

Math 104A - Intro to Numerical Analysis

NUMERICAL SOLUTION OF ODE

JEAN-HYUN PARK

UNIVERSITY OF CALIFORNIA SANTA BARBARA

FALL 2022



Numerical solution of ODE

Numerical solution of ODE

Intro

PROBLEM OF INTEREST

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

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

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

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

- $(\dot{})$ denotes time derivative $\frac{d}{dt}()$.
- \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**.

EXISTENCE AND UNIQUENESS OF EXACT SOLUTION

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

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

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 $t \in [a, b]$,

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

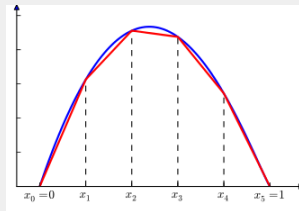
the (IVP) has a unique solution on $[a, b]$.

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

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

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

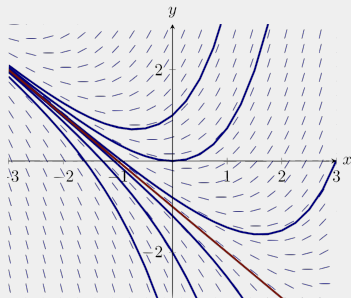
- To make the statement true, we end up needing to classify functions finer and finer.
- **Subjective question:** Lipschitz functions are very important class. Would you come up with a more intuitive, informal description?



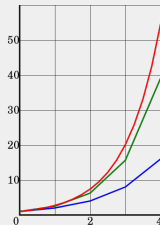
CONCRETE PICTURES OF WHAT WE WILL DO

What does a numerical solution look like?

t_0	t_1	t_2	t_3	\dots
x_0	x_1	x_2	x_3	\dots



(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),
 $t_n = t_0 + nh$ ($n = 0, 1, \dots, N$)
- 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

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

Example: 4th order Taylor method

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

- 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.

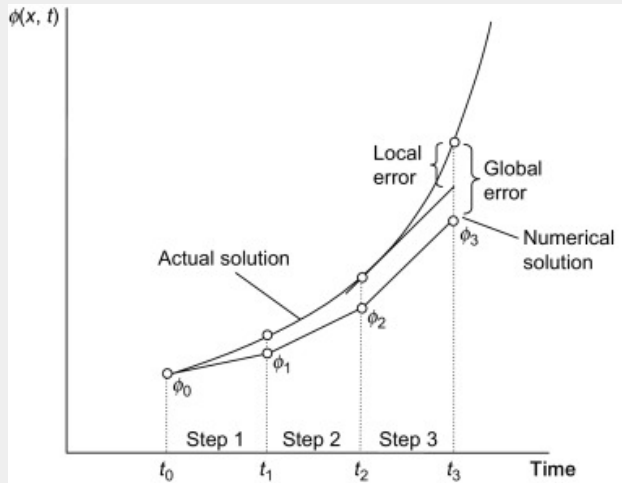
$$\underbrace{x(t+h)}_{\text{target}} - \underbrace{x(t) + hx'(t) + \frac{h^2}{2!}x''(t) + \frac{h^3}{3!}x'''(t) + \frac{h^4}{4!}x^{(4)}(t)}_{\text{approximation}} = \frac{h^5}{5!}x^{(5)}(\xi)$$

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:** $x_{n+1} = x_n + \frac{h}{2}(f(t_n, x_n) + f(t_{n+1}, x_{n+1}))$
- **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.



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

Before we begin

- Putting off ComHW2 to Thu (Dec 1) and moving Fri OH 4:10-5:10PM to Wed 12:00-1:00PM.
- Join ESCI!! You can also leave comments on iclicker.
- Some answers to “How would you choose a method for IVPs?” (11AM)
 - ▶ Based on initial conditions and how accurate I must be. But if I can get it done easily I’m using explicit euler if else I’ll se trapezoida
 - ▶ Depending on the context and the goal of accuracy and computation power needed.
 - ▶ I would use the explicit euler method to solve. it is the quickest/simplest one to solve as it requires the least amount of computation.
 - ▶ If I have an IVP, I would see the function form and decide the solving method. To see if it is separable or replaceable?
 - ▶ how difficult x_{n+1} is to calculate in terms of f like if it is easy or hard to find the root for the implicit versions
 - ▶ it would depend on how annoying f is. the bigger it is the harder to solve for x_{n+1}
 - ▶ It depends on what what nodes are provided to you.

