



Absolutstabilitet

1 Introduktion

För att en numerisk ODE-metod ska vara användbar måste den vara konvergent, dvs den numeriska lösningen ska närma sig den exakta lösningen när steglängden går mot noll. Det är ett ganska naturligt krav. Det betyder emellertid inte att metoden alltid fungerar bra för en *fix* steglängd även om denna är ganska liten. Följande enkla exempel visar att man i allmänhet också måste ställa krav på steglängden för att metoden ska bli praktiskt användbar.

Exempel 1: Vi vill lösa ODEn

$$y' = -25y, \quad y(0) = 1,$$

med två konvergenta metoder: framåt och bakåt Euler. Resultaten med steglängden $h = 0.1$ för de båda metoderna ges i Figur 1. Överst visas bakåt Euler. Den numeriska lösningen (streck markerat med cirklar) är kvalitativt lik den exakta lösningen ($y(t) = \exp(-25t)$, andra strecket). För framåt Euler, som visas nederst, blir den numeriska lösningen dock helt fel. Vi har alltså två *konvergenta* metoder som ger helt olika resultat när vi använder ett fixt h . Vi behöver ett sätt att skilja det “bra” beteendet för bakåt Euler från det “dåliga” beteendet för framåt Euler. Därför introducerar vi begreppet *absolutstabilitet*.

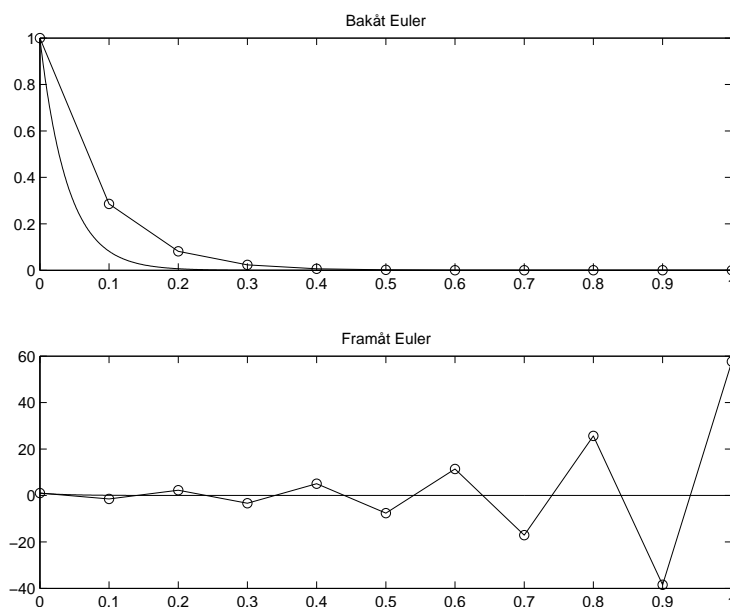


Figure 1. Steglängd $h = 0.1$.

(Notera att resultaten ovan inte motsäger det faktum att metoderna är konvergenta. När vi låter h bli mindre blir resultaten för båda metoderna lika den exakta lösningen. Om vi till exempel tar $h = 0.01$ får vi resultaten i Figur 2.)

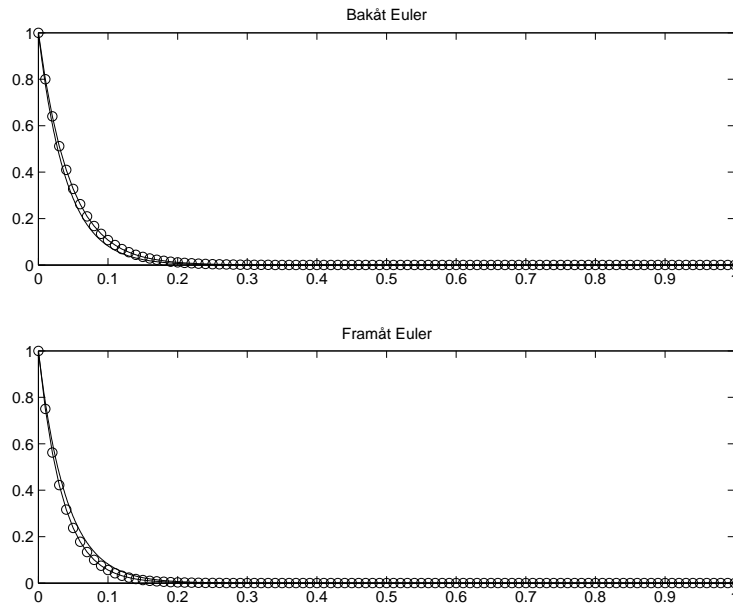


Figure 2. Steglängd $h = 0.01$.

