Gruppeopgave 2

Modellering og Simulering af Fysiske Systemer

Davy Eskildsen, Emil Johanson, Thorbjørn S. Kaiser, Mathias Svensson

9 December 2010

Indhold

1 Analyse af modellerings og simulerings processer

K. Erleben - Simulering og modellering af robotter og mennesker

Aktiklen har en kort gennemgang af problemet i den virkelige verden. Derefter gennemgå den de den matematiske model for problemet ved hjælp af en idealiserings proces. Herefter beskrives den næste proces, diskretisering, hvor ved der kommer frem mod den diskrete model. Til sidst sluttets der af med at fortælle lidt mere om hvordan virkeligheden ser ud.

Der er ingen simulering og derfor ingen resultater, hvilket også medføre at der ikke kan laves validering. Aktiklen kommer kon med en generel udgave af algoritmen og ikke en implementeret i et program, hvilket er skyld i at der ikke kan laves simulering og verificering.

Ved at implemetere algoritmen i et program ville vi kunne lave en simulering og derved kunne få nogle resultater som kunne validere på om vi med programmet kunne styre en skelet til at tage en kaffekop.

Hsu og Keyser - Piles of Objects

(a) Artiklen beskriver en mere effektiv metode til at simulere ensformige der ligger i “bunker”. Dog er formålet ikke at opnå den samme konfiguration af objekterne som den tilsvarende startkonfiguration vil føre til – derimod blot at opnå en effekt som ser realistisk ud.

Modellen der bruges er en standard fysiksimulation med friktion, tyngdekraft og lignende. Det videnskabelige arbejde går på, at have lavet en ændring i diskretiseringen.

I denne diskretisering kan objekterne deaktiveres, hvorved de ikke vil bevæge sig før de bliver vækket igen senere – enten fordi at bunken begynder at skride, eller fordi bunken bliver ramt af et objekt med høj fart. Denne deaktivering gør, at disse objekter (eventuelt midlertidigt) ikke behøves simmuleret. På denne måde opnår man en performanceforbedring, som eventuelt kan bruges til at bruge mere realistiske friktionssimmuleringer.

De verificerer at modellen er acceptabel, ved at sammenligne deres optimerede diskretisering med en mere standardiseret diskretisering. Valideringen består hovedsageligt i eyeballing.

(b) Det videnskabelige arbejde i artiklen består i at eftervise hypotesen “en simulation hvori vores optimering udnyttes vil være omtrent lige så præcis som en fuld simulation der ikke gør”.

Vi mener egentlig ikke der er noget umiddelbart kritisabelt i artiklen – men en stor del af artiklen indeholder i højere grad en beskrivelse af deres optimering end noget egentlig videnskabeligt arbejde.

2 Bold simulator

2.a)

```
function [x, y, vx, vy] = ball_step(x, y, vx, vy, dt, m, g)

x = x + dt*vx;
y = y + dt*vy;
vy = vy - dt*g;

end
```

2.b)

Funktionen var fuldt implementeret.

2.c)

1

Til verificering af den diskrete model kan man fx. løse de givne ligninger for den matematiske model eksakt og se hvorvidt denne og den diskrete approksimation udvikler sig tilnærmelsesvist ens.

Dette er naturligvist ikke tilfældet for alle differentialligninger, men det er her muligt da ligningen er tilpas simpel:

$$x(t) = t \cdot v_{x,0} + x_0 x'(t) \qquad = v_{x,0} \tag{1}$$

$$x''(t) = 0 \tag{2}$$

$$\tag{3}$$

$$y(t) = -\frac{1}{2} \cdot g \cdot t + v_{y,0} \cdot t + y_0 y'(t) \qquad = -g \cdot t + v_{y,0} \tag{4}$$

