Stability Analysis: multistep methods (I)

single ODE $$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha.$$

Consider multistep method, $w_0 = \alpha$, $w_1 = \alpha_1$, $\cdots$, $w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$\begin{aligned} w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ &\quad + h\, F\left(t_j, w_{j+1}, w_j, \cdots, w_{j+1-m}\right), \quad x_j = a + j\, h. \end{aligned}$$

Stability Analysis: multistep methods (I)

single ODE $\quad \dfrac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha.$

Consider multistep method, $w_0 = \alpha$, $w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$\begin{aligned} w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ &\quad + h\, F\left(t_j, w_{j+1}, w_j, \cdots, w_{j+1-m}\right), \quad x_j = a + j\, h. \end{aligned}$$

- local truncation error

$$\textbf{LTE } \tau_{j+1}(h) \stackrel{def}{=} \frac{y(t_{j+1}) - (a_{m-1}\, y(t_j) + a_{m-2}\, y(t_{j-1}) + \cdots + a_0\, y(t_{j+1-m}))}{h} - F\left(t_j, y(t_{j+1}), y(t_j), \cdots, y(t_{j+1-m})\right).$$

Stability Analysis: multistep methods (I)

single ODE $\qquad \dfrac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha.$

Consider multistep method, $w_0 = \alpha$, $w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$\begin{aligned} w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ &\quad + h\, F\left(t_j, w_{j+1}, w_j, \cdots, w_{j+1-m}\right), \quad x_j = a + j\, h. \end{aligned}$$

- local truncation error

$$\textbf{LTE } \tau_{j+1}(h) \stackrel{def}{=} \frac{y(t_{j+1}) - (a_{m-1}\, y(t_j) + a_{m-2}\, y(t_{j-1}) + \cdots + a_0\, y(t_{j+1-m}))}{h} - F\left(t_j, y(t_{j+1}), y(t_j), \cdots, y(t_{j+1-m})\right).$$

Assumptions on F

- If $f \equiv 0$, then $F \equiv 0$.

$$\begin{aligned} &\left|F\left(t_j, u_{j+1}, u_j, \cdots, u_{j+1-m}\right) - F\left(t_j, \widehat{u}_{j+1}, \widehat{u}_j, \cdots, \widehat{u}_{j+1-m}\right)\right| \\ &\quad \le L\left(|u_{j+1} - \widehat{u}_{j+1}| + \cdots + |u_{j+1-m} - \widehat{u}_{j+1-m}|\right) \end{aligned}$$

Stability Analysis: multistep methods (II)

- *Definition*: **consistency**

$$\mathbf{lim}_{h\to 0}\mathbf{max}_{m\le j\le N}\,|\tau_j(h)| = 0, \quad \mathbf{lim}_{h\to 0}\mathbf{max}_{0\le j\le m-1}\,|y(t_j)-\alpha_j| = 0.$$

Stability Analysis: multistep methods (II)

- *Definition*: **consistency**

$$\mathbf{lim}_{h\to 0}\mathbf{max}_{m\le j\le N}\,|\tau_j(h)| = 0, \quad \mathbf{lim}_{h\to 0}\mathbf{max}_{0\le j\le m-1}\,|y(t_j)-\alpha_j| = 0.$$

- *Definition*: **convergence**

$$\mathbf{lim}_{h\to 0}\mathbf{max}_{0\le j\le N}\,|y(t_j)-w_j| = 0$$

Stability Analysis: multistep methods (II)

- *Definition*: **consistency**

$$\lim_{h\to 0}\max_{m\le j\le N}|\tau_j(h)| = 0, \quad \lim_{h\to 0}\max_{0\le j\le m-1}|y(t_j)-\alpha_j| = 0.$$

- *Definition*: **convergence**

$$\lim_{h\to 0}\max_{0\le j\le N}|y(t_j)-w_j| = 0$$

Stability is a much bigger issue

Stability Analysis: multistep methods (III)

single ODE $$\frac{dy}{dt} = f(t, y) = 0, \quad a \le t \le b, \quad y(a) = \alpha.$$

Multistep method with $w_0 = \alpha$, $w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$\begin{aligned} w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ &\quad + h\, F\left(t_j, w_{j+1}, w_j, \cdots, w_{j+1-m}\right), \\ &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m}. \quad (F \equiv 0) \end{aligned}$$

Stability Analysis: multistep methods (III)

single ODE $\quad \frac{dy}{dt} = f(t, y) = 0, \quad a \le t \le b, \quad y(a) = \alpha.$

Multistep method with $w_0 = \alpha$, $w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$\begin{aligned} w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ &\quad + h\, F\left(t_j, w_{j+1}, w_j, \cdots, w_{j+1-m}\right), \\ &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m}. \quad (F \equiv 0) \end{aligned}$$

- Assume $\alpha = \alpha_1 \cdots = \alpha_{m-1}$,

Stability Analysis: multistep methods (III)

single ODE $$\frac{dy}{dt} = f(t,y) = 0, \quad a \le t \le b, \quad y(a) = \alpha.$$

Multistep method with $w_0 = \alpha$, $w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$\begin{aligned} w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ &\quad + h\, F\left(t_j, w_{j+1}, w_j, \cdots, w_{j+1-m}\right), \\ &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m}. \quad (F \equiv 0) \end{aligned}$$

- Assume $\alpha = \alpha_1 \cdots = \alpha_{m-1}$,

Minimum requirements

- $w_{j+1} \equiv \alpha$ for all j.
- w_{j+1} remains close to α in finite precision.

Finite recurrence relations (I)

Given $w_0 = \alpha_0,\ w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$w_{j+1} = a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m}.$$

Finite recurrence relations (I)

Given $w_0 = \alpha_0,\ w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$w_{j+1} \quad = \quad a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m}.$$

- To solve for w_j for all j, **assume**

$$\frac{w_{k+1}}{w_k} = \lambda \quad \text{for all} \quad k.$$

- Recurrence becomes

$$\mathbf{P}(\lambda) \quad = \quad 0, \quad \mathbf{P}(\lambda) \stackrel{def}{=} \lambda^m - \left(a_{m-1}\, \lambda^{m-1} + a_{m-2}\, \lambda^{m-2} + \cdots + a_0\right).$$

- thus λ must be a root of $\mathbf{P}(\lambda)$.
- $\lambda = 1$ should be a root of $\mathbf{P}(\lambda)$.

Finite recurrence relations (II)

- Recurrence relation

$$w_{j+1} \quad = \quad a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m}.$$

- **characteristic polynomial**

$$\mathbf{P}(\lambda) \quad \stackrel{def}{=} \quad \lambda^m - \left(a_{m-1}\,\lambda^{m-1} + a_{m-2}\,\lambda^{m-2} + \cdots + a_0\right).$$

- If $\mathbf{P}(\lambda)$ has m **distinct** roots $\lambda_1, \cdots, \lambda_m$, then

$$w_j = c_1\,\lambda_1^j + c_2\,\lambda_2^j + \cdots + c_m\,\lambda_m^j, \quad j = 0, 1, \cdots m-1, m, \cdots$$

  for constants $c_1, c_2, \cdots c_m$ determined by the equations for $0 \leq j \leq m-1$.

