

# Trekroppsproblemet

Simon Sigurdhsson `ssimon@student...`  
Robin Persson Söderholm `robper@student...`

15 oktober 2009

## **Sammanfattning**

En relativt kort inblick i trekroppsproblemet.

# Innehåll

<b>1</b>	<b>Bakgrund</b>	<b>3</b>
1.1	Fysiken bakom problemet . . . . .	3
<b>2</b>	<b>Vårt problem</b>	<b>3</b>
2.1	Skala . . . . .	6
<b>3</b>	<b>Resultat</b>	<b>7</b>
3.1	Stabilitet . . . . .	7
3.1.1	En kropp står still . . . . .	7
3.1.2	Två kroppar står stilla . . . . .	7
3.2	Instabilitet . . . . .	8
3.2.1	Singulariteter . . . . .	8
3.2.2	Degradering till tvåkroppssystem . . . . .	9
3.3	Triviala fall . . . . .	9
3.3.1	Tre stationära kroppar . . . . .	10
3.3.2	Extrema starthastigheter . . . . .	10
<b>4</b>	<b>Diskussion</b>	<b>10</b>
4.1	Angående symmetri och stabila system . . . . .	10
4.2	Instabila system . . . . .	11
4.3	Modellens exakthet . . . . .	11
<b>A</b>	<b>Indata tillhörande figurer</b>	<b>12</b>

# Figurer

1	Stabila trekroppssystem . . . . .	6
2	Stabila system . . . . .	7
3	Två stillastående kroppar . . . . .	7
4	Instabila system . . . . .	8
5	En singularitet . . . . .	8
6	Degraderade system . . . . .	9

# 1 Bakgrund

Ända sedan *Newton* formulerade lagarna för klassisk fysik har man försökt att använda dessa i ett mer astronomiskt syfte; att försöka beräkna planetbanor. Generellt kallas det problem man då försöker lösa för  $n$ -kroppsproblemet. För  $n = 1$  har detta problem en trivial lösning, och för  $n = 2$  kan man analytiskt få fram en entydig lösning (i alla fall om man heter *Bernoulli*). Vill man sedan titta på  $n = 3$  stöter man på problem: det finns ingen (känd) exakt analytisk lösning.

Trekroppsproblemet har därför ställt till med en del problem, och eftersom det är ett relativt intressant problem för t.ex. astronomer, har man därför varit tvungen att numeriskt beräkna lösningar, något som inte är särskilt svårt. Man kan nämligen dela upp systemet i tre mindre system, som man kan behandla mer eller mindre separat. Det handlar ju i princip bara om en stegvis uppdatering av ett antal variabler beroende på tiden. Tyvärr introducerar detta lite mindre fel i resultatet, men med tillräckligt små tidssteg kan dessa fel ignoreras.

## 1.1 Fysiken bakom problemet

Fysiken bakom problemet är mycket enkel och baseras på en enda formel (1), som vi sedan överför till vektoriell form och summerar för varje objekt, vilket ger oss den totala kraften som påverkar varje objekt.

$$F = G \frac{m_1 m_2}{|\mathbf{x}_2 - \mathbf{x}_1|} \quad (1)$$

Eftersom vi sedan vill ha hastigheter, positioner och accelerationer istället för krafter måste vi även använda den triviala likheten  $F = ma$  för att med fördel kunna sätta upp ett fint ekvationssystem som ger oss precis det vi vill ha.

## 2 Vårt problem

Vi börjar med att konstatera att vi har ett ganska (o)trevligt ekvationssystem (2).

$$\ddot{\mathbf{q}}_j = \sum_{k \neq j} \frac{m_k (\mathbf{q}_k - \mathbf{q}_j)}{|\mathbf{q}_k - \mathbf{q}_j|^3} \quad j, k \in \{1, 2, 3\} \quad (2)$$

Detta ger oss tre andra gradens differentialekvationer, vilka vi inte kan lösa analytiskt. Vi söker helt enkelt hjälp hos MATLAB och dess fina ode-funktioner. Olyckligtvis tycker dessa funktioner inte riktigt om andraderiva-

tor, vilket man dock enkelt kan åtgärda genom att dela upp ekvationerna i sex första gradens differentialekvationer (3).

