



MATHEMATICA-TILLFÄLLE 1

Mekanik, vt 2019

Marcus Berg, 2019-04-02

Övning 0: Uppvärmning

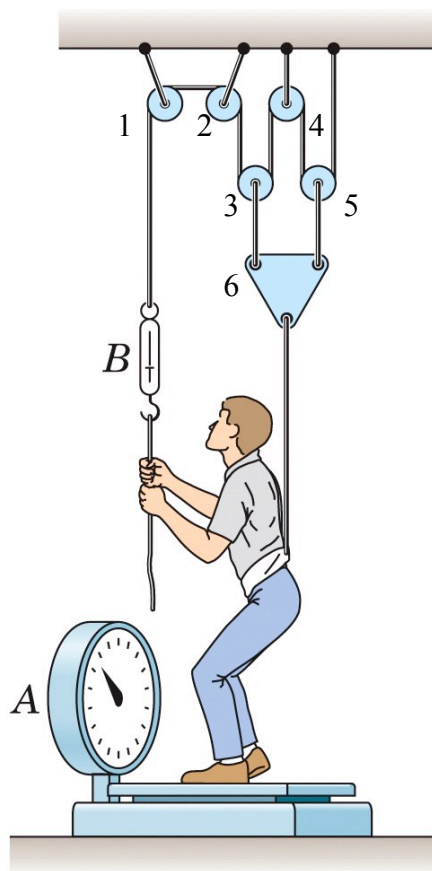
Jag har lagt upp en Mathematica-fil (*notebook*) för det här på Canvas som heter `Mathematica0.nb`. Har du tid kan du börja med **inte använda den filen**, utan skriv in nedanstående själv i ditt eget tomma arbetsblad, så långt du hinner och orkar, för att öva.

Tips: Med Mathematica kan man råka tilldela variabler värden och sedan glömma att de redan har värden. Menyvalet “*Evaluation/Quit Kernel/Local*” nollställer Mathematicas **kärna**, som knappen C på en miniräknare (alternativ: `ClearAll` och `Remove["Global`*"]`). Kärnan avslutas *inte* för att man stänger filen! Man kan sedan starta om kärnan för hand, eller om man utför ett kommando så startar den om automatiskt (kan ta en sekund). Ett annat allmänt tips är att tänka i **regler**, som i uppgift “c” nedan. Notera också de små klamrarna längst till höger i arbetsbladet, som man kan manipulera uttryck med. Hitta måsklamrar `{ }` på din dator, de finns!

- Starta nytt arbetsblad “New Document”. Börja med att göra en **rubrik** genom att trycka Cmd-3 (Mac) eller Alt-3 (PC). Tryck sedan **pil ner** för att börja. För mindre textkommentarer rekommenderar jag 7 istället för 3. För menyålskare ligger alla alternativ under “Format”.
- Din första Mathematica-beräkning: skriv in `1+1` och tryck **shift-Return**.
- Din första användning av **transformationsregler**: `x + y /. {x->1, y->1}`
- Gör en Taylor-utveckling (eller McLaurin-utveckling, eller Laurent-utveckling ... i allmänhet kallas det bara “serieutveckling”): `Series[Sin[x], {x, 0, 10}]`; Prova några olika funktioner, t.ex. e^x som skrivs `Exp[x]`. Hur hög ordning (“10” i exemplet) vågar du gå till?
- Gör en oändlig summering: `Sum[1/n^2, {n, 1, Infinity}]`;
Jämför med ändligt antal termer i summan (“trunkering”): `Sum[1/n^2, {n, 1, 10}]`
och sedan `%/N` på nästa rad. Här står `%` för “förra uttrycket” och `//N` för numerisk utvärdering av ett symboliskt uttryck som ett bråk. Använd hjälpsidorna.
- Använd kommandot `simplify` på tre olika trigonometriska uttryck, t.ex. $\cos^2 x + \sin^2 x$, $\sin x \cos x$, $1 + \tan^2 x$. Tycker du det funkar bra?
Om det inte alltid funkar som du vill, prova `TrigReduce` istället för `simplify`.
- och gör en plott av en fördefinierad funktion: `Plot[ArcSin[x], {x, -1, 1}]`
- Derivera funktionen i föregående uppgift med `D[ArcSin[x], x]`. Vad händer?
- Integrera svaret i föregående uppgift med `Integrate[%, x]`. Vad händer?
- Ta någon enkel funktion av två variabler, t.ex. $x^2 + y^2$ eller $\sin(xy)$, och gör `Plot3D`. (Senare: prova `ContourPlot` och `DensityPlot` och skriv ut eller spara som PDF.)
- Skriv `DSolve[(y'[x])^2 == y[x], y[x], x]` Förstår du vad som händer?
Notationen `C[1]` betyder “integrationskonstant 1”.
- Prova kommandot `Manipulate[Factor[x^n+1], {n, 2, 10, 1}]` (taget från hjälpsidan för “Manipulate”). Du får ett skjutbart “reglage” som du ändrar parametern n med. Försök göra samma sak med en plott av sinusfunktionen $\sin(wx+d)$, där du ändrar w och d .

Övning 1: Statik ger linjära ekvationssystem

Här är uppgift 3/7 ur gamla boken (Meriam & Kraige), som uppvärmning för uppgift 6-64 i nya boken (Hibbeler). En mekanikstudent vill väga sig själv på en känslig våg och gör därför följande underliga upphängning. Han drar med 76 N i dynamometern B och läser av 286 N på vågen. Vad är studentens tyngd?



Problem 3-7
© John Wiley & Sons, Inc. All rights reserved.

Jag har skrivit in ekvationerna för de 9 objekten i en Mathematica-fil `Mathematica1.nb`, ladda ned den från Canvas, men läs inte bara igenom! Uppgiften är att **prova**: vad händer om man

- tar bort krafter, t.ex. från friläggning av trissa 5 ser vi direkt att $\mathbf{H}=\mathbf{C}$, men vad händer om vi aldrig anger den regeln, utan låter Mathematica försöka lösa systemet ändå?
- sätter okända krafter med "fel tecken", t.ex. mg uppåt? (Uppenbart fel här, bara som exempel.)
- Lägg till regeln för \mathbf{H} igen (jfr. uppgift a), men gör ett felaktigt antagande, t.ex. $\mathbf{H}=\mathbf{0}$?
- ändrar hur hårt studenten drar (kraft \mathbf{C}), men har samma tyngd, vad blir kraften på vågen (\mathbf{N})?

Sättet jag skrev in ekvationssystemet på är "robot-mässigt", dvs. utan att tänka. Det finns en viss poäng med det i programmering: det är i efterhand lätt att läsa kod om ingen djup tanke har behövts för att skriva den. Men det intressanta som kvarstår när du skall göra uppgift 6-64 är ändå: finns det något smartare sätt än att lösa alla ekvationer? (Vänta med att svara till kap.6!)