UPPSALA UNIVERSITET

Institionen för informationsteknologi
Beräkningsvetenskap

Besöksadress:
Lägerhyddsvägen 1, hus 10

Postadress:
Box 337
751 05 Uppsala

Telefon:
018–471 0000 (växel)

Telefax:
018–51 19 25

Hemsida:
http://www.it.uu.se

Department of Information Technology
Scientific Computing

Visiting address:
Lägerhyddsvägen 1, hus 10

Postal address:
Box 337
SE-751 05 Uppsala
SWEDEN

Telephone:
+46 18–471 0000 (switch)

Telefax:
+46 18–51 19 25

Web page:
http://www.it.uu.se

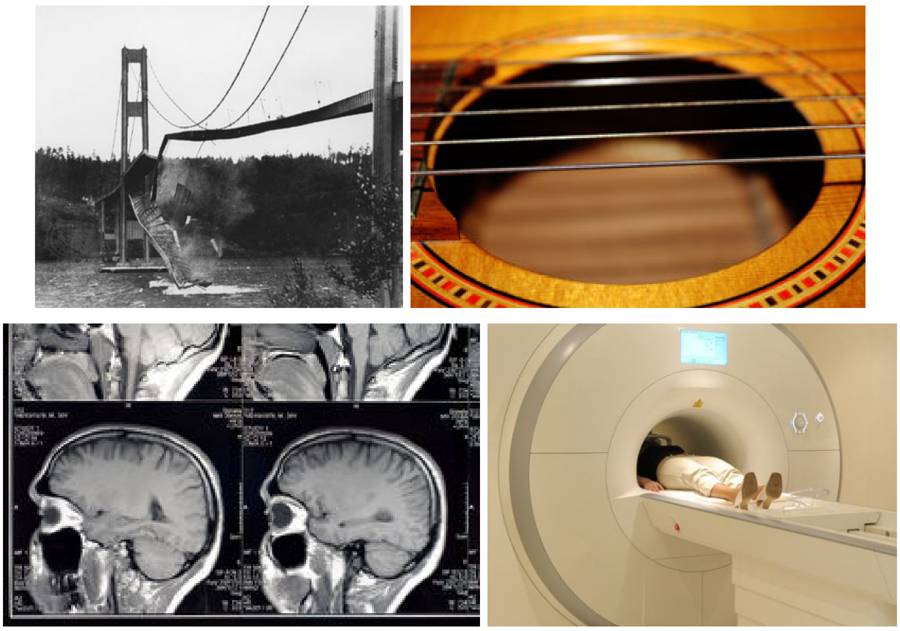
Introduktion till Beräkningsvetenskap

Projekt 1: Kaotisk pendel

*I detta projekt ska du lösa en ODE, som beskriver en dämpad och driven pendel, med numeriska metoder. Du kommer att skriva egna ODE lösare och även testa på en av Pythons (SciPy) egna ODE lösare, `solve_ivp`. Du kommer analysera stabilitet, noggrannhet, resonans och söka efter kaotiskt beteende hos pendeln. Nyckelbegrepp är **stabilitet** och **konvergens**.*

Bakgrund

Resonans, även kallat självsvängning eller egensvängning är ett allmänt fenomen hos oscillerande eller vibrerande system som innebär att även en svag periodisk yttre störning (pådrivande kraft) nära systemets egenfrekvens kan leda till att systemets svängningsamplitud, accelerationer och energiinnehåll ökar kraftig. Fenomenet har stor teknisk betydelse bland annat ur säkerhetssynpunkt, tex vid konstruktioner av roterande maskiner, byggnader och broar (se tex Tacoma Narrows Bridge). Men fenomenet resonans utnyttjas även vid konstruktion av musikinstrument och viss mätutrustning, tex vid Magnetisk ResonansTomografi (MRT, MRI eller MR). I Figur 1 illustreras några exempel där resonans förekommer.



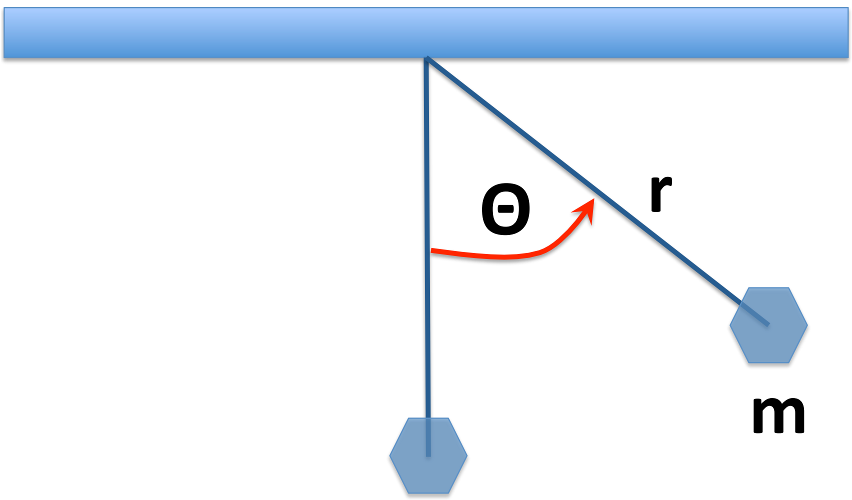
Figur 1: Olika konstruktioner och tillämpningar med koppling till resonans.