$$\begin{aligned} \mathbf{x}_j &= \sum_{k \neq j} \frac{m_k(\mathbf{x}_{k+3} - \mathbf{x}_{j+3})}{|\mathbf{x}_{k+3} - \mathbf{x}_{j+3}|^3} & j, k \in \{1, 2, 3\} \\ \mathbf{x}_j &= \mathbf{x}_{j-3} & j \in \{4, 5, 6\} \end{aligned} \quad (3)$$

Där  $\mathbf{x}_i$  är tre element långa bitar av en stor vektor  $\mathbf{x}$  som vi skickar till `ode`. Elementen  $\mathbf{x}_1 \dots \mathbf{x}_3$  motsvarar hastigheter och  $\mathbf{x}_4 \dots \mathbf{x}_6$  motsvarar positioner, och högerledet i varje ekvation är en derivata.

Eftersom problemet inte är styvt men kräver stor noggrannhet väljer vi att använda `ode113`, en metod som fungerar bra till problem som kräver stor noggrannhet.

Vi överför i varje fall våra ekvationer till en någorlunda trevlig MATLAB-kod (se Listing 1).

Listing 1: Differentialekvationerna i MATLAB-form

```
q1f = @(q1, q2, q3)((m2*(q2-q1)/norm(q2-q1)^3 +
    m3*(q3-q1)/norm(q3-q1)^3));
q2f = @(q1, q2, q3)((m1*(q1-q2)/norm(q1-q2)^3 +
    m3*(q3-q2)/norm(q3-q2)^3));
q3f = @(q1, q2, q3)((m1*(q1-q3)/norm(q1-q3)^3 +
    m2*(q2-q3)/norm(q2-q3)^3));
qa110 = [qv10 qv20 qv30 q10 q20 q30]';
qa11 = @(t, qq)([q1f(qq(10:12), qq(13:15),
    qq(16:18)); q2f(qq(10:12), qq(13:15),
    qq(16:18)); q3f(qq(10:12), qq(13:15),
    qq(16:18)); qq(1:3); qq(4:6); qq(7:9)]);
```

En kort förklaring kan vara lämplig. Vi börjar med att definiera enskilda funktioner för de  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$  vi definierade ovan; dessa är de ekvationer vi vill lösa. Sedan skapar vi en lämplig funktion `qa11` som lägger ihop alla våra ekvationer  $\mathbf{x}_1 \dots \mathbf{x}_6$ , och som även tar en tidsparameter för att vi sedan ska kunna skicka denna funktion till `ode113`.

Givetvis måste man även definiera de startvillkor som kommer att ingå i `qa110`, som vi använder som startvillkor i `ode113`. Vi kan nu enkelt använda `ode113` och `plot` för att rita upp lite trevliga plottar. Är man ambitiös kan man även animera dessa, vilket vi gjort (se Listing 2).

Listing 2: Beräkning och animering av resultatet

```
[T, Y] = ode113(qa11, 0:.01:500, qa110);
```

```

for j=15:1:size(T)
    hold on
    axis([-2 2 -2 2 -2 2])
    plot3(Y((j-14):j,10), Y((j-14):j,11),
        Y((j-14):j,12), 'r-');
    plot3(Y((j-14):j,13), Y((j-14):j,14),
        Y((j-14):j,15), 'k-');
    plot3(Y((j-14):j,16), Y((j-14):j,17),
        Y((j-14):j,18), 'b-');
    plot3(Y(j,10), Y(j,11), Y(j,12), 'r*');
    plot3(Y(j,13), Y(j,14), Y(j,15), 'k*');
    plot3(Y(j,16), Y(j,17), Y(j,18), 'b*');
    F(j-14) = getframe;
    clf;
end
movie(F,20,18)

```

Detta är relativt grundläggande MATLAB-kod och bör inte behöva förklaras. För den som är intresserad av att titta på animationerna (de kommer ju av förståeliga skäl inte finnas i animerad form i denna rapport) så kommer de att ligga på <http://web.student.chalmers.se/~ssimon/FFM233/projekt/>.

