Numerisk løsning av ODL

Arne Morten Kvarving

Department of Mathematical Sciences
Norwegian University of Science and Technology

5. November 2007

Problem og framgangsmåte

- Vi vil finne en tilnærming til løsningen av

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y(t))$$
$$y(t_0) = y_0.$$

- Jeg kaller vanligvis den uavhengig variabelen for t, mens Kreyzig bruker x. Grunnen: disse problemene kalles for initialverdiproblem, som får meg til å tenke tid, og x får meg intuitivt til å tenke rom.
- Vi vet at løsningen er gitt ved

$$y(t) = y_0 + \int_{t_0}^{t} f(s, y(s))\,\mathrm{d}s.$$

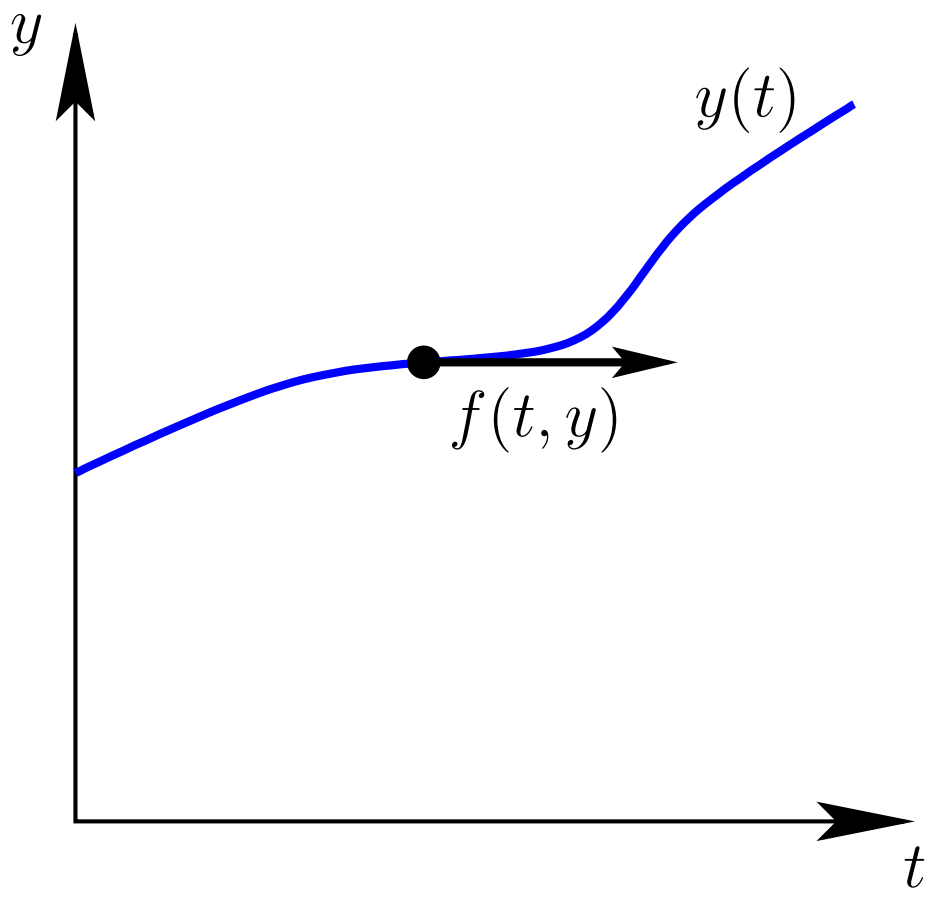
- Fristende å bruke regler for numerisk integrasjon som vi har sett på til nå. Disse kun for autonome system:

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(y(t)),\ y(t_0) = y_0.$$

Problem og framgangsmåte

- Fins to hovedklasser av metoder:
  - Ettskrittsmetoder. Her bruker vi bare y-verdien ved ett tidsnivå for å tilnærme verdien ved neste nivå.
  - Flerskrittsmetoder. Disse bruker y-verdier ved flere tidsnivå for å tilnærme løsningen ved neste. Er ikke pensum.
- I tillegg kan disse metodene deles inn i to deler igjen:
  - Eksplisitte metoder. Dette er metoder hvor vi kun bruker kjente data for å tilnærme løsningen ved neste nivå.
  - Implisitte metoder. Dette er metoder hvor vi er nødt til å løse et ligningssystem ved hvert tidsnivå. Generelt så er dette likningssystemet ikkelineært og må løses med en metode for ikkelineære ligningsysstemer slik som Newtons metode. Dette gjør disse metoden veldig dyre å bruke i praksis. Men i enkelte tilfeller lønner det seg allikevel - for *stive ligninger*.

