TMV156/TMV155E Inledande matematik E, 2009

MATLAB — INLÄMNINGSUPPGIFT

Instruktioner

Den här uppgiften är obligatorisk. Den ska lämnas i lådan utanför Fredrik Lindgrens kontor nummer MV:L2104 i Matematiska Vetenskaper senast kockan 16.00 tisdagen den 3:e november 2009. Det är alltså tisdag i läsvecka 2 i läsperiod 2.

Det är okej att samarbeta, men du ska lämna in en egen rapport och du ska skriva kommentarerna själv (se nedan). Om några rapporter ser för lika ut så kan rapportförfattarna kallas in på (läx-) förhör.

Samtliga upgifter ska skrivas i en och samma fil utom funktionsfilen `derivera.m` som måste skrivas i en egen fil för att det ska fungera. Dock ska innehållet i filen klistras in och bortkommenteras precis under den `for`-loop från vilken den anropas. En rapport ska genereras med Matlabs funktion `publish`. Skriver man `publish('filnamn')` i kommandofönstret så genereras ett `html`-dokument med all text och kod i filen `filnmamn.m` samt figurer som den ritar och resultat som den skriver ut i kommandofönstret. Vi får, om vi gör rätt, också en innehållsförteckning. Jag hoppas att följande programsnutt är självförklarande. Klistra in det i en `m`-fil och kör den med `publish` så listar du nog ut hur det fungerar!

```
%% Dokumentets titel
% Om man börjar med två procenttecken följt av mellanslag och Dokumentets
% Titel så genererar det dokumentets titel.
%% Uppgift 1 Att skriva lite text och göra några beräkningar
% Om man har två procenttecken igen så genereras en rubrik en rubrik, har
% man bara ett blir det löpande text på detta vis.

%Om vi sätter ; efter en tilldelning så skrivs inte värdet ut, vare sig i
%kommandofönstret eller i den fil som Matlabs publish genererar

f=@(x)(x-2).^-1; %Skapar funktionshandtag och den anonyma funktionen 1/(x-2).
a=3;    %Vi deklarerar variabeln a och ger den värdet 3

%Om vi faktiskt vill veta svaret så låter vi Matlab skriva ut det:

y1=f(a) %Räknar ut 1/(a-2) för a=3 och tilldelar variabeln y3 detta värde.
%% Uppgift 2 Rita grafen till funktionen f(x)=(1)/(x-2)
% Nedan ritar vi grafen vi till y=1/(x-2) på intervallet [-1,4].
close all
x=linspace(-1,4); %Skapar punkter i definitionsmängden [-1,4]
y=f(x); %Skapar funktionsvärdena för alla punkter i x
plot(x,y,'^:r') %Ritar grafen y=1/(x-2)
```

Kommandot `publish('filnamn','doc')` genererar en `doc`-fil med samma innehåll om man får tro publish-filens hjälptext. Det går också att generera `Latex`-kod med mera. Hur som helst så ska dokumentet skrivas ut och lämnas in i pappersformat i enlighet med instruktionerna ovan.

All programkod ska noggrannt *kommenteras*, dvs det ska skrivas en förklaring i anslutning till varje kommando i koden. Så här skulle det kunna se ut:

```
x=linspace(-5,5,1000); %Genererar 1000 tal jämt fördelade mellan -5 och 5
fkn=@(z)(z.^2); %Ett funktionshandtag som pekar på den anonyma funktionen z.^2
```

Date: 2009–10–09, Fredrik Lindgren, Matematiska vetenskaper, Chalmers tekniska högskola.

```
y=fkn(x); %Räknar ut kvadraten på varje tal i vektorn x och sparar dem
          % i vektorn y.
plot(x,y) %Ritar grafen av funktionen y=x^2 på intervallet [-5,5].
```

1. Uppgifter

I alla uppgifter nedan ska du arbeta med funktionen

$$R(x) = \frac{x^3 - 6x^2 + 9x - 3}{(x-1)^3}$$

. Notera att om $p(x) = x^3 - 6x^2 + 9x - 3$ och $q(x) = (x-1)^3$ så är

$$R(x) = \frac{p(x)}{q(x)}$$

.

1.1. Rita grafen.

1.1. Rita grafen. Den första delen består av att rita R:s graf på ett sätt så att *alla intressanta aspekter framgår*. Välj skalor så att det verkligen är fallet.

Grafen och dess asymptoter: Först ska du rita grafen av funktionen R samt eventuella asymptoter. Asymptoterna ska ritas med en annan linje*typ*. Du vill säkert inte slösa utskriftskvot på färgutskrifter så figuren ska vara gjord på ett sådant sätt att all väsentlig information framgår i svartvitt format.

Kritiska punkter, inflektionspunkter och nollställen: Funktionen R har två kritiska punkter. Hitta dessa med `fminbnd` och markera dem med romber (diamanter) i figuren från förra uppgiften. Finn också inflektionspunkter (analytiskt) och markera dessa med cirklar. För tydlighetens skull så ska även rötterna markeras med kryss. Om $p(x)$ och $q(x)$ inte har några gemensamma rötter så är rötterna till $R(x)$ desamma som rötterna till $p(x)$. Hitta de senare med `roots`.

