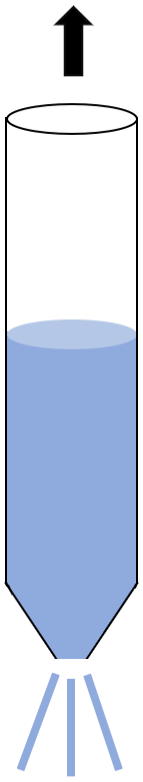
SF1545
Kursansvarig: Ninni Carlsund
Uppgift #17

August Regnell
970712-9491

Rapport: Vattenraketen i provflygning



Sammanfattning

Denna rapport behandlar lösningen av ett problem där en vattenrakets maximala höjd vid uppskjutning ska beräknas genom att variera mängden vatten i raketen och storleken på dess mynning. Två ekvationssystem, vilka har erhållits i samband med kursomgång 2018 av SF1545 på KTH, användes för att beskriva hur en vattenraket gjord av en PET-flaska flyger efter avfyring. Beräkningarna har skett i MatLab där koden allena har skrivits av författaren. Optimeringen gav att den maximala höjden raketen når är 73.3999 meter då flaskhalsens tvärsnitt är 1/3.594 av flaskkroppens och flaskan är fylld till 25.45%.

Innehållsförteckning

1 Nomenklatur

Storheter	Beskrivning	Enhet
N	Trycket i flaskan vid t = 0	atm
γ	Värmekapacitetsförhållande (eng. heat capacity ratio)	-
p_A	Lufttrycket utanför flaskan	Pa
g	Gravitationskonstanten	m/s^2
R	Flaskkroppens radie	m
L	Flaskans längd	m
L_0	Den längd som motsvarar den tomma flaskans vikt	m
C_f	Friktionskoefficient	m^{-1}
C_{DR}	Motståndskoefficient	-
C_D	Koefficient som motsvarar energiförlusten i utströmning av vattnet	-
t	Tiden från att mynningen har öppnats	s
X	Vattennivåns avstånd från flaskans mynning	m
X(0)	Vattennivåns avstånd från flaskans mynning vid t = 0	m
Z	Flaskans höjd över marken	m
ρ	Vattnets densitet	kg/m^3
α	Förhållandet mellan flaskhalsens och flaskkroppens tvärsnittsarea (1 / α)	-
Fyllnadsgrad	Vattennivån vid t = 0 i relation till flaskans längd	-

Tabell 1

2 Problembeskrivning

En vattenraket skapas genom att fylla en PET-flaska med längden L och radien R med vatten och luft med högt tryck. Luften har trycket N vid start, oavsett hur fylld flaskan är. När flaskans mynning öppnas vid tiden t = 0 kommer trycket trycka ut vattnet vilket enligt Newtons andra lag kommer få raketen att accelerera uppåt.

Raketens kraftsituation beskrivs av två olika system beroende på mängden kvarvarande vatten i flaskan – X. *System 1* gäller då X > 0, dvs. så länge det finns vatten i flaskan, medan *system 2* gäller då X = 0 och Z > 0, ie. då allt vatten har lämnat flaskan och flaskan befinner sig i luften.

$$
\begin{aligned}
f &= \frac{p_A}{\rho}\left(N\left(\frac{L-X(0)}{L-X}\right)^{\gamma}-1\right)\\
X\dot{u}+(X+L0)\dot{w} &= (\alpha-1)u^2-g(L0+X)-C_{DR}w|w|\\
X\dot{u}+X\dot{w} &= -f-gX-C_fXu|u|+C_D\cdot u^2(\alpha^2-1)\\
\dot{X} &= u\\
\dot{Z} &= w
\end{aligned}
$$

System 1

$$
\begin{aligned}
\dot{u} &= 0\\
\dot{w} &= -g-C_{DR}w|w|\\
\dot{X} &= 0\\
\dot{Z} &= w
\end{aligned}
$$

System 2

I tabellen nedan presenteras de konstanter som är angivna i uppgiftbeskrivningen.

Konstant	Värde
N	10 atm
γ	1.4
p_A	100 000 Pa
g	9.81 m/s^2

R	0.05 m
L	0.5 m
L_0	0.006 m
C_f	0.001 m^{-1}
C_{DR}	0.01
C_D	1.2
ρ	1000 kg/m^3

Tabell 2

Uppgiften blir att med ekvationssystemen ovan kalkylera den högsta höjden raketen når och dess tillhörande kombination av α och fyllnadsgrad.

3 Kort lösningsbeskrivning

Genom att lösa ut $\dot{u}$ och $\dot{w}$, andraderivatan för X respektive Z, ur *system 1* och *system 2* kan MatLabs inbyggda ODE-lösare användas för att beräkna raketens bana för ett givet α och fyllnadsgrad. Interpolation av andra grad används för att beräkna den maximala höjden med de givna förhållandena.

Programmet upprepar denna process för ett antal kombinationer av α och fyllnadsgrad där resultaten sparas för varje iteration.

Då endast ett finit antal kombinationer kan testas genomförs även styckvis interpolation av de framtagna värdena över en tredimensionell yta för att öka chansen att hitta det optimala resultatet.

