

# Numeriska metoder och MATLAB för lösning av differentialekvationer

Joel Nilsson, ME06  
joelni at kth.se

Martin Axelsson, ME06  
maxels at kth.se

Fredrik Lundgren, ME06  
flundg at kth.se

DN1215 - Numeriska metoder för ME - Laboration 2

2008-02-29

### **Sammanfattning**

En ingenjör som skall lösa problem idag måste kunna använda sig utav numeriska metoder och datorer för att lösa problem. Ett populärt och bra verktyg till detta är MATLAB. Det har undersökts hur polynominterpolation görs med hjälp av MATLAB och hur dess inbyggda lösare för differentialekvationer fungerar. Resultaten visar att storleken på fel och beräkningstider varierar kraftigt beroende på vilka metoder och lösare man tillämpar.

## Innehåll

<b>1</b>	<b>Inledning</b>	<b>2</b>
<b>2</b>	<b>Problem och syfte</b>	<b>3</b>
2.1	Polynominterpolation . . . . .	3
2.2	Arenstorf-banan . . . . .	3
2.3	Kemisk reaktionskinetik . . . . .	4
<b>3</b>	<b>Metod och utförande</b>	<b>5</b>
3.1	Polynominterpolation . . . . .	5
3.2	Arenstorf-banan . . . . .	5
3.3	Kemisk reaktionskinetik . . . . .	6
<b>4</b>	<b>Resultat</b>	<b>7</b>
4.1	Polynominterpolation . . . . .	7
4.2	Arenstorf-banan . . . . .	7
4.3	Kemisk reaktionskinetik . . . . .	8
<b>5</b>	<b>Diskussion och slutsats</b>	<b>9</b>
5.1	Polynominterpolation . . . . .	9
5.2	Arenstorf-banan . . . . .	9
5.3	Kemisk reaktionskinetik . . . . .	9
<b>A</b>	<b>Teoretiska frågor med svar</b>	<b>10</b>

## 1 Inledning

Att arbeta som ingenjör innebär ofta att ställa upp matematiska modeller av verkligheten. Modeller som sedan skall lösas vilket i många fall kan innebära att beräkningarna blir så komplexa att hjälpmedel som datorer måste användas.

Numeriska metoder används för att utföra beräkningar på mätdata och rena siffror, vilket i princip alltid är förekommande när verkligheten analyseras. För att bygga modeller utifrån detta används interpolation för att räkna fram polynomfunktioner, eller differentialekvationer som beskriver fenomenet. MATLAB är för detta ett synnerligen bra verktyg ty det har inbyggda lösare för differentialekvationer och det går snabbt att lösa ekvationssystem för att bestämma polynom.

Frågeställningen som byggdes upp på detta var:

Hur utför man polynominterpolation med MATLAB?

Hur formuleras differentialekvationer av högre ordning som system av första ordning?

Hur löses dessa system med MATLAB?

Hur implementerar man själv algoritmer, som Runge-Kutta?

Vilka för- respektive nackdelar har de olika metoderna?

Syftet var att förstå hur differentialekvationer löses med numeriska metoder i MATLAB.

Däremot var inte syftet att presentera färdig MATLAB-kod utan endast teorin och matematiken bakom. Att sen implementera det i MATLAB (eller liknande verktyg) är ett kapitel för sig.

## 2 Problem och syfte

Problemet är att lösa differentialekvationer med numeriska metoder i MATLAB. Både med MATLABs inbyggda funktioner samt att implementera en egen.

Syftet är att förstå hur differentialekvationer löses numeriskt och hur MATLABs funktioner beter sig. Vilka för- respektive nackdelar de har.

### 2.1 Polynominterpolation

Polynominterpolation är approximationer av kända funktioner eller mätdata med polynomserier utav en given grad. En högre grad medför generellt en högre noggrannhet, men beräkningarna tar längre tid och funktionen kan bete sig underligt och ge orimliga svar.

Första problemet är att bestämma andragradspolynomet  $p_2(t)$  som approximerar funktionen

$$f(t) = \sqrt{t}, \quad t = 0, 1, 4 \quad (2.1)$$

För att sedan jämföra polynomets värde med funktionens värde vid  $t = 2$ , det vill säga hur stort felet är.

