UPPSALA UNIVERSITET

Institutionen för informationsteknologi
Beräkningsvetenskap
Emanuel H. Rubensson
Universitetslektor

Besöksadress:
MIC hus 2, Polacksbacken
Lägerhyddvägen 2

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
Emanuel H. Rubensson
Associate Professor

Visiting address:
MIC bldg 2, Polacksbacken
Lägerhyddvägen 2

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

Beräkningsvetenskap I

Projekt: Räckvidd för elbil

Du utvecklar en kartapp och färddator för elektriska fordon. Appen ska utökas med funktioner för återstående körsträcka längs en given rutt för aktuell laddning av fordonets batteri.

Bilars energiförbrukning beror till stor del på hastigheten. Räckvidden längs med olika rutter kan därför variera kraftigt. I det här projektet ska ni ta fram en uppskattning av återstående körsträcka (räckvidd) givet information om energiförbrukning vid olika hastigheter samt förväntad hastighet längs med given rutt.

För 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 och förväntad hastighet längs med given rutt beräknad utifrån gällande hastighetsbestämmelser, historiska hastighetsdata, och/eller aktuell trafikinformation.

Tesla Roadster 2.5 at AutoRAI Amsterdam 2011 by Overlaet / Wikimedia Commons / CC-BY-SA-3.0

Del 1, approximation

(a) *Elkonsumtion*: Din första uppgift är att skriva en matlab-funktion som returnerar elkonsumtionen för given hastighet.

Ladda ned filen `roadster.mat` som innehåller data med elkonsumtion för ett antal hastigheter för Tesla Roadster. Du kan se innehållet i filen med kommandot `whos('-file','roadster.mat')`. 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`. Informationstexten kan skrivas ut med kommandot `fprintf(readme)`. Plotta sedan energiförbrukning som funktion av hastighet
`plot(speed_kmph, consumption_Whpkm, 'x')`.

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

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

Man ska kunna anropa funktionen för godtycklig hastighet inom ett rimligt spann av hastigheter, tex upp till 200 km/h. Du har 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 med hjälp av interpolation eller någon annan form av kurvanpassning. Välj någon interpolationsmetod från kursen och implementera den i din funktion `consumption`. Du kan använda matlabs funktionalitet för interpolation, se tips i slutet av detta dokument.

Plotta konsumtionen som funktion av hastighet (i samma figur som ovan) och bekräfta att din interpolerande funktion antar rimliga värden. Annars kan du behöva justera interpolationen eller byta metod. Räkna också ut förväntad räckvidd (i km) för fixerad hastighet givet fulladdat batteri med en kapacitet på 55 000 Wh. Använd kommandot `subplot` för att under grafen för elkonsumtion lägga in en plot över räckvidd som funktion av (fixerad) hastighet. Vid vilken hastighet är räckvidden som längst?

(b) *Hastighet*: Din andra uppgift är att skriva en matlab-funktion som returnerar hastighet givet tillryggalagd sträcka längs en given rutt.

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`. Ladda ned filerna och utforska innehållet på samma sätt som för `roadster.mat`.

Du ska nu skriva en matlab-funktion som returnerar hastighet $v(x)$ (km/h) för given tillryggalagd sträcka x (km). Funktionen ska ha funktionshuvudet

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

där `route` är namnet på `.mat`-filen där hastighetsdata för rutten ligger lagrad så att funktionen kan anropas tex `v=velocity(x,'speed_anna')`. Precis som för elkonsumtionen så ska man kunna anropa funktionen för godtyckligt värde inom ett rimligt spann, i detta fall godtycklig tillryggalagd sträcka kortare än ruttens totala längd. Demonstrera din approximation med lämpliga plottar.

Del 2, integration

(a) *Ankomsttid*: Du ska nu använda funktionen `velocity` från del 1 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 rutten ges av

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

Du kan anta att $v(s)$ är positiv. Formulera någon av integrationsmetoderna från kursen 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 integrationsmetoden. Notera att du skall göra en egen implemention av integrationsmetoden du valt, utan anrop tex 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? Undersök hur många delintervall n som krävs för att få korrekt antal minuter i den numeriska integrationen.

(b) *Total elkonsumtion*: Du ska nu använda båda funktionerna `velocity` och `consumption` från del 1 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