4 Mer detaljerad lösningsbeskrivning

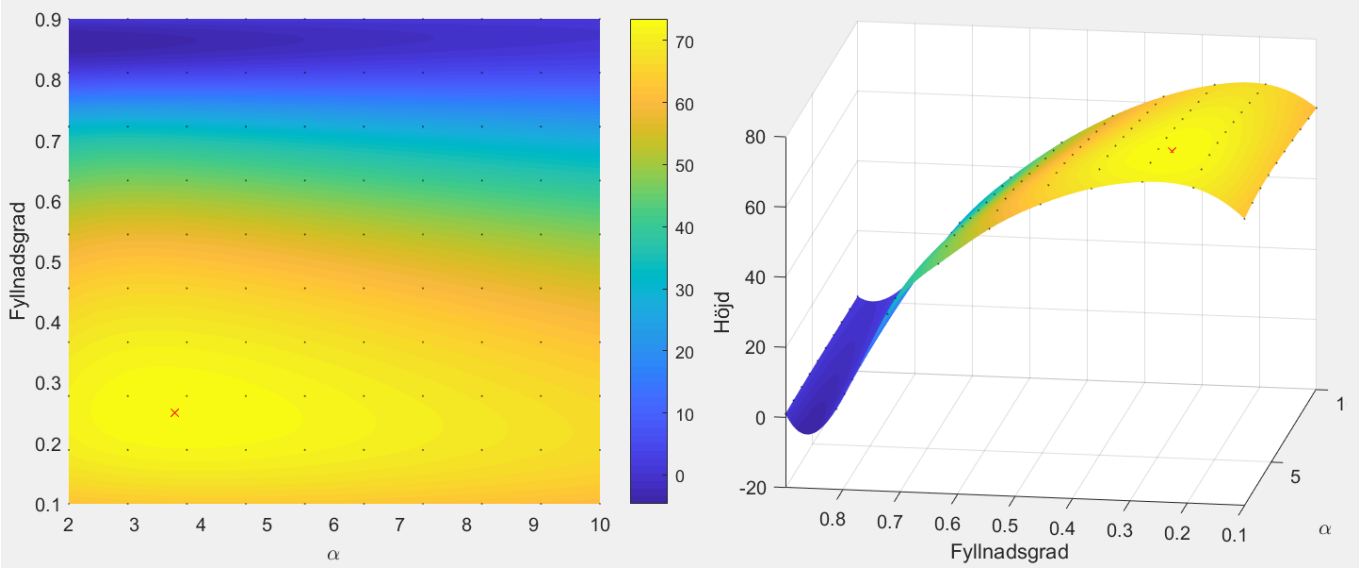
MatLab-koden som har skrivits består av 15 filer; en funktionsfil, en huvudfil, en värdefil, en intervallhalveringsfil, en ODE-lösningsfil, tre interpolationsfiler, två plottingfiler, en körningsfil och fyra felskattningsfiler.

Funktionsfilen innehåller en omskrivning av *system 1* och *system 2* sådana att de blir system av differentialekvationer av första grad med derivatan och andraderivatan av X respektive Z i vänsterled. Med hjälp av if-satser med villkor för X och Z byter programmet mellan de två omskrivna systemen. För att undvika problemet med att andraderivatan för X når oändligheten byter programmet system vid $X \leq 10^{-9}$ istället för $X = 0$.

Värdefilen returnerar de konstanter som behövs för att genomföra beräkningen.

Körningsfilen och de fyra felskattningsfilerna använder huvudfilen, vilken beskrivs nedan, för att genomföra en respektive flera beräkningar av den maximala höjden med olika villkor.

Inledningsvis grovlokaliseras den maximala höjden med stora intervall med stora steglängder för α och fyllnadsgraden. Undersökningen visar att α ligger på intervallet [3 4] och fyllnadsgraden på intervallet [0.2 0.3]. Resultatet presenteras i *figur 1*. Den beräknade maximala höjden markeras i figuren med ett rött kryss och de höjder som beräknats med *ode45* med svarta punkter.



Figur 1

Hädanefter kommer alla beräkningar påbörjas med α på intervallet [3 4] och fyllnadsgrad på intervallet [0.2 0.3].

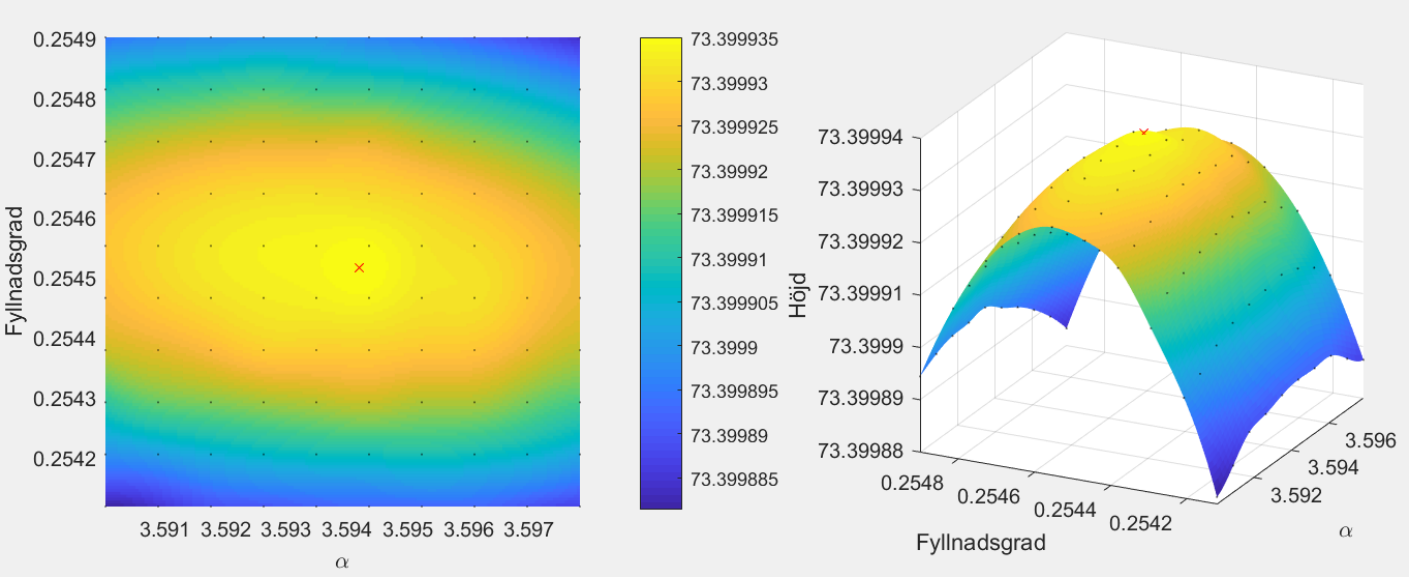
Först kallar huvudfilen på ODE-lösningsfilen som i sin tur kallar på en den första interpolationsfilen. ODE-lösningsfilen använder MatLabs inbyggda ODE-lösare *ode45* för att kalla på funktionsfilen för olika kombinationer av α och fyllnadsgrad med t på intervallet [0 10]. t = 10 valdes som ändpunkt på intervallet då det är efter raketen slagit i marken igen. Detta kommer returnera en vektor med värden på Z under raketens flygtur, på vilken den första interpolationsfilen genomför interpolation av andra grad för att erhålla raketens maximala höjd. Valet av grad två kommer från att Z(t)-grafen efterliknar ett polynom av andra grad, se *bilaga 1*.