Nästa steg är att lägga till punkten  $t = 3$  och bestämma ett tredjegradspolynom,  $p_3(t)$ , till samma funktion. För att återigen jämföra värdena och felen.

I det andra problemet är det gammafunktionen som ska approximeras med ett fjärdegradspolynom,  $p_4(t)$ . Gammafunktionen är känd och definieras som:

$$\Gamma(t) = \int_0^\infty \tau^{t-1} e^{-\tau} d\tau, \quad t > 0 \quad (2.2)$$

Om argumentet till gammafunktionen är ett heltal så förenklas funktionen till:

$$\Gamma(n) = (n-1)! \quad (2.3)$$

Polynomet och funktionsvärdena skall ritas i en figur och det relativa felet i en annan.

### 2.2 Arenstorf-banan

$$y_1'' = y_1 + 2y_2' - \hat{\mu} \frac{y_1 + \mu}{N_1} - \mu \frac{y_1 - \hat{\mu}}{N_2}, \quad (2.4)$$

$$y_2'' = y_2 + 2y_1' - \hat{\mu} \frac{y_2}{N_1} - \mu \frac{y_2}{N_2}, \quad (2.5)$$

där

$$N_1 = \sqrt{(y_1 + \mu)^2 + y_2^2}, \quad N_2 = \sqrt{y_1 - \hat{\mu}^2 + y_2^2}$$

och

$$\mu = 0,012277471, \quad \hat{\mu} = 1 - \mu$$

Begynnelsevärdena

$$y_1(0) = 0,994, \quad y_1'(0) = 0, \quad y_2(0) = 0, \quad y_2'(0) = -2,00158510637908252240537862224$$

ger en periodisk bana. Periodtiden är

$$T = 17,0652166$$

Till att börja med skall differentialekvationerna 2.4 och 2.5 formuleras om till ett system av första ordningen. Detta system skall sedan lösas med valfri ODE-lösare i MATLAB. Noggrannheten i problemet är given till  $10^{-8}$ . Slutligen skall en fasplot göras.

Nästa problem är att på egen hand implementera Runge-Kuttan av fjärde ordning (RK4) i MATLAB. Algoritmen skall sedan användas för att lösa det ursprungliga problemet, vid olika steglängder. Steglängden är  $h = T/n$  med start på  $n = 1000$ , för att sedan dubblera  $n$  tills man får en bra figur.

Slutligen skall problemet lösas med en ännu högre noggrannhet. Noggrannheten som sökes är  $\pm 1$  km för  $|y(0) - y(T)|$ .

### 2.3 Kemisk reaktionskinetik

$$y' = \begin{bmatrix} y_1' \\ y_2' \\ y_3' \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & 0 \\ k_1 & -k_2 & 0 \\ 0 & k_2 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (2.6)$$

Konstanterna  $k_1$  och  $k_2$  bestämmer reaktionshastigheten för  $y_1(t)$  respektive  $y_2(t)$ .

Problemet är att lösa differentialekvation 2.6 vilken är formulerad som ett system av första ordning. Detta med hjälp av (i MATLAB) en Runge-Kutta-metod, en Adams-metod och en metod för styva system. Metoderna skall jämföras med avseende på dess effektivitet, mätt antingen i antalet funktionsanrop eller i tid.

## 3 Metod och utförande

### 3.1 Polynominterpolation

Allmänt ges ett polynom av

$$\sum_{k=1}^n c_k p^{k-1} \quad (3.1)$$

där  $n - 1$  är polynomets gradtal och  $n$  är antalet interpolationspunkter.

Polynomets konstanter beräknas lämpligen med matriser på följande sätt

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} t_1^0 & t_1^1 & t_1^2 & \dots & t_1^{n-1} \\ t_2^0 & t_2^1 & t_2^2 & \dots & t_2^{n-1} \\ \dots & \dots & \dots & \dots & \dots \\ t_n^0 & t_n^1 & t_n^2 & \dots & t_n^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (3.2)$$

vilket utförs mycket enkelt i MATLAB.