Finite recurrence relations (III)

- Example recurrence relation

$$w_{j+1} \quad = \quad 3\, w_j - 2\, w_{j-1}. \quad (m = 2.)$$

- **characteristic polynomial**

$$\mathbf{P}(\lambda) \quad = \quad \lambda^2 - 3\,\lambda^1 + 2 = (\lambda - 1)(\lambda - 2).$$

- Roots of $\mathbf{P}(\lambda)$ are 1 and 2.
- recurrence solution

$$w_j = c_1 + c_2\, 2^j, \quad j = 0, 1, 2, 3, \cdots$$

where

$$w_0 = c_1 + c_2, \quad w_1 = c_1 + 2\, c_2, \quad \text{or}$$

$$c_1 = 2 w_0 - w_1, \quad c_2 = w_1 - w_0.$$

Finite recurrence relations (IV)

- Example recurrence relation

$$w_{j+1} \;=\; 2\,w_j - 1\,w_{j-1}. \quad (m = 2.)$$

- **characteristic polynomial**

$$\mathbf{P}(\lambda) \;=\; \lambda^2 - 2\,\lambda^1 + 1 = (\lambda - 1)^2.$$

- Roots of $\mathbf{P}(\lambda)$ are 1 and 1.
- recurrence solution

$$w_j = c_1 + j\,c_2, \quad j = 0, 1, 2, 3, \cdots$$

where

$$w_0 = c_1, \quad w_1 = w_1 - w_0.$$

Root conditions

Multistep method with $w_0 = \alpha$, $w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$\begin{aligned} w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ &\quad + h\, F\left(t_j, w_{j+1}, w_j, \cdots, w_{j+1-m}\right), \\ \mathbf{P}(\lambda) &= \lambda^m - \left(a_{m-1}\, \lambda^{m-1} + a_{m-2}\, \lambda^{m-2} + \cdots + a_0\right). \end{aligned}$$

Root conditions

Multistep method with $w_0 = \alpha,\ w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$
\begin{aligned}
w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\
&\quad + h\, F\left(t_j, w_{j+1}, w_j, \cdots, w_{j+1-m}\right), \\
\mathbf{P}(\lambda) &= \lambda^m - \left(a_{m-1}\, \lambda^{m-1} + a_{m-2}\, \lambda^{m-2} + \cdots + a_0\right).
\end{aligned}
$$

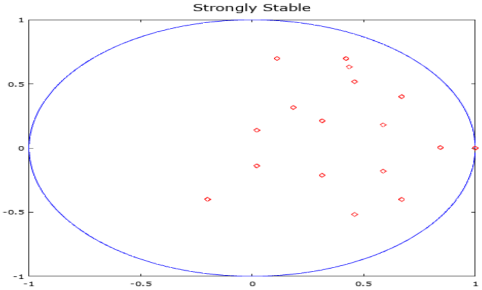
root condition: every root λ_i of $\mathbf{P}(\lambda)$ must satisfy $|\lambda_i| \leq 1$

Assume multistep method satisfies root condition.

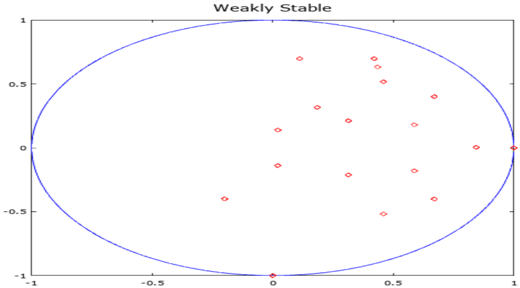
- **strongly stable**: $\lambda = 1$ is only root of $\mathbf{P}(\lambda)$ with magnitude 1.
- **weakly stable**: $\mathbf{P}(\lambda)$ has more than 1 distinct root with magnitude 1.

Otherwise method is **unstable.**

- **strongly stable**:



- **weakly stable**:



single ODE $$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha.$$

Theorem: Assume multistep method with $w_0 = \alpha$, $w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$\begin{aligned} w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ &\quad + h\, F\left(t_j, w_{j+1}, w_j, \cdots, w_{j+1-m}\right). \end{aligned}$$

single ODE $$\frac{dy}{dt} = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha.$$

Theorem: Assume multistep method with $w_0 = \alpha$, $w_1 = \alpha_1, \cdots, w_{m-1} = \alpha_{m-1}$,

- for $j = m-1, m, m+1, \cdots$

$$\begin{aligned} w_{j+1} &= a_{m-1} w_j + a_{m-2} w_{j-1} + \cdots + a_0 w_{j+1-m} \\ &\quad + h F\left(t_j, w_{j+1}, w_j, \cdots, w_{j+1-m}\right). \end{aligned}$$

Assume the method is **consistent**, then

- The method is stable $\iff$ it satisfies root condition $\iff$ it is convergent.

example fourth-order Adams-Bashforth method

- 4-step and fourth-order

$$w_{j+1} = w_j + h\,F\,(t_j, w_j, w_{j-1}, w_{j-2}, w_{j-3})$$

where $F\,(t_j, w_j, w_{j-1}, w_{j-2}, w_{j-3}) =$

$$\frac{h}{24}\,(55f(t_j, w_j) - 59f(t_{j-1}, w_{j-1}) + 37f(t_{j-2}, w_{j-2}) - 9f(t_{j-3}, w_{j-3}))\,.$$

example fourth-order Adams-Bashforth method

- 4-step and fourth-order

$$w_{j+1} \quad = \quad w_j + h\,F\left(t_j, w_j, w_{j-1}, w_{j-2}, w_{j-3}\right)$$

where $F\left(t_j, w_j, w_{j-1}, w_{j-2}, w_{j-3}\right) =$

$$\frac{h}{24}\left(55f(t_j, w_j) - 59f(t_{j-1}, w_{j-1}) + 37f(t_{j-2}, w_{j-2}) - 9f(t_{j-3}, w_{j-3})\right).$$

Solution: $m = 4$,

$$\mathbf{P}(\lambda) = \lambda^4 - \lambda^3 = \lambda^3\,(\lambda - 1).$$

Roots of $\mathbf{P}(\lambda)$ are $0, 0, 0, 1$, satisfying root condition: **strongly stable**.

example fourth-order Milne's method

- 4-step and fourth-order

$$w_{j+1} = w_{j-3} + h\,\widehat{F}\,(t_j, w_j, w_{j-1}, w_{j-2}, w_{j-3})$$

where

$$\widehat{F}\,(t_j, w_j, w_{j-1}, w_{j-2}, w_{j-3})$$
$$= \frac{4h}{3}\,(2f(t_j, w_j) - f(t_{j-1}, w_{j-1}) + 2f(t_{j-2}, w_{j-2}))\,.$$

example fourth-order Milne's method

- 4-step and fourth-order

$$w_{j+1} \quad = \quad w_{j-3} + h\,\widehat{F}\,(t_j, w_j, w_{j-1}, w_{j-2}, w_{j-3})$$

where
$$\widehat{F}\,(t_j, w_j, w_{j-1}, w_{j-2}, w_{j-3}) = \frac{4h}{3}\,(2f(t_j, w_j) - f(t_{j-1}, w_{j-1}) + 2f(t_{j-2}, w_{j-2}))\,.$$

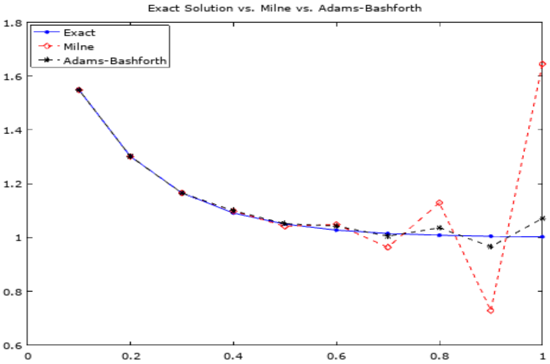
Solution: $m = 4$,

$$\mathbf{P}(\lambda) = \lambda^4 - 1 = (\lambda - 1)(\lambda + 1)(\lambda - \sqrt{-1})(\lambda + \sqrt{-1}).$$

Roots of $\mathbf{P}(\lambda)$ are $\pm 1, \pm\sqrt{-1}$, satisfying root condition:
weakly stable as all roots have magnitude 1.