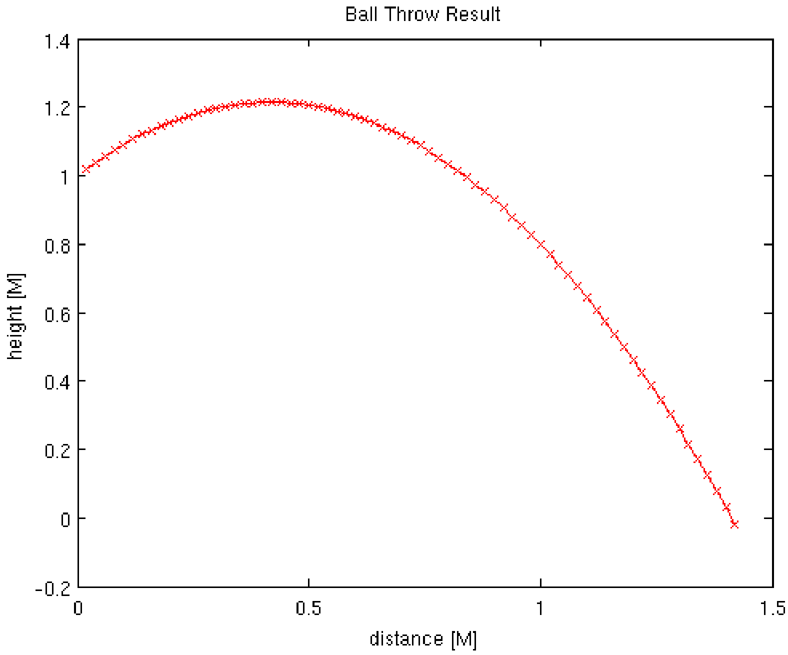
$$y''(t) = -g \tag{5}$$

$$\tag{6}$$

Til validering af simulatoren kunne man optage kastet af en bold (her gerne i et rum med vakuum...) og måle dennes position henad tidsaksen og aflede hastigheden til de forskellige punkter. Derefter kunne man plotte disse data ind i simulatoren, og se om denne giver et tilfredstillende output, altså et output der passer til den indsamlede data.

Man kunne også, i mangel på kamera, opstille nogle fornuftige hypoteser om hvordan simulatoren skal opføre sig når man "skruer" på de givne variable. Nogle eksempler:

- Ændring af m burde ikke have nogen indflydelse på vores simulation, da modellen er for simple til at massen spiller en rolle.
- Ændring af g burde gøre at hastigheden ud af y-aksen (v_y) per tidsenhed (Δt) ændres, hvor en højere g skulle resultere i en større ændring imod en lavere g.
- Ændring af Δt burde kun have betydning for hvor "finkornet" vores simulation er, dvs. at hastigheden med hvilken tiden flyder ikke burde have nogen indflydelse på den her underliggende model, men kun på hvor store stridt vi tager i simulationen. Da simulationen er en diskret approksimation betyder dette at en mindre Δt vil give en højere præcision, dvs. flere datapunkter fra startkast til at bolden rammer jorden.



Figur 1: Basiseksperiment

2

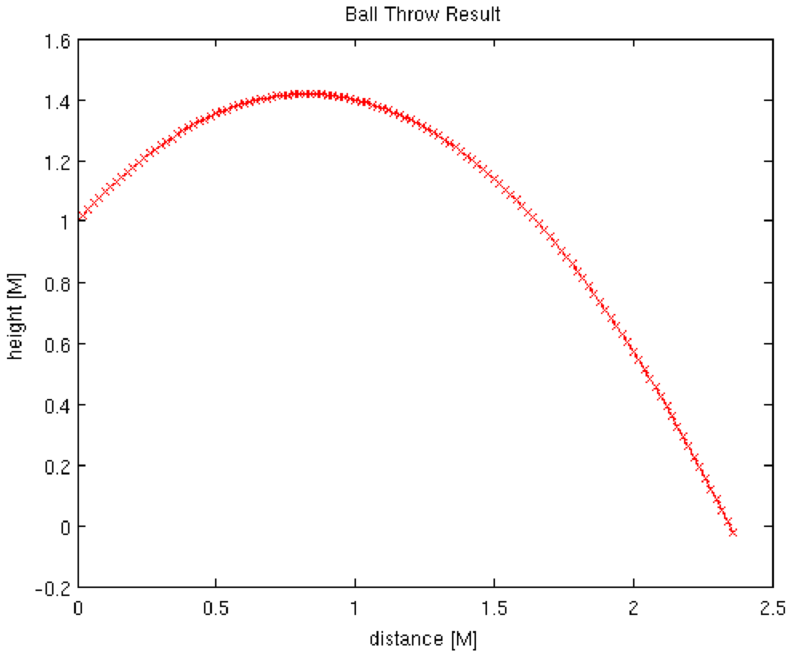
For at afprøve vores hypoteser starter vi først med at have et basiseksperiment, hvor $m = 1, g = 9.81, \Delta t = 0.01, x_0 = 0, y_0 = 1, v_{x,0} = 2, v_{y,0} = 2$. Resultatet kan ses på figur 1.