I första problemet ger ekvationerna 3.1 och 3.2 följande:

$$p_2(t) = c_1 + c_2 t + c_2 t^2 \quad (3.3)$$

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} t_1^0 & t_1^1 & t_1^2 \\ t_2^0 & t_2^1 & t_2^2 \\ t_3^0 & t_3^1 & t_3^2 \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (3.4)$$

Där  $t_n$  är de kända interpolationspunkterna och  $y_n$  är de exakta värdena från den kända funktionen  $f(t)$  (ekvation 2.1).

Felet ges av differensen mellan  $p_2(2)$  och  $f(2)$ .

För att förbättra noggrannheten av interpolationen till  $f(t)$  läggs ytterligare en interpolationspunkt till. Nu fås istället ett tredjegradspolynom:

$$p_3(t) = c_1 + c_2 t + c_2 t^2 + c_3 t^3 \quad (3.5)$$

Även här ges felet av differensen mellan  $p_3(2)$  och  $f(2)$ .

I nästa problem där gammafunktionen är given, den ska interpoleras med hjälp av ett fjärdegradspolynom. Detta görs på motsvarande sätt som föregående problem. När polynomet är framtaget skall det relativa felet plottas i en figur, det relativa felet definieras som:

$$r = \frac{p_4(t) - \Gamma(t)}{\Gamma(t)} \quad (3.6)$$

### 3.2 Arenstorf-banan

Med hjälp av variabelsubstitutioner skrivs differentialekvationerna 2.4 och 2.5 om till ett system av första ordningen.

$$\begin{aligned} u_1 &= y_1 \\ u_2 &= y_1' \\ u_3 &= y_2 \\ u_4 &= y_2' \\ u_2' &= y_1'' \\ u_4' &= y_2'' \end{aligned}$$

Då blir de ursprungliga differentialekvationerna, 2.4 och 2.5

$$u'_2 = u_1 + 2u_4 - \hat{\mu} \frac{u_1 + \mu}{N_1} - \mu \frac{u_1 - \hat{\mu}}{N_2}, \quad (3.7)$$

$$u'_4 = u_3 + 2u_2 - \hat{\mu} \frac{u_3}{N_1} - \mu \frac{u_3}{N_2}, \quad (3.8)$$

och första derivatan av funktionen  $y(t)$  blir då

$$y(t)' = \begin{bmatrix} u_2 \\ u'_2 \\ u_4 \\ u'_4 \end{bmatrix} \quad (3.9)$$

I MATLAB skrivs detta ekvationssystem som en funktion vilken kan anropas av MATLABs ode-lösare.

Runge-Kuttas metod av fjärde ordningen formuleras som följer

$$\begin{aligned} k_1 &= f(t_k, y_k) \\ k_2 &= f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}k_1\right) \\ k_3 &= f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}k_2\right) \\ k_4 &= f(t_k + h, y_k + hk_3) \\ y_{k+1} &= y_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \quad (3.10)$$

Hur detta implementeras i MATLAB ligger utanför denna rapportens innehåll.

### 3.3 Kemisk reaktionskinetik

Ekvationssystemet 2.6 löses med hjälp av olika ode-lösare i MATLAB. Lösningens implementering i MATLAB är analog med den i problemet med Arenstorf-banan. För att se hur ämnena reagerar ritas deras funktioner i en figur. Då blir det möjligt att grafiskt se när systemet når jämvikt och tidsintervallet kan således optimeras. Beräkningstiden och antalet funktionsanrop för de olika lösarna sparas för senare jämförelse.



## 4 Resultat

### 4.1 Polynominterpolation

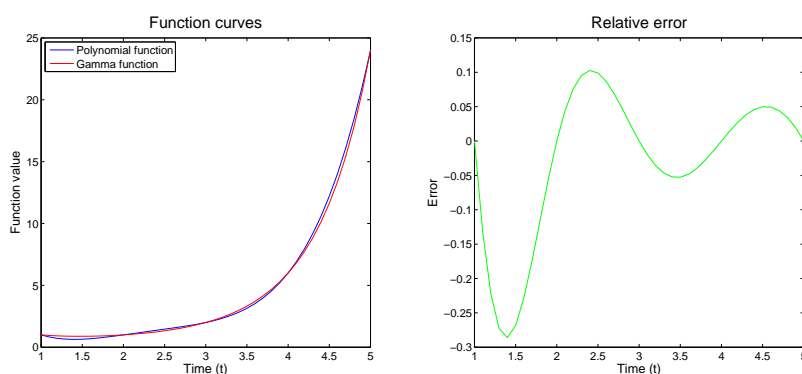
Konstanterna till andragradspolynomet  $p_2(t)$  bestämdes till:  $c_1 = 0$ ,  $c_2 = 1,1667$  och  $c_3 = -0,1667$ .