Example: Adams-Bashforth vs. Milne

$$\begin{aligned} \frac{dy}{dt} &= -6y+6, \quad 0 \le t \le 1, \quad y(0)=2, \\ \text{exact solution } y(t) &= 1+e^{-6t}, \quad h=0.1. \end{aligned}$$

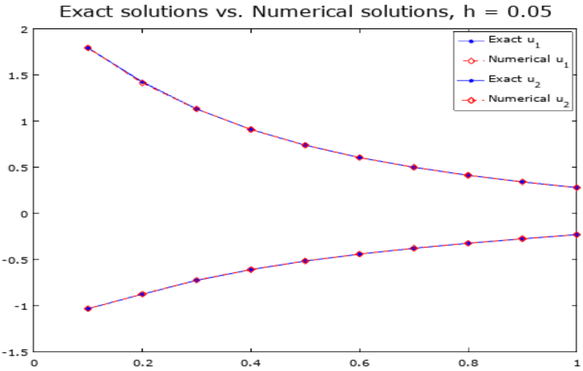


Example: Stiff ODEs (I), $h = 0.05$

$$\frac{d}{dt}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 9u_1 + 24u_2 + 5\cos t - \frac{1}{3}\sin t, \\ -24u_1 - 51u_2 - 9\cos t + \frac{1}{3}\sin t \end{pmatrix}, \quad \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 4 \\ 2 \end{pmatrix},$$

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} 2e^{-3t} - e^{-39t} + \frac{1}{3}\cos t \\ -e^{-3t} + 2e^{-39t} - \frac{1}{3}\cos t \end{pmatrix},$$

exact solution, zombie term e^{-39t}

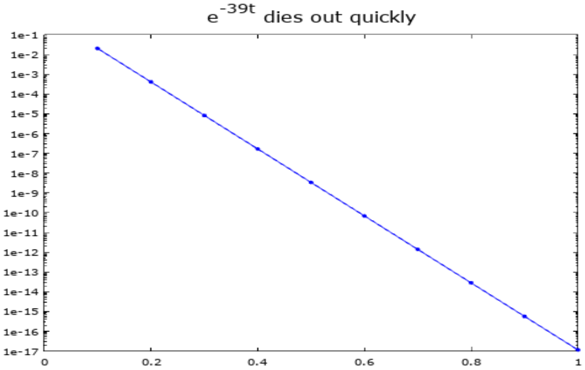


Transient e^{-39t}: little effect on solution, big effect on h

$$\frac{d}{dt}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 9u_1 + 24u_2 + 5\cos t - \frac{1}{3}\sin t, \\ -24u_1 - 51u_2 - 9\cos t + \frac{1}{3}\sin t \end{pmatrix}, \quad \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 4 \\ 2 \end{pmatrix},$$

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} 2e^{-3t} - e^{-39t} + \frac{1}{3}\cos t \\ -e^{-3t} + 2e^{-39t} - \frac{1}{3}\cos t \end{pmatrix},$$

exact solution, zombie term e^{-39t}



Example: Stiff ODEs (III), $h = 0.1$

$$\frac{d}{dt}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 9u_1 + 24u_2 + 5\cos t - \frac{1}{3}\sin t, \\ -24u_1 - 51u_2 - 9\cos t + \frac{1}{3}\sin t \end{pmatrix}, \quad \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \frac{1}{3}\begin{pmatrix} 4 \\ 2 \end{pmatrix},$$

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} 2e^{-3t} - e^{-39t} + \frac{1}{3}\cos t \\ -e^{-3t} + 2e^{-39t} - \frac{1}{3}\cos t \end{pmatrix},$$

exact solution, zombie term e^{-39t}

t_j	numerical u_1	numerical u_2
0.1	-2.6452	7.8445
0.2	-18.452	38.876
0.3	-87.472	176.48
0.4	-934.07	789.35
0.5	-1760	3520
0.6	-7848.6	15698
0.7	-34990	69980
0.8	$-1.5598e+05$	$3.1196e+05$
0.9	$-6.9533e+05$	$1.3907e+06$
1.0	$-3.0997e+06$	$6.1994e+06$

Test equations for stiff ODEs

$$\frac{dy}{dt} \quad = \quad \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- test equation has transient solution $y(t) = \alpha\, e^{\lambda\, t}$ that decays to 0 as $t \to 0$.
- test equation has steady-state solution $y(t) = 0$.

Desired of numerical methods:

- **convergence:**
$$\mathbf{lim}_{h\to 0} \left| y(t_j) - w_j \right| = 0.$$
- **numerical stability:** small error in α leads to small error in w_j.

Euler's Method (I)

$$\frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- Euler's method: $w_0 = \alpha$,

$$w_{j+1} = w_j + h\,\lambda\, w_j = (1 + \lambda\, h)\, w_j, \quad j = 0, 1, \cdots ,$$

Euler's Method (I)

$$\frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- Euler's method: $w_0 = \alpha$,

$$w_{j+1} = w_j + h\,\lambda\, w_j = (1 + \lambda\, h)\, w_j, \quad j = 0, 1, \cdots,$$

- solving for w_j:

$$\begin{aligned} w_j &= (1 + \lambda\, h)^j\, w_0 = (1 + \lambda\, h)^j\, \alpha, \\ |y(t_j) - w_j| &= \left| e^{\lambda\, h\, j} - (1 + \lambda\, h)^j \right| |\alpha|\,. \end{aligned}$$

Euler's Method (I)

$$\frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- Euler's method: $w_0 = \alpha$,

$$w_{j+1} = w_j + h\,\lambda\, w_j = (1 + \lambda\, h)\, w_j, \quad j = 0, 1, \cdots ,$$

- solving for w_j:

$$\begin{aligned} w_j &= (1 + \lambda\, h)^j\, w_0 = (1 + \lambda\, h)^j\, \alpha, \\ |y(t_j) - w_j| &= \left| e^{\lambda\, h\, j} - (1 + \lambda\, h)^j \right| |\alpha|\,. \end{aligned}$$

- for convergence, need

$$|1 + \lambda\, h| < 1, \quad \text{or} \quad -2 < \lambda\, h < 0.$$

Euler's Method (II)

- Euler's method: $w_0 = \alpha$, and for $j = 0, 1, \cdots$,

$$w_{j+1} = (1 + \lambda h) w_j = (1 + \lambda h)^{j+1} \alpha.$$

Euler's Method (II)