För att på ett bra sätt kunna presentera resultaten i pappersform använder vi en annan approach; vi plottar med en färgförändring som beror på kroppens hastighet (se Listing 3).

Listing 3: Plottning av problemet med avtagande färg

```

maxd = 0;
for j=2:1:size(T)
    if(norm(Y(j,10:11) - Y(j-1,10:11)) > maxd); maxd =
        norm(Y(j,10:11) - Y(j-1,10:11)); end
    if(norm(Y(j,13:14) - Y(j-1,13:14)) > maxd); maxd =
        norm(Y(j,13:14) - Y(j-1,13:14)); end
    if(norm(Y(j,16:17) - Y(j-1,16:17)) > maxd); maxd =
        norm(Y(j,16:17) - Y(j-1,16:17)); end
end
for j=2:1:size(T)
    hold on
    axis([-2 2 -2 2 -2 2])
    c = norm(Y(j,10:11) - Y(j-1,10:11))/maxd;
    plot3(Y((j-1):j,10), Y((j-1):j,11),
        Y((j-1):j,12), 'Color', [1 1/2-c*1/2
        1/2-c*1/2]);

```

```

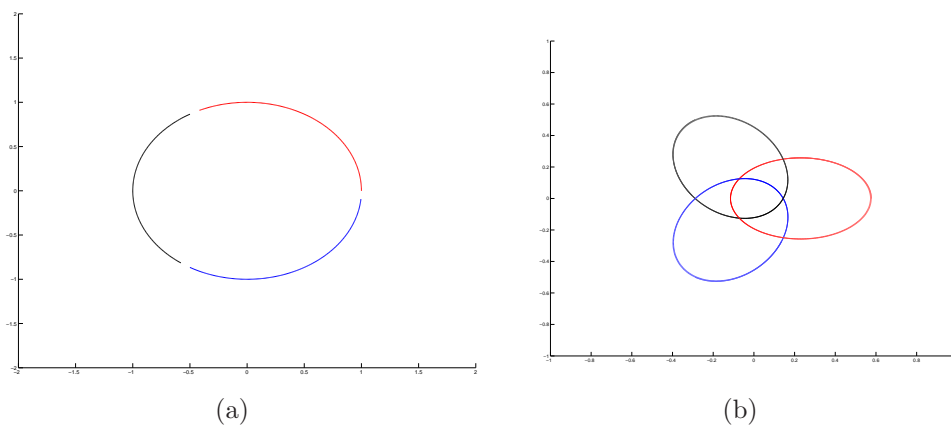
c = norm(Y(j,13:14) - Y(j-1,13:14))/maxd;
plot3(Y((j-1):j,13), Y((j-1):j,14),
Y((j-1):j,15), 'Color', [1/2-c*1/2 1/2-c*1/2
1/2-c*1/2]);
c = norm(Y(j,16:17) - Y(j-1,16:17))/maxd;
plot3(Y((j-1):j,16), Y((j-1):j,17),
Y((j-1):j,18), 'Color', [1/2-c*1/2 1/2-c*1/2
1
]);
end

```

Även detta är relativt enkel MATLAB-kod. Vi plottar även svagare färger när kropparna rör sig långsamt, för att enklare kunna visualisera kropparnas hastighet vid olika tidpunkter.

## 2.1 Skala

Något man bör beakta är hur stor skala man vill ha på indatan. Man kan ju givetvis försöka återskapa delar av vårt solsystem genom att använda de faktiska värden vi har gällande avstånd, massa och hastighet. Detta är dock högst onödigt, eftersom man med mycket mindre värden kan framkalla om inte exakta så i alla fall approximativa system som beter sig i princip likadant. Detta är en klar fördel eftersom man då kan ta mindre tidssteg utan att `ode113` blir sur och påstår sig hitta singulariteter som inte finns. Detta ger större exakthet och mindre beräkningsintensiva problem att lösa.

