \in C^{n+1}[a, b]$, and $p \in \Pi_n$ interpolating f at the nodes. For each $x \in [a, b]$, there is $\xi_x \in (a, b)$ such that

$$f(x) - p(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{i=0}^{n} (x - x_i)$$