Eulers metode



I ethvert punkt gir diffligningen retningen.

Eulers metode

- Naiv metode: I ethvert punkt beskriver f tangenten til løsningen.
- Vi går framover langs denne tangenten med et skritt av lengde h.
- Dette gir metoden

$$y_{n+1} = y_n + hf(t_n, y_n).$$

  Kalles for Eulers metode.
- Kan også motiveres fra Taylorutvikling:

$$\begin{aligned} y(t_{n+1}) &= y_n + hy'_n + \frac{h^2}{2}y''_n + \mathcal{O}\left(h^3\right) \\ &= y_n + hf(y_n) + \frac{h^2}{2}y''_n(t_n) + \mathcal{O}\left(h^3\right). \end{aligned}$$

Eulers metode

- Fra Taylorutviklingen ser vi at i ett steg har vi en feil som er $\mathcal{O}\left(h^2\right)$. Dette kalles for *lokalfeilen*: Anta at vi starter med eksakt verdi $y_n = y(t_n)$,

$$|y(t_n + h) - y_{n+1}| = \frac{h^2}{2} y''(t_n) + \mathcal{O}\left(h^3\right) = \mathcal{O}\left(h^2\right)$$

- Denne feilen forplanter seg siden vi tar mange steg.
- Kan vises at denne forplantningen gir oss en orden mindre i *globalfeilen*. Anta at vi starter med $y_0 = y(t_0)$ og gjør n steg av lengde h:

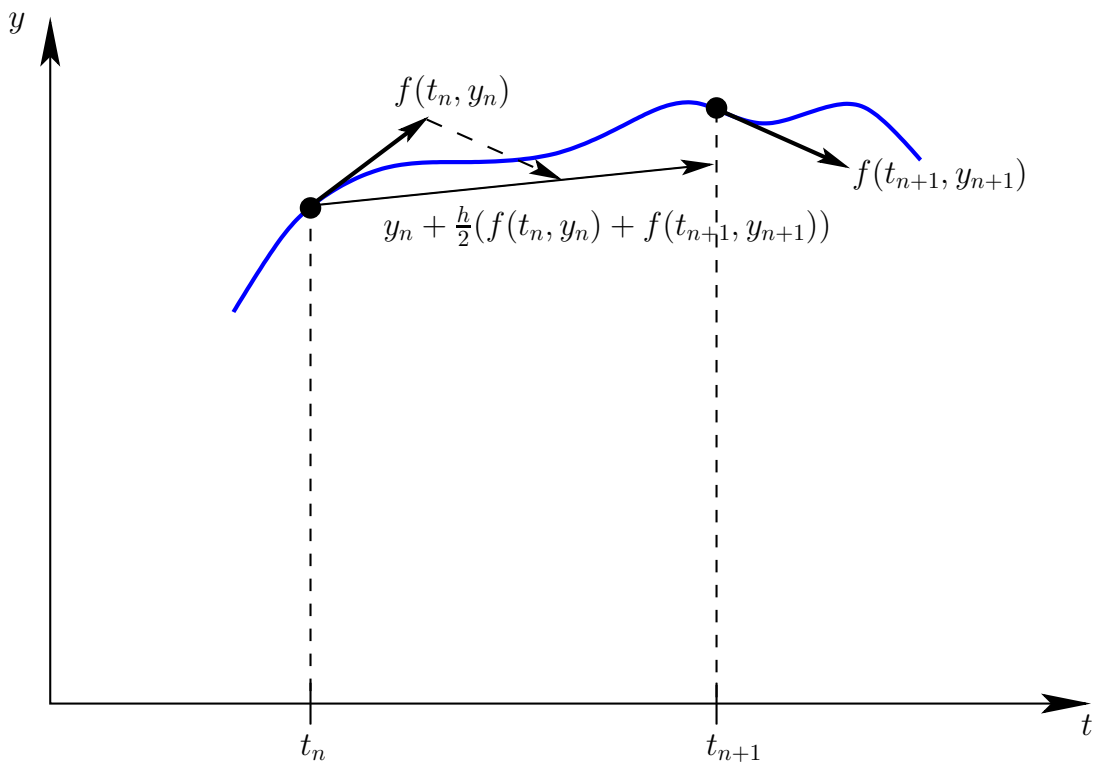
$$|y(t_0 + nh) - y_n| = \mathcal{O}\left(h\right)$$

Altså er Eulers metode en førsteordens metode.