Vi starter ud med at skrue på massen m til hhvs. 0.5 og 2, en fordobling og en halvering. Vi får her som ventet præcis den samme kurve som basiseksperimentet, hvorfor vi ikke fremviser to ekstra ens figurer.

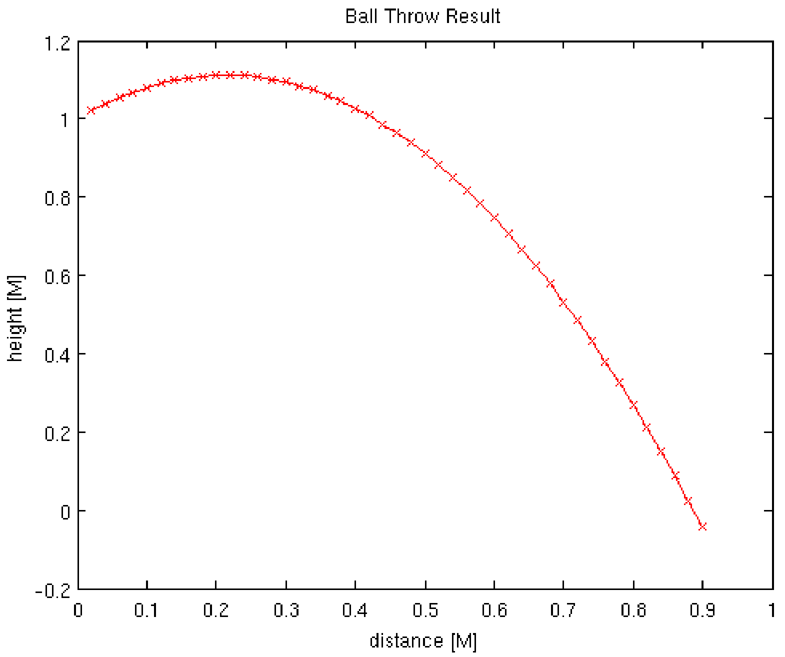
Vi skruer nu på tyngdeaccelerationen g til hhvs. 4.905 og 19.62, igen en fordobling og en halvering. Resultatet kan ses på figur 2 og 3. Vi ser her at ved lavere tyngdeacceleration flyver bolden længere og ved højere tyngdeacceleration falder bolden hurtigere til jorden.

Vi skruer nu til sidst på Δt til hhvs. 0.08 og 0.00125, her gange hhvs. dividere vi med 8 for at gøre forskellen mere tydelig. Resultatet kan ses på figur 4 og 5. Det er tydeligt at der er færre hhvs. flere datapunkter når vi gør Δt større hhvs. mindre.