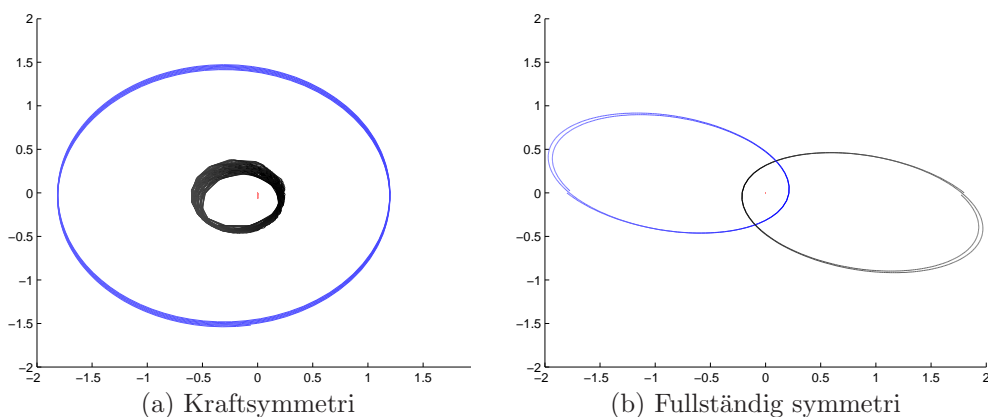


Figur 1: Med en del räknande kan man konstruera stabila trekroppssystem där alla kroppar rör sig; här har vi utgått från en liksidig triangel.

## 3 Resultat

### 3.1 Stabilitet

Stabilitet i trekroppsproblemet är *mycket* svårt att uppnå när alla kroppar har en begynnelsehastighet, man måste i princip konstruera system för att uppnå stabilitet. Ett bra exempel på sådana konstruktioner kan ses i figur 1. Man kan också göra en del antaganden för att uppnå stabila system.



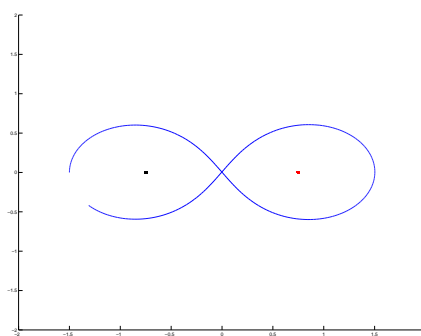
Figur 2: Stabila system utnyttjar ofta olika symmetrier för att hålla sig stabila; antingen kraftsymmetri (a) eller fullständig symmetri (b).

#### 3.1.1 En kropp står still

Antar man att en av kropparna inte har någon begynnelsehastighet kan man enkelt få fram system som blir stabila och till viss del efterliknar solsystem. Antar man dessutom att den stillastående planeten har betydligt större massa än de andra två blir det ännu lite enklare. I figur 2 visas två olika symmetriska och synnerligen stabila system.

#### 3.1.2 Två kroppar står stilla

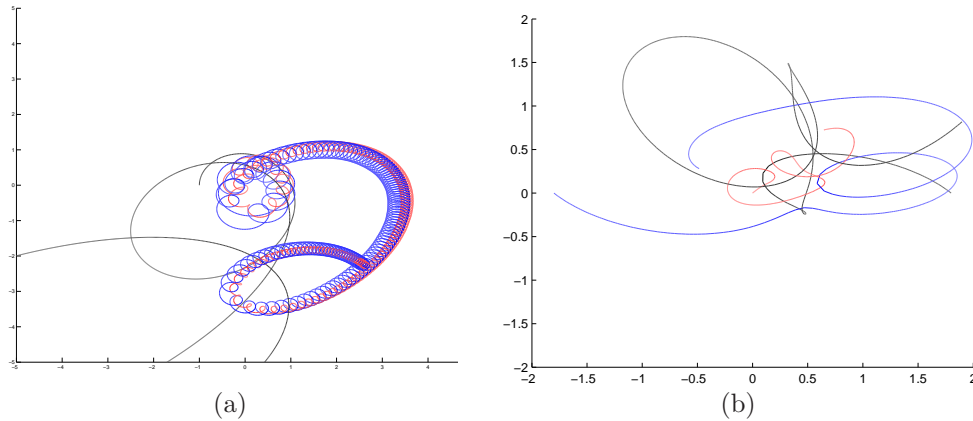
Fallet då två av kropparna står stilla (även känt som *Eulers* trekroppsproblem) kan lösas analytiskt m.h.a.