- Euler's method: $w_0 = \alpha$, and for $j = 0, 1, \cdots$,

$$w_{j+1} \;=\; (1 + \lambda\, h)\, w_j = (1 + \lambda\, h)^{j+1}\, \alpha.$$

- Now assume a round-off error of δ in w_0:

$$\widehat{w}_0 = \alpha + \delta.$$

Euler's Method (II)

- Euler's method: $w_0 = \alpha$, and for $j = 0, 1, \cdots$,

$$w_{j+1} = (1 + \lambda\, h)\, w_j = (1 + \lambda\, h)^{j+1}\, \alpha.$$

- Now assume a round-off error of δ in w_0:

$$\widehat{w}_0 = \alpha + \delta.$$

- Euler's method numerically produces, for $j = 0, 1, \cdots$,

$$\widehat{w}_{j+1} = (1 + \lambda\, h)\, \widehat{w}_j = (1 + \lambda\, h)^{j+1} \left(\alpha + \delta\right).$$

Euler's Method (II)

- Euler's method: $w_0 = \alpha$, and for $j = 0, 1, \cdots$,

$$w_{j+1} = (1 + \lambda\, h)\, w_j = (1 + \lambda\, h)^{j+1}\, \alpha.$$

- Now assume a round-off error of δ in w_0:

$$\widehat{w}_0 = \alpha + \delta.$$

- Euler's method numerically produces, for $j = 0, 1, \cdots$,

$$\widehat{w}_{j+1} = (1 + \lambda\, h)\, \widehat{w}_j = (1 + \lambda\, h)^{j+1}\, (\alpha + \delta)\,.$$

- round-off error at step $(j + 1)$:

$$\widehat{w}_{j+1} - w_{j+1} = (1 + \lambda\, h)^{j+1}\, \delta.$$

Euler's Method (II)

- Euler's method: $w_0 = \alpha$, and for $j = 0, 1, \cdots$,
$$w_{j+1} = (1 + \lambda\, h)\, w_j = (1 + \lambda\, h)^{j+1}\, \alpha.$$

- Now assume a round-off error of δ in w_0:
$$\widehat{w}_0 = \alpha + \delta.$$

- Euler's method numerically produces, for $j = 0, 1, \cdots$,
$$\widehat{w}_{j+1} = (1 + \lambda\, h)\, \widehat{w}_j = (1 + \lambda\, h)^{j+1}\, (\alpha + \delta)\,.$$

- round-off error at step $(j + 1)$:
$$\widehat{w}_{j+1} - w_{j+1} = (1 + \lambda\, h)^{j+1}\, \delta.$$

- for numerical stability, need
$$|1 + \lambda\, h| < 1, \quad \text{or} \quad -2 < \lambda\, h < 0.$$

Multistep Methods (I)

$$\frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- general multistep method, for $j = m-1, m, \cdots$,

$$\begin{aligned} w_{j+1} = {} & a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ & + h\, \left(b_m\, f(t_{j+1}, w_{j+1}) + b_{m-1}\, f(t_j, w_j) + \cdots + b_0\, f(t_{j+1-m}, w_{j+1-m})\right) \end{aligned}$$

Multistep Methods (I)

$$\frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- general multistep method, for $j = m-1, m, \cdots$,

$$\begin{aligned} w_{j+1} = & \, a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ & + h\, \left(b_m\, f(t_{j+1}, w_{j+1}) + b_{m-1}\, f(t_j, w_j) + \cdots + b_0\, f(t_{j+1-m}, w_{j+1-m})\right) \end{aligned}$$

- for test equation,

$$\begin{aligned} w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ & \quad + \lambda h\, \left(b_m\, w_{j+1} + b_{m-1}\, w_j + \cdots + b_0\, w_{j+1-m}\right), \quad \text{or} \end{aligned}$$

$$(1 - \lambda h\, b_m)\, w_{j+1} - (a_{m-1} + \lambda h\, b_{m-1})\, w_j - \cdots - (a_0 + \lambda h\, b_0)\, w_{j+1-m} = 0.$$

Multistep Methods (I)

$$\frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- general multistep method, for $j = m-1, m, \cdots$,

$$w_{j+1} = a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m}$$
$$+h\, \left(b_m\, f(t_{j+1}, w_{j+1}) + b_{m-1}\, f(t_j, w_j) + \cdots + b_0\, f(t_{j+1-m}, w_{j+1-m})\right)$$

- for test equation,

$$\begin{aligned} w_{j+1} &= a_{m-1}\, w_j + a_{m-2}\, w_{j-1} + \cdots + a_0\, w_{j+1-m} \\ &\quad +\lambda h\, \left(b_m\, w_{j+1} + b_{m-1}\, w_j + \cdots + b_0\, w_{j+1-m}\right), \quad \text{or} \end{aligned}$$

$$(1 - \lambda h\, b_m)\, w_{j+1} - (a_{m-1} + \lambda h\, b_{m-1})\, w_j - \cdots - (a_0 + \lambda h\, b_0)\, w_{j+1-m} = 0.$$

- **characteristic polynomial**

$$\mathbf{Q}\,(z, \lambda\, h) \stackrel{def}{=} (1 - \lambda h\, b_m)\, z^m - (a_{m-1} + \lambda h\, b_{m-1})\, z^{m-1} - \cdots - (a_0 + \lambda h\, b_0)\,.$$

$$\text{test equation} \quad \frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- general multistep method for test equation,

$$(1 - \lambda h\, b_m)\, w_{j+1} - (a_{m-1} + \lambda h\, b_{m-1})\, w_j - \cdots - (a_0 + \lambda h\, b_0)\, w_{j+1-m} = 0.$$

$$\text{test equation} \quad \frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- general multistep method for test equation,

$$(1 - \lambda h\, b_m)\, w_{j+1} - (a_{m-1} + \lambda h\, b_{m-1})\, w_j - \cdots - (a_0 + \lambda h\, b_0)\, w_{j+1-m} = 0.$$

- *assume* distinct roots $\beta_1, \beta_2, \cdots, \beta_m$ of

$$\mathbf{Q}\,(z, \lambda\, h) = (1 - \lambda h\, b_m)\, z^m - (a_{m-1} + \lambda h\, b_{m-1})\, z^{m-1} - \cdots - (a_0 + \lambda h\, b_0) = 0.$$

test equation $\quad \dfrac{dy}{dt} \;=\; \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$

- general multistep method for test equation,

$$(1 - \lambda h\, b_m)\, w_{j+1} - (a_{m-1} + \lambda h\, b_{m-1})\, w_j - \cdots - (a_0 + \lambda h\, b_0)\, w_{j+1-m} = 0.$$

- *assume* distinct roots β_1, β_2, $\cdots$, β_m of

$$\mathbf{Q}\,(z, \lambda\, h) = (1 - \lambda h\, b_m)\, z^m - (a_{m-1} + \lambda h\, b_{m-1})\, z^{m-1} - \cdots - (a_0 + \lambda h\, b_0) = 0.$$

- there must exist constants $c_1, c_2, \cdots, c_m$,

$$w_j = c_1\ (\beta_1)^j + c_2\ (\beta_2)^j + \cdots + c_m\ (\beta_m)^j\,, \quad \text{for} \quad j = 0, 1, 2, \cdots$$

$$\text{test equation} \quad \frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- general multistep method for test equation,

$$(1 - \lambda h\, b_m)\, w_{j+1} - (a_{m-1} + \lambda h\, b_{m-1})\, w_j - \cdots - (a_0 + \lambda h\, b_0)\, w_{j+1-m} = 0.$$