$$E(x) = \int_0^x c(v(s))ds \tag{2}$$

där $c(v)$ är elkonsumtion per km (Wh/km) som funktion av hastighet v (km/h) och $v(s)$ är förväntad hastighet längs med rutten. Formulera som ovan någon av integrationsmetoderna från kursen 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 = total_consumption(x, route, n)
```

Hur mycket energi gör Anna och Elsa totalt av med längs med sina respektive rutter? Undersök hur många delintervall n som krävs för att få korrekt antal wattimmar i den numeriska integrationen.

(c) *Konvergensstudie*: För åtminstone en av de två integralerna (1) och (2) skall du nu göra en empirisk undersökning av noggrannhetsordningen för din metod.

Räkna ut integrationsfelet för en serie beräkningar där antalet delintervall dubbleras successivt, dvs för en serie $n, 2n, 4n, 8n, 16n, \ldots$. Om du inte känner till det exakta integralvärdet kan du jämföra med integralvärdet med halverad steglängd. Alltså, för beräkningen med n delintervall jämför du med värdet med $2n$ delintervall, etc.

Plotta integrationsfelet som funktion av antalet delintervall n eller steglängd $h = (x - 0)/n$. Här är det vanligtvis lämpligt att plotta i log-skala, det kan du göra genom att istället för `plot` använda kommandot `loglog` som ger log-skala både på x- och y-axeln. Plotta också hjälplinjer som visar vilken lutning på kurvan som svarar mot olika noggrannhetsordningar. Tex, om du använder en andra ordningens metod plottar du en hjälplinje som visar

$O(1/n^2)$- eller $O(h^2)$-beteende. Hitta en linje som stämmer överens med dina resultat.

Stämmer den empiriskt bestämda noggrannhetsordningen med teorin för den integrationsmetod du använt? Om inte, varför?

Del 3, hitta roten

(a) *Sträcka*: Antag nu att föraren vill veta hur långt längs rutten hon förväntas ha kommit på en viss tid. Utgå från ekvationen för ankomsttid och formulera den icke-linjära ekvation vars rot ger förväntad tillryggalagd sträcka efter T timmar. Formulera sedan Newtons metod för att lösa ekvationen. Fundera på vad som skulle vara en bra startgissning för att hitta roten. Skriv en matlab-funktion där du implementerar Newtons metod inklusive startgissning. Funktionen skall ha funktionshuvudet

```
function x = distance(T, route)
```

Hur långt kommer Anna och Elsa längs sina respektive rutter på 30 min? *Tips*: Använd funktionerna `time_to_destination` och `velocity`.

(b) *Räckvidd*: Antag nu att batteriet är laddat till C Wh. Formulera den icke-linjära ekvation vars rot ger bilens räckvidd. Formulera sedan Newtons metod för att lösa ekvationen. Fundera på vad som skulle vara en bra startgissning för att hitta roten. Skriv en matlab-funktion där du implementerar Newtons metod inklusive startgissning. Funktionen skall ha funktionshuvudet

```
function x = reach(C, route)
```

Hur långt skulle Anna och Elsa komma på sina respektive rutter med en batteriladdning på $C = 10000$ Wh?

Tips: Använd funktionerna `total_consumption`, `velocity` och `consumption`.

Tips och referens

Interpolation i matlab: Kort lathund för interpolation i matlab. För mer information använd `help` eller `doc`, tex `help spline` eller `doc spline`.

`c = polyfit(x,y,n)`**:** Ger vektor med koefficienter $[c_1, c_2, \ldots, c_{n+1}]$ för ett polynom

$$p(x) = c_1 x^n + c_2 x^{n-1} + \cdots + c_n x + c_{n+1} \tag{3}$$

av grad `n` som bäst approximerar datat `y` i punkterna `x` i minsta kvadratmening.

`yy = polyval(c,xx)`**:** Returnerar värdet på polynomet `c` i punkterna `xx`.

`pp = spline(x,y)`**:** Ger det styckvisa polynom som är resultatet av kubisk splineinterpolation med datavärden `y` i punkterna `x`.

`yy = ppval(pp,xx)`**:** Returnerar värdet på det styckvisa polynomet `pp` i punkterna `xx`.

`yy = spline(x,y,xx)`**:** Kort för `yy = ppval(spline(x,y),xx)`. Dvs evaluera, i punkterna `xx`, spline som interpolerar punkterna x, y.

`yy = interp1(x,y,xx,METHOD)`**:** Generell funktion för interpolation i 1 dimension. Interpolerar punkterna givna av vektorerna `x`, `y` med interpolationsmetod `METHOD` och returnerar värdet på den interpolerande funktionen i punkterna `xx`. Not.: anropet `yy = interp1(x,y,xx, 'spline')` är ekvivalent med `yy = spline(x,y,xx)`.

`yy = interp1(x,y,xx,METHOD,NaN)`**:** Extra argument `NaN` användbart för att upptäcka anrop med oväntad input. Värden utanför intervallet som spänns upp av `x` ersätts med `NaN`.

Noggrannhetsordning: Låt $\tilde{u}_h$ vara en numerisk approximation av ett exakt värde u, där h är steglängd eller liknande i en numerisk metod. Den numeriska metoden har noggrannhetsordning p om det finns ett tal C oberoende av h så att

$$|\tilde{u}_h - u| \leq Ch^p, \tag{4}$$

åtminstone för tillräckligt små h.

Analysens fundamentalsats: Antag att funktionen f är kontinuerlig på intervallet $[a, b]$ och låt

$$F(t) = \int_a^t f(x)dx, \quad a \leq x \leq b \tag{5}$$

Då gäller att $F'(x) = f(x),\ x \in (a, b)$.