Figur 3: Detta system består av två helt stillastående kroppar, och kan lösas analytiskt.

elliptiska integraler. Detta är inte särskilt intressant för oss, men en fin plot av detta fall kan ses i figur 3.

### 3.2 Instabilitet

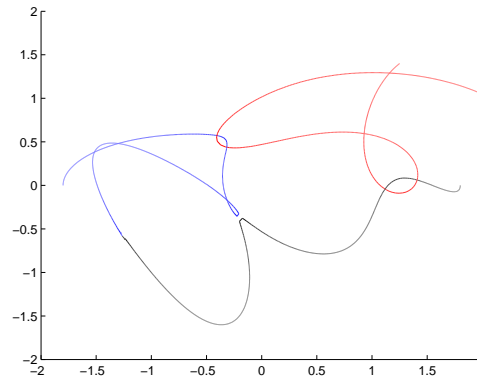


Figur 4: Instabila system kan ofta ge upphov till mycket intressanta mönster (a), men ibland blir det bara nonsens (b).

Trekroppsproblemet är ofta mycket instabilt, speciellt när man inte har någon symmetri, vilket gör att man ofta hamnar i lite smått kaotiska system. Figur 4a visar ett system som *nästan* har degraderat till ett system bestående av en “dubbelplanet” och en vanlig kropp. Figur 4b visar ett mindre intressant system som är betydligt mer kaotisk. Kör man denna simulering en längre tid så kommer det sannolikt att sluta med en singularitet.

#### 3.2.1 Singulariteter

En singularitet uppstår då två planeter krockar med varandra. Inte “krock” i det avseendet att de kommer nära varandra och deras banor hakar i varandra, som normalt avses med begreppet inom astronomi, utan att de faktiskt kolliderar med varandra. Detta leder även till en singularitet i själva



Figur 5: I detta system uppstår en singularitet när den svarta och den blå kroppen kommer allt för nära varandra.



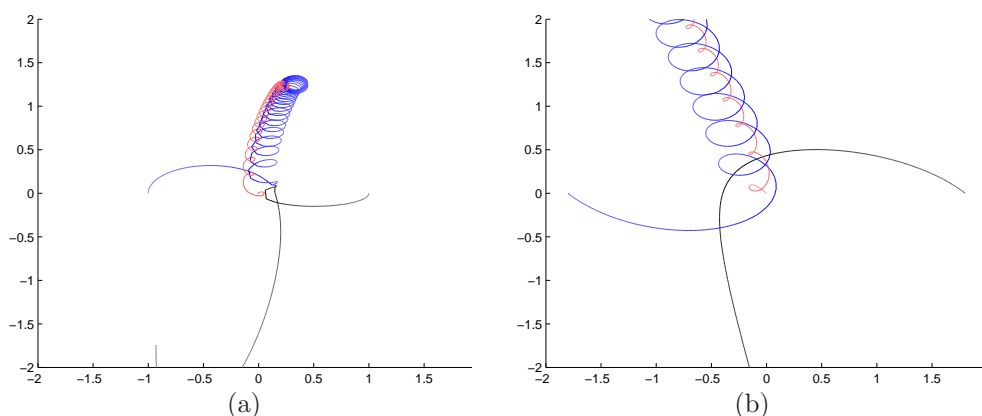
differentialekvationen, eftersom  $|\mathbf{q}_k - \mathbf{q}_j|^3$  då går mot 0, vilket gör att  $\ddot{\mathbf{q}}_j$  går mot  $\infty$ . Ett bra exempel på detta syns i figur 5.

Detta händer i princip alltid (förr eller senare) i instabila system; det ligger i instabila systems natur att förfalla till "enklare" system, och för trekroppssystemet finns det endast två olika sådana; en singularitet (som leder till ett enkroppssystem, oftast) eller en degradering till ett tvåkroppssystem.

### 3.2.2 Degradering till tvåkroppssystem

Detta resultat uppnås när en av kropparna av någon anledning avlägsnar sig så mycket från de andra två att dess effekt på systemet i princip kan försummas, som i figur 6. Detta sker i oftast p.g.a. en så kallad *gravitational slingshot*. Detta kräver dock att man har en planet som är betydligt tyngre än den man skickar iväg.