- *assume* distinct roots $\beta_1, \beta_2, \cdots, \beta_m$ of

$$\mathbf{Q}\,(z, \lambda\, h) = (1 - \lambda h\, b_m)\, z^m - (a_{m-1} + \lambda h\, b_{m-1})\, z^{m-1} - \cdots - (a_0 + \lambda h\, b_0) = 0.$$

- there must exist constants $c_1, c_2, \cdots, c_m$,

$$w_j = c_1\, (\beta_1)^j + c_2\, (\beta_2)^j + \cdots + c_m\, (\beta_m)^j, \quad \text{for} \quad j = 0, 1, 2, \cdots$$

- for convergence and numerical stability, we must require

$$|\beta_k| < 1, \quad k = 1, 2, \cdots, m.$$

$$\text{test equation} \quad \frac{dy}{dt} = \lambda y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- *assume* distinct roots $\beta_1, \beta_2, \cdots, \beta_m$ of

$$\mathbf{Q}(z, \lambda h) = (1 - \lambda h\, b_m)\, z^m - (a_{m-1} + \lambda h\, b_{m-1})\, z^{m-1} - \cdots - (a_0 + \lambda h\, b_0) = 0.$$

$$\text{test equation} \quad \frac{dy}{dt} = \lambda y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- *assume* distinct roots $\beta_1, \beta_2, \cdots, \beta_m$ of

$$\mathbf{Q}(z, \lambda h) = (1 - \lambda h\, b_m)\, z^m - (a_{m-1} + \lambda h\, b_{m-1})\, z^{m-1} - \cdots - (a_0 + \lambda h\, b_0) = 0.$$

- for convergence and numerical stability, we must require

$$|\beta_k| < 1, \quad k = 1, 2, \cdots, m.$$

test equation $\frac{dy}{dt} = \lambda y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$

- *assume* distinct roots $\beta_1, \beta_2, \cdots, \beta_m$ of

$$\mathbf{Q}(z, \lambda h) = (1 - \lambda h\, b_m)\, z^m - (a_{m-1} + \lambda h\, b_{m-1})\, z^{m-1} - \cdots - (a_0 + \lambda h\, b_0) = 0.$$

- for convergence and numerical stability, we must require

$$|\beta_k| < 1, \quad k = 1, 2, \cdots, m.$$

- **region R of absolute stability**

$$\mathbf{R} \stackrel{def}{=} \{ \lambda h \in \mathbb{C} \mid \quad |\beta_k| < 1 \quad \text{for all zeros } \beta_k \text{ of } \mathbf{Q}(z, \lambda h). \}$$

region R of absolute stability: Euler's method

$$\text{test equation} \quad \frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- Euler's method

$$w_{j+1} = w_j + h\, f(t_j, w_j) = (1 + \lambda\, h)\, w_j.$$

- **characteristic polynomial**

$$\mathbf{Q}\,(z, \lambda\, h) = z - (1 + \lambda h) = 0, \quad \text{with root} \;\; \beta_1 = 1 + \lambda h.$$

region R of absolute stability: Euler's method

$$\text{test equation} \quad \frac{dy}{dt} \;=\; \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- Euler's method

$$w_{j+1} = w_j + h\, f(t_j, w_j) = (1 + \lambda\, h)\, w_j.$$

- **characteristic polynomial**

$$\mathbf{Q}\,(z, \lambda\, h) = z - (1 + \lambda h) = 0, \quad \text{with root} \;\; \beta_1 = 1 + \lambda h.$$

- **region R of absolute stability**

$$\mathbf{R} = \{\; \lambda\, h \in \mathbb{C} \;\mid\; \; |1 + \lambda h| < 1. \}$$

region R of absolute stability: Implicit Trapezoid

$$\text{test equation} \quad \frac{dy}{dt} = \lambda y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- Implicit Trapezoid

$$w_{j+1} = w_j + \frac{h}{2}\left(f\left(t_{j+1}, w_{j+1}\right) + f\left(t_j, w_j\right)\right) = w_j + \frac{\lambda\, h}{2}\left(w_{j+1} + w_j\right).$$

- **characteristic polynomial**

$$\mathbf{Q}\left(z, \lambda\, h\right) = \left(1 - \frac{\lambda\, h}{2}\right) z - \left(1 + \frac{\lambda\, h}{2}\right) = 0, \quad \text{with root} \quad \beta_1 = \frac{1 + \frac{\lambda\, h}{2}}{1 - \frac{\lambda\, h}{2}}.$$

region R of absolute stability: Implicit Trapezoid

$$\text{test equation} \quad \frac{dy}{dt} = \lambda\, y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- Implicit Trapezoid

$$w_{j+1} = w_j + \frac{h}{2}\left(f\left(t_{j+1}, w_{j+1}\right) + f\left(t_j, w_j\right)\right) = w_j + \frac{\lambda\, h}{2}\left(w_{j+1} + w_j\right).$$

- **characteristic polynomial**

$$\mathbf{Q}\left(z, \lambda\, h\right) = \left(1 - \frac{\lambda\, h}{2}\right) z - \left(1 + \frac{\lambda\, h}{2}\right) = 0, \quad \text{with root} \quad \beta_1 = \frac{1 + \frac{\lambda\, h}{2}}{1 - \frac{\lambda\, h}{2}}.$$

- **region R of absolute stability**

$$\mathbf{R} = \{\, \lambda\, h \in \mathbb{C} \mid \quad \mathbf{Re}\left(\lambda h\right) < 0. \}$$

region R of absolute stability: Implicit Trapezoid

$$\text{test equation} \quad \frac{dy}{dt} = \lambda y, \quad y(0) = \alpha, \quad \text{for} \quad \lambda < 0.$$

- Implicit Trapezoid

$$w_{j+1} = w_j + \frac{h}{2}\left(f(t_{j+1}, w_{j+1}) + f(t_j, w_j)\right) = w_j + \frac{\lambda h}{2}\left(w_{j+1} + w_j\right).$$

- **characteristic polynomial**

$$\mathbf{Q}(z, \lambda h) = \left(1 - \frac{\lambda h}{2}\right) z - \left(1 + \frac{\lambda h}{2}\right) = 0, \quad \text{with root} \quad \beta_1 = \frac{1 + \frac{\lambda h}{2}}{1 - \frac{\lambda h}{2}}.$$

- **region R of absolute stability**

$$\mathbf{R} = \{ \lambda h \in \mathbb{C} \mid \mathbf{Re}(\lambda h) < 0. \}$$

- *Definition*: Implicit Trapezoid is **A-stable**.

