



Institutionen för
informationsteknologi
Beräkningsvetenskap

Besöksadress:
MIC hus 2, Polacksbacken
Lägerhyddsvägen 2

Postadress:
Box 337
751 05 Uppsala

Telefon:
018-471 0000 (växel)

Telefax:
018-52 30 49

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

Department of
Information Technology
Scientific Computing

Visiting address:
MIC bldg 2, Polacksbacken
Lägerhyddsvägen 2

Postal address:
Box 337
SE-751 05 Uppsala
SWEDEN

Telephone:
+46 18-471 0000 (switch)

Telefax:
+46 18-52 30 49

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

Miniprojekt – Elbil

Del 1, Introduktion

Du är med i ett projekt där man utvecklar olika funktionaliteter till en färddator för elektriska fordon. Färddatorn ska utökas med funktioner för återstående körsträcka utifrån förbrukning längs en körd rutt samt aktuell laddning av fordonets batteri.

I utvecklingssyfte kommer ni att använda data över energiförbrukningen för den eldrivna sportbilen Tesla Roadster och hastighetsdata längs med en viss rutt som ni tagit fram med hjälp av gps-spårare. När den nya funktionaliteten är klar så ska den kunna användas med data för andra bilmodeller. Notera att ni ska skriva egna funktioner och inte använda Matlabs inbyggda funktioner för minsta kvadratanpassning, interpolation, integrering och ekvationslösning. Färddatorn har begränsat utrymme och det skulle bli för dyrt med licenskostnader om generell programvara inkluderades i varje bil.

(a) Konsumtionsdata: Ladda ned filen `roadster.mat` som innehåller data med elkonsumtion för ett antal hastigheter för Tesla Roadster. Filen innehåller två vektorer med namnen `speed_kmph` och `consumption_Whpkm` med hastighet (km/h) och energiförbrukning (Wh/km) samt en textsträng `readme` med information om innehållet. Läs in filen med kommandot `load`. Informations-texten kan skrivas ut med kommandot `fprintf(readme)`. Plotta sedan energiförbrukningen som funktion av hastigheten.

(b) Ruttdata: Dina kollegor Anna och Elsa har kört mellan två städer längs varsin rutt. Med hjälp av gps-spårare har de registrerat position och hastighet. De har sedan bearbetat datat för att få fram hastigheten för ett antal punkter längs med rutten. De har sparat datat för de två rutterna i två filer `speed_anna.mat` och `speed_elsa.mat`. Läs in filerna och plotta hastigheten som en funktion av sträckan för de båda rutterna.

Förberedande uppgift: Filerna ovan ger data i diskreta punkter men vi kommer senare att behöva data för mellanliggande värden. För det kommer vi att använda linjär interpolation och behöver då veta mellan vilka två punkter som datat ligger. Skriv nu en Matlabfunktion som hittar det första elementet i en vektor för vilket gäller att det är större än eller lika med det sökta värdet. Funktionen ska då returnera positionen i vektorn. Då vet vi att värdet ligger mellan denna position i



vektorn och föregående position. Som specialfall får man hantera fallet då värdet är lika med det första värdet i vektorn och då returnera nästa position istället. Funktionen ska ha följande form:

```
function index=findpos(vector, val)
```

Du ska kunna anropa funktionen som

```
>> x=11:20;  
>> index=findpos(x,13.2)
```

och få svaret `index = 4`. Lägg gärna in varningsmeddelanden i funktionen om det sökta värdet ligger utanför vektorn. Du kan anta att vektorns värden ligger i stigande ordning och vi kan då implementera sökningen med hjälp av en while-loop som går igenom vektorns element från början och avbryter när vi har hittat ett värde i vektorn som är större än eller lika med värdet som vi angav. Metoden kallas då för *linjär sökning*. Fundera på om det finns mer effektiva algoritmer än linjär sökning och implementera gärna det (dock inget krav).

