DN1212 Numeriska metoder och grundläggande programmering FN1 08-11-18

Hedvig Kjellström
hedvig@csc.kth.se

Denna föreläsning

- Om numeriska metoder
- Om programmering (Staffan Romberger)
- Information om kursen
- Grundläggande begrepp
  - Potensserier, taylorutvecklingar
  - Linearisering
  - Gränsvärdesprocesser
  - Feluppskattning
  - Överbestämda ekvationssystem, minsta-kvadratanpassning
- Ni behöver inte anteckna, föreläsningen kommer finnas på kursens hemsida: http://www.csc.kth.se/utbildning/kth/kurser/DN1212/numpm09/

Vad är numeriska metoder?

- Numerisk metod:
  - En algoritm (följd av regler) som löser ett kontinuerligt matematiskt problem med ett ändligt antal aritmetiska operationer (+, -, *, /)
  - $\Rightarrow$ **Approximativ lösning i sifferform**
- Numerisk analys:
  - Konstruktionen av numeriska metoder genom att ersätta matematikens oändliga processer med ändliga processer
  - Analys av metoderna för att bilda sig en uppfattning om hur noggranna de är
  - **Alltid approximationer, måste veta hur stora felen är**

Varför numeriska metoder?

- Finns en mängd problem som inte går att lösa analytiskt (med papper och penna)
- Ex 1: Ickelinjär ekvationslösning

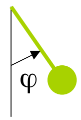
$$\int_1^2 \frac{\sin x}{x} dx = ?$$

Numerisk integrering

(kan inte hitta primitiv funktion)

- Ex 2: Diffekvationen för pendelrörelse

$$\frac{d^2\varphi}{dt^2} + \frac{g}{L}\sin\varphi = 0$$



Numerisk lösning av diffekvationer

Varför numeriska metoder?

- Ex 3: Stora linjära ekvationssystem $Ax = b$
  om A = 2x2, OK
  om A = 100x100, jobbigt…

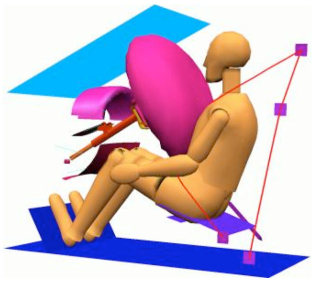
Numerisk matrisalgebra

Tillämpningar

- Man vill ofta matematiskt modellera ett fysikaliskt problem
- Modellen blir för komplicerad att lösa analytiskt (med papper och penna)

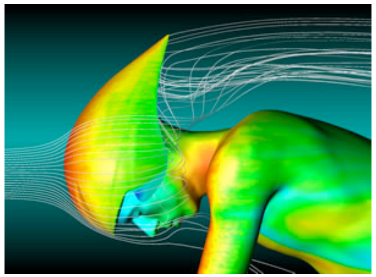
Ex 1: Krocksimulering

- Modell för hur revbenen påverkas av bältestrycket vid en krock
- Fördelar mot verkliga experiment:
  Vid exp kanske bara kan mäta tryck vid fåtal punkter - här överallt
  Kan här göra en massa simuleringar vid olika hastigheter med olika kroppsformer - biligare än exp



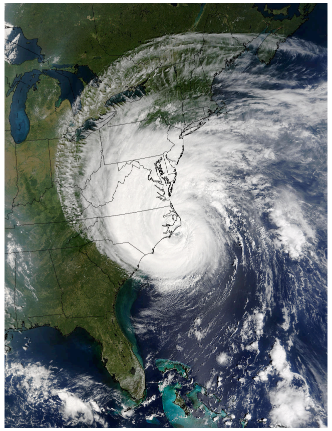
Ex 2: Strömningsmekanik

- Beräkning av strömningsfältet runt en cyklist samt tryckfördelningen på kroppen
- Bra vid utveckling av cyklar och hjälmar