Eulers metode

- Sammenfaller med rektangelregelen for numerisk integrasjon.
- “Brukes aldri i praksis” - Erwin Kreyzig.
- Ikke sant. Brukes av og til når det er nødvendig (dyr f..)
- “Førsøkes unngått” bedre.
- Ofte utgangspunkt for konstruksjon av andre metoder.

Heuns metode: Forbedret Euler



Vi bruker snittet av to tangenter.

Heuns metode: Forbedret Euler

- Euler bruker bare tangent i startpunktet.
- Hvis dere husker trapesregelen - bruk snittet av start og slutt på intervallet.
- Dette gir trapesmetoden for ODL:

$$y_{n+1} = y_n + \frac{h}{2}\left(f\left(t_n, y_n\right) + f\left(t_{n+1}, y_{n+1}\right)\right)$$

  Som vi ser er dette et eksempel på en implisitt metode.
- Men hva skjer hvis vi erstatter $f(t_{n+1}, y_{n+1})$ med $f(t_{n+1}, y_n + hf(t_n, y_n))$? Vi approksimerer altså løsningen i sluttidspunktet vha et vanlig Eulersteg.

Heuns metode: Forbedret Euler

- Dette gir Heuns metode:

$$
\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_{n+1}, y_n + hk_1) \\
y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2)
\end{aligned}
$$

  Legg merke til måten vi skriver opp metoden på - grunnen vil dere skjønne senere.

- Ser på Taylorutviklingen:

$$
\begin{aligned}
|y(t_n + h) - y_n| &= hy'(t_n) + \frac{h^2}{2}y''(t_n) + \mathcal{O}\left(h^3\right) \\
&= hf(y_n) + \frac{h^2}{2}f'(y_n) + \mathcal{O}\left(h^3\right)
\end{aligned}
$$

Heuns metode: Forbedret Euler

- Den deriverte kan approksimeres ved

$$f'(y_n) = \frac{f(y_{n+1}) - f(y_n)}{h} + \mathcal{O}(h)$$
$$\approx \frac{f(y_n + hf(y_n)) - f(y_n)}{h} + \mathcal{O}(h)$$

  Feilen vi gjør ved å approksimere $f(y_{n+1})$ vha et Eulersteg er som vi vet $\mathcal{O}(h^2)$ og pga h i nevneren blir denne feilen også $\mathcal{O}(h)$ så den blir "spist" opp av $\mathcal{O}(h)$ leddet.

- Sette vi dette inn i Taylorutviklingen finner vi

$$|y(t_n + h) - y_n| = hf(y_n)$$
$$+ \frac{h^2}{2}\left(\frac{f(y_n + hf(y_n)) - f(y_n)}{h}\right)$$
$$+ \frac{h^2}{2}\mathcal{O}(h) + \mathcal{O}(h^3).$$

Heuns metode: Forbedret Euler

- Fra dette ser vi at Heuns metode har en tredjeordens lokalfeil, og følgelig en andre ordens globalfeil - den er en metode av orden 2.
- Metoden kan også skrives på formen

$$y_{n+1}^* = y_n + hf(t_n, y_n)$$
$$y_{n+1} = y_n + \frac{h}{2}\left(f\left(t_n, y_n\right) + f\left(t_{n+1}, y_{n+1}^*\right)\right)$$

  Når metoden er skrevet på denne formen ser vi at det er en såkalt *predictor-corrector*-metode.
- Først "forutsier" vi en løsning i det første steget, så "retter" vi på denne i det andre.

Runge-Kutta-metoder

- Systematisering av steget Euler $\rightarrow$ Heun.
- Vi beregner $k_1, k_2, k_3, \cdots, k_s$ verdier inni intervallet. Disse blir beregnet av metoder med lavere orden.
- Til slutt bruker vi en vektet sum av disse k'ene.
- Genrelt kan metodene skrives som

$$k_r = f\left(t_n + c_r h, y_n + h\sum_{j=1}^{r} a_{rj} k_j\right), \quad r = 1, 2, 3, \cdots, s$$

$$y_{n+1} = y_n + h\sum_{r=1}^{s} b_r k_r$$

Her kalles s antall *nivåer*.