Följande resultat fås:  $p_2(2)=1,6667$  att jämföra med  $f(2) = \sqrt{2} \approx 1,4142$  vilket ger ett fel på 0,2525.

Till tredjegradspolynomet  $p_3(t)$  bestämdes motsvarande konstanter till:  $c_1 = 0$ ,  $c_2 = 1,3453$ ,  $c_3 = -0,3900$  och  $c_4 = 0,0447$

Det ger följande resultat:  $p_3(2) = 1,4880$  vilket medför att felet är på 0,0738. Skillnaden mellan felen i båda fallen blir alltså 0,1786.

Konstanterna till fjärdegradspolynomet  $p_4(t)$  bestämdes till:  $c_1 = 1,0400$ ,  $c_2 = 0$ ,  $c_3 = 0,0167$ ,  $c_4 = -0,1000$  och  $c_5 = 0,0433$ .

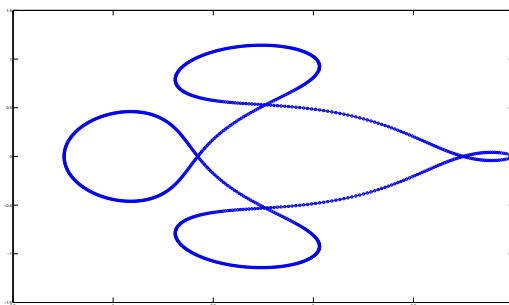


Figur 1: Funktionskurvorna och relativa felet

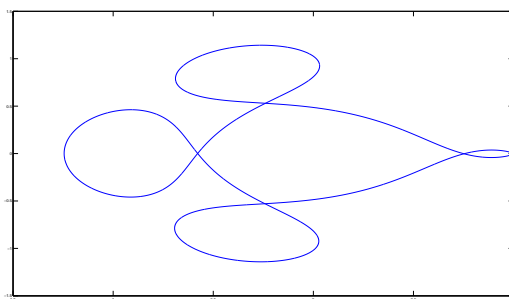
### 4.2 Arenstorf-banan

Lösaren (MATLABs ode45) behövde 426 steg för att lösa systemet.

RK4 behövde 16 000 steg för att lösa samma system.



Figur 2: Faskurva till arenstorf-bana löst med MATLABs `ode45`



Figur 3: Faskurva till arenstorf-bana löst med egen implementerad RK4-metod

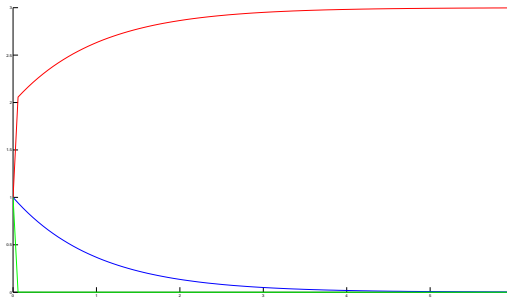
För att få den noggrannhet som eftersöktes i det sista delproblemet krävdes det mellan 64 000 till 128 000 till RK4.

### 4.3 Kemisk reaktionskinetik

Tabell 1: ODE-lösarnas prestanda

lösare	tid (s)	funktions anrop (antal)
<code>ode45</code>	1,3	11 947
<code>ode113</code>	1,6	8 244
<code>ode23tb</code>	1,4	2 829

`ode45` är en Runge-Kutta-metod (av ordning fem), `ode113` är en Adams-metod och `ode23tb` är en metod för styva system.



Figur 4: Kurvor över det kemiska reaktionsförloppet.