UPPSALA UNIVERSITET

Vi kommer i detta projekt att studera ett oscillerande system (här en ODE) som beskriver en driven och dämpande pendel. Kaotiska egenskaper hos system är ett annat intressant fenomen som studeras i kaosforskning, tex dynamiska system. Kaosforskningen sträcker sig över flera ämnesområden till exempel matematik, fysik, ekonomi, meteorologi och ekologi. Kaotiska system är system där små förändringar i begynnelsevillkor (starttillstånd) ger stora och på sikt oförutsägbara skillnader i tillstånd, ett fenomen som har fått namnet Fjärilseffekten. En förutsättning för att ett system ska kunna bete sig kaotiskt är att det inte är linjärt. Vi kommer här att studera ett ickelinjärt system (ODE) som beskriver pendel-rörelsen och studera när kaos uppkommer och hur det kan se ut.

Matematisk beskrivning av en pendel

Om vi endast tar hänsyn till gravitationskraften beskrivs pendelns rörelse av $\theta'' = -\frac{g}{R}\sin(\theta)$, där θ är vinkeln som pendeln har (se Figur 2). Pendeln har massan m och dess längd är r. För att påbörja pendelrörelsen startar vi pendeln (vid tiden $t = 0$) från vila (dvs $\theta' = 0$) från en given vinkel $\theta = \theta_0$. $\theta' = 0$ och $\theta = \theta_0$ är våra begynnelsevillkor, dvs pendelns position och hastighet vid tiden $t = 0$.



Figur 2: Illustration av en pendel med massa m och längd r.

I verkligheten måste vi även ta hänsyn till friktionskraft, som vi här antar är proportionell mot pendels hastighet, $b\,\theta'$, där b är en friktionsparameter. Till sist lägger vi på en drivande kraft $F_0 \cos(\omega t)$, där F_0 är amplituden och ω frekvensen hos den drivande kraften. Den fullständiga beskrivningen av pendelrörelsen ges av följande ODE,

$$\theta'' = -\frac{g}{r}\sin(\theta) - \frac{b}{m}\theta' + \frac{F_0}{m\,r}\cos(\omega t)\,. \tag{1}$$

UPPSALA UNIVERSITET

I ekvation (1) har vi 6 parametrar: m, r, g, b, F_0 och ω. Genom att införa $2\beta = \frac{b}{m}$, $\omega_0^2 = \frac{g}{r}$ och $\gamma = \frac{F_0}{m\,r\,\omega_0^2}$, kan vi skriva om (1) med endast 4 parametrar (ω_0, ω, β och γ),

$$\theta'' = -\omega_0^2 \sin(\theta) - 2\,\beta\theta' + \gamma\omega_0^2 \cos(\omega\,t)\ . \tag{2}$$

För små pendelrörelser, dvs $|\theta| << 1$, kan vi linearisera (2) genom att Taylor-utveckla kring $\sin(0)$, vilket ger oss $\sin(\theta) \simeq \theta$ då $|\theta| << 1$. Om vi samtidigt sätter friktionen till noll, dvs $b = 0$, får vi följande förenklade (linjära) ODE,

$$\theta'' = -\omega_0^2\theta + \gamma\omega_0^2 \cos(\omega\,t)\ . \tag{3}$$

Parametern ω_0 brukar benämnas systemets naturliga (eller egen) frekvens. (Egenvärdena till (3) ges av $\pm i\omega_0$, där i är imaginära enheten). Om den drivande kraften sätts nära (eller lika med) systemets naturliga frekvens uppstår resonans, vilket kan vara mycket destruktivt och är något man i praktiska tillämpningar vill undvika (tex i en centrifug eller en bro).

Uppgifter

Här följer 4 obligatoriska uppgifter (Deluppgift 1-4), och en slutlig frivillig uppgift. Dessa uppgifter testar centrala delar av kursen, och är en bra övning inför den slutliga examinationen.

