

# Sammanfattning av SF1544 Numeriska metoder, grundkurs

Yashar Honarmandi  
yasharh@kth.se

17 januari 2019

## **Sammanfattning**

Detta är en sammanfattning av SF1544 Numeriska metoder, grundkurs.

## Innehåll

1	Användbar matte	1
2	Grundläggande definitioner för numeriska metoder	3
3	Lösning av ekvationer	3
4	Interpolation	7
5	Optimering	7
6	Lösning av ordinarie differentialekvationer	8

# 1 Användbar matte

**Allmän begränsning av globalt fel** Betrakta begynnelsevärdesproblemet

$$\begin{aligned}\frac{dy}{dt}(t) &= f(t, y(t)), \\ y(a) &= b,\end{aligned}$$

löst på  $[a, T]$ , där  $f$  är Lipschitzkontinuerlig. Betrakta en numerisk lösning med lokalt fel begränsad av  $Mh^{p+1}$ . Då begränsas det globala felet av

$$|y(T) - y_N| \leq \frac{e^{L(T-a)}M}{L}h^p.$$

**Bevis** Vi inför  $y(t; t_n)$  som den exakta lösningen som startar i  $(t_n, y_n)$ . Det globala felet ges då av

$$\begin{aligned}|y(T) - y_N| &= |y(T) - y(T; t_{N-1}) + y(T; t_{N-1}) + \dots - y(T; t_1) + y(T; t_1) - y_N| \\ &\leq |y(T) - y(T; t_{N-1})| + |y(T; t_{N-1}) - y(T; t_{N-2})| + \dots + |y(T; t_1) - y_N|.\end{aligned}$$

Den första termen ges simpelthen av det lokala felet. Satsen om entydighet av lösning för en sådan differentialekvation ger vidare

$$|y(T; t_i) - y(T; t_{i-1})| \leq e^{L(T-t_i)}|y(t_i; t_i) - y(t_i; t_{i-1})|.$$

Det som står kvar i absolutbeloppstecknet är det lokala felet, eftersom den vänstra termen är exakt och den högra kommer från en iteration. Detta ger

$$|y(T; t_i) - y(T; t_{i-1})| \leq e^{L(T-t_i)}Mh^{p+1} = e^{L(N-i)h}Mh^{p+1}$$

och vidare

$$\begin{aligned}|y_N - y(T)| &\leq Mh^{p+1} + Mh^{p+1}e^{Lh} + \dots + Mh^{p+1}e^{L(N-1)h}. \\ &= Mh^{p+1}\frac{1 - e^{LNh}}{1 - e^{Lh}} \\ &= Mh^{p+1}\frac{e^{LNh} - 1}{e^{Lh} - 1} \\ &\leq Mh^{p+1}\frac{e^{LNh}}{Lh},\end{aligned}$$

och beviset är klart.

**Möjlighet för polynominterpolation** Givet  $n + 1$  punkter  $(x_i, y_i)$ , där  $x_i \neq x_j$  om  $i \neq j$ , finns det ett entydigt polynom  $p$  av grad (högst)  $n$  så att  $p(x_i) = y_i \ \forall i = 1, \dots, n + 1$ .

**Bevis** För att visa existensen av  $p$ , kan vi välja det som

$$p(x) = \sum_{i=1}^{n+1} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}.$$

Denna metoden kallas för Lagrangeinterpolation. Vi ser att  $p$  har grad  $n$  och

$$p(x_k) = \sum_{i=1}^{n+1} y_i \prod_{j \neq i} \frac{x_k - x_j}{x_i - x_j}.$$

För alla termer i summan där  $i \neq k$  kommer det finnas en faktor  $x_k - x_k$  i nämnaren, och dessa ger inget bidrag. För  $i = k$  blir produkten bara 1, och  $p(x_k) = y_k$ .

För att visa att  $p$  är entydig, antag att  $q$  är ett annat interpolationspolynom och bilda  $v = q - p$ , med grad högst  $n$ . Detta ger att  $v$  har  $n + 1$  nollställen. Detta är endast möjligt om  $v = 0$ , och därmed måste  $p$  vara unikt.