## 5 Diskussion och slutsats

### 5.1 Polynominterpolation

I första deluppgiften konstaterade vi direkt att felet blev stort när vi endast hade andragradspolynom som interpolerade funktionen. Tredjegradspolynomet förbättrade approximationen avsevärt jämfört med det första.

Gammafunktionens interpolation visar i grafen att vid interpolationspunkterna är det relativa felet lika med noll. Detta ses som självklart när den här typen av interpolation används, felet mellan interpolationspunkterna kan däremot variera kraftigt. När vi plottade de olika funktionerna mot varandra kunde vi tydligt se att desto längre från sista interpolationspunkten vi kom, desto större blev felet.

### 5.2 Arenstorf-banan

Den egna RK4-metoden vi skrev gav visserligen samma resultat som MATLABs `ode45`, men använde ett mycket större antal steg för att lösa problemet.

### 5.3 Kemisk reaktionskinetik

Metoden för styva system verkade i detta fall vara det bästa. Vi kunde konstatera att i det här problemet inte skiljde mycket mellan de olika metoderna i beräkningstid. Antalet funktionsanrop samt misslyckade försök varierade dock kraftigt, men metoden för styva system fungerade bäst.

## A Teoretiska frågor med svar

- 1 Sant eller falskt: Det finns ett oändligt antal matematiska funktioner som interpolerar en given mängd datapunkter.

*Sant.*

- 2 Sant eller falskt: Då en kontinuerlig funktion interpoleras med ett polynom i ekvidistanta punkter i ett givet intervall, så konvergerar alltid polynomet mot funktionen då antalet interpolationspunkter växer.

*Falskt.*

- 3 Varför är interpolation med polynom av hög grad ofta dåligt?

*Högre grad på polynomet medför att beräkningarna tar längre tid mot ett polynom av lägre grad. Dessutom blir noggrannheten sämre vid ändpunkterna.*

- 4a Har en differentialekvation (ensam) generellt en unik lösning?

*Nej.*

- 4b Om den har det, varför? Om inte, vilken ytterligare information behövs för att en unik lösning skall finnas?

*Ett begynnelsevärde så att konstanten/konstanterna i lösningen kan bestämmas vilket då ger en unik lösning.*

- 5a Vad menas med första ordningens ODE?

*Att det inte finns någon derivata av högre ordning än ett i ekvationen.*

- 5b Varför skrivs en ODE av högre ordning vanligen ner till en ODE av första ordningen?

*För att en ODE av första ordningen är enklare att lösa (även som ett system) och det är lätt att skriva algoritmer som utför jobbet.*

- 6 Skriv följande system som ett första ordningens system:

$$\begin{cases} y_1'' = \frac{-GM y_1}{\sqrt[3]{y_1^2 + y_2^2}} \\ y_2'' = \frac{-GM y_2}{\sqrt[3]{y_1^2 + y_2^2}} \end{cases}, \quad \begin{bmatrix} u_1 = y_1 \\ u_2 = u_1' \\ u_3 = y_2 \\ u_4 = u_3' \end{bmatrix} \Rightarrow \begin{cases} u_2' = \frac{-GM y_1}{\sqrt[3]{y_1^2 + y_2^2}} \\ u_4' = \frac{-GM y_2}{\sqrt[3]{y_1^2 + y_2^2}} \end{cases}$$

- 7 Beskriv i ord skillnaden mellan det lokala och det globala felet som uppkommer då ett begynnelsevärdesproblem för ett ODE löses numeriskt.

*Lokala felet är skillnaden mellan det exakta och det beräknade värdet. Det globala felet är summeringen av de lokala felen.*

- 8 Vad menas med en numerisk metod av ordning  $p$  då en ODE löses?

*Noggrannheten - Högre ordning  $p$  ger generellt en bättre noggrannhet*

- 9 Vad menas med styv ODE? Varför kan en styv ODE vara svår att lösa numeriskt?

*En ODE som är styv kan bli ostabil vid många numeriska lösningar, om inte steglängden görs extremt liten. De är svåra att lösa numeriskt för att vissa metoder är bättre än andra och det kan vara svårt att avgöra vilken som skall tillämpas.*

- 10 Vilken är den enklaste numeriska metoden för att lösa en styv ODE?

*Adams-Moulton.*