2 Definition

Betrakta *testproblemet*

$$\begin{cases} y' = \lambda y, \\ y(0) = 1, \end{cases} \quad \lambda \in \mathbb{C}, \quad \operatorname{Re} \lambda < 0. \quad (1)$$

Lösningen $y(t) = \exp(\lambda t) \rightarrow 0$ när $t \rightarrow \infty$ eftersom $\operatorname{Re} \lambda < 0$. Det minsta man kan begära är att den numeriska lösningen ska uppföra sig på samma sätt. Vi definierar därför:

Definition: En numerisk lösning $\{u_n\}$ till (1) är *absolutstabil* för ett fixt h om

$$\lim_{n \rightarrow \infty} u_n = 0.$$

Stabiliteten kommer att bero på produkten $h\lambda$ och vi definierar också ett stabilitetsområde för dessa tal:

Definition: *Stabilitetsområdet* \mathcal{A} (“region of absolute stability”) för en numerisk metod är de komplexa tal $h\lambda$ för vilka den numeriska lösningen av (1) är absolutstabil.

Här följer två exempel på stabilitetsområden.

Exempel 2: Framåt Euler. När vi tillämpar metoden på (1) får vi

$$u_{n+1} = u_n + h\lambda u_n, \quad u_0 = 1.$$

Efter omskrivning ger detta

$$u_{n+1} = (1 + h\lambda)u_n = (1 + h\lambda)^2 u_{n-1} = \dots = (1 + h\lambda)^{n+1} u_0 = (1 + h\lambda)^{n+1}.$$

För att följderna $\{u_n\}$ ska gå mot noll måste således $h\lambda$ uppfylla

$$|1 + h\lambda| < 1. \quad (2)$$

Området \mathcal{A} är därför en cirkel i det komplexa talplanet med radie ett och centrum i -1 ; se Figur 3, vänster. När λ är *reell* reducerar (2) till uttrycket

$$-1 < 1 + h\lambda < 1 \quad \Leftrightarrow \quad h < 2/|\lambda|.$$

Exempel 3: Bakåt Euler. Här har vi

$$u_{n+1} = u_n + h\lambda u_{n+1}, \quad u_0 = 1,$$

vilket leder till

$$u_{n+1} = \frac{1}{1 - h\lambda} u_n = \frac{1}{(1 - h\lambda)^2} u_{n-1} = \cdots = \frac{1}{(1 - h\lambda)^{n+1}} u_0 = \frac{1}{(1 - h\lambda)^{n+1}}.$$

För bakåt Euler måste därför

$$|1 - h\lambda| > 1,$$

för att följderna $\{u_n\}$ ska gå mot noll. Men eftersom realdelen av λ är negativ kommer detta villkor alltid vara uppfyllt. Stabilitetsområdet \mathcal{A} är därför hela vänstra halvan av det komplexa talplanet; se Figur 3, höger. (Ibland definierar man \mathcal{A} även för λ med positiv realdel. För bakåt Euler blir då \mathcal{A} hela komplexa talplanet förutom en cirkel med radien ett, centrerad i $+1$.)