**Fel för linjär interpolation** Antag att  $f \in C^2$ . Låt  $p$  vara det linjära polynomet som interpolerar punkterna  $(x_0, f(x_0))$  och  $(x_0 + h, f(x_0 + h))$ . Då gäller  $|f(x) - p(x)| \leq Ch^2$ ,  $x_0 \leq x \leq x_0 + h$ , där

$$C = \max_{x_0 \leq z \leq x_0 + h} \frac{\left| \frac{d^2 f}{dx^2}(z) \right|}{8}.$$

**Bevis** Antag  $x_0 = 0$ , och bilda  $g = f - p$ .  $g$  är kontinuerlig på  $[0, h]$ , och antar därmed ett största och minsta värde på detta intervallet. Antag att den antar sitt största värde i  $x = a$ . Taylorutveckling kring  $a$  ger

$$g(x) = g(a) + \frac{dg}{dx}(a)(x - a) + \frac{1}{2} \frac{d^2 g}{dx^2}(c)(x - a)^2$$

för något  $c \in [a, x]$ . Vi ser att andra termen måstse bli 0, ty  $a$  är en max-punkt. Evaluering av Taylorutvecklingen i  $x = 0$  eller  $x = h$  (den som är närmast  $a$ ) ger  $g(x) = 0$  och

$$g(a) = -\frac{1}{2} \frac{d^2 g}{dx^2}(c)(x - a)^2, |g(a)| \leq \frac{1}{2} \max_{x_0 \leq z \leq x_0 + h} \left| \frac{d^2 g}{dx^2}(z) \right| \left( \frac{1}{2} h \right)^2.$$

Vi vet att  $\frac{d^2 f}{dx^2}(x) = \frac{d^2 g}{dx^2}(x)$  eftersom  $p$  är linjär, vilket ger

$$|g(a)| \leq \max_{x_0 \leq z \leq x_0 + h} \frac{\left| \frac{d^2 f}{dx^2}(z) \right|}{8} h^2.$$

## 2 Grundläggande definitioner för numeriska metoder

### Reduktionsfaktor

**Kvadratisk konvergens** En numerisk metod vars feltermar uppfyller

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n^2} = M$$

är kvadratisk konvergent.

## 3 Lösning av ekvationer

**Fixpunktsmetoden** Betrakta ekvationen

$$x = g(x).$$

Fixpunktsmetoden är en enkel iterationsmetod för att lösa denna ekvationen, med den enkla iterationsformeln

$$x_{n+1} = g(x_n).$$

En pseudokod-beskrivning av en lösning med startvillkor  $x_0$  där det itereras tills lösningen stämmer med en tolerans  $\epsilon$  är:

```
define g(x)
input x0
input t
while abs(x - g(x)) > t
    x = g(x)
end
return x
```

**Konvergens** Om  $g \in C^1$ ,  $\left| \frac{dg}{dx}(\alpha) \right| < 1$  och  $\alpha$  är en fixpunkt, finns det en omgivning till  $\alpha$  så att om  $x_0$  är i denna omgivningen, går  $x_n \rightarrow \alpha$ . Metoden konvergerar linjärt med reduktionsfaktor  $S = \left| \frac{dg}{dx}(\alpha) \right|$ .

För att visa detta, skriver vi

$$x_{n+1} - \alpha = g(x_n) - g(\alpha) = \frac{dg}{dx}(\alpha)(x_n - \alpha),$$

där vi har använt medelvärdesatsen och det faktum att  $\alpha$  är en fixpunkt. Vidare, eftersom  $g \in C^1$  finns det en omgivning till  $\alpha$  så att  $\left| \frac{dg}{dx}(x) \right| \leq \frac{S+1}{2}$ . Om  $x_0$  är i denna, är

$$e_{n+1} \leq \frac{S+1}{2} e_n.$$

Detta implicerar att  $x_n \rightarrow \alpha$  och att

$$\frac{e_{n+1}}{e_n} \rightarrow S.$$

**Fixpunktsmetoden för system** Betrakta ekvationssystemet

$$\mathbf{x} = g(\mathbf{x}).$$

Vi löser även detta med fixpunktsmetoden, och använder iterationsformeln

$$\mathbf{x}_{n+1} = g(\mathbf{x}_n).$$

En pseudokod-beskrivning av en lösning med startvillkor  $\mathbf{x}_0$  där det itereras tills lösningen stämmer med en tolerans  $\mathbf{t}$  är:

```
define g(x)
input x0
input t
while abs(x - g(x)) > t
    x = g(x)
end
return x
```

där allt nu är listor. Toleransen ser kanske lite annorlunda ut, vafan vet jag.