ROYAL INSTITUTE OF TECHNOLOGY

Ex 3: Vädersimuleringar

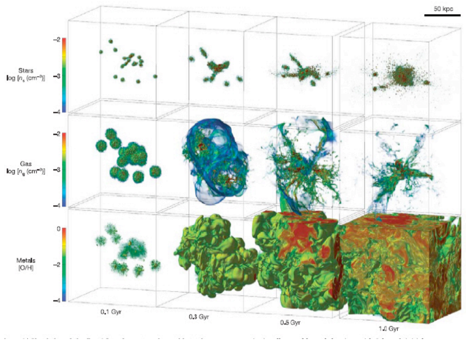


- Alla väderprognoser bygger på numeriska beräkningar av vädermodeller

ROYAL INSTITUTE OF TECHNOLOGY

Ex 4: Astrofysik

- Simulering av det första giga-året i en galax (10^{11} ggr solens massa)
- Utförd med superdatorn ”Earth Simulator” i Japan
- Simuleringen bygger på vetenskapligt framtagna modeller för universums sammansättning
- (Nature 440 644, http://physicsworld.com/cws/article/news/24536)



ROYAL INSTITUTE OF TECHNOLOGY

Ex 5: Filmindustrin

- Simulering av stormigt vatten (Titanic)
- Matematisk modell för hur vågor rör sig
- Fördel mot verklig inspelning
  Pengar…

ROYAL INSTITUTE OF TECHNOLOGY

Poäng med numeriska metoder

- Kan lösa väldigt komplicerade problem genom att kombinera flera enkla metoder!
- Problemen skrivs om på en form som passar för datorberäkningar
- Måste alltså kunna programmera - och det handlar första halvan av kursen om!

Programmeringsdelen

Staffan Romberger, föreläsare

ROYAL INSTITUTE OF TECHNOLOGY

Om DN1212

- Mål: Efter genomgången kurs ska studenten kunna
  - identifiera olika matematiska problem och skriva om dem på en form som är lämplig för numerisk behandling
  - välja lämplig numerisk metod för behandling av det givna problemet
  - motivera val av metod genom att redogöra för fördelar och begränsningar
  - välja en algoritm som leder till effektiva beräkningar och implementera den i ett programspråk lämpat för beräkningar t ex Matlab
  - presentera resultaten på ett relevant och illustrativt sätt
  - göra tillförlitlighetsbedömning av resultaten
  - bryta ner större problem i hanterliga delar och skriva egna funktioner för dessa i programspråket
  - använda styr- och datastrukturer
  - hantera filer på olika sätt, både vid inläsning och utskrift
  - använda färdiga funktioner ur programspråkets bibliotek (t ex Matlabs bibliotek) för beräkning, visualisering och effektiv programmering
  - skriva välstrukturerade program i programspråket

numme

programmering

ROYAL INSTITUTE OF TECHNOLOGY

Om DN1212

- Kursaktiviteter:
  - 6 föreläsningar, 6 salsövningar om programmering
  - 10 föreläsningar, 10 salsövningar om numeriska metoder
  - 13 datorsalspass
- Examination:
  - 5 datorlaborationer, redovisas muntligt i grupper om två
  - 1 projekt, redovisas skriftligt
  - 1 tenta (27 maj 9-12)
- Datorlabbar i grupper om två
  - Inte fler än två - men det går bra att jobba ensam

Om ni söker en labbkompis, kom fram till tavlan i rasten på denna föreläsning

ROYAL INSTITUTE OF TECHNOLOGY

Om DN1212

- Betyg enl tentaresultat (se kurs-PM)
- Bonuspoängsystem till tentan:
  - Lab 2 redovisad innan 11/12 ger 1 bp
  - Lab 3 redovisad innan 29/1 ger 1 bp
  - Lab 4 redovisad innan 25/3 ger 1 bp
  - Lab 5 redovisad innan 7/5 ger 1 bp
- 11/12, 29/1, 25/3, 7/5 kommer det vara **lång kö** - redovisa tillfället innan för att slippa vänta!
- Bonuspoängen gäller tom mars 2010
- Kursomfång: 9 hp ≈ 240 timmar