De maximala höjderna för varje kombination av α och fyllnadsgrad sparas i en matris vilken den andra interpolationsfilen använder för att genomföra tredimensionell spline-interpolation. Spline valdes då det är den interpolationsmetod med högst grad av kontinuitet (C^2) som *griddedInterpolant* kan använda. På detta sätt går det att hitta ett bättre värde på den maximala höjden trots att endast ett finit antal kombinationer av α och fyllnadsgrad har beräknats med *ode45*. De funktioner som används för den tredimensionella interpolationen är MatLabs *ndgird* och *griddedInterpolant*. *ndgrid* skapar utifrån två vektorer två matriser där den första matrisen har den första vektorn som kolumner och den andra matrisen har den

andra vektorn som rader[1]. Om den första vektorn är *n* lång och den andra är *m* lång kommer matriserna bli *n* x *m*. *griddedInterpolant* skapar en interpolerad yta utifrån tre matriser på grid-form[2]. Interpolationen kommer itereras med högre noggrannhet tills normen av förändringen av α och fyllnadsgrad understiger en (absolut) interpolationstolerans. Den tredje interpolationsfilen ger information om de två interpolationerna som genomförts.

Processen i de två föregående styckena kommer därefter upprepas av huvudfilen ett givet antal gånger med kortare och kortare intervall för α och fyllnadsgrad. Vid varje ny iteration kommer intervallhalveringsfilen se till att intervallängden för de två variablerna halveras samtidigt som de nya intervallen centreras kring det värde på α respektive fyllnadsgrad som gav den maximala höjden vid föregående iteration. Antalet steg i intervallen för α och fyllnadsgrad hålls konstant, men till följd av att intervallängden minskar kommer även steglängden göra det. Den minskade steglängden möjliggör en lägre tolerans för normen av förändringen av α och fyllnadsgrad utan att minne-utnyttjandet blir för stort, varför programmet förändrar interpolationstoleransen när antalet iterationer ökar.

Figur 2 visar resultatet efter 8 intervallhalveringar med den maximala höjden markerat med ett rött kryss och de höjder som beräknats med *ode45* med svarta punkter. Grafen har skapats av den andra plottingfilen, vilken har använt sig av MatLabs *plot3* och *surf*.



Figur 2

I *figur* 3 i *bilaga 1* presenteras även en graf, skapad av den första plottingfilen, för vattennivån i raketen – X(t) och raketens höjd över marken – Z(t) för den optimala kombinationen av α och fyllnadsgrad.

[1] https://se.mathworks.com/help/matlab/ref/ndgrid.html
[2] https://se.mathworks.com/help/matlab/ref/griddedinterpolant.html

5 Tillförlitlighetsbedömning

För det första kan det uppstå fel i resultatet till följd av den använda modellen. Detta fel är svårt att skatta men går att observera när α närmar sig $\alpha = 1$. Då kommer raketen vara en cylinder utan botten, något modellen inte är tänkt att beskriva, vilket leder till opålitliga resultat. Det är av denna anledning startpunkten för intervallet för α vid grovlokaliseringen är $\alpha = 2$.

För det andra kan fel i de angivna konstanterna i uppgiftlydelsen leda till fel i resultatet. Därför har konstanterna var för sig störts med 1% innan en beräkning av den maximala höjden, vars resultat kan användas för att skata systemets känslighet. Resultatet presenteras i *tabell 3* i *bilaga 1*. Störningen ger en total avvikelse från maximal höjd på 3.99%, vilket tyder på att systemet är relativt okänsligt.

För det tredje kan fel uppstå på grund av ett för litet antal intervallhalveringar. *Tabell 4* i *bilaga 1* presenterar optimeringsresultatet för olika antal intervallhalveringar. Tabellen visar att α stabiliserar kring 3.594, fyllnadsgrad kring 0.2545 och den maximala höjden kring 73.3999 meter när antalet intervallhalveringar överstiger fyra.

För det fjärde kan fel möjligen uppstå i *ode45*. Detta kan skattas genom att variera ODE-lösarens relativa tolerans. *Tabell 5* i *bilaga 1* presenterar det optimerade resultatet för olika *RelTol* för *ode45*. Tabellen visar att α stabiliserar kring 3.594, fyllnadsgrad kring 0.2545 och den maximala höjden kring 73.3999 meter när den relativa toleransen understiger 10^{-6}.

För det femte kan fel uppstå till följd av när bytet sker till *system 2*. Därför utfördes beräkningar för olika bytestillfällen. Beräkningarna presenteras i *tabell 6* i *bilaga 1*. Tabellen visar att α stabiliserar kring 3.594, fyllnadsgrad kring 0.2545 och den maximala höjden kring 73.3999 meter när bytet sker senare än $X = 10^{-7}$ meter.

För det sjätte kan valet av interpolationsmetod påverka resultatet. Därför genomfördes även en beräkning där den tvådimensionella interpolationen var av grad tre, istället för grad två. Skillnaden mellan de två resultaten var $2.53*10^{-7}$ meter. Det utfördes även beräkningar för olika metoder för *griddedInterpolant*, vilket presenteras i *tabell 7* i *bilaga 1*. De olika metoderna hade gemensamt fyra säkra siffror för α, fyra för fyllnadsgrad och sju för den maximala höjden.

Till sist kan även fel uppstå till följd av att den tredimensionella interpolationen avslutas för tidigt. Det är av denna anledning interpolationen upprepas med högre och högre noggrannhet tills normen av korrigeringen av α och fyllnadsgrad understiger en interpolationstolerans som är beroende av antalet intervallhalveringar. Intressant är att den tredimensionella interpolationen endast bidrar med $1.73*10^{-7}$ meter vid den slutgiltiga beräkningen, vilken genomfördes med 8 intervallhalveringar, byte till *system 2* då $X \leq 10^{-9}$, en relativ tolerans på 10^{-8} för *ode45* och en interpolationstolerans på $0.5*10^{-6}$. Bidraget är dock positivt samt tillfredsställer interpolationstoleransen vilket gör att interpolationen kan anses vara lyckad.