Deluppgift 1

Man kan visa att

$$\theta(t) = c_1 \cos(\omega_0\,t) + c_2 \sin(\omega_0\,t) + \frac{\gamma\,\omega_0^2 \cos(\omega\,t)}{\omega_0^2 - \omega^2}\ , \tag{4}$$

är en analytisk lösning till (3), då $\omega \neq \omega_0$. Konstanterna c_1 och c_2 ges av begynnelsevillkoren. Sätt begynnelsevillkoren till $\theta = \theta' = 0$ och bestäm c_1 och c_2. För fallet då $\omega = \omega_0$ ges den analytiska lösningen till (3) av,

$$\theta(t) = d_1 \cos(\omega_0\,t) + \frac{\gamma\,\omega_0}{2} t\ \sin(\omega_0\,t)\ . \tag{5}$$

Sätt begynnelsevillkoren till $\theta = \theta' = 0$ och bestäm d_1. Det finns alltså en linjär tillväxt då vi driver pendeln med egenfrekvensen $\omega = \omega_0$ (vilket brukar benämnas resonans). Observera dock att vi här antagit små rörelser och försummat friktionskrafter, vilket betyder att det fullständiga systemet beskrivet av (2) inte nödvändigtvis behöver uppvisa resonans då $\omega = \omega_0$. Skapa en funktion i Python, som givet en tidsvektor returnerar den analytiska lösningen i motsvarande punkter. De övriga 3 parametervärdena (ω_0, ω och γ) ska tilldelas utanför funktionen, och ska vara inparametrar till funktionen. Skriv funktionen så att den automatiskt väljer rätt analytisk lösning, (4) eller (5), beroende på om $\omega \neq \omega_0$ eller $\omega = \omega_0$. Redovisa plottar för de analytiska lösningarna (mellan $t = 0$ och $t = 10$) då: 1) $\omega = 2\,\pi$, $\omega_0 = 3\,\pi$, $\gamma = 0.1$, och 2) $\omega_0 = \omega = 3\,\pi$, $\gamma = 0.1$. Använd $N = 1000$ intervall för en kontinuerlig kurva.

UPPSALA
UNIVERSITET

Deluppgift 2: ***Konvergens***

Skriv ett program i Python som löser (3) med RK4. Sätt $\omega = 2\pi$, $\omega_0 = 3\pi$, $\gamma = 0.1$. Sätt begynnelsevillkoren till $\theta = \theta' = 0$. Använd den analytiska lösningen (4) för att kontrollera felet (c_1 och c_2 bestämdes i deluppgift 1). Lös fram till sluttiden $T = 9.9$. Låt N beteckna antalet steg (intervall). Tidssteget (k) ges då enligt $k = \frac{T}{N}$. Använd $N = 200$ och plotta både lösningen och absolutbeloppet av felet som funktion av tiden. Redovisa dessa två figurer. Felet vid tiden T då vi använt N steg betecknar vi som $e_N = |y(T) - y_N|$, där $y(T)$ är den analytiska lösningen vid $t = T$, och y_N den numeriska lösningen framtagen med RK4, vid sluttiden T, dvs efter N steg med tidssteget $k = \frac{T}{N}$.

Nu ska beräkningskoden ni skrivit verifieras, dvs att den RK4 metod ni implementerat verkligen löser (3). Man kan visa att RK4 har noggrannhets-ordning 4, vilket betyder att felet e_N vid en given sluttid $t = T$ ska minska som k^4, där k är tidssteget. Om man mäter felen vid $t = T$ där vi använt N och $2N$ steg kan man uppskatta metodens noggrannhets-ordningen (konvergensen), som vi betecknar med $q(2N)$, med följande formel:

$$q(2N) = log_2 \left(\frac{e_N}{e_{2N}} \right) . \tag{6}$$

Bestäm nu e_{200}, e_{400} och e_{800}, och använd sedan formel (6) för att bestämma $q(400)$ och $q(800)$. Redovisa resultatet i form av en tabell med de olika N i första kolumn, motsvarande e_N i andra kolumn och slutligen $q(N)$ i sista kolumn (se Tabell 1, för tydligt redovisad tabell-mall). Här bör man se att $q(400)$ och $q(800)$ blir ungefär 4 (avrundat till närmsta heltal), annars är det någon bugg i koden.

N	e_N	q(N)
200	e_{200}	
400	e_{400}	$q(400)$
800	e_{800}	$q(800)$

Tabell 1: Mall för en tydlig tabell över feluppskattning och konvergens. Ersätt med beräknade resultaten.

Deluppgift 3: ***Numerisk stabilitet***

Beräkna återigen (3) med din RK4 implementering fram till $t = 40$ med begynnelsevillkoren $\theta = \theta' = 0$. Testa först fallet då $\omega_0 = \omega = 3\pi$ och $\gamma = 0.1$. Använd $N = 800$ och redovisa den numeriska lösningen (mellan t=0 och t=40). Använd sedan $\omega = 2\pi$, $\omega_0 = 3\pi$, $\gamma = 0.1$ och redovisa lösningen då $N = 800$. Redovisa figurer på de två olika lösningarna. Kommentera skillnader i lösningarnas utseende, och försök förklara orsakerna.