Ett annat, triviale fall som också ger ett tvåkroppssystem är när en av planeterna har en oerhört stor begynnelsehastighet eller ligger extremt långt från de andra två.



Figur 6: Två olika system som båda degraderas till tvåkroppssystem p.g.a. en så kallad *gravitational slingshot*.

### 3.3 Triviala fall

Trekroppssystemet har som de flesta andra problem även ett litet antal triviala fall. Sådana fall intresserar naturligtvis inte någon, eftersom de är just triviala.

### 3.3.1 Tre stationära kroppar

Om man antar att de tre kroppar som ingår i problemet har så liten massa och ligger så långt från varandra att krafterna de påverkar varandra med är försumbara, då får man ett trivialt fall av trekroppsproblemet. Tyvärr så är det här problemet inte någon man normalt är intresserad av.

### 3.3.2 Extrema starthastigheter

Antar man istället att alla tre kropparna har hastigheter sådana att de avlägsnar sig mycket snabbt ifrån varandra, då får man ett annat trivialt fall. Inte heller detta problem är något man normalt är intresserad av.

## 4 Diskussion

Även om det generella trekroppsproblemet ännu inte lösts analytiskt gör det inte livet svårt för de som vill räkna på sådana problem. Med dagens teknologi är det lätt att beräkna numeriska lösningar, och allt eftersom processorer blir snabbare går det även snabbare att lösa problemen.

När vi beräknat lösningar till våra startvärden, vilket vi i nästan alla fall har gjort på intervallet  $[0, 5]$  och med ett steg på 0.01, så har beräkningarna oftast inte tagit mer än ett par sekunder. Behöver man beräkna på större tidsintervall kommer lösningstiden öka icke-linjärt, men det bör man överleva.

För vissa begynnelsevärden kan problemet bli betydligt svårare för `ode113` att lösa. I dessa fall är problemet styvt och då bör man egentligen använda en vettigare metod, exempelvis `ode15s`.

### 4.1 Angående symmetri och stabila system

Vill man ha stabila trekroppsproblem måste man i princip få någon form av symmetri. Förutsätter man att en av kropparna står stilla får man två relativt enkla krav för att få symmetri i problemet;  $m_2|\mathbf{x}_1 - \mathbf{x}_2| = m_3|\mathbf{x}_1 - \mathbf{x}_3|$ , där  $\mathbf{x}_{2,3}$  är de kroppar som snurrar runt ett masscentrum liggandes i  $\mathbf{x}_1$ , samt att  $\mathbf{x}_2$  och  $\mathbf{x}_3$  ligger på *motsatta sidor* av  $\mathbf{x}_1$ . Det är sådana krav vi har ställt i systemet i figur 2a.

Vissa extrakrav måste också ställas för att det ska bli stabilt;  $\mathbf{x}_{2,3}$  måste ha vettiga hastigheter för att det inte ska sluta med en singularitet (eller ett icke-problem), och  $\mathbf{x}_1$  måste ha så stor massa att de andra kropparna inte påverkar dess rörelse.

Antar man däremot att alla tre kropparna rör på sig blir det mycket svårare att hitta en symmetri. Undantagsfallen är då det är så trivialt att

man kan lägga hela problemet i ett plan som rör sig och då få fram problemet ovan, t.ex. att alla kroppar har samma hastighet i  $\hat{z}$  (eller motsvarande för ett roterande plan) – i dessa fall kan man anta att sagda plan ligger still och helt enkelt hänvisa till lösningen ovan.

Ett bra sätt att uppnå symmetri i det fallet är att lägga kropparna i hörnen på en liksidig triangel. Mer generellt kan man säga att om man vill ett enklare fall av  $n$ -kroppsproblemet så ska alla kroppar ligga jämnt fördelade på en sfär.

Även i det fallet måste man ställa några extra krav; till exempel måste begynnelsehastigheterna vara vinkelräta mot vektorn  $\mathbf{q}-\mathbf{o}$  där  $\mathbf{o}$  är masscentrum, och dessutom vara riktade så att kropparna “roterar” i samma riktning. Detta är givetvis enklare att räkna på om man förutsätter att alla kroppar har samma massa, men det är inte nödvändigt.