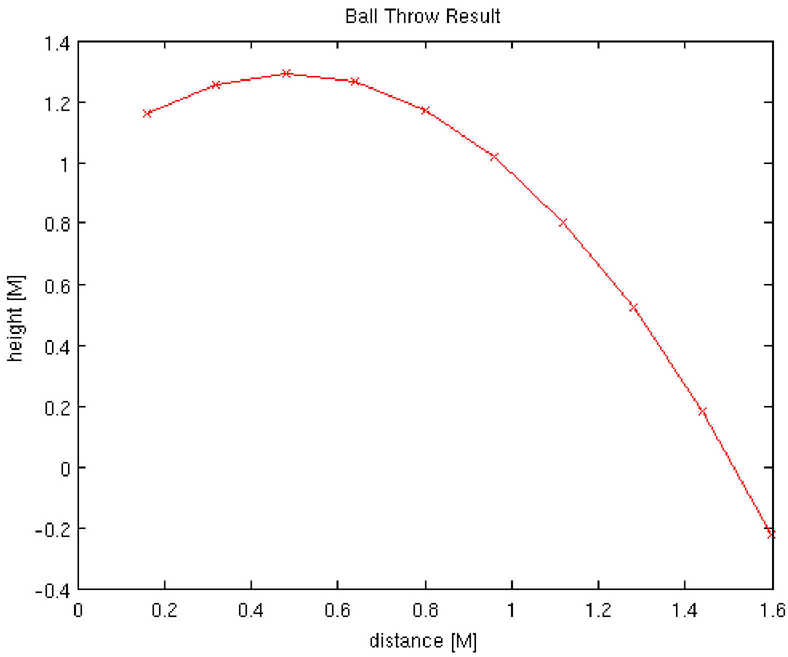
Vores hypoteser om vores simulation holder altså. Kurven for boldens bane opføre sig som forventet under forskellige fysiske forhold, og simulationen kan siges at være god ud fra vores forsimplede model af verdenen. Den holder selvfølgelig ikke i mere komplicerede sammenhæng (fx. med vindmodstand mm.) men det er heller ikke det den er bygget til at simulere.