Runge-Kutta-metoder

- Finnes uendelig mange metoder av vilkårlig orden (og finne metoder med høy orden er langt fra trivielt dog) og dette er metoder som OFTE brukes i praksis.
- Disse metodene kan klassifiseres ved
  - matrisen $\underline{A}$ (hvor mye vi skal bruke av hver k-verdi på hvert nivå).
  - vektoren $\underline{c}$ (på hvilket tidspunkt prøver vi å tilnærme tangenten på dette nivået?).
  - vektoren $\underline{b}$ (vektene av hver k-verdi i siste steg).
- Denne informasjonen presenteres ofte i et *RK-tableaux* på formen

$$\begin{array}{c|c} \underline{c} & \underline{A} \\ \hline & \underline{b}^T \end{array}$$

Runge-Kutta-metoder

- Vi skal kun se på eksplisitte metoder. Dette betyr at
  - Det første nivået vil alltid være kun en evaluering i starttidspunktet - det er den eneste informasjonen vi har tilgjengelig.
  - Matrisen $\underline{A}$ må være strengt nedretriangulær (ingen diagonal heller).
- Våre tableaux blir altså på formen

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & \cdots \\ c_2 & a_{21} & 0 & \vdots & \vdots \\ c_3 & a_{31} & a_{32} & 0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \\ \hline & b_1 & b_2 & b_3 & \cdots \end{array}$$

Runge-Kutta-metoder

- Altså kan metodene være skrives på formen

$$k_1 = f\left(t_n, y_n\right)$$

$$k_r = f\left(t_n + c_r h, y_n + h\sum_{j=1}^{r-1} a_{rj} k_j\right), \quad r = 2, 3, \cdots, s$$

$$y_{n+1} = y_n + h\sum_{r=1}^{s} b_r k_r$$

Runge-Kutta-metoder

- La oss skrive opp tableauxet for Heuns metode, dvs metoden

$$
\begin{aligned}
k_1 &= f(t_n, y_n) \\
k_2 &= f(t_{n+1}, y_n + hk_1) \\
y_{n+1} &= y_n + \frac{h}{2}(k_1 + k_2)
\end{aligned}
$$

- Vi har $s = 2$ nivå. Vi har $c_1 = 0$ (siden en eksplisitt metode), $c_2 = 1$, $b_1 = \frac{1}{2}$ og $b_2 = \frac{1}{2}$.
- Siden dette er en eksplisitt tonivå metode har vi kun en koeffisient i matrisen $\underline{A}$, nemlig $a_{21} = 1$. Dette gir tilsammen tableauxet

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

Runge-Kutta-metoder

- Her skal vi se på en veldig ofte brukt metode, ERK4.
- Som navnet antyder er dette en eksplisitt Runge-Kutta metode av orden 4.
- Tableauxet er gitt ved

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

Runge-Kutta-metoder

- Hvis vi nå bruker informasjonen kan vi skrive metoden eksplisitt som

$$
\begin{aligned}
k_1 &= f\left(t_n, y_n\right) \\
k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \\
k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right) \\
k_4 &= f\left(t_n + h, y_n + hk_3\right) \\
y_{n+1} &= y_n + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)
\end{aligned}
$$

- Denne metoden sammenfaller med Simpsons metode for autonome systemer.

Feilkontroll - adaptive løsere

- Av stor interesse å ha kontroll på hvor stor feil vi gjør.
- For eksempel kan vi nok en gang bruke motivasjonen fra adaptiv integrasjon: Vi vil gjerne bruke liten h der løsningen variérer mye, mens vi kan nøye oss med større h der løsningen varierer mindre.
- To fremgangsmåter:
  - Som i numerisk integrasjon - bruk forskjellige h.
  - Bruk to forskjellige metoder med forskjellig orden.