Testa nu att successivt använda färre antal steg (gör detta för fallet då $\omega = 2\pi$, $\omega_0 = 3\pi$, $\gamma = 0.1$) : $N = 400$, $N = 200$, $N = 150$, $N = 135$,

$N = 134$, $N = 133$, $N = 132$, $N = 131$. Sammanfatta dina iaktagelser och förklara vad som händer då $N < 133$.

Testa slutligen att beräkna lösningen med Euler framåt (RK1), där vi använder $N = 10000$. Förklara varför lösningen inte blir noggrann, trots det mycket korta tidssteget.

Deluppgift 4: *Kaos*

Skriv ett program i Python som löser (2) med den (i *SciPy*) inbyggda ODE lösaren `solve_ivp` där vi använder RK45 metoden (`method='RK45'`). Sätt begynnelsevillkoren till $\theta = \theta' = 0$ och parametrarna till $\omega = 2\pi$, $\omega_0 = \frac{3}{2}\omega$ och $\beta = \frac{1}{4}\omega_0$. Räkna fram till sluttiden $T = 40$. I python kan anropet till `solve_ivp` se ut som följande (där vi skapat en funktion som heter F, och där vi tilldelar värden för alla parametrar),

```
import numpy as np
import scipy.integrate as ode

f = np.array([0,0])
tspan = (0,T)
SOL = ode.solve_ivp(F, tspan, f, ...
    method='RK45',args=(omega0,omega,beta,gamma),rtol=1e-12, ...
    atol=1e-12)
```

Redovisa lösningen θ som funktion av tiden då $\gamma = 1$. Stega nu γ från 1.04 i steg om 0.01 fram till 1.16, och redovisa dina iakttagelser. Om du tittar närmare på lösningen då $1.06 \leq \gamma$ kan man se att lösningen efter en viss tid följer en periodisk lösning med perioden 1, vilket motsvarar perioden hos den drivande kraften med $\omega = 2\pi$. Om du tittar närmare på lösningen då $\gamma = 1.07$ kan man se att lösningen efter en viss tid ställer in sig, men hoppar mellan två olika lösningar som vardera har perioden 2 (men olika amplitud), vilket kallas *bifurkation* (och brukar analyseras med ett bifurkationsdiagram inom teorin för dynamiska system). För $\gamma \geq 1.07$ börjar alltså systemet att uppvisa ett kaotiskt beteende. Redovisa lösningarna då $\gamma = 1.06$, $\gamma = 1.07$, $\gamma = 1.16$.

Valfri uppgift: *Konvergens för ickelinjär model*

Denna uppgift är ej obligatorisk, men rekommenderas. Vi ska här öva på att verifiera en beräkningsmetod, när en analytisk lösning saknas. Skriv ett program i Python som löser (2) med RK4. Sätt begynnelsevillkoren till $\theta = \theta' = 0$ och parametrarna till $\omega = 2\pi$, $\omega_0 = \frac{3}{2}\omega$, $\beta = \frac{1}{4}\omega_0$ och $\gamma = 1.06$. Räkna fram till sluttiden $T = 1$. Då en analytisk lösning saknas kan man istället ersätta denna med en noggrann numerisk lösning, en så kallad *referenslösning*, där man använt ett tillräckligt litet tidssteg (i detta fall) och att metoden är stabil. Detta bygger på antagandet att den numeriska metoden som används för att räkna fram *referenslösningen* konvergerar mot en antagen lösning till den matematiska (i detta fall ODE) modellen.

Om man tror på sin egen beräkningskod kan man använda denna för att beräkna referenslösningen, men ett säkrare alternativ är att använda en redan verifierad beräkningskod. I detta fall använder vi oss av implementeringen från deluppgift 4, där vi använder `solve_ivp`.

Felet vid tiden T då vi använt N steg betecknar vi som tidigare med $e_N = |y(T) - y_N|$, men där nu $y(T)$ är framräknad med `solve_ivp`. y_N är den numeriska lösningen framtagen med RK4 med tidssteget $k = \frac{T}{N}$. Bestäm nu e_{200}, e_{400} och e_{800}, och använd sedan formel (6) för att bestämma $q(400)$ och $q(800)$. Redovisa resultatet i form av en tabell (med de olika N i första kolumn, motsvarande e_N i andra kolumn och slutligen $q(N)$ i sista kolumn (där första värdet då sätts tom). Här bör man, om referenslösningen är korrekt, se att $q(400)$ och $q(800)$ blir ungefär 4 (avrundat till närmsta heltal) om man implementerat metoden korrekt.