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, \dots)

1. Advance one step using Taylor's method.

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

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):

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

Recall Taylor expansion of x requires

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

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:

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

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

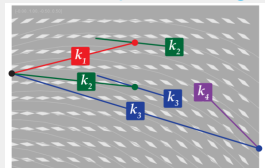
# function eval.	1	2	3	4	5	6	7	8
Max order of accuracy	1	2	3	4	4	5	6	6

An important example: *The (classical) RK4*

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

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?

OPTIONAL COMPUTATIONAL PROJECT

Simulate the Lorenz system with several method and compare the trajectories. (explicit Euler, implicit Euler, RK4)

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

$$\begin{cases} \dot{x}_t = y - x, \\ \dot{y}_t = -xz + \frac{1}{9}x - y, \\ \dot{z}_t = xy - 2z, \end{cases} \quad \begin{bmatrix} x(0) \\ y(0) \\ z(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (2)$$

- Consider changing the initial condition. I have chosen it randomly.

Numerical solution of ODE

Multistep Methods

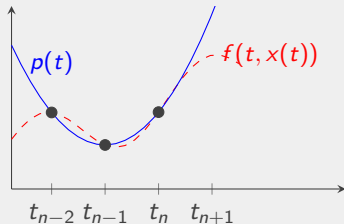
MULTISTEP METHODS

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 $x_n, x_{n-1}, \dots, x_{n-k+1}$ at $t_n, t_{n-1}, \dots, t_{n-k+1}$.

- 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, $|x_{n+1} - x(t_{n+1})| = \mathcal{O}(h^{k+1})$ as $h \rightarrow 0$.

Proof.

Board Work. □

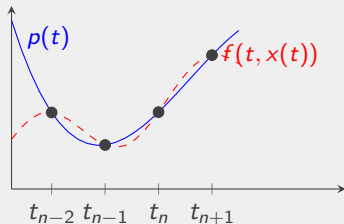
■ Recall: Let

$x_0, x_1, \dots, x_n \in [a, b]$ 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

$$f(x) - p(x) =$$

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

MULTISTEP METHODS - ADAM-MOULTON



Same 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 $(t_{n-2}, f_{n-2}), (t_{n-1}, f_{n-1}), (t_n, f_n), (t_{n+1}, f_{n+1})$. ($f_j := f(t_j, x(t_j))$)

3. Obtain a method by labeling $x_j \approx x(t_j)$:

$$x_{n+1} = x_n + \textcolor{red}{A}f_{n+1} + Bf_n + Cf_{n-1} + Df_{n-2}$$

■ Question: What is the degree of $p(t)$?

Example: 3 step Adam-Moulton method (AM3)

$$x_{n+1} = x_n + h \left(\frac{9}{24} f(t_{n+1}, x_{n+1}) + \frac{19}{24} f(t_n, x_n) - \frac{5}{24} f(t_{n-1}, x_{n-1}) + \frac{1}{24} f(t_{n-2}, x_{n-2}) \right)$$

- Question: Is this possible?
- Question: Guess the order of accuracy.

ADAM-MOULTON METHOD EQUIPPED WITH AN ITERATIVE METHOD

Issue: We need to know x_{n+1} to compute x_{n+1} ! (Implicit)

$$x_{n+1} = x_n + h \left(\frac{9}{24} f(t_{n+1}, x_{n+1}) + \frac{19}{24} f(t_n, x_n) - \frac{5}{24} f(t_{n-1}, x_{n-1}) + \frac{1}{24} f(t_{n-2}, x_{n-2}) \right)$$

Idea: Recast the method as a fixed point problem.

1. Relabel what we are after, x_{n+1} , say, z to emphasize that it is the real unknown.
2. Treat everything else, which is already known, as data and lump it into a single function, say, ϕ .
3. Find the solution, z such that

$$z = \phi(z) := C_1 h f(t_{n+1}, z) + C_2.$$

That is, $z_{m+1} = \phi(z_m)$ for $m = 0, 1, 2, \dots$.