Del 2, Interpolation och minstakvadratanpassning

(a) Elkonsumtion: Du ska nu skriva en matlab-funktion som returnerar elkonsumtionen $c(v)$ (Wh/km) för given hastighet v (km/h) och bilmodell. Funktionen ska ha funktionshuvudet

```
function c = consumption(v, carmodel)
```

Man ska kunna anropa funktionen för en godtycklig vektor v av positiva hastigheter (> 0) och få ut förbrukningsdata för bilmodellen `carmodel`. Som bilmodell har vi Tesla Roadster vars förbrukningsdata finns i filen `roadster.mat`. Anropet till funktionen kan då med $v = 30$ km/h se ut som

```
>> c = consumption(30, 'roadster.mat');
```

Alternativt ska vi också kunna anropa funktionen med en vektor av hastighetsvärden, t.ex.:

```
>> v=[96.4 99.1 100.3 102.7 97.0];  
>> c = consumption(v, 'roadster.mat');
```

Då ska funktionen returnera en vektor c med motsvarande förbrukningsdata. Funktionen behöver alltså först ladda in filen `carmodel` för att du ska få tillgång till förbrukningsdata. Du har i `roadster.mat` endast tillgång till data för ett



ändligt antal hastigheter. Du behöver därför approximera energiförbrukningen för hastigheter som inte finns med i filen. Det kan du göra genom att anpassa en matematisk modell till förbrukningsdata. Som approximerande funktion kan du använda $c(x) = a_1x^{-1} + a_2 + a_3x + a_4x^2$ och hitta koefficienterna i funktionen mha minsta kvadratmetoden. Funktionen `consumption` ska alltså först bestämma funktionen som beskriver förbrukningsdata i filen `carmodel` och därefter beräkna energiförbrukningen för de hastigheter som anges i vektorn `v`. Man ska kunna anropa funktionen för en enskild hastighet och för en vektor av olika hastigheter och den ska då returnera motsvarande förbrukningsdata som ett enskilt värde respektive en vektor av värden. (Vektorargumentet behöver vi senare för effektivitet när vi behöver beräkna förbrukningen för ett stort antal hastighetsvärden och då inte vill behöva göra om minsta kvadratanpassningen för varje värde.)

(b) Hastighet: Du ska nu skriva en Matlabfunktion som returnerar en hastighet v (km/h) för en given position x (km) längs en rutt. Funktionen ska ha funktionshuvudet

```
function v = velocity(x, route)
```

där `route` är namnet på `.mat`-filen där hastighetsdata för rутten ligger lagrad. Eftersom hastigheten varierar väldigt mycket och ojämnt kan vi inte approximera datavärdena med en enkel funktion. Det är också svårt att använda styckvisa polynom över flera punkter då hastiga inbromsningar och accelerationer sker längs rутten. Vi kan dock anta att varje punktvärde har tillräcklig noggrannhet för våra syften. Det räcker då med att du använder linjär interpolation mellan punktvärdena. Med hjälp av din funktion `findpos` kan du nu hitta de två punkter som ligger närmast x och därefter implementera antingen Lagrange interpolationspolynom eller Newtons interpolationspolynom och beräkna det sökta värdet. Man ska kunna anropa funktionen för ett godtyckligt värde inom den givna rутten. Anrop till funktionen med negativa värden eller värden bortom rутtens längd bör resultera i felmeddelande via `error`. Implementera funktionen så att den fungerar för så väl skalära värdena som för vektorvärden på x .

Del 3, Integration

(a) Ankomsttid: Du ska nu använda funktionen `velocity` från Del 2(b) för att uppskatta hur lång tid det tar att färdas en viss sträcka längs en viss rutt. Tiden $T(x)$ (h) det tar att färdas x km längs med rутten ges av

$$T(x) = \int_0^x \frac{1}{v(s)} ds. \quad (1)$$

Du kan anta att $v(s)$ är positiv. Formulera trapetsmetoden för att uppskatta integralen och skriv en matlab-funktion som returnerar tid $T(x)$ (h) för given tillryggalagd sträcka x (km). Funktionen skall ha funktionshuvudet

```
function T = time_to_destination(x, route, n)
```



där `route` är som tidigare och n är antalet delintervall i trapetsmetoden. Notera att du skall göra en egen implementation av trapetsmetoden, utan anrop till matlabs `integral`. (Men du kan använda `tex integral` för att verifiera dina resultat.)

Hur lång tid tar det för Anna och Elsa att färdas hela sträckan längs sina respektive rutter? Hur ska man resonera när man väljer antalet delintervall? Undersök t.ex. hur många delintervall n som krävs för att få korrekt antal minuter i den numeriska integrationen om vi bara betraktar diskretiseringsfelet.