Feilkontroll - variasjon av steglengde

- Bruker samme metode, først med steglengde $2h$, så med steglengde h.
- Som et eksempel ser vi her på ERK4.
  - Først bruker vi steglengde $2h$ for å oppnå tilnærming $\tilde{y}$. Dette gir oss

$$y(2h) - \tilde{y} = \frac{1}{2} C (2h)^5 = 16\epsilon$$

    Dvs vi har $\epsilon = Ch^5$. Vi har faktoren $\frac{1}{2}$ siden vi gjør halvparten så mange steg med lengde $2h$.
  - Så gjør vi to steg med lengde h for å oppnå $\tilde{\tilde{y}}$. Dette gir oss

$$y(2h) - \tilde{\tilde{y}} = \epsilon$$

  - Vi har nå to uttrykk for $y(2h)$ og vi finner

$$y(2h) = \tilde{y} + 16\epsilon = \tilde{\tilde{y}} + \epsilon \Rightarrow$$

$$\epsilon = \frac{1}{15} (\tilde{\tilde{y}} - \tilde{y})$$

Feilkontroll - variasjon av steglengde

- Disse metodene er veldig dyre å bruke i praksis. Hvorfor? Jo, fordi vi gjør en god del ekstra arbeid bare for å være i stand til å estimere feilen vi gjør i ett steg.
- Ønskelig å finne metoder hvor feilestimatet er tilnærmet "gratis", altså at vi ikke må gjøre mye ekstra arbeid bare for å estimere feilen.
- En måte å oppnå dette på er å variere orden istedet for steglengde.

Feilkontroll - variasjon av orden

- Eksempel: RKF45

$$\text{RKF4} : \tilde{y} = y(h) + \mathcal{O}\left(h^5\right)$$
$$\text{RKF5} : \tilde{\tilde{y}} = y(h) + \mathcal{O}\left(h^6\right)$$
$$\tilde{\tilde{y}} - \tilde{y} \approx Ch^5.$$

  Vi antar her at $Ch^6 << Ch^5$. Dermed har vi et feilestimat for løsning gitt ved $RKF4$.

- Vi kan parre mange metoder, for eksempel Euler-Heun. Det viktige er at vi kan evaluere metoden med høyere orden billig. For Runge-Kutta-metoder betyr dette at vi leter etter metoder som bruker de samme k-verdiene, dvs den eneste forskjellen på metodene bør være vektoren $\underline{b}$.

Oppsummering - eksplisitte metoder

Metode	Funksjonsevalueringer	Globalfeil	Lokalfeil
Euler	1	1	2
Heun	2	2	3
RK4	4	4	5
RKF4	5	4	5
RKF5	6	5	6
RKF	6	4	5

Legg merke til at RKF4 er en dårlig metode for "ren" 4. orden da vi bruker 5 funksjonsevalueringer.

Oppsummering - eksplisitte metoder

- Alle metodene vi har sett på fungerer fint for system av førsteordens differensialligninger.
- Formlene er akkurat de samme - du må bare gjøre alt på vektoriell form.
- Fungerer også for høyere ordens differensialligninger.
- Hvis vi har gitt en skalar m'te orden differensialligning

$$y^{(m)} = f\left(t, y, y', y'', \cdots, y^{(m-1)}\right),$$

  kan vi skrive den om til et system av førsteordens differensialligninger slik at metodene våre kan anvendes.
- Derfor lite fokus på metoder for høyereordens ligninger. Ett unntak er Runge-Kutta-Nyström metoder.

Hvordan lage system ut av en høyereordens ligning

- Gitt en skalar m'te orden differensialligning

$$y^{(m)} = f\left(t, y, y', y'', \cdots, y^{(m-1)}\right).$$

- Vi definerer vektorkomponenter

$$\begin{aligned} y_1 &= y \\ y_2 &= y' \\ y_3 &= y'' \\ &\vdots \\ y_m &= y^{(m-1)}. \end{aligned}$$

Dette er i bunn og grunn bare en ny navngivning.

Hvordan lage system ut av en høyereordens ligning

- Innfører vi dette får vi vårt system med ligninger:

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ &\vdots \\ y_m' &= f\left(t, y_1, y_2, y_3, \cdots, y_m\right) \end{aligned}$$

- En anvendelse av dette med spesiell interesse: Hvordan skrive en ikke-autonom ligning som et autonomt system. Vi innfører

$$\begin{aligned} y_1 &= t \\ y_2 &= y \end{aligned}$$

og kan dermed skrive ligningen vår som

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} 1 \\ f\left(\underline{y}\right) \end{pmatrix}.$$

Runge-Kutta-Nyström metoder

- Betrakter ligningen

$$y'' = f\left(t, y, y'\right)$$

- Vi lager et system ut av det

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= f\left(t, y_1, y_2\right) \end{aligned}$$

- Anta at f ikke avhenger av y'. Det viser seg da at hvis vi anvender (spesielle) RK-metoder på systemet så vil to eller flere nivå sammenfalle, og dermed blir metodene billigere. Disse metodene kalles for RKN-metoder.