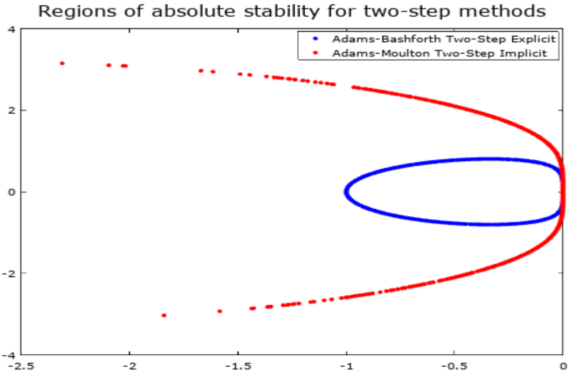
regions of absolute stability:

- **Adams-Bashforth Two-Step Explicit Method**

$$w_{j+1} = w_j + \frac{h}{2}\left(3f(t_j, w_j) - f(t_{j-1}, w_{j-1})\right).$$

- **Adams-Moulton Two-Step Implicit Method**

$$w_{j+1} = w_j + \frac{h}{12}\left(5f(t_{j+1}, w_{j+1}) + 8f(t_j, w_j) - f(t_{j-1}, w_{j-1})\right).$$



Implicit Trapezoid with Newton Iteration

- Implicit Trapezoid

$$w_{j+1} = w_j + \frac{h}{2}\left(f(t_{j+1}, w_{j+1}) + f(t_j, w_j)\right).$$

Implicit Trapezoid with Newton Iteration

- Implicit Trapezoid

$$w_{j+1} = w_j + \frac{h}{2}\left(f\left(t_{j+1}, w_{j+1}\right) + f\left(t_j, w_j\right)\right).$$

- w_{j+1} is root to

$$\mathbf{F}\left(w\right) \stackrel{def}{=} w - w_j - \frac{h}{2}\left(f\left(t_{j+1}, w\right) + f\left(t_j, w_j\right)\right) = 0.$$

Implicit Trapezoid with Newton Iteration

- Implicit Trapezoid

$$w_{j+1} = w_j + \frac{h}{2}\left(f(t_{j+1}, w_{j+1}) + f(t_j, w_j)\right).$$

- w_{j+1} is root to

$$\mathbf{F}(w) \stackrel{def}{=} w - w_j - \frac{h}{2}\left(f(t_{j+1}, w) + f(t_j, w_j)\right) = 0.$$

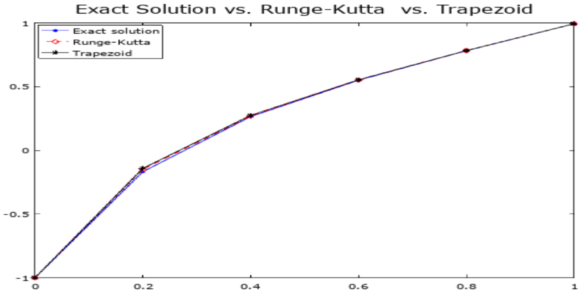
- Newton Iteration: $w_{j+1}^{(0)} = w_j$ (no predictor); for $\ell = 0, 1, \cdots$

$$\begin{aligned} w_{j+1}^{(\ell+1)} &= w_{j+1}^{(\ell)} - \frac{\mathbf{F}\left(w_{j+1}^{(\ell)}\right)}{\mathbf{F}'\left(w_{j+1}^{(\ell)}\right)} \\ &= w_{j+1}^{(\ell)} - \frac{w_{j+1}^{(\ell)} - w_j - \frac{h}{2}\left(f(t_{j+1}, w_{j+1}^{(\ell)}) + f(t_j, w_j)\right)}{1 - \frac{h}{2}\frac{\partial f}{\partial y}(t_{j+1}, w_{j+1}^{(\ell)})} \end{aligned}$$

Implicit Trapezoid example (I)

$$\begin{aligned} \frac{dy}{dt} &= 5e^{5t}(y-t)^2+1, \quad 0 \le t \le 1, \quad y(0)=-1, \\ y(t) &= t-e^{-5t} \quad \text{exact solution} \end{aligned}$$

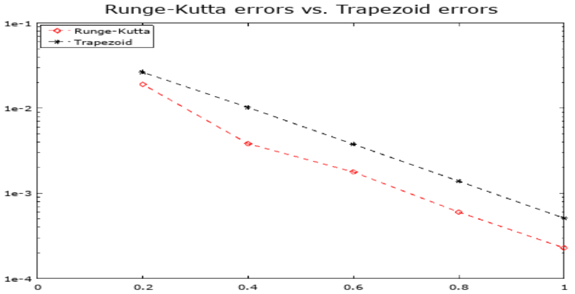
- Implicit Trapezoid vs. $4^{\mathbf{th}}$ order Runge-Kutta , $h=0.2$



Implicit Trapezoid example (I)

$$\frac{dy}{dt} = 5e^{5t}(y-t)^2 + 1, \quad 0 \le t \le 1, \quad y(0) = -1,$$
$$y(t) = t - e^{-5t} \quad \text{exact solution}$$

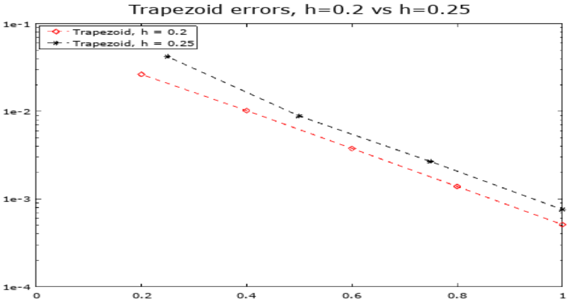
- Implicit Trapezoid vs. 4^{th} order Runge-Kutta , $h = 0.2$



Implicit Trapezoid example (II)

$$\frac{dy}{dt} = 5e^{5t}(y-t)^2 + 1, \quad 0 \le t \le 1, \quad y(0) = -1,$$
$$y(t) = t - e^{-5t} \quad \text{exact solution}$$

- Implicit Trapezoid, $h = 0.25$



Implicit Trapezoid example (II)

$$\begin{aligned} \frac{dy}{dt} &= 5e^{5t}(y-t)^2 + 1, \quad 0 \leq t \leq 1, \quad y(0) = -1, \\ y(t) &= t - e^{-5t} \quad \text{exact solution} \end{aligned}$$

▶

t_j	4^{th} order Runge Kutta
0.0	-1.0
0.25	0.4014315
0.5	3.4374753
0.75	$1.44639e+23$
1.0	**Inf**

Linear Equations: example of ellipse $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

- **forward**: given a^2 and b^2, ellipse uniquely determined.

Linear Equations: example of ellipse $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

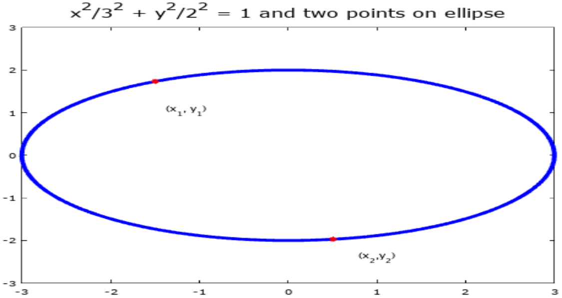
- **forward**: given a^2 and b^2, ellipse uniquely determined.
- **backward**: given points (x_1, y_1) and (x_2, y_2) on ellipse, uniquely determine a^2 and b^2.

$$\frac{x_1^2}{a^2} + \frac{y_1^2}{b^2} = 1,$$
$$\frac{x_2^2}{a^2} + \frac{y_2^2}{b^2} = 1.$$