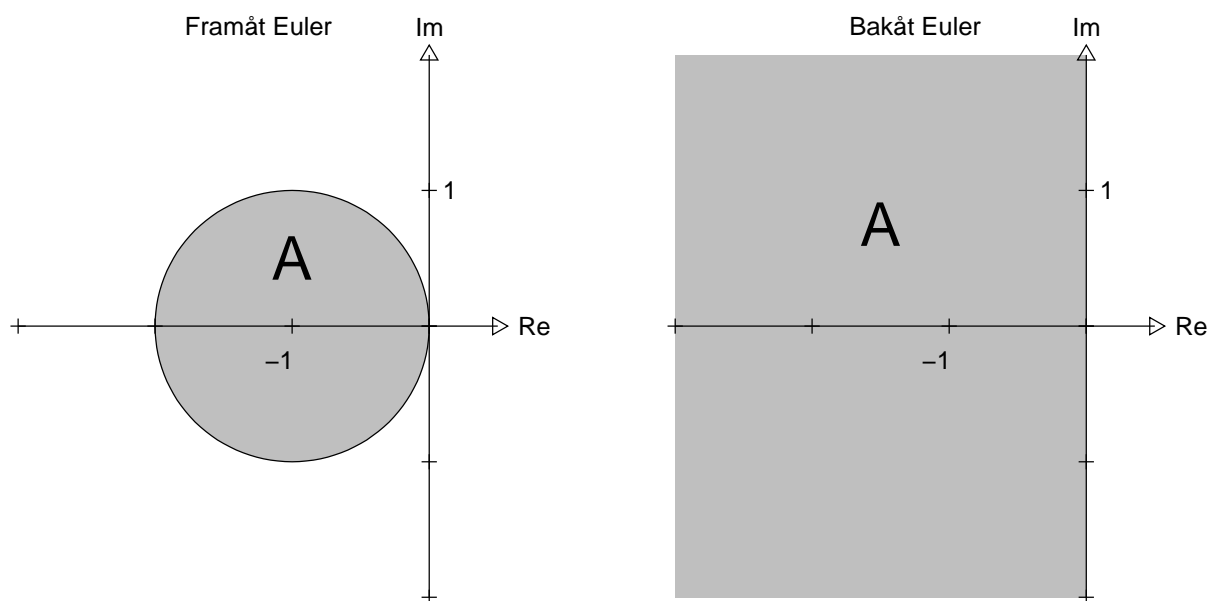


Figure 3. Stabilitetsområden för framåt och bakåt Euler.

Denna analys förklarar det inledande exemplet. I det första fallet, där $h = 0.1$, får vi att $h\lambda = -2.5 \notin \mathcal{A}$ för framåt Euler, varför lösningen växer obegränsat. I det andra fallet hade vi $h = 0.01$ och $h\lambda = -0.25 \in \mathcal{A}$. Lösningen för framåt Euler är då absolutstabil. Eftersom det inte finns någon stabilitetsgräns för bakåt Euler ger den metoden en stabil lösning i båda fallen.

Stabilitetsområdena för framåt och bakåt Euler är typexempel på stabilitetsområden för explicita respektive implicita metoder. Explicita metoder har ett begränsat stabilitetsområde vilket betyder att det alltid finns en stabilitetsgräns för h ; normalt måste h vara tillräckligt litet för att metoden ska vara absolutstabil. Framåt Euler är tex absolutstabil för problemet

ovan om $h < 2/25 = 0.08$. Implicita metoder, å andra sidan, har vanligtvis ett obegränsat stabilitetsområde och är ofta absolutstabila för alla h . (Det finns dock exempel på implicita metoder med begränsat stabilitetsområde.)

Vi avslutar denna del med ett lite mer komplicerat fall.

Exempel 4: Runge–Kutta 4. Om vi tillämpar metoden på testproblemet får vi följande för k_j -värdena i metoden,

$$\begin{aligned}k_1 &= \lambda u_n, \\k_2 &= \lambda \left(u_n + \frac{h}{2} k_1 \right) = \left(\lambda + \frac{h}{2} \lambda^2 \right) u_n, \\k_3 &= \lambda \left(u_n + \frac{h}{2} k_2 \right) = \left(\lambda + \frac{h}{2} \lambda^2 + \frac{h^2}{4} \lambda^3 \right) u_n, \\k_4 &= \lambda (u_n + h k_3) = \left(\lambda + h \lambda^2 + \frac{h^2}{2} \lambda^3 + \frac{h^3}{4} \lambda^4 \right) u_n,\end{aligned}$$

Tillsammans ger det sedan iterationsformeln

$$u_{n+1} = u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) = \left(1 + h\lambda + \frac{(h\lambda)^2}{2} + \frac{(h\lambda)^3}{6} + \frac{(h\lambda)^4}{24} \right) u_n.$$