(b) Total elkonsumtion: Du ska nu använda dina funktioner `velocity` och `consumption` från Del 2 för att uppskatta elkonsumtionen för en given sträcka längs en viss rutt. Den totala elkonsumtionen $E(x)$ i wattimmar efter en tillryggalagd sträcka på x km ges av

$$E(x) = \int_0^x c(v(s)) ds \quad (2)$$

där $c(v)$ är elkonsumtion per km (Wh/km) som funktion av hastighet v (km/h) och $v(s)$ är förväntad hastighet längs med ruten. Formulera nu Simpsons metod för att uppskatta integralen och skriv en matlab-funktion som returnerar elkonsumtion $E(x)$ (Wh) för given tillryggalagd sträcka x (km). Funktionen skall ha funktionshuvudet

```
function E = tot_consumption(x, carmodel, route, n)
```

Hur mycket energi gör Anna och Elsa totalt av med längs med sina respektive rutter? Hur ska man välja antalet delintervall n i Simpson?

(c) Återstående räckvidd: Nu vill vi veta hur långt vi kommer med kvarvarande energi i batteriet. Skriv en funktion som utifrån laddningsmängd vid start, förbrukad total energi samt beräknad medelförbrukning per kilometer returnerar kvarstående räckvidd i km. Funktionen ska ha följande funktionshuvud:

```
function r = range(x, carmodel, route, charge, n)
```

där `charge` är batteriladdningen vid start och de övriga inparameterna som ovan. Funktionen ska först beräkna totala energiförbrukningen och medelförbrukning av den tillryggalagda sträckan x . Genom att anta att medelförbrukningen sedan förblir densamma ska funktionen beräkna den återstående sträckan r som batteriladdningen räcker till.



Del 4, Ekvationslösning

Räckvidd längs given rutt: En intressant information när man ska köra med sin elbil är räckvidden givet en destination, dvs när man måste stanna för att ladda batteriet. Det är en funktionalitet som kräver att vi i förväg vet vilken rutt man tar och hur fort man kör längs rutten. Denna information skulle kunna hämtas från bilens GPS-system som beräknar närmaste vägen och där aktuella hastighetsbegränsningar finns tillgängliga för alla vägar och vägavsnitt. I nuläget kan vi för enkelhets skull använda färddata för de givna rutterna. Frågeställningen blir då följande. Om man antar att batteriet är laddat till C Wh, hur länge räcker den laddningen längs en viss rutt? Dvs, beräkna vid vilken position x som $E(x) = C$. Fundera på hur man skulle kunna lösa detta, och skriv sedan en funktion som löser problemet. För den här deluppgiften kan du använda dig av Matlabs inbyggda lösare `fzero` men du uppmuntras till att skriva en egen lösare.

Diskussion

Diskutera felkällor, vilket innefattar numeriska fel i metoder, mätfel i data och modellfel pga förenklingar och brister i den fysikaliska modellen. Diskutera också hur man kan uppskatta och beräkna de olika felen samt vilka förbättringar av metoder, modeller och antaganden som skulle kunna göras. Du behöver inte beräkna felen numeriskt, det räcker med att du resonerar kring detta då det kan vara väldigt svårt att uppskatta storleken på vissa fel. Det viktiga är att du reflekterar över användbarheten och ger problemet en verklighetsanknytning.

Rapport

Redovisningen av hela elbilsprojektet (del 1 – 4) ska ha formen av en rapport. Rapporten kan skrivas på svenska eller engelska och ni ska använda ett textbehandlingsystem, t.ex. Latex eller Word. Rapporten ska ta upp följande punkter:

Rapportmall:

1. *Problembeskrivning*, en presentation av uppgiften med en introduktion där man sätter in problemet i sammanhang och ger en motivering till varför man vill lösa uppgiften. Utgå ifrån att läsaren inte har tillgång till uppgiftsbeskrivningen och beskriv problemställningen med egna ord.
2. *Lösningsmetoder*, en beskrivning av de numeriska lösningsmetoderna med teori och analys samt deras implementation.
3. *Resultat och problemlösning*, en presentation av resultaten med relevanta plottar och tabeller samt kommentarer och observationer gällande resultaten. Projektet består av flera deluppgifter och det är lämpligt att låta avsnittet bestå av ett delavsnitt per delproblem.
4. *Diskussion*, med eventuella förklaringar till observationer samt en redogörelse av felkällor, begränsningar och brister. Ni kan också diskutera eventuella förbättringar och utökningar av modell och metoder.



UPPSALA
UNIVERSITET

5. *Sammanfattning*, med de viktigaste resultaten och slutsatserna.
6. *Referenser*, här förtecknas de källor som ni refererar till i rapporten. Det kan vara litteraturhänvisningar, websidor, mm.
7. *Appendix*, här redovisas era Matlab-program. Annat material som ni vill redovisa och som ni inte tycker passar i den löpande texten, kan också placeras här.