6 Resultat och Diskussion

Raketen når enligt den genomförda optimeringen en maximal höjd på 73.3999 meter när α = 3.594 och fyllnadsgrad = 0.2545. α och fyllnadsgrad har beräknats med 4 siffrors noggrannhet medan den maximala höjden har beräknats med 6 siffrors noggrannhet. De absoluta felgränserna är alltså; $E_{\alpha} = 0.5*10^{-3}$, $E_{\text{fyllnadsgrad}} = 0.5*10^{-4}$ och $E_{\text{maximal höjd}} = 0.5*10^{-4}$ meter.

Resultatet anses rimligt enligt den använda modellen men läsaren bör vara medveten om att vattenraketer gjorda av PET-flaskor sällan når sådana höga höjder. Modellen bör därför ses över innan framtida undersökningar så att den till exempel inte underskattar energiförluster eller överskattar trycket som rimligen kan uppnås i flaskan. Ett förslag till förbättring är att göra L0 beroende av α, då en större öppning skulle göra flaskan lättare då mängden plast skulle vara lägre. Vattenraketer gjorda av starkare material kan dock nå en betydligt högre höjd, världsrekordet är idag 830 meter[3].

7 Egen arbetsinsats

Under förbehandlingen av ekvationssystemen har en del diskussion skett med Ninni Carlsund. I övrigt har samtligt arbete enbart genomförts av författaren.

8 Referenslista

Mathworks – Griddedinterpolant

https://se.mathworks.com/help/matlab/ref/griddedinterpolant.html

Mathworks – Ndgrid

https://se.mathworks.com/help/matlab/ref/ndgrid.html

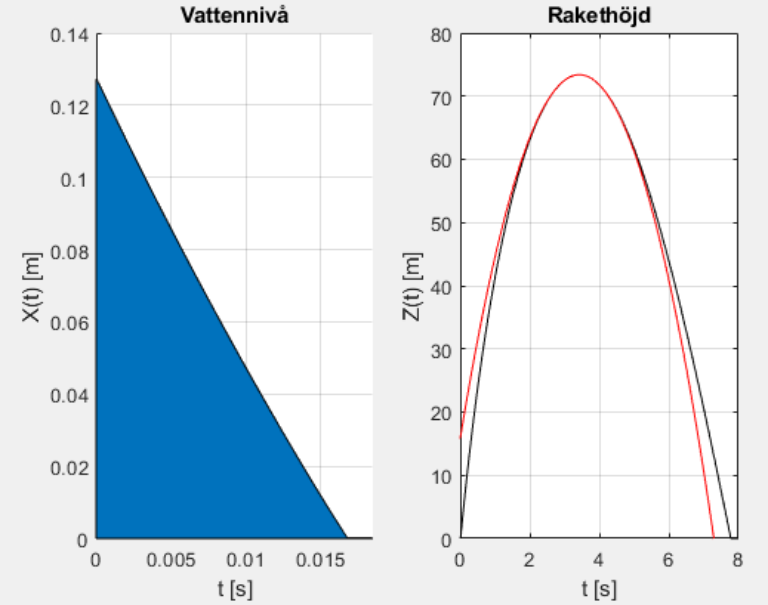
Wikipedia – Water Rocket

https://en.wikipedia.org/wiki/Water_rocket

[3] https://en.wikipedia.org/wiki/Water_rocket

9.1 Bilaga 1 – Figurer och tabeller

9.1.1 Figur 3



Figur 3

Figuren visar hur vattennivån i flaskan – X(t) och flaskans höjd – Z(t) varierar med tiden vid de optimerade α och fyllnadsgrad. Den svarta kurvan är den som beräknats av *ode45* medan den röda är andragradspolynomet som interpolerades kring den svarta kurvans maximum.

9.1.2 Tabell 3

Konstant	α [%]	Fyllnadsgrad [%]	Höjd [%]
N	0.0618	0.0979	0.607
γ	0.0398	0.567	0.194
p_A	0.0206	$7.59*10^{-5}$	0.523
g	0.0348	0.00104	0.522
L	0.177	0.211	0.230
L_0	0.109	0.231	0.227
C_f	0	0	$1.33*10^{-6}$

C_{DR}	0.538	0.0873	0.579
C_D	0.346	0.0355	0.585
ρ	0.0274	$3.22*10^{-5}$	0.522
Totalt	1.36	1.23	3.99

Tabell 3

Tabellen visar hur α, fyllnadsgrad respektive maximal höjd procentuellt avviker (absolutbeloppet) från normal beräkning när konstanterna var och en ökas med 1%. Beräkningarna genomfördes med en relativ tolerans på 10^{-8} för *ode45*, en interpolationstolerans på $0.5*10^{-6}$, 8 intervallhalveringar och startintervall för α och fyllnadsgrad på [3 4] respektive [0.2 0.3] och byte till *system 2* då $X \leq 10^{-9}$.

9.1.3 Tabell 4

Antal intervallhalveringar	α	Fyllnadsgrad	Maximal höjd
0	3.5934	0.25450	73.39993434
1	3.5936	0.25452	73.39993409
2	3.5935	0.25453	73.39993476
3	3.5935	0.25453	73.39993415
4	3.5934	0.25452	73.39993452
5	3.5935	0.25453	73.39993397
6	3.5939	0.25451	73.39993442
7	3.5942	0.25452	73.39993505
8	3.5943	0.25452	73.39993472

Tabell 4

Tabellen visar beräknat α, fyllnadsgrad respektive maximal höjd vid olika antal intervallhalveringar. Beräkningarna genomfördes med en relativ tolerans på 10^{-8} för *ode45*, startintervall för α och fyllnadsgrad på [3 4] respektive [0.2 0.3] och byte till *system 2* då $X \leq$

10^{-9}. Interpolationstoleransen minskar när antalet intervallhalveringar ökar och är som lägst $0.5*10^{-6}$ vid 8 intervallhalveringar. Den exakta interpolationstoleransen vid varje antal intervallhalveringar återges i MatLab-koden i *tredimensionell interpolation* i *bilaga 2*.