Stabilitetsområdet för Runge–Kutta 4 utgörs därför av de tal z i vänstra halvan av komplexa talplanet, för vilka $|P(z)| < 1$ där

$$P(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}.$$

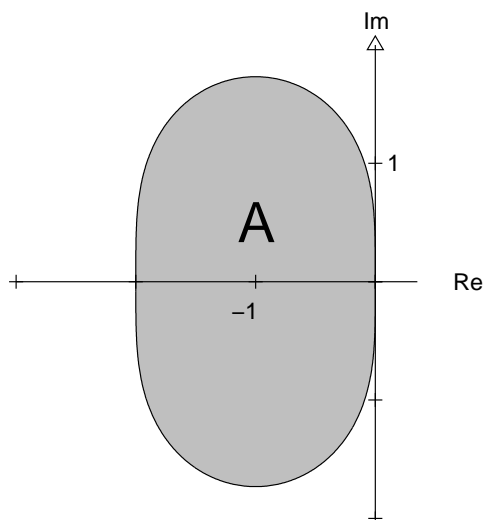
Området är plottat till höger i Figur 4. Stabilitetsgränsen för reella λ ges genom att söka lösningar till $P(x) = 1$ och $P(x) = -1$ med $x \in \mathbb{R}$. Det visar sig att det bara finns två sådana rötter: $x = 0$ och $x \approx -2.7853$. Det betyder att λh måste ligga i intervallet $(-2.7853, 0)$, dvs stabilitetsgränsen ges av $2.7853/|\lambda|$. Med ett allmänt λ kan man bestämma stabilitetsgränsen för h genom att hitta rötter till $R(h) := |P(h\lambda)|^2 = 1$, där $R(h)$ är ett reellt polynom i h .

3 Absolutstabilitet och störningar

Begreppet absolutstabilitet har en vidare tillämpning än bara för det enkla testproblemet ovan. I allmänhet säger vi att en numerisk lösning eller numerisk metod är absolutstabil om *effekten av en liten störning i begynnelsedata u_0 försvinner i limes $n \rightarrow \infty$* . Detta är av stor vikt, eftersom alla numeriska beräkningar är behäftade med små fel. Absolutstabilitet betyder att den numeriska lösningen beter sig kvalitativt på samma sätt som den exakta lösningen. I praktiken är absolutstabilitet ett krav för att en numerisk metod ska gå att använda.

Notera att stabilitetsområdet \mathcal{A} är definierat helt i termer av det triviala testproblemet (1). Det kan dock användas även för allmänna ODEr för att avgöra om en metod är absolutstabil. Vi illustrerar detta med ett par exempel nedan. Härledningen av varför effekten av en liten störning faktiskt försvinner när $n \rightarrow \infty$ under de givna villkoren ges senare i sektion 4.

Runge–Kutta 2



Runge–Kutta 4

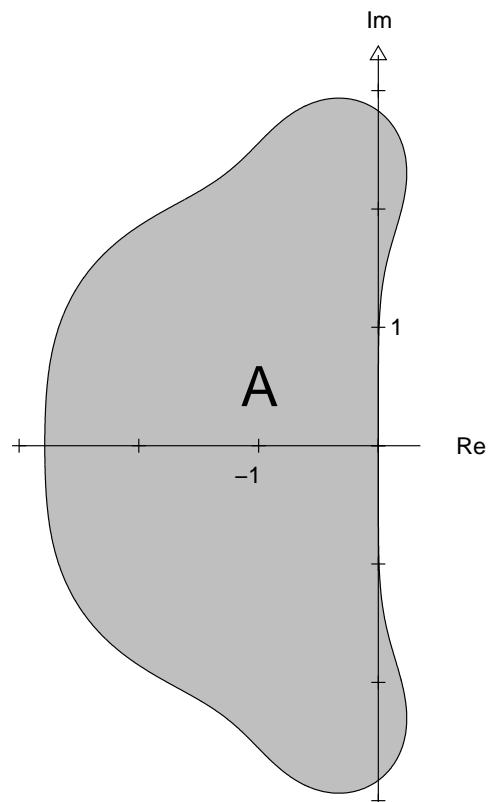


Figure 4. Stabilitetsområden för Runge–Kutta 2 (“explicita trapetsregeln”) och Runge–Kutta 4.

Exempel 5: Linjär skalär ODE med variabla koefficienter

$$\begin{cases} y' = a(t)y + b(t), \\ y(0) = y_0, \end{cases} \quad a_{\min} \leq a(t) \leq a_{\max} < 0.$$

I detta fall blir villkoret för absolutstabilitet att

$$\boxed{ha(t) \in \mathcal{A}, \quad \forall t.}$$

(Mer precist krävs att $ha(t)$ ligger likformigt inuti i \mathcal{A} för alla t . Se sektion 4 nedan.)