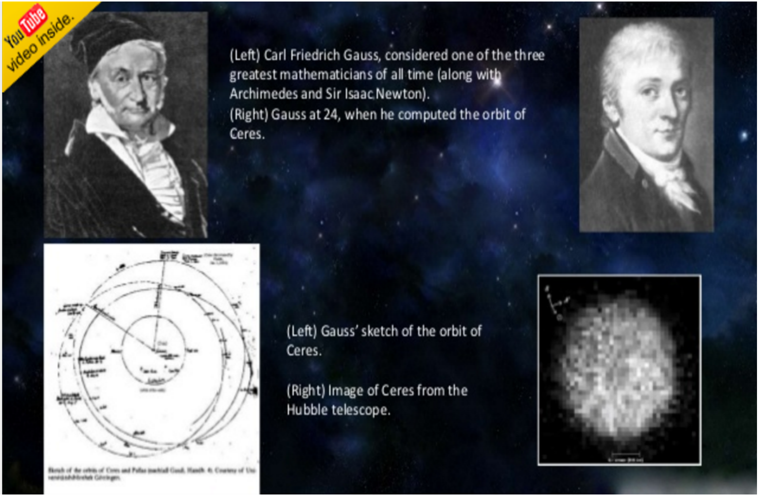
Linear Equations: example of ellipse $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

- **forward**: given a^2 and b^2, ellipse uniquely determined.
- **backward**: given points (x_1, y_1) and (x_2, y_2) on ellipse, uniquely determine a^2 and b^2.

$$\frac{x_1^2}{a^2} + \frac{y_1^2}{b^2} = 1,$$
$$\frac{x_2^2}{a^2} + \frac{y_2^2}{b^2} = 1.$$



Carl Gauss: World's first numerical analyst



Solving Linear Equations

- example: solving for x_1, x_2, x_3, x_4 in system of linear equations

$$\begin{array}{lrcrcrcrcr}
E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\
E_2: & 2x_1 & + & x_2 & - & x_3 & + & x_4 & = & 1, \\
E_3: & 3x_1 & - & x_2 & - & x_3 & + & 2x_4 & = & -3, \\
E_4: & -x_1 & + & 2x_2 & + & 3x_3 & - & x_4 & = & 4.
\end{array}$$

Solving Linear Equations

- example: solving for x_1, x_2, x_3, x_4 in system of linear equations

$$\begin{array}{lrcrcrcrcr} E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\ E_2: & 2x_1 & + & x_2 & - & x_3 & + & x_4 & = & 1, \\ E_3: & 3x_1 & - & x_2 & - & x_3 & + & 2x_4 & = & -3, \\ E_4: & -x_1 & + & 2x_2 & + & 3x_3 & - & x_4 & = & 4. \end{array}$$

Solution techniques (elementary operations)

- Multiply equation E_i by any constant $\lambda \neq 0$, with the resulting equation used in place of E_i.
  $((\lambda\, E_i) \rightarrow (E_i))$

Solving Linear Equations

- example: solving for x_1, x_2, x_3, x_4 in system of linear equations

$$\begin{array}{lrcrcrcrcr}
E_1 : & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\
E_2 : & 2x_1 & + & x_2 & - & x_3 & + & x_4 & = & 1, \\
E_3 : & 3x_1 & - & x_2 & - & x_3 & + & 2x_4 & = & -3, \\
E_4 : & -x_1 & + & 2x_2 & + & 3x_3 & - & x_4 & = & 4.
\end{array}$$

Solution techniques (elementary operations)

- Multiply equation E_i by any constant $\lambda \neq 0$, with the resulting equation used in place of E_i.
  $((\lambda\, E_i) \to (E_i))$
- Multiply equation E_j by any constant λ and add to equation $E_i, i \neq j$, with the resulting equation used in place of E_i.
  $((E_i + \lambda\, E_j) \to (E_i))$

Solving Linear Equations

- example: solving for x_1, x_2, x_3, x_4 in system of linear equations

$$\begin{array}{lrcrcrcrcr}
E_1 : & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\
E_2 : & 2x_1 & + & x_2 & - & x_3 & + & x_4 & = & 1, \\
E_3 : & 3x_1 & - & x_2 & - & x_3 & + & 2x_4 & = & -3, \\
E_4 : & -x_1 & + & 2x_2 & + & 3x_3 & - & x_4 & = & 4.
\end{array}$$

Solution techniques (elementary operations)

- Multiply equation E_i by any constant $\lambda \neq 0$, with the resulting equation used in place of E_i.
  $((\lambda\, E_i) \to (E_i))$
- Multiply equation E_j by any constant λ and add to equation $E_i, i \neq j$, with the resulting equation used in place of E_i.
  $((E_i + \lambda\, E_j) \to (E_i))$
- Equations E_i and E_j can be transposed in order.
  $((E_i) \leftrightarrow (E_j))$

Solving Linear Equations: example (I)

- example: solving for x_1, x_2, x_3, x_4 in system of linear equations

$$\begin{array}{lrcrcrcrcr}
E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\
E_2: & 2x_1 & + & x_2 & - & x_3 & + & x_4 & = & 1, \\
E_3: & 3x_1 & - & x_2 & - & x_3 & + & 2x_4 & = & -3, \\
E_4: & -x_1 & + & 2x_2 & + & 3x_3 & - & x_4 & = & 4.
\end{array}$$

- eliminate x_1 from E_2, $(E_2 - 2E_1) \rightarrow (E_2)$:

$$(\,2x_1 + x_2 - x_3 + x_4\,) - 2\,(\,x_1 + x_2 + 3x_4\,) = 1 - 2 \times 4.$$

- new system of equations, same solution in x_1, x_2, x_3, x_4

$$\begin{array}{lrcrcrcrcr}
E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\
\mathbf{E_2}: & & - & x_2 & - & x_3 & - & 5x_4 & = & -7, \\
E_3: & 3x_1 & - & x_2 & - & x_3 & + & 2x_4 & = & -3, \\
E_4: & -x_1 & + & 2x_2 & + & 3x_3 & - & x_4 & = & 4.
\end{array}$$

Solving Linear Equations: example (II)

- $$\begin{array}{lrcrcrcrcr} E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\ E_2: & & - & x_2 & - & x_3 & - & 5x_4 & = & -7, \\ E_3: & 3x_1 & - & x_2 & - & x_3 & + & 2x_4 & = & -3, \\ E_4: & -x_1 & + & 2x_2 & + & 3x_3 & - & x_4 & = & 4. \end{array}$$
- eliminate x_1 from E_3, E_4:

$$(E_3 - 3E_1) \to (E_3), \quad (E_4 - (-1)E_1) \to (E_4).$$

- new system of equations, same solution in x_1, x_2, x_3, x_4

$$\begin{array}{lrcrcrcrcr} E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\ E_2: & & - & x_2 & - & x_3 & - & 5x_4 & = & -7, \\ \mathbf{E_3}: & & - & 4x_2 & - & x_3 & - & 7x_4 & = & -15, \\ \mathbf{E_4}: & & + & 3x_2 & + & 3x_3 & + & 2x_4 & = & 8. \end{array}$$

- equations E_2, E_3, E_4 no longer contain x_1.