**Intervallhalveringsalgoritmen** Betrakta ekvationen

$$f(x) = 0.$$

Intervallhalveringsalgoritmen utgår från två punkter  $a, b$  så att  $f(a)f(b) < 0$ , och gör följande:

1. Beräkna funktionsvärdet i punkten  $m = \frac{b+a}{2}$ .
2. Om  $f(a)f(m) < 0$ , sätt  $a = m$ . Annars, sätt  $b = m$ .
3. Sluta iterationen när intervallbredden  $\frac{b-a}{2}$  är mindre än den givna toleransen.
4. Returnera  $\frac{b+a}{2}$ .

En pseudokod-beskrivning av en lösning med startvillkor  $a$  och  $b$  där det itereras tills lösningen stämmer med en tolerans  $\mathbf{t}$  är:

```

define f(x)
input a, b
input t
while (b - a)/2 > t
    m = (a + b)/2
    if f(a)f(m) < 0
        a = m
    else
        b = m
    end
end
return (a + b)/2

```

**Newton-Rhasonsmetoden** Betrakta ekvationen

$$f(x) = 0.$$

Newton-Rhasons metod utgår från tangenten till  $f$ . Om man startar i  $x_0$ , har tangenten ekvation  $t(x) = \frac{df}{dx}(x_0)(x - x_0) + f(x_0)$ . Dens nollställe ges av

$$x = x_0 - \frac{f(x_0)}{\frac{df}{dx}(x_0)}.$$

Från detta gör vi iterationen

$$x_{n+1} = x_n - \frac{f(x_n)}{\frac{df}{dx}(x_n)}$$

som avslutas när  $|x_{n+1} - x_n|$  är mindre än någon tolerans. Vi ser att detta är en variant av fixpunktsmetoden med  $g(x) = x - \frac{f(x)}{\frac{df}{dx}(x)}$ .

En pseudokod-beskrivning av en lösning med startvillkor  $x_0$  där det itereras tills lösningen stämmer med en tolerans  $t$  är:

```

define f(x)
define fderiv(x)
input x_0
input t
while f(x_0) > t
    x = x - f(x)/fderiv(x)
end
return x

```

**Konvergens** Om  $\alpha$  är ett nollställe till  $f$ , är det även ett nollställe till  $g$ . Vi har för  $f \in C^2$  och  $\frac{df}{dx}(\alpha) \neq 0$  att

$$\frac{dg}{dx}(\alpha) = 1 - \frac{\frac{df}{dx}(\alpha)}{\frac{df}{dx}(\alpha)} + \frac{f(\alpha)\frac{d^2f}{dx^2}(\alpha)}{\frac{df}{dx}(\alpha)^2} = 0.$$

Därmed, om dessa villkor uppfylls, är Newton-Rhasons metod kvadratisk konvergent med konstant  $M = \frac{\frac{d^2f}{dx^2}(\alpha)}{2\frac{df}{dx}(\alpha)}$ .

För att visa detta, konstaterar vi att lokal konvergens följer av beviset som gjordes för fixpunktsmetoden. Vi Taylorutvecklar vidare nära  $x_n$  och får

$$\begin{aligned} f(\alpha) &= f(x_n) + \frac{df}{dx}(x_n)(\alpha - x_n) + \frac{1}{2}\frac{d^2f}{dx^2}(c_n)(\alpha - x_n)^2, \\ \frac{f(\alpha)}{\frac{df}{dx}(x_n)} &= \frac{f(x_n)}{\frac{df}{dx}(x_n)} + \alpha - x_n + \frac{1}{2}\frac{\frac{d^2f}{dx^2}(c_n)}{\frac{df}{dx}(x_n)}(\alpha - x_n)^2 = 0. \end{aligned}$$

Detta skriver vi om till

$$x_n - \frac{f(x_n)}{\frac{df}{dx}(x_n)} - \alpha = \frac{1}{2}\frac{\frac{d^2f}{dx^2}(c_n)}{\frac{df}{dx}(x_n)}(\alpha - x_n)^2.$$

Detta implicerar

$$\frac{e_{n+1}}{e_n^2} = \left| \frac{\frac{d^2f}{dx^2}(c_n)}{2\frac{df}{dx}(x_n)} \right| \rightarrow M,$$

och beviset är klart.

**Newton-Rhasons metod för system** Betrakta ekvationen

$$\mathbf{f}(\mathbf{x}) = 0.$$

Metoden är den samma för ett enda system. Den utgår från den linjära approximationen

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_0) + d\mathbf{f}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$$

och ger iterationen

$$\mathbf{x}_{n+1} = \mathbf{x}_n - d\mathbf{f}(\mathbf{x}_n)^{-1}\mathbf{f}(\mathbf{x}_n).$$

Notera att beräkningsmässigt är det svårt att hitta en inversmatris, så det är smartare att hitta en vektor  $\mathbf{h}$  så att  $d\mathbf{f}(x_n)\mathbf{h} = \mathbf{f}(x_n)$  och använda den i stället.