Exempel 6: Linjärt system

$$\begin{cases} \mathbf{y}' = A\mathbf{y}, \\ \mathbf{y}(0) = \mathbf{y}_0, \end{cases} \quad \mathbf{y} \in \mathbb{R}^d, \quad A \in \mathbb{R}^{d \times d}.$$

Villkoret här är att

$$\boxed{h\lambda_k \in \mathcal{A}, \quad k = 1, \dots, d,}$$

där λ_k är egenvärdena till A . Notera att λ_k i allmänhet är komplexa även om A är reell. Detta är den viktigaste anledningen till att man definierar stabilitetsområdet i komplexa talplanet och inte nöjer sig med bara reella λ .

Exempel 7: Allmän ODE

$$\begin{cases} \mathbf{y}' = \mathbf{f}(\mathbf{y}), \\ \mathbf{y}(0) = \mathbf{y}_0, \end{cases} \quad \mathbf{y} \in \mathbb{R}^d, \quad \mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d. \quad (3)$$

För allmänna olinjära ODEer som denna är villkoret att

$$\boxed{h\lambda_k(\mathbf{y}(t)) \in \mathcal{A}, \quad k = 1, \dots, d, \quad \forall t,} \quad (4)$$

där $\lambda_k(\mathbf{y})$ är egenvärdena till jakobianen $J(\mathbf{y}) = \frac{\partial \mathbf{f}}{\partial \mathbf{y}}$. Observera att (4) ska gälla likformigt för alla lösningsvärden $\mathbf{y}(t)$ som genomlöps. Stabilitet beror därför på lösningen själv.

Vi säger att en numerisk ODE-metod är absolutstabil för (3) om villkoret i (4) är uppfyllt.

För de enkla linjära fallen i Exempel 5 och 6 ovan förenklas (4) till de villkor vi nämnt tidigare. I det skalära fallet i Exempel 5 är $J = f' = a(t)$. I fallet med system, Exempel 6, är $J = A$.

4 Härledning av stabilitetsvillkoren

Vi illustrerar mekanismen för framåt Euler. Slutsatserna vi drar är dock generella och gäller även för mer sofistikerade metoder.

4.1 Linjär skalär ODE med variabla koefficienter

I detta fall har vi

$$\begin{cases} y' = a(t)y + b(t), \\ y(0) = y_0, \end{cases} \quad a_{\min} \leq a(t) \leq a_{\max} < 0. \quad (5)$$

Notera att $a(t)$ är reell och strikt negativ. En ostörd lösning $\{u_n\}$ med framåt Euler ges då av

$$u_{n+1} = u_n + h[a(t_n)u_n + b(t_n)], \quad u_0 = y_0. \quad (6)$$

Vi betraktar också en lösning $\{z_n\}$ för vilken begynnelsevärdet är stort med ett litet tal δ ,

$$z_{n+1} = z_n + h[a(t_n)z_n + b(t_n)], \quad z_0 = y_0 + \delta. \quad (7)$$

Låt nu $e_n := z_n - u_n$ beskriva effekten av den lilla störningen. Subtraktion av (6) från (7) ger en iterationsformel för $\{e_n\}$,

$$e_{n+1} = e_n + ha(t_n)e_n, \quad e_0 = \delta.$$

En omskrivning liknande den i Exempel 2 ovan ger

$$e_{n+1} = [1 + ha(t_n)]e_n = [1 + ha(t_n)][1 + ha(t_{n-1})]e_{n-1} = \dots = \delta \prod_{k=0}^n [1 + ha(t_k)].$$

Om nu $ha(t) \in \mathcal{A}$ likformigt för alla t , dvs om $|1 + ha(t)| < 1 - \varepsilon$ för alla t och något $\varepsilon > 0$, kommer absolutbeloppet av varje faktor i produkten vara strikt mindre än ett. Det betyder att

$$|e_{n+1}| \leq \delta \prod_{j=0}^n (1 - \varepsilon) = \delta(1 - \varepsilon)^{n+1} \rightarrow 0 \quad \text{när } n \rightarrow \infty.$$

Vi konstaterar att i detta fall är alltså

$$\boxed{ha(t) \in \mathcal{A}, \quad \forall t, \text{ likformigt,}} \quad (8)$$

tillräckligt för att effekten av en liten störning ska försvinna när $n \rightarrow \infty$.