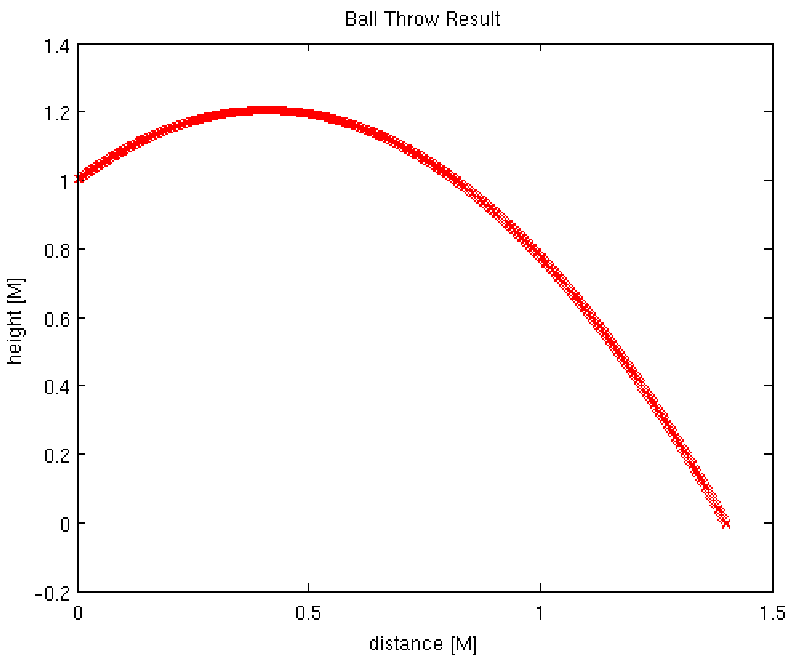
Figur 2: $g = 4.905$



Figur 3: $g = 19.62$



Figur 4: $\Delta t = 0.08$



Figur 5: $\Delta t = 0.00125$

3 Kant finder

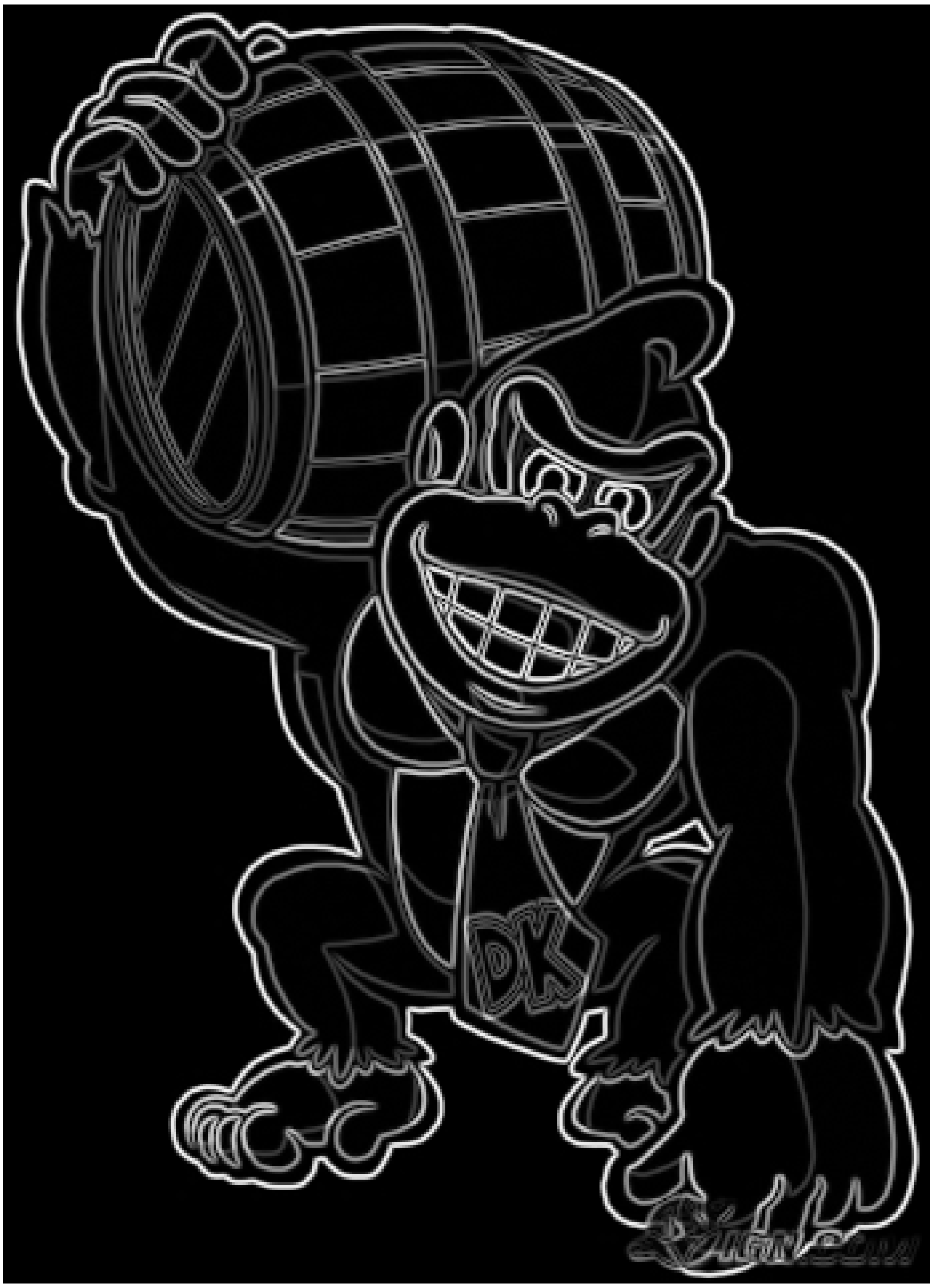
Verificering og validering

Vi kan køre billeder igennem vores kant-finde og derved se om billedernes kanter stemmer overens. Dette kan give en ide om hvor god vores kant-finder er. Se billeder:

Figur 6: Originalt lena billed

Figur 7: Lena billed efter kant finder funktionen

Figur 8: Originalt Donkey Kong billed



Figur 9: Donkey Kong billed efter kant finder funktionen