  ⇒ 7 timmar per vecka utöver schemalagda aktiviteter

ROYAL INSTITUTE OF TECHNOLOGY

Om DN1212

- Kursledare
  Hedvig Kjellström (kursansvarig, föreläsare numerikdelen)
  Staffan Romberger (föreläsare programmeringsdelen)

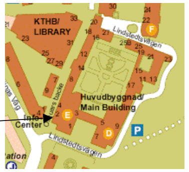
- Övningsledare
  Grupp 1: Staffan Romberger (programmering) och Beatrice Frock (numme)
  Grupp 2: Hedvig Kjellström
  Grupp 3: Jon Häggblad

- Samt datorsalshandledare, både CSC och M-teknologer

Om DN1212

- Litteratur:

  Kursbunt (övningshäfte, kortfattad nummebok, matlabhäfte, kurs-PM, extenta, lab 1-3)
  Köpes (150:-) på CSC/Nadas studentexp, Osquars backe 2, plan 2 (må-fr 9.30-12, må-to 13-15)



  Stephen J Chapman, Matlab Programming for Engineers (PEng)
  Köpes (ca 320:-) på Kårbokhandeln

  Peter Pohl, Grundkurs i numeriska metoder (GNM)
  Köpes (ca 320:-) på Kårbokhandeln

Om DN1212

- Kursutvärdering kommer ske i slutet av kursen - men kommentarer är välkomna när som helst under kursen. Prata med Hedvig eller Staffan!

- Mer om kursen
  I kurs-PM
  På hemsidan:
  http://www.csc.kth.se/utbildning/kth/kurser/DN1212/numpm09/

- Kolla hemsidan en gång i veckan för nyheter, labblydelser, föreläsningsanteckningar mm.

Matematiska grunder för numeriska metoder

Potensserier - taylorutveckling

ROYAL INSTITUTE OF TECHNOLOGY

- Betrakta en funktion *f(x)*
- I allmänhet inte möjligt att tex hitta uttryck för *f′(x)* eller hitta rötterna (nollställena) till *f(x)*
  (Däremot kan tex $f'(x_0)$ för en specifik punkt x_0 beräknas)
- Skriv om *f(x)* på en form som är lättare att använda för numeriska metoder

- Taylorutveckling av f:

$$f(x) = f(x_0) + (x - x_0) \cdot f'(x_0) + \frac{(x-x_0)^2}{2!} \cdot f''(x_0) + \ldots$$

dvs

$$f(x) = \sum_{i=0}^{\infty} \frac{(x-x_0)^i}{i!} \cdot f^{(i)}(x_0)$$

Potensserier - taylorutveckling

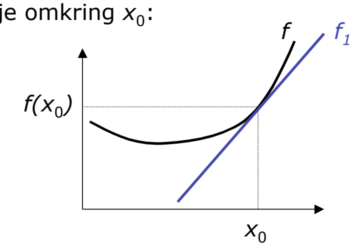
ROYAL INSTITUTE OF TECHNOLOGY

- Approximation till *f(x)*: ta bara med de första *n* termerna i taylorutvecklingen

$$f_n(x) = \sum_{i=0}^{n} \frac{(x-x_0)^i}{i!} \cdot f^{(i)}(x_0)$$

- Tex, approximera *f(x)* med rät linje omkring x_0:

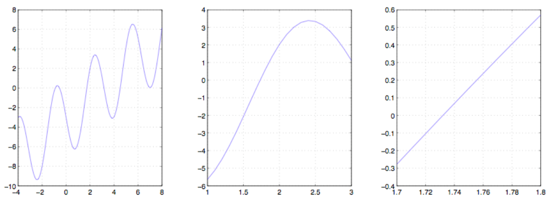
$$f_1(x) = f(x_0) + (x - x_0) \cdot f'(x_0)$$