Beskrivande text: Slutligen bör figuren kompletteras med namn på båda axlarna samt en titel på figuren och förklaring av graferna (funktionsgrafen och asymptoterna) med hjälp av `legend`.

1.2. Numerisk derivering. Vi vet att derivatan av en funktion ges av gränsvärdet

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{1}$$

om gränsvärdet existerar. Det föranleder oss att för ett litet h approximera derivatan med *differenskvoten*

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}. \tag{2}$$

Ju mindre h desto bättre torde approximationen bli. Det skulle vara sant om vi hade tillgång till hur många decimaler som helst. Det har vi nu inte i Matlab så när h blir tillräckligt litet så kommer avrundningsfelet bli större och större. Hur stort h bör vara beror på funktionen som ska deriveras och är något man kan tänkas vilja bestämma som användare. Det gäller att den *symmetriska differenskvoten* (3) ger en bättre approximation än den i Ekvation (2).

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h} \tag{3}$$

Funktionsfil med numerisk derivering: Din uppgift är att skriva en funktion `derivera.m` som beräknar den symmetriska differenskvoten av en matematisk funktion f i en punkt x. Funktionen ska ha *tre inargument*: Ett funktionshandtag f, ett tal x som är punkten i vilken derivatan ska approximeras samt steglängden h. Funktionen ska ha *ett utargument*: Det approximativa värdet av derivatan i punkten x.

Hjälptext: Du ska dessutom skriva en hjälptext i början av filen (innan function ...=derivera...). Där ska det tydligt framgå vad användaren behöver ge för typ av inargument och vilka utargument han kan tänkas få tillbaka. Följ gärna exemplet från programmet `bisekt.m` nedan:

```
% bisect - bisection algorithm for the scalar equation f(x) = 0
%
%   Syntax:
%           x = bisect(f, int, tol)
%   Arguments:
%           f - function handle pointing to a function file
%           int - 1x2 matrix specifying an interval int = [a,b]
%           tol - a tolerance
%   Returns:
%           x - an approximate solution in the interval int = [a,b]
%   Description:
%           The program bisect uses the bisection algorithm to
%           compute an approximate solution of the scalar equation
%           f(x) = 0 in the interval int = [a,b].  The function file must
%           contain the function y = f(x).  The function values f(a)
%           and f(b) must have opposite signs.  Then the program
%           computes an approximate solution x with the error
%           |x-x_exact| < tol.  It returns an empty value for x if
%           f(a) and f(b) have the same sign.
%   Examples:
%          x = bisect(@sin,[1,4],1e-7)   computes pi to 6 decimals
%
%-----------------------------------------------------------------------

function x = bisect(f, int, tol)
%Här börjar programkoden...
```

Texten i hjälpfilen ska alltså skrivas ut i kommandofönstret om du skriver

```
help derivera
```

vid Matlab-prompten (alltså vid tecknet >> i kommandofönstret, där man skriver kommandon om man inte gör det i en m-fil). Det går bra att skriva på svenska.

Fördefinierad steglängd: Om funktionsvärdet $f(x)$ är i storleksordningen 1 så kan man visa att $h = 10^{-5}$ är ett bra val av steglängd för den symmetriska differenskvoten. Den person som vill använda din fil kanske inte vill/orkar fundera över steglängden.

Lägg till följande kod i början av din fil så behöver inte användaren mata in något h-värde om hon inte vill. För att detta ska fungera måste steglängden i din funktionsfil ha namnet h.

```
% if-else-satsen nedan testar om användaren har matat in något
%h-värde. Om det inte är fallet så sätter den h till 10^-5.
%Den testar också om användaren matat in för få eller för många
%inargument. Då returnerar den ett felmeddelande och avbryter
%programmet.
%NARGIN (Number of ARGuments IN) är en funktion som returnerar
%information om hur många variabler användaren anropade funktionen med.
argument=nargin;
if argument==2
   h=1e-5;
elseif argument==3
   %Om det är tre inargument så gör vi ingenting alls utom att
   %beräkna derivatan.
else
   error('Felaktigt antal inargument!')
end
```

Beräkna den numeriska derivatan av R: Din sista uppgift blir att på det intervallet du plottade funktionen R beräkna den approximativa derivatan med hjälp av funktionsfilen `derivera.m`. Generara en vektor med tal i definitionsmängden $[a, b]$:

```
x=linspace(a,b,1000);
```

Skapa sedan en vektor där du ska spara derivatorna, vi ger alla element värdet noll till att börja med:

```
y_prim=zeros(size(x));
```

Nu ska du i en for-loop anropa din egenhändigt skrivna deriveringsfil med ett x-värde i taget, $x(n)$ och spara värdet på derivatan på motsvarande plats i vektorn y, $y(n) = ...$ Rita sedan grafen av den approximativa derivatan och den exakta i en ny figur. Ge gärna någon kort kommentar om hur graferna ser ut och säg något om skillnaden!

LYCKA TILL! /FREDRIK