9.1.4 Tabell 5

Relativ tolerans	α [-]	Fyllnadsgrad [-]	Höjd [m]
10^{-3}	3.5945	0.25372	73.41002462
10^{-4}	3.6003	0.25475	73.40167256
10^{-5}	3.5944	0.25453	73.40005567
10^{-6}	3.5932	0.25449	73.39993949
10^{-7}	3.5938	0.25451	73.39993518
10^{-8}	3.5943	0.25453	73.39993472
10^{-9}	3.5941	0.25452	73.39993492
10^{-10}	3.5939	0.25450	73.39993517
10^{-11}	3.5937	0.25449	73.39993450

Tabell 5

Tabellen visar beräknat α, fyllnadsgrad respektive maximal höjd vid olika relativa toleranser för *ode45*. Beräkningarna genomfördes med en interpolationstolerans på $0.5*10^{-6}$, 8 intervallhalveringar, startintervall för α och fyllnadsgrad på [3 4] respektive [0.2 0.3] och byte till *system 2* då $X \leq 10^{-9}$.

9.1.5 Tabell 6

≥ X [m]	α [-]	Fyllnadsgrad [-]	Höjd [m]
10^{-4}	3.5932	0.25576	73.151336
10^{-5}	3.6007	0.25440	73.375030
10^{-6}	3.5968	0.25454	73.397576

10^{-7}	3.5932	0.25451	73.399698
10^{-8}	3.5944	0.25451	73.399914
10^{-9}	3.5943	0.25452	73.399935
10^{-10}	3.5937	0.25452	73.399936

Tabell 6

Tabellen visar beräknat α, fyllnadsgrad respektive maximal höjd när tillfället för byte av *system 2* varieras. Beräkningarna genomfördes med en relativ tolerans på 10^{-8} för *ode45*, en interpolationstolerans på $0.5*10^{-6}$, 8 intervallhalveringar och startintervall för α och fyllnadsgrad på [3 4] respektive [0.2 0.3].

9.1.6 Tabell 7

Metod[4]	**Kontinuitet**	**α [-]**	**Fyllnadsgrad [-]**	**Höjd [m]**
Spline	C^2	3.5943	0.25453	73.39993472
Cubic	C^1	3.5944	0.25449	73.39993447
Makima	C^1	3.5943	0.25452	73.39993474

Tabell 7

Tabellen visar beräknat α, fyllnadsgrad respektive maximal höjd för olika tredimensionella interpolationsmetoder, dvs. den metod *griddedInterpolant* använder. Beräkningarna genomfördes med en relativ tolerans på 10^{-8} för *ode45*, en interpolationstolerans på $0.5*10^{-6}$, byte till *system 2* då $X \leq 10^{-9}$, 8 intervallhalveringar och startintervall för α och fyllnadsgrad på [3 4] respektive [0.2 0.3].

[4] https://se.mathworks.com/help/matlab/ref/griddedinterpolant.html

9.2 Bilaga 2 – MatLab-kod

Filerna ligger i den ordning som de presenteras i *Mer detaljerad lösningsbeskrivning*.

9.2.1 Funktionsfilen

```
function f = systemet(t,u,varden)

    % Bryter ner indata för att göra det lättare att förstå koden
    fyllnadsgrad = varden(1);
    alpha = varden(2);
    N = varden(3);
    gamma = varden(4);
    pA = varden(5);
    g = varden(6);
    L = varden(7);
    L0 = varden(8);
    Cf = varden(9);
    CDR = varden(10);
    CD = varden(11);
    roh = varden(12);
    X0 = varden(13);
    systembyte = varden(14);

    % Bryter ned u för att göra det lättare att förstå ekvationerna
    X = u(1:2); % [X X']
    Z = u(3:end); % [Z Z']

    % ----- System 1 -----
    if X(1) > systembyte % När det finns vatten i flaskan
        dz = Z(2);
        ddz = ((alpha-1)*((X(2).^2))-g*(L0+X(1))-CDR*Z(2).*abs(Z(2))+...
            (pA/roh)*(N*(((L-X0)./(L-X(1))).^gamma)-1)+g*X(1)+...
            Cf*X(1).*X(2).*abs(X(2))-CD*(X(2).^2)*(alpha^2-1))/L0;

        dx = X(2);
        ddx = ((alpha-1)*((X(2).^2))-g*(L0+X(1))-CDR*Z(2)*abs(Z(2))-(X(1)+L0)*ddz)./(X(1));

    % ----- System 2 -----
    elseif Z(1) > 0 % När vattnet i flaskan är slut
        dz = Z(2);
        ddz = -g-CDR*Z(2).*abs(Z(2));

        dx = 0;
        ddx = 0;
    % ----- System 3 -----
    else % När pet-flaskan träffat marken igen
        dz = 0; ddz = 0; dx = 0; ddx = 0;
    end
    f = [dx ddx dz ddz]';
end
```

9.2.2 Huvudfilen

```
% ----- Intervall -----
intervall_alpha = intervall_alpha_start;
intervall_fyllnadsgrad = intervall_fyllnadsgrad_start;
intervall_t = [0 10];

% ----- Beräkningen -----
for uppdatering = 0:intervall_uppdateringar
    intervall_halvering;
    diff_med_IP;
    tredimensionell_IP;
    if visa_IP_info
        IP_info;
    end
end

% ---- Plotting ----
if plotta_3D
    tredimensionell_plotting;
end
if plotta_2D
    tvadimensionell_plotting;
end

% ----- Variabelbyte -----
max_alpha = senaste_max_alpha;
max_fyllnadsgrad = senaste_max_fyllnadsgrad;
max_hojd = senaste_max_hojd;
```