Hvorfor bruke implisitte metoder - stive ligninger

- Betrakt ligningen (kjent som Dahlquists testligning)

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \lambda y$$
$$y(0) = y_0$$

- Vi vet at den eksakte løsningen er gitt som

$$y(t) = y_0 e^{\lambda t}.$$

  Denne løsningen eksisterer for $t \to \infty$ hvis og bare hvis $\mathrm{Re}\ \lambda < 0$. Dette er oppførsel vi gjerne vil gjenskape i metodene våre.

- Hva skjer hvis vi anvender Eulers metode på denne ligningen?

$$y_{n+1} = y_n + hf(t_n, y_n) = y_n + h\lambda y_n = (1 + h\lambda)\, y_n$$

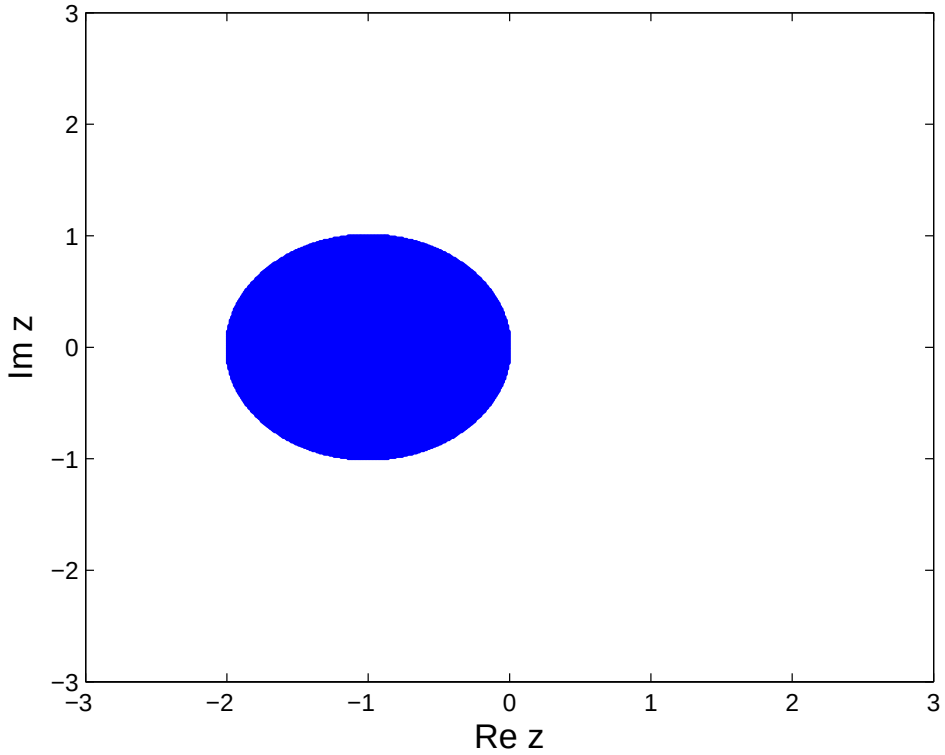