Kommentarer:

- $b(t)$ påverkar inte effekten av störningen.
- Man kan också göra en analys där man har en störning i varje steg, dvs då den störda lösningen $\{z_n\}$ istället för (7) ges av

$$z_{n+1} = z_n + h[a(t_n)z_n + b(t_n) + \delta_n], \quad z_0 = y_0 + \delta_0,$$

för små värden $\{\delta_n\}$. Härledningen blir då mer komplicerad och effekten av störningen går inte alltid mot noll, men under samma villkor som ovan på $ha(t)$ kan man visa att effekten är begränsad av storleken på δ_n ,

$$|e_n| \leq C \max |\delta_n|.$$

4.2 Linjärt system av ODEer

Betrakta nu det linjära systemet

$$\begin{cases} \mathbf{y}' = A\mathbf{y}, \\ \mathbf{y}(0) = \mathbf{y}_0, \end{cases} \quad \mathbf{y} \in \mathbb{R}^d, \quad A \in \mathbb{R}^{d \times d}. \quad (9)$$

På samma sätt som i förra exemplet får vi

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + hA\mathbf{u}_n, & \mathbf{u}_0 &= \mathbf{y}_0, \\ \mathbf{z}_{n+1} &= \mathbf{z}_n + hA\mathbf{z}_n, & \mathbf{z}_0 &= \mathbf{y}_0 + \delta, \end{aligned}$$

och med $\mathbf{e}_n := \mathbf{z}_n - \mathbf{u}_n$,

$$\mathbf{e}_{n+1} = \mathbf{e}_n + hA\mathbf{e}_n, \quad \mathbf{e}_0 = \delta.$$

Antag att A är en diagonaliserbar matris. Då kan den skrivas

$$A = S\Lambda S^{-1}$$

där Λ är en diagonalmatris med egenvärdena λ_n till A på diagonalen, och kolumnerna i S innehåller motsvarande egenvektorer \mathbf{v}_n ,

$$\Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{pmatrix}, \quad S = \begin{pmatrix} | & | & \cdots & | \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_d \\ | & | & \cdots & | \end{pmatrix}.$$

Gör nu ett variabelbyte och sätt $\mathbf{q}_n := S^{-1}\mathbf{e}_n$. Det ger

$$\mathbf{q}_{n+1} = S^{-1}\mathbf{e}_{n+1} = S^{-1}\mathbf{e}_n + hS^{-1}(S\Lambda S^{-1})\mathbf{e}_n = \mathbf{q}_n + h\Lambda\mathbf{q}_n.$$

Betrakta de enskilda komponenterna i $\mathbf{q}_n = (q_n^1, q_n^2, \dots, q_n^d)^T$. Eftersom Λ är diagonal kopplas ekvationerna isär och vi får

$$q_{n+1}^k = q_n^k + h\lambda_k q_n^k, \quad k = 1, \dots, d.$$

Men detta är precis samma form som framåt Euler applicerat på testproblemet (Exempel 2 ovan) och vi vet att $q_n^k \rightarrow 0$ när $h\lambda_k \in \mathcal{A}$. Vi får alltså att hela vektorn

$$\mathbf{q}_n \rightarrow 0 \quad \text{om } h\lambda_k \in \mathcal{A} \text{ för } k = 1, \dots, d.$$

Eftersom $\mathbf{e}_n = S\mathbf{q}_n$ går naturligtvis även \mathbf{e}_n mot noll när \mathbf{q}_n går mot noll och vi konstaterar att i detta fall är

$$\boxed{h\lambda_k \in \mathcal{A}, \quad k = 1, \dots, d,} \quad (10)$$

tillräckligt för att effekten av en liten störning ska försvinna när $n \rightarrow \infty$.