Approximation med räta linjer

ROYAL INSTITUTE OF TECHNOLOGY

- De flesta numeriska metoder innehåller approximation med rät linje
  lokal linearisering runt x_0 = 1-gradens taylorutveckling runt x_0
- Måste välja skala: Hur långt från x_0 är approx giltig?
- Exempel: Hitta nollställen till *f(x)* = *x* - 4 *sin*(2*x*) - 3
  Approximera $f(x) \approx f_1(x)$ och beräkna $f_1(x) = 0$
  Men $f(x) \approx f_1(x)$ gäller bara i närheten av x_0



Gränsvärdesprocesser

ROYAL INSTITUTE OF TECHNOLOGY

- Serie av beräkningar som konvergerar mot lösningen till problemet (det har ni sett: gräns = limes, tex $\lim_{\delta x \to 0} \frac{\delta y}{\delta x} = \frac{dy}{dx}$)
- Man stannar när man nått önskad noggrannhet (kommit tillräckligt nära lösningen)
- Två typer av processer:

Rekursion

Approximationer till en *följd av storheter* beräknas succesivt med en och samma beräkningsprocess

Tex, beräkna y_t för t = 1, 2, … från en diffekvation $\frac{dy}{dt} = f(y)$

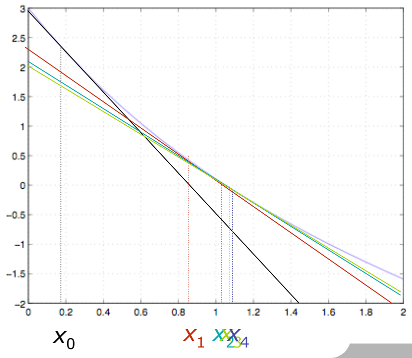
Iteration

Approximationer till en *enda storhet* beräknas succesivt med en och samma beräkningsprocess

Tex, beräkna lösningen till *f(x)* = 0 med approximationer x_n som kommer allt närmare den verkliga lösningen när *n* ökar

Exempel på iteration

- Hitta lösningar till $x = 3\,e^{-x}$
- Iterativ metod för ekvationslösning: Newton-Raphson
  - Kan formulera problemet som att vi vill hitta rötterna (nollställena) till $f(x) = 3\,e^{-x} - x$
  - Börja i x_0 För n=1,2,...
  - 1. Lokal linjär approx $f_1(x)$
  - 2. $x_{n+1} : f_1(x) = 0$
  - Stanna när $\delta x = x_{n+1} - x_n < \Delta$
- Mer om Newton-Raphson senare i kursen

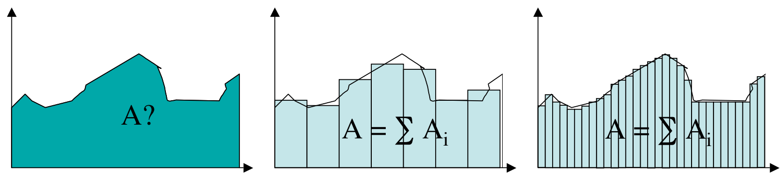


Beräkningsfel och effektivitet

- Numerisk analys handlar ju om att hitta approximativa lösningar till problem som inte går att lösa analytiskt
  - Hur bra är approximationen?
  - Hur lång tid tar den att beräkna?
  - Ofta en avvägning: en oändligt bra approximation tar oändligt lång tid

Måste kunna uppskatta felet!