  Anvender denne rekursivt og får

$$y_n = (1 + h\lambda)^n\, y_0.$$

Hvorfor bruke implisitte metoder - stive ligninger

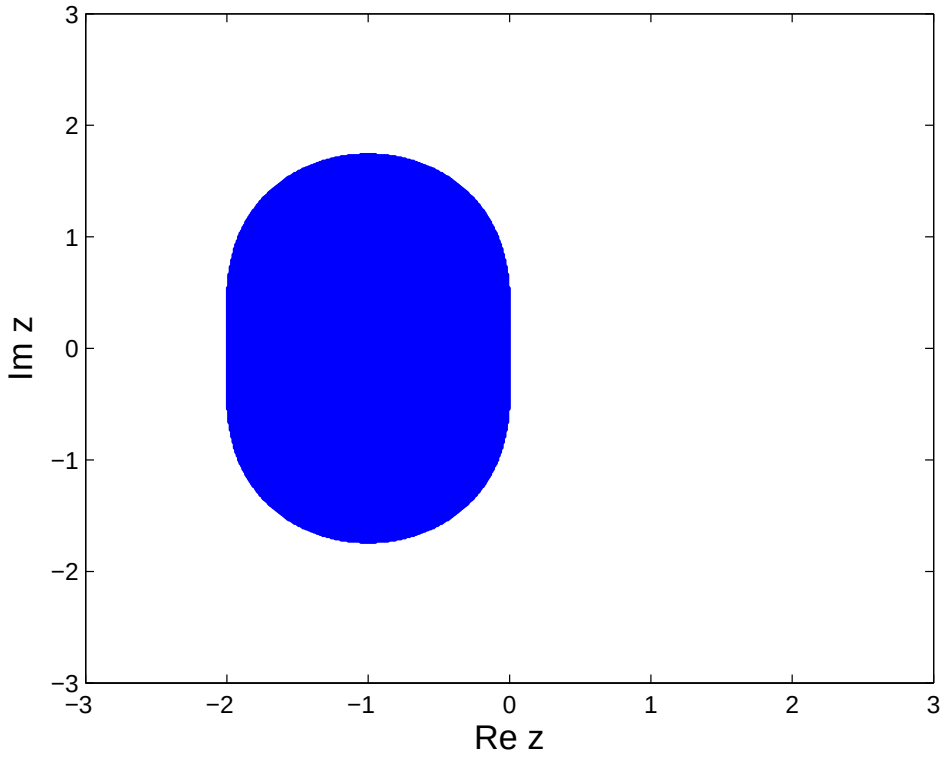
- Så hvis metoden skal gi en løsning for $t \to \infty \Leftrightarrow n \to \infty$ så må
$$|1 + h\lambda| < 1.$$
- Vi setter nå $z = h\lambda$. Området hvor vi får en stabil løsning er plottet i figuren under. Husk, den eksakte løsningen blir dempet for alle $\lambda \in \mathbb{C}^-$ mens metoden bare demper i den lille fylte området.

Hvorfor bruke implisitte metoder - stive ligninger



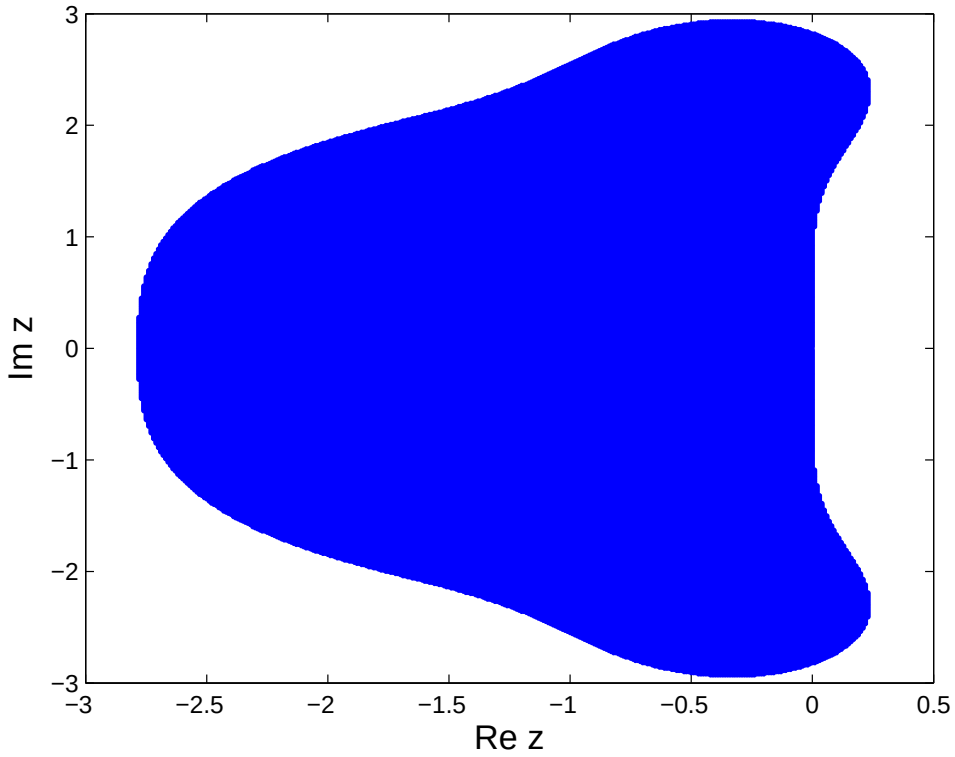
Stabilitetsområde for Eulers metode.