## 4 Interpolation

Det fundamentala interpolationsproblemet går ut på att hitta en kurva som bäst möjligt passar med vissa datapunkter. Kom i håg satsen om möjlighet för polynominterpolation.

**Polynominterpolation - första försök** Vi gör först en naiv ansats

$$p(x) = \sum_{i=0}^n c_i x^i$$

och anpassar konstanterna så att  $p$  antar rätt värden i datapunkterna. Detta ger oss ett linjärt system

$$X\mathbf{c} = \mathbf{y},$$

där  $\mathbf{c}$  är en vektor med alla koefficienter,  $\mathbf{y}$  är en vektor med alla  $y$ -värden och  $X_{ij} = x_j^{i-1}$ .  $X$  kallas för en Vandermonde-matris. Om man har många datapunkter, kan detta dock ge upphov till ett illakonditionerat system.

**Newtons interpolationsmetod** Vi gör en ny ansats

$$\begin{aligned} p(x) &= d_0 + d_1(x - x_0) + d_2(x - x_0)(x - x_1) + \dots \\ &= \sum_{i=0}^n d_i \prod_{j<i} (x - x_j). \end{aligned}$$

Vi ser att  $p(x_0) = y_0$ , och man kan från detta få de nästa koefficienterna.

**Minsta kvadratmetoden** Minsta kvadratmetoden är en metod för approximation av överbestämda ekvationssystem, dvs. system med fler ekvationer än obekanta. Sådana system har inget interpolationspolynom.

Sådana ekvationssystem kan formuleras som  $A\mathbf{x} = \mathbf{b}$ . Minsta kvadratlösningen är den lösningen som minimerar  $\|A\mathbf{x} - \mathbf{b}\|$ . Mer specifikt, om vi söker en funktion  $f(x) = \sum_{i=1}^n c_i \phi_i(x)$ , där funktionerna  $\phi_i$  kan väljas som vi vill, är  $A_{ij} = \phi_j(x_i)$ ,  $\mathbf{x}_i = c_i$  och  $\mathbf{b}_i = y_i$ .

Minstakvadratlösningen till detta systemet löser normalekvationerna

$$A^T A \mathbf{x} = A^T \mathbf{b}.$$

Om kolumnerna i  $A$  är linjärt oberoende, har detta en lösning.

## 5 Optimering

Denna delen kommer handla om att försöka hitta minima till en funktion över ett rum. Att hitta maxpunkter är det samma som att byta tecken och hitta minimum, så vi fokuserar därför på att hitta minima.

**Gyllene snitt-sökning** Antag att  $f$  är kontinuerlig och har endast ett lokalt minimum på  $[a, b]$ , och betrakta två punkter  $x_1$  och  $x_2$ . Om  $f(x_1) \leq f(x_2)$  finns minimumspunkten på  $[a, x_2]$ , och om  $f(x_1) \geq f(x_2)$  finns minimumspunkten på  $[x_1, b]$ . Detta gäller eftersom derivatan endast byter tecken en gång på  $[a, b]$ , och vi med hjälp av funktionsvärdena i två punkter får information om derivatan imellan dessa punkterna. Man kan få en ganska säker metod vid att välja både  $x_1$  och  $x_2$  nära mitten av intervallet.

Metoden kan förbättras vid att återanvända funktionsvärden i kommande iterationer. Sätt  $a = b_0, b = 1$ . Vi väljer  $x_1$  och  $x_2$  symmetriskt och likformigt, så att  $x_2 = 1 - x_1$  och

$$\frac{x_1}{x_2} = \frac{x_2}{1}.$$

Vi kan lösa detta och få

$$x_1 = \frac{3 \pm \sqrt{5}}{2}, \quad x_2 = \frac{\sqrt{5} - 1}{2}.$$

Vi ser att  $x_2$  är det gyllene snitt. Med detta val gäller att intervallängden avtar med en faktor  $g$  per iteration.

**Newtons metod** Newtons metod för att hitta minima till en funktion  $f(\mathbf{x})$  är att använda Newtons metod för att lösa ekvationssystemet  $\vec{\nabla} f(\mathbf{x}) = \mathbf{0}$ .