- Exempel: numerisk integration



Felkällor

- Så, var kommer felen ifrån i approximationen?
  - Viktigt att veta källan!
- Modellfel
  - Fysikalisk/matematisk modell är en förenklad bild av verkligheten (astrofysikexemplet i början)
  - Tex, system (pendel) som beskrivs av diffekvationen

$$\frac{d^2\varphi}{dt^2} + \frac{g}{L}\sin\varphi = 0$$

  - Kan förenklas som

$$\frac{d^2\varphi}{dt^2} + \frac{g}{L}\varphi = 0$$

  - Men det fungerar bara för små φ

Felkällor

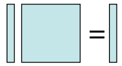
- Fel i indata
  - När vi ska simulera från pendelexemplet kanske startvinkeln φ_0 är felaktig - vi kanske mätt upp den slarvigt
  - Viktigt att veta hur det påverkar simuleringsresultatet φ_t
  - Vi talar om ett systems **kondition/stabilitet/konditionstal**: fel i utdata = konditionstal * fel i indata
  - Mer om konditionstal senare i kursen
- Trunkeringsfel
  - Uppkommer då vi avbryter gränsprocesserna
  - Tex, vi låter inte δx gå ända till 0 när vi approximerar en derivata: $\lim_{\delta x \to 0} \frac{\delta y}{\delta x} = \frac{dy}{dx}$
- Avrundningsfel
  - Datorn representerar tal med ett ändligt antal bitar (32 eller 64)
  - Spelar roll om vi har storheter med väldigt olika värden: $x = 2^{31}\,y$

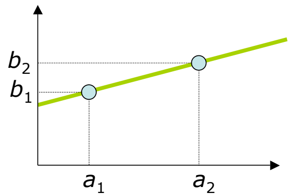
Noggrannhetsordning

- Många numeriska metoder har ett trunkeringsfel *E* som beror av en steglängd *h:*
  $E \approx c\, h^p$
  där *p* är metodens *noggrannhetsordning*, dvs dess "effektivitet"
  $\therefore \log E \approx \log c + p * \log h$
  där *p* är lutningen hos den räta linjen i en loglog-plot
- Mer om felanalys senare i kursen

Överbestämda ekv.system, minstakvadratmetoden

- En av de vanligaste numeriska beräkningarna i ingenjörs-tillämpningar är att lösa linjärt ekvationssystem A**x** = **b**
- Om A är kvadratisk med det(A) ≠ 0, gausselimination
  Tex, anpassning av rät linje till två punkter:
  $b = b_0 + b'*a$



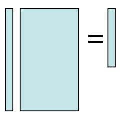


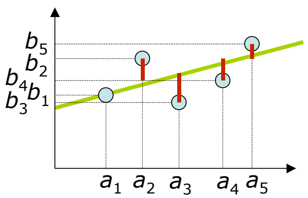
$$\begin{bmatrix} 1 & a_1 \\ 1 & a_2 \end{bmatrix} \begin{bmatrix} b_0 \\ b' \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

I Matlab: `x = b\A;`

Överbestämda ekv.system, minstakvadratmetoden

- Vad händer om det är fler än två punkter?
- Överbestämt ekvationssystem A**x** = **b**:
  Kan inte hitta en rät linje som går genom alla punkter (a_i, b_i)
  Vi får ta en approximation $b \approx b_0 + b'*a$, dvs lösa A**x** ≈ **b**
  Bra approximation är den som minimerar *summan av kvadraterna av avstånden* mellan linjen och punkterna





$$\begin{bmatrix} 1 & a_1 \\ 1 & a_2 \\ 1 & a_3 \\ 1 & a_4 \\ 1 & a_5 \end{bmatrix} \begin{bmatrix} b_0 \\ b' \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}$$

- Mer om detta senare i kursen

I Matlab: `x = b\A;`

Härnäst…

- Kolla hemsidan
  http://www.csc.kth.se/utbildning/kth/kurser/DN1212/numpm09/
- Imorgon 13-15 i M1: Föreläsning 1 om programmering
  Läs PEng kap 1-2 (repetition från MJ1102)
- Inför numerikdelarna av Lab 1 och 2
  Läs GNM 1, 4:1A, 4:1D