9.2.3 Värdefilen

```
function [konstanter, konstanter_namn] = hamta_varden()

    % ----- Värden från uppgiften -----
    N = 10; % [] Starttryck
    gamma = 7/5; %[] Cp/Cv
    pA = 100000; %[Pa] Lufttryck (100 KPa)
    g = 9.81; %[m/s^2] Tyngdaccelerationen
    L = 0.5; %[m] Flaskans längd
    L0 = 0.006; %[m] Flaskan tom
    Cf = 0.001; %[m^-1] Inre friktion i vattnet
    CDR = 0.01; %[] Motståndskoefficient
    CD = 1.2; %[] Förlust i utströmningen
    roh = 1000; %[kg/m^3] Vattnets densitet
    konstanter = [N gamma pA g L L0 Cf CDR CD roh];

    % Konstanternas namn som strängar
    konstanter_namn = {'N','gamma','pA','g','L','L0','Cf','CDR','CD','roh'};
end
```

9.2.4 Intervallhalvering

```
if uppdatering > 0 % Halverar intervalllängden och centrerar kring
    % senaste optimeringen
    mitten_alpha = senaste_max_alpha;
    spridning_alpha = (intervall_alpha(2) - intervall_alpha(1))/4;
    % Delar med fyra för att halvera intervallet
    intervall_alpha = [mitten_alpha - spridning_alpha mitten_alpha + spridning_alpha];

    mitten_fyllnadsgrad = senaste_max_fyllnadsgrad;
    spridning_fyllnadsgrad = (intervall_fyllnadsgrad(2)-intervall_fyllnadsgrad(1))/4;
    intervall_fyllnadsgrad = [mitten_fyllnadsgrad-spridning_fyllnadsgrad
mitten_fyllnadsgrad+spridning_fyllnadsgrad];
end

iter_per_var = iter_per_var_start;

alphan = linspace(intervall_alpha(1),intervall_alpha(2),iter_per_var);
fyllnadsgrader = linspace(intervall_fyllnadsgrad(1),intervall_fyllnadsgrad(2),iter_per_var);
```

9.2.5 ODE-lösning

```
% Variabler som behövs nollställas vid varje iteration
senaste_ode45_max_hojd = 0;
max_hojd_matris = [];

i = 1; % Håller koll på vilket index nuvarande alpha har

for alpha = alphan
    j = 1; % Håller koll på vilket index nuvarande fyllnadsgrad har

    for fyllnadsgrad = fyllnadsgrader
        X0 = konstanter(5)*fyllnadsgrad; % L*fyllnadsgrad
        varden = [fyllnadsgrad alpha konstanter X0 systembyte];
        % Data som funktionen behöver

        f = @(t,u) systemet(t,u,varden);
        % Sparar funktionen som f

        u_start = zeros(4,1);
        u_start(1) = X0;
        % Startvärden för [X X' Z Z']'
        % Alla är noll förutom X

        % Löser systemet av differentialekvationer för t = [0 10]
        [tut, uut] = ode45(f,intervall_t,u_start,opts);

        tvadimensionell_IP;

        max_hojd_matris(i,j) = max_hojd;

        if max_hojd > senaste_ode45_max_hojd
            % För att hålla koll på hur mycket inerpolationen bidrog med
            differens_interpolation_max_hojd = max_hojd - utan_interpolation_ode45_max_hojd;
```

```
            if plotta_2D % Ifall datan vill användas till plotting efteråt
                max_tut = tut;
                max_uut = uut;
            end

            senaste_ode45_max_hojd = max_hojd;
        end
        j = j + 1;
    end
    i = i + 1;
end
```

9.2.6 Tvådimensionell interpolation

```
% Hittar max-värde för Z för varje kombination av alpha och
% fyllnadsgrad mha interpolation av grad 2
[utan_interpolation_ode45_max_hojd, index]= max(uut(:,3));
index_max = (index-1:index+1)'; % Tar fram index för de
% punkter som ska användas för interpolation

c = polyfit(tut(index_max),uut(index_max,3),2); % IP grad 2
t_max = roots(polyder(c)); % Tar fram t vid max_hojd
max_hojd = max(polyval(c,t_max));
```

9.2.7 Tredimensionell Interpolation

```
alphan_last = alphan; % Sparar sista alpha och fyllnadsgrad
% innan tredimensionell interpolation
fyllnadsgrader_last = fyllnadsgrader; % last ska vara låst

[interpolerad_alpha_matris,interpolerad_fyllnadsgrad_matris] = ndgrid(alphan,fyllnadsgrader);
% Gör om alphan och fyllnadsgrader till matriser där den första har
% alphan som kolumner och den andra fyllnadsgrader som rader
interpolerad_matris =
griddedInterpolant(interpolerad_alpha_matris,interpolerad_fyllnadsgrad_matris,max_hojd_matris,
IP_metod);
% Använder de matriser som togs fram med ndgrid och de beräknade max
% höjderna till att genomföra den tredimensionella interpolationen

% När intervallen blir mindre och ode45toleransen lägre klarar datorn
% av en lägre interpolationstolerans utan att ram-minnet tar slut
if uppdatering == 0 || ode45_tolerans > 1e-5
    IP_tolerans = 0.5e-3;
elseif uppdatering > 0
    IP_tolerans = 0.5e-4;
elseif uppdatering > 2
    IP_tolerans = 0.5e-5;
elseif uppdatering > 5
    IP_tolerans = 0.5e-6;
end

% Dummies
interpolations_fel = 100;
senaste_max_alpha = 100;
senaste_max_fyllnadsgrad = 100;
```

```
% ---- Tredimensionell interpolation ----
while norm(interpolations_fel) > IP_tolerans
    % Uppdatering av alphan och fyllnadsgrader med tätare intervall
    iter_per_var = iter_per_var * mult_av_iter_per_var;
    fyllnadsgrader =
linspace(intervall_fyllnadsgrad(1),intervall_fyllnadsgrad(2),iter_per_var);
    alphan = linspace(intervall_alpha(1),intervall_alpha(2),iter_per_var);

    % Tar fram en matris med interpolerade höjder utifrån större
    % matriser med tätare element av alpha och fyllnadsgrad
    [interpolerad_alpha_matris,interpolerad_fyllnadsgrad_matris] =
ndgrid(alphan,fyllnadsgrader);
    interpolerad_hojd_matris =
interpolerad_matris(interpolerad_alpha_matris,interpolerad_fyllnadsgrad_matris);

    % Hittar den interpolerade max höjden och det alpha och
    % fyllnadsgrad som tillhör
    [interpolerad_max_hojd, index_max_hojd] = max(interpolerad_hojd_matris(:));
    interpolerad_max_alpha = interpolerad_alpha_matris(index_max_hojd);
    interpolerad_max_fyllnadsgrad = interpolerad_fyllnadsgrad_matris(index_max_hojd);

    % Beräkning av felvektorn
    interpolations_fel = [interpolerad_max_alpha interpolerad_max_fyllnadsgrad]' - ...
        [senaste_max_alpha senaste_max_fyllnadsgrad]';

    % Sparar värdena för att kunna beräkna felet
    senaste_max_hojd = interpolerad_max_hojd;
    senaste_max_alpha = interpolerad_max_alpha;
    senaste_max_fyllnadsgrad = interpolerad_max_fyllnadsgrad;

end
```