4.3 Allmän ODE

Vi avslutar med det allmänna fallet,

$$\begin{cases} \mathbf{y}' = \mathbf{f}(\mathbf{y}), \\ \mathbf{y}(0) = \mathbf{y}_0, \end{cases} \quad \mathbf{y} \in \mathbb{R}^d, \quad \mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d.$$

För olinjära ODEer som denna är det svårt att ge rigorösa villkor som leder till att effekten av en liten störning försvinner i limes $n \rightarrow \infty$. Genom att anta att störningen hela tiden är liten kan man dock ge en tumregel som fungerar mycket bra i praktiken. Den kan härledas informellt på följande vis. Som tidigare får vi

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + h\mathbf{f}(\mathbf{u}_n), & \mathbf{u}_0 &= \mathbf{y}_0, \\ \mathbf{z}_{n+1} &= \mathbf{z}_n + h\mathbf{f}(\mathbf{z}_n), & \mathbf{z}_0 &= \mathbf{y}_0 + \boldsymbol{\delta}, \end{aligned}$$

och med $\mathbf{e}_n := \mathbf{z}_n - \mathbf{u}_n$,

$$\mathbf{e}_{n+1} = \mathbf{e}_n + h[\mathbf{f}(\mathbf{u}_n) - \mathbf{f}(\mathbf{z}_n)], \quad \mathbf{e}_0 = \boldsymbol{\delta}.$$

För små värden på \mathbf{e}_n har vi att

$$\mathbf{e}_{n+1} \approx \mathbf{e}_n + hJ(\mathbf{u}_n)\mathbf{e}_n, \quad \mathbf{e}_0 = \boldsymbol{\delta}.$$

där $J(\mathbf{y})$ är jakobianen till \mathbf{f} . För ett fixt n är detta på samma form som i Exempel 6. Villkor (10) ska då gälla. På samma sätt som i Exempel 5 måste det vidare gälla uniformt för alla n . Slutgiltiga villkoret blir därför

$$\boxed{h\lambda_k(\mathbf{y}(t)) \in \mathcal{A}, \quad k = 1, \dots, d, \quad \forall t.}$$

5 Styva ODE

Begreppet absolutstabilitet är speciellt viktigt för så kallade *styva* ODEer ("stiff ODEs"). Styva problem innehåller tidsskalor av olika storleksordning, där de snabba tidsskalorna inte bidrar signifikant till lösningens uteseende. Detta är vanligt i tillämpningar som tex diffusionsproblem och kemiska reaktioner. För styva problem är explicita metoder ineffektiva, och problemen hanteras istället vanligtvis med implicita metoder.

Exempel 8: Betrakta den enkla linjära ODEn

$$y' = -y, \quad y(0) = 1,$$

med lösning $y(t) = e^{-t}$. Vi vill använda Framåt Euler för lösa denna med ett fel mindre än 10^{-2} i intervallet $[0, 1]$. Det visar sig att $h = 0.05$ är en tillräckligt liten steglängd. Betrakta nu istället följande 2×2 -system,

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} -10000 & 9999 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad y_1(0) = y_2(0) = 1. \quad (11)$$

Det är lätt att kolla att lösningen ges av $y_1(t) = y_2(t) = e^{-t}$, dvs komponentvis samma som förra fallet. Om vi vill få ett fel mindre än 10^{-2} även i detta fall fungerar det dock inte alls att använda Framåt Euler med samma steglängd, $h = 0.05$ som i förra fallet. Lösningen blir instabil. För att få en stabil lösning måste vi ta en 250 gånger kortare steglängd, $h < 0.0002$, vilket gör att lösningen blir 250 gånger dyrare att beräkna. Vi får då också ett fel som är ca 250 gånger mindre än i första fallet, $\approx 4 \cdot 10^{-5}$.

Exemplet visar det typiska problemet med styva ODE när man löser dem med explicita metoder. På grund av stabilitetskravet måste h väljas väldigt litet. Resultatet blir lösningar som är kostsamma att beräkna och som är "onödigt" noggranna.

Implicita metoder, som tex Bakåt Euler, har normalt inte någon stabilitetsgräns. De är stabila för alla h . Man kan därför välja h enbart med tanke på noggrannhetskraven, vilket ofta gör beräkningarna billigare, trots att implicita metoder är dyrare att implementera.

Exempel 9: När man använder Bakåt Euler med steglängd $h = 0.05$ för att lösa (11) får man ett fel $\approx 10^{-2}$. Den extra kostnaden att göra ett steg med Bakåt Euler jämfört med Framåt Euler uppvägs med marginal av att man kan ta 250 gånger färre steg.