**Gradientmetoden** Vi vet att  $f$  avtar snabbast i riktningen  $-\vec{\nabla} f(\mathbf{x})$ . Vi gör därför iterationen

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \gamma_n \vec{\nabla} f(\mathbf{x}_n).$$

$\gamma_n$  kan väljas konstant eller så att den minimerar  $f(\mathbf{x}_{n+1})$ . Jämförd med Newtons

## 6 Lösning av ordinarie differentialekvationer

**Eulers metod (framåt)** Betrakta begynnelsevärdesproblemet

$$\begin{aligned} \frac{dy}{dt}(t) &= f(t, y(t)), \\ y(a) &= b. \end{aligned}$$

Eulers metod går ut på att

1. Dela upp intervallet vi vill lösa problemet på i diskreta steg, där steg  $n$  finns i punkten  $t_n = a + nh$ , där  $h$  är steglängden.

2. Linjarisera problemet till

$$y_{n+1} - y_n = f(t_n, y_n)h,$$

där  $y_n = y(t_n)$ .

En pseudokod-beskrivning av en lösning med startvillkor  $t_0$  och  $y_0$ , steglängd  $h$  och  $N$  steg är:

```
define f(t, y)
input t0 and y0
input h and N
t = t0
y = y0
for 1 < i < N
    y = y + f(t, y)*h
    t = t + h
end
```

**Felanalys** Om vi betraktar det första steget i iterationen, har man lokalt

$$\begin{aligned} y(t_1) &= y(t_0) + \frac{dy}{dt}(t_0)(t_1 - t_0) + \frac{1}{2} \frac{d^2y}{dt^2}(\alpha)(t_1 - t_0)^2 \\ &= y(t_0) + hf(t_0, y_0) + \frac{1}{2} \frac{d^2y}{dt^2}(\alpha)h^2. \end{aligned}$$

Om andraderivatan av  $y$  är begränsad, ger detta

$$|y_1 - y(t_1)| \leq Mh^2,$$

och det lokala felet är  $O(h^2)$ .

Det globala felet kan nu uppskattas som det lokala felet multiplicerat med antal steg. Om vi försöker lösa ekvationen på intervallet  $[a, T]$  med  $N$  steg, har man

$$Nh = T - a,$$

och det globala felet kan uppskattas som

$$|y_N - y(t_N)| \approx h^2 \frac{T - a}{h} = Ch.$$

**Eulers metod för system av differentialekvationer** Betrakta begynnelsevärdesproblemet

$$\begin{aligned} \frac{d\mathbf{y}}{dt}(t) &= \mathbf{f}(t, \mathbf{y}(t)), \\ \mathbf{y}(a) &= \mathbf{b}. \end{aligned}$$

Eulers metod går ut på att

1. Dela upp intervallet vi vill lösa problemet på i diskreta steg, där steg  $n$  finns i punkten  $t^n = a + nh$ , där  $h$  är steglängden.
2. Linjarisera problemet till

$$y^{n+1} - y_n = \mathbf{f}(t^n, y^n)h,$$

där  $\mathbf{y}^n = \mathbf{y}(t^n)$ .

En pseudokod-beskrivning av en lösning med startvillkor  $\mathbf{t}_0$  och  $\mathbf{y}_0$ , där denna är en lista med  $M$  element, steglängd  $h$  och  $N$  steg är:

```

define f(t, y)
input t0 and y0
input h and N
t = t0
y = y0
for 1 < i < N
    for 1 < j < M
        y[j] = y[j] + f[j](t, y)*h
    end
    t = t + h
end

```

Observera att  $\mathbf{f}$  nu är en lista av  $M$  funktioner, och kom ihåg att högre ordningens ekvationer med en funktion kan skrivas som ett system av differentialekvationer.

**Eulers metod bakåt** Betrakta begynnelsevärdesproblemet

$$\begin{aligned}\frac{dy}{dt}(t) &= f(t, y(t)), \\ y(a) &= b.\end{aligned}$$

Eulers metod bakåt går ut på att

1. Dela upp intervallet vi vill lösa problemet på i diskreta steg, där steg  $n$  finns i punkten  $t_n = a + nh$ , där  $h$  är steglängden.
2. Linjarisera problemet till

$$y_{n+1} - y_n = f(t_{n+1}, y_{n+1})h,$$

där  $y_n = y(t_n)$ . Detta ger en ekvation i  $y_{n+1}$  som måste lösas numeriskt.