Hvorfor bruke implisitte metoder - stive ligninger



Stabilitetsområde for Heuns metode.

Hvorfor bruke implisitte metoder - stive ligninger



Stabilitetsområde for RK4.

Hvorfor bruke implisitte metoder - stive ligninger

- Dette betyr at hvis λ er en stor verdi må vi bruke veldig liten h for å holde oss innenfor stabilitetsområdet.
- Hvordan relateres dette til generelle ligninger? La oss se for oss at vi har et system med ligninger på formen

$$\frac{\mathrm{d}\underline{y}}{\mathrm{d}t} = \underline{A}\,\underline{y}$$

og at matrisen $\underline{A}$ kan diagonaliseres, dvs vi kan finne $\underline{Q}$ og $\underline{\Lambda}$ slik at

$$\underline{Q}\,\underline{\Lambda}\,\underline{Q}^{T} = \underline{A}\,.$$

hvor $\underline{Q}$ er en ortogonal matrise med egenvektorene til $\underline{A}$ som kolonner og $\underline{\Lambda}$ en diagonal matrise med egenverdiene til $\underline{A}$ langs diagonalen.

Hvorfor bruke implisitte metoder - stive ligninger

- Setter så dette inni i ligningen:

$$\begin{aligned}\frac{\mathrm{d}\underline{y}}{\mathrm{d}t} &= \underline{A}\,\underline{y} \\ &= \underline{Q}\,\underline{\Lambda}\,\underline{Q}^{T}\underline{y}\end{aligned}$$

Siden $\underline{Q}$ er ortogonal er inversen gitt ved $\underline{Q}^{-1} = \underline{Q}^{T}$. Multipliserer med denne fra venstre:

$$\begin{aligned}\underline{Q}^{T}\frac{\mathrm{d}\underline{y}}{\mathrm{d}t} &= \underline{\Lambda}\,\underline{Q}^{T}\underline{y} \Rightarrow \\ \frac{\mathrm{d}\underline{z}}{\mathrm{d}t} &= \underline{\Lambda}\,\underline{z}\end{aligned}$$

hvor vi har satt $\underline{z} = \underline{Q}^{T}\underline{y}$.

Hvorfor bruke implisitte metoder - stive ligninger

- Vi har altså en ligning av Dahlquist type langs hver egenverdi av matrisen $\underline{A}$ (husk, $\underline{\Lambda}$ er diagonal og følgelig er det ingen koblinger mellom ligningene).
- Vi må holde oss innenfor stabilitetsområdet for alle egenverdier - hvilket betyr at den største egenverdien dikterer steglengden vi kan bruke.
- Hvis $\underline{A}$ har en egenverdi med veldig stor verdi kalles ligningen for *stiv*.

Hvorfor bruke implisitte metoder - stive ligninger

- Hva skjer så for en implisitt metode? La oss betrakte trapesmetoden anvendt på testligningen:

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

$$\left(1 - \frac{h\lambda}{2}\right) y_{n+1} = \left(1 + \frac{h\lambda}{2}\right) y_n \Rightarrow$$

$$y_n = \left(\frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}}\right)^n y_0$$

- Området hvor dette gir en stabil løsning inkluderer hele det venstre halvplanet. Dvs, metoden vår gir en stabil løsning overalt hvor den eksakte løsningen eksisterer. Denne egenskapen kalles *A-stabilitet* og det kan vises at ingen eksplisitte metoder kan være A-stabile. Dette betyr altså at vi har *ingen* restriksjoner på hvilke steglengder vi kan bruke.

Hvorfor bruke implisitte metoder - oppsummert

- Noen ganger er restriksjonene på steglengden vi får når vi bruker en eksplisitt metode så streng at de blir nærmest ubrukelige i praksis.
- Vi tyr da til implisitte løsere. Selv om vi må løse et system av ligninger for hvert steg, blir den totale algoritmen billigere siden vi kan velge steglengde kun basert på nøyaktighetsbetraktninger.