5.1 Teori

Enligt standard felanalys för ODE-lösare beror det lokala trunkationsfelet på hur snabbt den exakta lösningen varierar. I fallet Framåt Euler är det tex begränsat av lösningens andraderivata: i steg n är lokala trunkationsfelet $\tau_n \sim h^2 y''(\xi)/2$ för något $\xi \in (t_n, t_{n+1})$. En viktad summa av de lokala felen ger sedan storleksordningen hos det globala felet. Mer precist blir felet $y(t_n) - u_n \approx Ch$ där konstanten C väsentligen bara beror på den exakta lösningen $y(t)$, via de lokala trunkationsfelen. I Exempel 8 har vi precis samma exakta lösning e^{-t} i båda fallen och man skulle kunna tro att man även i (11) skulle få ett fel mindre än 10^{-2} med steglängden $h = 0.05$. Feluppskattningen och resonemanget ovan gäller dock bara så länge lösningen är stabil.

Från teorin om absolutstabilitet vet vi att eftersom egenvärdena till systemmatrisen i (11) är $\lambda_1 = -10000$ och $\lambda_2 = -1$, måste steglängden väljas mindre än $2/|\lambda_1| = 0.0002$ för att vara stabil. Om vi tar h aningens mindre än 0.0002 gäller återigen feluppskattningen $y(t_n) - u_n \approx Ch$, med samma C , och vi får ett fel som är ca $0.05/0.0002 = 250$ gånger mindre än i första fallet.

Vi kan sammanfatta situationen på följande sätt. I båda fallen finns en *noggrannhetsgräns* $h < h_{\text{nog}} = 0.05$ som måste uppfyllas för att lösningen ska bli tillräckligt noggrann. Det finns också en *stabilitetsgräns* $h < h_{\text{stab}}$ som måste uppfyllas för att lösningen ska vara stabil. I första fallet är $h_{\text{stab}} = 2 \gg h_{\text{nog}}$ och valet av steglängd begränsas av noggrannheten. I det andra fallet är emellertid $h_{\text{stab}} \ll h_{\text{nog}}$ och valet av steglängd begränsas istället av stabiliteten. Lösningen blir "onödigt" noggrann och dyr med Framåt Euler. I dessa fall när stabilitetskravet är mycket mer restriktivt än noggrannhetskravet kallar vi problemet styvt.

Betrakta nu det mer allmänna fallet när ODEn är ett linjärt system

$$\mathbf{y}' = A\mathbf{y},$$

där vi antar att alla egenvärden λ_j till A är reella och negativa. Om \mathbf{x}_j är egenvektorn till λ_j kan lösningen skrivas

$$\mathbf{y}(t) = \sum_{j=1}^d c_j \mathbf{x}_j e^{\lambda_j t},$$

för några konstanter c_j som bestäms av begynnelsedata. Från formen ser vi att vi kan tolka λ_j som systemets olika tidsskalor. Dessa problem är ofta styva när $\min_j |\lambda_j| \ll \max_j |\lambda_j|$, dvs när man har tidsskalor av mycket skilda storleksordningar.

Låt oss göra en icke-rigorös analys som förklarar detta. När $|\lambda_j|$ är stor dör $\mathbf{x}_j \exp(\lambda_j t)$ av snabbt i tiden eftersom $\lambda_j < 0$. Lösningens utseende bestäms efter en kort tid därför väsentligen av de termer som motsvarar små egenvärden. Det betyder att noggrannhetsgränsen bestäms av de *små* egenvärdena. Typiskt blir $h_{\text{nog}} \sim 1/|\lambda_j|$ för något av de små egenvärdena. Stabilitetsgränsen bestäms ju dock av de *stora* egenvärdena, $h_{\text{stab}} = 2/\max_j |\lambda_j|$. Olikheten $h_{\text{stab}} \ll h_{\text{nog}}$ blir därför under dessa antaganden $|\lambda_k| \ll \max_j |\lambda_j|$ för något litet $|\lambda_k|$.

Notera slutligen att allmänna olinjära problem

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}),$$

efter linjärisering runt en punkt lokalt beter sig som ett linjärt system av typen ovan. En sådan ODE kommer därför typiskt vara styv om Jakobianmatrisen $J(\mathbf{y}(t))$ har egenvärden av olika storleksordningar, $\min_j |\lambda_j(J(\mathbf{y}(t)))| \ll \max_j |\lambda_j(J(\mathbf{y}(t)))|$.