## 4.2 Instabila system

Vi har tidigare påstått att det endast finns två resultat i ett instabilt system; en singularitet eller en divergens. Detta följer i princip ur definitionen av ett stabilt system; ett stabilt system är ett sådant som efter en viss tid börjar oscillera och därmed fastnar i ett repeterande mönster. Detta gör att stabila system inte hamnar i en singularitet eller divergens. Instabila system hamnar därmed per definition inte i ett sådant oscillerande läge, och för eller senare kommer det att sluta med att kropparna antingen rör sig allt för långt från varandra eller att de kolliderar.

## 4.3 Modellens exakthet

Metoden är väldigt exakt, förutsatt att man använder rätt metod för att numeriskt lösa differentialekvationerna och att man tar lagom stora tidssteg.

Något man bör ta i beaktande är dock att metoden baseras på *Newtons* klassiska mekanik, och att den därför bortser från relativitetsteori. Relativitetsteori är som bekant extra relevant när man jobbar med astronomi, och därför kan det vara värt att skriva om problemet för att ta detta i beaktande.

## A Indata tillhörande figurer

	$m_{\text{röd}}$	$m_{\text{svart}}$	$m_{\text{blå}}$	$\mathbf{q}_{\text{röd}}$	$\mathbf{q}_{\text{svart}}$	$\mathbf{q}_{\text{blå}}$	$\dot{\mathbf{q}}_{\text{röd}}$	$\dot{\mathbf{q}}_{\text{svart}}$	$\dot{\mathbf{q}}_{\text{blå}}$
<b>Figur 1a</b>	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	(1,0,0)	$(-\frac{1}{2}, \frac{\sqrt{3}}{2}, 0)$	$(-\frac{1}{2}, -\frac{\sqrt{3}}{2}, 0)$	(0,1,0)	$(-\frac{\sqrt{3}}{2}, -\frac{1}{2}, 0)$	$(\frac{\sqrt{3}}{2}, -\frac{1}{2}, 0)$
<b>Figur 1b</b>	1	1	1	$(\frac{1}{\sqrt{3}}, 0, 0)$	$(-\frac{1}{2\sqrt{3}}, \frac{1}{2}, 0)$	$(-\frac{1}{2\sqrt{3}}, -\frac{1}{2}, 0)$	$(0, \frac{1}{\sqrt{3}}, 0)$	$(-\frac{1}{2}, -\frac{1}{2\sqrt{3}}, 0)$	$(\frac{1}{2}, -\frac{1}{2\sqrt{3}}, 0)$
<b>Figur 2a</b>	250	0.4	0.2	(0,0,0)	(-0.6,0,0)	(1.2,0,0)	(0,0,0)	(0, -15.8135,0)	(0,15.8198,0)
<b>Figur 2b</b>	8	2	2	(0,0,0)	(1.8,0,0)	(-1.8,0,0)	(0,0,0)	(-1,1,0)	(1, -1,0)
<b>Figur 3</b>	—	—	—	—	—	—	—	—	—
<b>Figur 4a</b>	10	5	5.5	(0,0,0)	(-1,0,0)	(1,0,0)	(0,0,0)	(0, $\pi$ ,0)	(0, $-\pi$ ,0)
<b>Figur 4b</b>	8	2	2	(0,0,0)	(1.8,0,0)	(-1.8,0,0)	(0.2,0.2,0)	(-1,1,0)	(1, -1,0)
<b>Figur 5</b>	1	2	2	(1.25,1.4,0)	(1.8,0,0)	(-1.8,0,0)	(-0.5, -0.5,0)	(0, -0.25,0)	(0,0.25,0)
<b>Figur 6a</b>	5	2	2	(0,0,0)	(1,0,0)	(-1,0,0)	(0,0,0)	(0, -0.5,0)	(0,1,0)
<b>Figur 6b</b>	8	2	2	(0,0,0)	(1.8,0,0)	(-1.8,0,0)	(-0.2,0.2,0)	(-1,1,0)	(1, -1,0)