9.2.8 Information från interpolationen

```
% Printar info angående resultatet av de två interpolationerna
disp(['Uppdatering ', num2str(uppdatering),' klar'])
disp(['Interpolation gav ', num2str(differens_interpolation_max_hojd),' m extra'])
disp(['3D interpolation gav ', num2str(senaste_max_hojd - senaste_ode45_max_hojd),' m extra'])
disp(['Totalt sett gav interpolation ', num2str(differens_interpolation_max_hojd +
senaste_max_hojd - senaste_ode45_max_hojd),' m extra'])
disp('-----------------------------------------------')
```

9.2.9 Tvådimensionell plotting

```
table(senaste_max_alpha,senaste_max_fyllnadsgrad,senaste_max_hojd,'VariableNames',{'Alpha','Fy
llnadsgrad','Hojd'})
% Presenterar resultatet på tabellform

figure(2)

% ----- Ploting av vattennivån i flaskan -----
t_efter_vatten_slut = find(uut(:,1) < 1e-9);
t_vatten_slut = tut(t_efter_vatten_slut(1));
% Vattnet tar aldrig slut då vi satt att programmet ska byta system
```

```
% då X < 1e-9

subplot(1,2,1)
% Gör grafen snygg
xlim([0 t_vatten_slut*1.1])
title('Vattennivå')
ylabel('X(t) [m]')
xlabel('t [s]')
grid on
hold on
area(tut,uut(:,1))

% ----- Animering av raketens flygtur -----
Z = @(x) spline(tut,uut(:,3),x);
% Använder spline för att kunna få jämna steg mellan punkterna

antal_steg = 150;
t_efter_nedslag = find(uut(:,3) < 0);
t_nedslag = tut(t_efter_nedslag(1));
t = linspace(0,t_nedslag,antal_steg);

subplot(1,2,2)
% Gör grafen snygg
xlim([0 ceil(t_nedslag)])
ylim([0 ceil(senaste_max_hojd/10)*10])
title('Rakethöjd')
ylabel('Z(t) [m]')
xlabel('t [s]')
grid on
hold on

% Själva animationen
for i = 1:antal_steg-1
    plot(t(i),Z(t(i)),'k.')
    drawnow
    pause(t(i+1)-t(i))
end
```

9.2.10 Tredimensionell plotting

```
grid on
hold on

% Plottar höjderna som togs fram av ode45
[fyllnadsgrader_matris, alphan_matris] = meshgrid(fyllnadsgrader_last,alphan_last);
plot3(alphan_matris,fyllnadsgrader_matris,max_hojd_matris,'k.','markersize',2);

% Plottar den interpolerade ytan
surf(interpolerad_alpha_matris,interpolerad_fyllnadsgrad_matris,interpolerad_hojd_matris,'edge
color','none')
colorbar

% Plottar den optimerade punkten
plot3(interpolerad_alpha_matris(index_max_hojd),
interpolerad_fyllnadsgrad_matris(index_max_hojd),
interpolerad_hojd_matris(index_max_hojd),'rx');
```

```
% Snyggar till figuren
xlabel('\alpha')
ylabel('Fyllnadsgrad')
zlabel('Höjd')
xlim([intervall_alpha(1),intervall_alpha(2)])
ylim([intervall_fyllnadsgrad(1),intervall_fyllnadsgrad(2)])
```

9.2.11 Körningsfilen

```
clear, close all, format long

% ----- Inställningar -----
ode45_tolerans = 1e-8;
opts = odeset('RelTol',ode45_tolerans);
intervall_uppdateringar = 8;
IP_metod = 'spline'; % Den interpolationsmetod griddedInterpolant använder
iter_per_var_start = 10; % Hur många punkter intervallen för alpha och fyllnadsgrad innehåller
mult_av_iter_per_var = 2; %Hur snabbt antalet punkter i intervallen för alpha och fyllnadsgrad
% ökar vid varje iteration vid den tredimensionella interpolationen
systembyte = 1e-9; % Vid det X det byts till system 2

plotta_2D = true;
plotta_3D = true;
visa_IP_info = true;

% ----- Inputs -----
intervall_alpha_start = [3 4];
intervall_fyllnadsgrad_start = [0.2 0.3];
intervall_t = [0 10];
konstanter = hamta_varden(); % Hämtar nödvändiga värden

% ----- Körning -----
main; % Kör huvudfilen
```