Solving Linear Equations: example (III)

- $$\begin{array}{lrrrrrrrrr} E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\ E_2: & & - & x_2 & - & x_3 & - & 5x_4 & = & -7, \\ E_3: & & - & 4x_2 & - & x_3 & - & 7x_4 & = & -15, \\ E_4: & & + & 3x_2 & + & 3x_3 & + & 2x_4 & = & 8. \end{array}$$
- Use E_2 to eliminate x_2 from E_3 and E_4 by performing

$$(E_3 - 4E_2) \to (E_3), \quad (E_4 - (-3)E_2) \to (E_4).$$

- new system of equations,

$$\begin{array}{lrrrrrrrrr} E_1: & x_1 & + & x_2 & & & + & 3x_4 & = & 4, \\ E_2: & & - & x_2 & - & x_3 & - & 5x_4 & = & -7, \\ \mathbf{E_3}: & & & & & 3x_3 & + & 13x_4 & = & 13, \\ \mathbf{E_4}: & & & & & & - & 13x_4 & = & -13. \end{array}$$

Solving Linear Equations: example (IV), **backward substitution**

$$\uparrow \quad \begin{array}{lrcrcrcrcrcr|lcll}
\mathbf{E_1}: & x_1 & + & x_2 & & & + & 3x_4 & = & 4 & x_1 & = & 4 - x_2 - 3x_4 & = & -1, \\
\mathbf{E_2}: & & - & x_2 & - & x_3 & - & 5x_4 & = & -7 & x_2 & = & \frac{-7 + x_3 + 5x_4}{-1} & = & 2, \\
\mathbf{E_3}: & & & & & 3x_3 & + & 13x_4 & = & 13 & x_3 & = & \frac{13 - 13x_4}{3} & = & 0, \\
\mathbf{E_4}: & & & & & & - & 13x_4 & = & -13 & x_4 & = & \frac{-13}{-13} & = & 1.
\end{array}$$

Solving Linear Equations with pivoting (I)

- solving for x_1, x_2, x_3, x_4

$$
\begin{array}{lrcrcrcrcr}
E_1: & x_1 & - & x_2 & + & 2x_3 & - & x_4 & = & -8, \\
E_2: & 2x_1 & - & 2x_2 & + & 3x_3 & - & 3x_4 & = & -20, \\
E_3: & x_1 & + & x_2 & + & x_3 & & & = & -2, \\
E_4: & x_1 & - & x_2 & + & 4x_3 & + & 3x_4 & = & 4.
\end{array}
$$

- eliminate x_1 from E_2, E_3 and E_4:

$$
(E_2 - 2E_1) \to (E_2), \; (E_3 - E_1) \to (E_3), \; (E_4 - E_1) \to (E_4).
$$

- new system of equations,

$$
\begin{array}{lrcrcrcrcr}
E_1: & x_1 & - & x_2 & + & 2x_3 & - & x_4 & = & -8, \\
\mathbf{E_2}: & & & & - & x_3 & - & x_4 & = & -4, \\
\mathbf{E_3}: & & & 2x_2 & - & x_3 & + & x_4 & = & 6, \\
\mathbf{E_4}: & & & & & 2x_3 & + & 4x_4 & = & 12.
\end{array}
$$

- Can NOT use E_2 to eliminate x_2 from E_3 and E_4

Solving Linear Equations with pivoting (II)

- $$\begin{array}{lrcrcrcrcr} E_1: & x_1 & - & x_2 & + & 2x_3 & - & x_4 & = & -8, \\ E_2: & & & & - & x_3 & - & x_4 & = & -4, \\ E_3: & & & 2x_2 & - & x_3 & + & x_4 & = & 6, \\ E_4: & & & & & 2x_3 & + & 4x_4 & = & 12. \end{array}$$

- Exchange E_2 and E_3: $(E_2) \leftrightarrow (E_3)$ (**pivoting**)

$$\begin{array}{lrcrcrcrcr} E_1: & x_1 & - & x_2 & + & 2x_3 & - & x_4 & = & -8, \\ \mathbf{E_2}: & & & 2x_2 & - & x_3 & + & x_4 & = & 6, \\ \mathbf{E_3}: & & & & - & x_3 & - & x_4 & = & -4, \\ E_4: & & & & & 2x_3 & + & 4x_4 & = & 12. \end{array}$$

Solving Linear Equations with pivoting (III)

- $$\begin{array}{rrrrrrrrrl} E_1: & x_1 & - & x_2 & + & 2x_3 & - & x_4 & = & -8, \\ E_2: & & & 2x_2 & - & x_3 & + & x_4 & = & 6, \\ E_3: & & & & - & x_3 & - & x_4 & = & -4, \\ E_4: & & & & & 2x_3 & + & 4x_4 & = & 12. \end{array}$$

- eliminate x_3 from E_4: $(E_4 + E_3) \rightarrow (E_4)$

$$\begin{array}{rrrrrrrrrl} E_1: & x_1 & - & x_2 & + & 2x_3 & - & x_4 & = & -8, \\ E_2: & & & 2x_2 & - & x_3 & + & x_4 & = & 6, \\ E_3: & & & & - & x_3 & - & x_4 & = & -4, \\ \mathbf{E_4}: & & & & & & & 2x_4 & = & 4. \end{array}$$

- backward substitution:

$$x_4 = 2, \quad x_3 = 2, \quad x_2 = 3, \quad x_1 = -7.$$

System of 2nd order ODEs

System of m second-order ODEs:

$$
\begin{aligned}
\frac{d^2 u_1}{dt^2} &= f_1(t, u_1, u_1', u_2, u_2', \cdots, u_m, u_m'), \\
\frac{d^2 u_2}{dt^2} &= f_2(t, u_1, u_1', u_2, u_2', \cdots, u_m, u_m'), \\
&\vdots \\
\frac{d^2 u_m}{dt^2} &= f_m(t, u_1, u_1', u_2, u_2', \cdots, u_m, u_m'), \quad a \leq t \leq b,
\end{aligned}
$$

with $2m$ initial conditions

$$
\begin{aligned}
u_1(a) &= \alpha_1, \;\; u_1'(a) = \alpha_1', \quad u_2(a) = \alpha_2, \;\; u_2'(a) = \alpha_2', \cdots, \\
u_m(a) &= \alpha_m, \;\; u_m(a)' = \alpha_m'.
\end{aligned}
$$

$u_1, u_2, \cdots, u_m$ could be vectors.

System of 2nd order ODEs: first-order form

$$\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{2m} \end{pmatrix} \stackrel{def}{=} \begin{pmatrix} u_1 \\ u_1' \\ u_2 \\ u_2' \\ \vdots \\ u_m \\ u_m' \end{pmatrix}, \quad \mathbf{f}(t, \mathbf{z}) \stackrel{def}{=} \begin{pmatrix} z_2 \\ f_1(t, z_1, z_2, z_3, \cdots, z_{2m-1}, z_{2m}) \\ z_4 \\ f_2(t, z_1, z_2, z_3, \cdots, z_{2m-1}, z_{2m}) \\ \vdots \\ z_{2m} \\ f_m(t, z_1, z_2, z_3, \cdots, z_{2m-1}, z_{2m}) \end{pmatrix}$$

System of $2m$ first-order ODEs: $\dfrac{d\mathbf{z}}{dt} = \mathbf{f}(t, \mathbf{z}), \quad a \leq t \leq b,$

with initial condition $\mathbf{z}(a) = \alpha \stackrel{def}{=} \begin{pmatrix} \alpha_1 \\ \alpha_1' \\ \alpha_2 \\ \alpha_2' \\ \vdots \\ \alpha_m \\ \alpha_m' \end{pmatrix}.$