ACCURACY OF ADAM-MOULTON METHOD

Theorem

LTE of k -step AM method is of order $k + 2$, that is,
 $|x_{n+1} - x(t_{n+1})| = \mathcal{O}(h^{k+2})$ as $h \rightarrow 0$.

Proof.

Board Work.



PREDICTOR-CORRECTOR METHOD

Issue: We need to know x_{n+1} to compute x_{n+1} ! (Implicit)

Idea: Make an explicit variant of Adam-Moulton method.

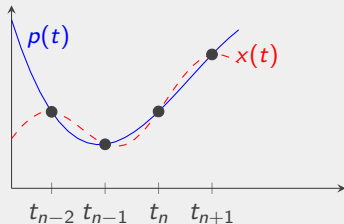
1. Choose one Adam-Bashforth method of one more step. (e.g, choose AB4 if AM3 is the main method).
2. Run the AB method (explicit) to *predict* the next x value, call it x_{n+1}^* .
3. Run the AM method using the predicted value on RHS in $f(t_{n+1}, x_{n+1}^*)$ instead of $f(t_{n+1}, x_{n+1})$ to *correct* the prediction.

That is, if AM3 is the main method,

$$x_{n+1} = x_n + h \left(\frac{9}{24} f(t_{n+1}, x_{n+1}^*) + \frac{19}{24} f(t_n, x_n) - \frac{5}{24} f(t_{n-1}, x_{n-1}) + \frac{1}{24} f(t_{n-2}, x_{n-2}) \right).$$

- The rationale behind this is to keep the same order of accuracy. Since AB4 involves four coefficients for $f_n, f_{n-1}, f_{n-2}, f_{n-3}$, our intuition says it is of order 4. On the other hand, AM3 involves four coefficients for $f_{n+1}, f_n, f_{n-1}, f_{n-2}$. Otherwise, you lose accuracy for no reason.

BACKWARD DIFFERENTIATION FORMULA (BDF)



Motivation: The methods of Adam's family are based on interpolating the slope functions. Can we obtain interpolating positions, x 's?

1. Given x_{n-2}, x_{n-1}, x_n at t_{n-2}, t_{n-1}, t_n resp., pretend to know x_{n+1} at t_{n+1} and find a polynomial $p(t)$ interpolating the four points.
2. Assume $x'(t) \approx p'(t)$, and argue $p'(t_{n+1}) \approx x'(t_{n+1}) = f_{n+1}$.

- Recall, from numerical differentiation, this approximation is really true if the solution is smooth enough.
- It literally looks backwards come up with a formula for the numerical differentiation.

BACKWARD DIFFERENTIATION FORMULA (BDF)

Examples:

$$\text{BDF1} \quad x_{n+1} - x_n = hf_{n+1}$$

$$\text{BDF2} \quad x_{n+1} - \frac{4}{3}x_n + \frac{1}{3}x_{n-1} = \frac{2}{3}hf_{n+1}$$

$$\text{BDF3} \quad x_{n+1} - \frac{18}{11}x_n + \frac{9}{11}x_{n-1} - \frac{2}{11}x_{n-2} = \frac{6}{11}hf_{n+1}$$

- BDF methods are all implicit. Recall $f_{n+1} := f(t_{n+1}, x_{n+1})$.
- In some sense, BDF methods are “dual” to Adam’s family: BDF minimizes #slope evaluations while Adam’s #history of x ’s
- BDFs are considered good option for ‘stiff’ problem. (This notion makes sense only in high dimensions.)

LINEAR MULTISTEP METHOD

$$\sum_{i=1}^{k+1} a_i x_i = h \sum_{i=1}^{k+1} b_i f_i$$

- Adam-Bashforth: $a_{k+1} = 1, a_k = -1$; all other a 's are zero; $b_{k+1} = 0$; other b 's are appropriately chosen
- Adam-Moulton: $a_{k+1} = 1, a_k = -1$; all other a 's are zero; $b_{k+1} \neq 0$; other b 's are appropriately chosen
- BDF: $b_{k+1} \neq 0$; other b 's are zero; all a 's are chosen appropriately
- In theory, we can make the most out of the degrees of freedom by tuning a 's and b 's to obtain as accurate method as possible. But we will see the method of the highest possible order is not a good option.
- There are many other linear multistep methods.
- **Question:** Guess how high the order of the method can be.