9.2.12 Felskattningsfil 1 – Konstanter

```
clear, close all, format long

% ----- Inställningar -----
ode45_tolerans = 1e-8;
opts = odeset('RelTol',ode45_tolerans);
intervall_uppdateringar = 8;
IP_metod = 'spline'; % Den interpolations metod griddedInterpolant använder
iter_per_var_start = 10; % Hur många punkter intervallen för alpha och fyllnadsgrad innehåller
mult_av_iter_per_var = 2; %Hur snabbt antalet punkter i intervallen för alpha och fyllnadsgrad
% ökar vid varje iteration vid den tredimensionella interpolationen
systembyte = 1e-9; % Vid det X det byts till system 2

plotta_2D = false;
plotta_3D = false;
visa_IP_info = false;

fel_procent = 0.01; % En-procentig förändring
```

```
% ----- Inputs -----
intervall_alpha_start = [3 4];
intervall_fyllnadsgrad_start = [0.2 0.3];
[konstanter_start, variabel_namn] = hamta_varden(); % Hämtar nödvändiga värden
konstanter = konstanter_start;
variabel_namn(11) = {'Total'};

% ----- Körning -----
main % Gör en vanlig beräkning för att ha som referens
referens = [max_alpha, max_fyllnadsgrad, max_hojd];
disp([num2str(100/(length(konstanter)+1)) '% klart'])

data = [];
for iter = 1:length(konstanter) % Genomför optimeringen för varje störd konstant
    konstanter = konstanter_start;
    konstanter(iter) = konstanter(iter) * (1+fel_procent);
    main;
    data = [data; abs(max_alpha/referens(1)-1), abs(max_fyllnadsgrad/referens(2)-1),
abs(max_hojd/referens(3)-1)];
    disp([num2str(100*(iter+1)/(length(konstanter)+1)) '% klart'])
end
data(11,:) = sum(data);

% ----- Resultat -----
table(referens(1),referens(2),referens(3),'VariableNames',{'Alpha','Fyllnadsgrad','Hojd'})
table(variabel_namn',100*data(:,1),100*data(:,2),100*data(:,3),'VariableNames',{'Variabel','Al
pha_avvikelse','Fyllnadsgrad_avvikelse','Hojd_avvikelse'})
```

9.2.13 Felskattningsfil 2 – Intervalluppdateringar

```
clear, close all, format long

% ----- Inställningar -----
ode45_tolerans = 1e-8;
opts = odeset('RelTol',ode45_tolerans);
IP_metod = 'spline'; % Den interpolations metod griddedInterpolant använder
iter_per_var_start = 10; % Hur många punkter intervallen för alpha och fyllnadsgrad innehåller
mult_av_iter_per_var = 2; %Hur snabbt antalet punkter i intervallen för alpha och fyllnadsgrad
% ökar vid varje iteration vid den tredimensionella interpolationen
systembyte = 1e-9; % Vid det X det byts till system 2

plotta_2D = false;
plotta_3D = false;
visa_IP_info = false;

antal_intervall_uppdateringar = 0:8;

% ----- Inputs -----
intervall_alpha_start = [3 4];
intervall_fyllnadsgrad_start = [0.2 0.3];
intervall_t = [0 10];
konstanter = hamta_varden(); % Hämtar nödvändiga värden

% ----- Körning -----
```

```
data = [];
for intervall_uppdateringar = antal_intervall_uppdateringar
    main;
    data = [data; intervall_uppdateringar max_alpha max_fyllnadsgrad max_hojd];
    disp([num2str((intervall_uppdateringar-
antal_intervall_uppdateringar(1)+1)/(length(antal_intervall_uppdateringar))*100) '% klart'])
end
data(:,5) = [0; diff(data(:,4))];

% ----- Resultat -----
array2table(data,'VariableNames',{'Intervall_uppdateringar','Alpha','Fyllnadsgrad','Hojd','Hoj
d_skillnad'})
```

9.2.14 Felskattningsfil 3 – ODE-tolerans

```
clear, close all, format long

% ----- Inställningar -----
intervall_uppdateringar = 8;
IP_metod = 'spline'; % Den interpolations metod griddedInterpolant använder
iter_per_var_start = 10; % Hur många punkter intervallen för alpha och fyllnadsgrad innehåller
mult_av_iter_per_var = 2; %Hur snabbt antalet punkter i intervallen för alpha och fyllnadsgrad
% ökar vid varje iteration vid den tredimensionella interpolationen
systembyte = 1e-9; % Vid det X det byts till system 2

plotta_2D = false;
plotta_3D = false;
visa_IP_info = false;

potenser = 3:11;

% ----- Inputs -----
intervall_alpha_start = [3 4];
intervall_fyllnadsgrad_start = [0.2 0.3];
intervall_t = [0 10];
konstanter = hamta_varden(); % Hämtar nödvändiga värden

% ----- Körning -----
data = [];
for potens = potenser % Gör optimeringen för olika ode45_toleranser
    ode45_tolerans = 10^(-potens);
    opts = odeset('RelTol',ode45_tolerans);
    main;
    data = [data; ode45_tolerans max_alpha max_fyllnadsgrad max_hojd];
    disp([num2str((potens-potenser(1)+1)/length(potenser)*100) '% klart'])
end

% ----- Resultat -----
array2table(data,'VariableNames',{'RelTol','Alpha','Fyllnadsgrad','Hojd'})
```

9.2.15 Felskattningsfil 4 – Systembyte

```
clear, close all, format long

% ----- Inställningar -----
ode45_tolerans = 1e-8;
opts = odeset('RelTol',ode45_tolerans);
intervall_uppdateringar = 8;
IP_metod = 'spline'; % Den interpolationsmetod griddedInterpolant använder
iter_per_var_start = 10; % Hur många punkter intervallen för alpha och fyllnadsgrad innehåller
mult_av_iter_per_var = 2; %Hur snabbt antalet punkter i intervallen för alpha och fyllnadsgrad
% ökar vid varje iteration vid den tredimensionella interpolationen

plotta_2D = false;
plotta_3D = false;
visa_IP_info = false;

potenser = 4:10;

% ----- Inputs -----
intervall_alpha_start = [3 4];
intervall_fyllnadsgrad_start = [0.2 0.3];
intervall_t = [0 10];
konstanter = hamta_varden(); % Hämtar nödvändiga värden

% ----- Körning -----
data = [];
for potens = potenser
    systembyte = 10^(-potens);
    main;
    data = [data; systembyte max_alpha max_fyllnadsgrad max_hojd];
    disp([num2str((potens-potenser(1)+1)/(length(potenser))*100) '% klart'])
end

% ----- Resultat -----
array2table(data,'VariableNames',{'Systembyte','Alpha','Fyllnadsgrad','Hojd'})
```