Teknillinen korkeakoulu Apiola
Matematiikka

Mat-1.192 Numeerinen ja symbolinen laskenta kevät 2004
`http://www.math.hut.fi/teaching/numsym/04/H/`

Laskuharjoitus 4 (viikko 9– , helmi–maalis 2004)

Jaossa: [CvL]-sivut ODE (CH 9)
Muista `http://www.math.hut.fi/teaching/numsym/matlab/moler/odes.pdf`
`http://www.math.hut.fi/teaching/numsym/matlab/CV/CHAPTER.9`
`http://www.math.hut.fi/teaching/L/ODE.html` ODE-”portaali”

Sekä [CvL]:ssä että Moler:ssa on laaja tehtävävalikoima. Erityisesti Molerista on mahdollisuus valita myös laajempia oppilaskohtaisia tehtäviä, joissa pääsee mukavasti käyttämään eri alueiden laskentavälineitä yhteistyössä.

Olkoot lyhyesti C ja M nuo viitteet alla olevien tehtävänumeroiden kohdalla.

Tässä kaikille tarkoitettuja tehtäviä. Kokoelmaa täydennetään niin ODE- kuin muidenkin alueiden osalta. Seminaariesitysten täydennyksenä (silloin kun ylimääräistä aikaa jää) voidaan pitää neuvontaharjoituksia, sovitaan myöhemmin, milloin harjoitustehtävien suorituksia käydään läpi.

1. M 7.1 (s. 34).
   ”Experimentally-osaan voit näppärästi käyttää Matlabin ode23:a tai ode23tx:ää.
   ”Algebraically-osaa ajatellen kirjoitin tiedoston loppuun `.../maple/ns04.mpl` pikku koodin ”, jonka laitan myös tehtävien perään. (Käytä hyväksesi, jos siltä tuntuu.)
2. M 7.2
   Tutki lisäksi algoritmin `ode23(tx)` askelpituuden valintastrategiaa samaan tyyliin kuin [CvL]:ssä

   s. 345 fig. 9.8 ja s. 347 fig. 9.11. Kokeile muutamalla eri AbsTol ja RelTol valinnalla.

   Huomaa, kuinka kätevä tässä on funktio `diff`.
3. Kummassakin lähteessä puhutaan 2:n kappaleen probleemasta. Kirjoita yleinen kahden kappaleen probleemaan liittyvä diffyhtälö. Massat m_1 ja m_2 olkoot hetkellä t pisteissä, joiden paikkavektorit ovat $\mathbf{r_1}$ ja $\mathbf{r_2}$.

   Kirjoita tätä yleistä muotoa vastaava m-tiedosto.

   Johda sitten muoto, joka esiintyy kummassakin viitteessä (”twobody”, ”Kepler”) (ja taitaa olla Matlabin demossakin). Tässä toinen kappale ajatellaan paljon massiivisemmaksi, ja se otetaan koordinaatiston origoksi.

   Käy huolellisesti läpi [CvL] ss. 343 – 348 Matlab-laskut ja piirrokset. Tee joitakin omia kokeiluja toleranssien ja menetelmien suhteen.

   Käytä alussa olevaa yleistä muotoa kahden samanmaassaisen (tms.) kappaleen liikeratojen tutkimiseen sopivilla alkuehdoilla. Tapahtuuko helposti massojen törmäyksiä? Ihan mukava olisi tehdä animaatio, mutta olkoon se ”vapaaehtoista”.

   (Tässä pikku johdatus aiheeseen ”three body problem”, josta voisi kehitellä aika laajankin esityksen, sopivia referenssejä on tarjolla.)
4. Ratkaise diffyhtälö $y' = \frac{1}{y^2} - yt, \quad y(1) = 1$

   sekä Taylorin 4:n asteen menetelmällä että Runge-Kutta 4:llä. Vertaa numeerisia ratkaisuja keskenään välillä $[1, 2]$ ja käytä askelpituuksia $h = 2^{-k}, k = 2, \ldots, 7$.

   Käytä Maplea hyväksesi. Voit samalla miettiä, miten implementoisit yleisen 4:nnen kertaluvun Taylorin menetelmän tai yleisemmin, yleisen n:nnen kertaluvun Taylorin menetelmän Maplella.

   Huomaa, että koko työ menee hyvin Maplessa, meillä on valmis rk4 tiedostossa `../maple/ns04.mpl`. Toisaalta rk4:n voi tehdä hyvin Matlabilla, vaikka omalla funktiolla tai CvL-funktiolla FixedRK.m
   `http://www.math.hut.fi/teaching/numsym/04/matlab/CV/CHAPTER.9/`

Tehtävän 1 Maple-apu

```
ode23step:=proc(f,t,y,h)
local s1,s2,s3,s4,tuusi,yuusi,euusi;
s1:=f(t,y);
s2:=f(t+h/2,y+h/2*s1);
s3:=f(t+3*h/4,y+3*h/4*s2);
tuusi:=t+h;
yuusi:=y+h*(2*s1+3*s2+4*s3)/9;
s4:=f(tuusi,yuusi);
euusi:=h/72*(-5*s1+6*s2+8*s3-9*s4);
[tuusi,yuusi,euusi];
end:
```

Tehtävä 3

Muistele Newtonin gravitaatiolakia: Voima on suoraan verrannollinen massojen tuloon ja kääntäen verrannollinen etäisyyden neliöön. Voit valita yksiköt niin, että gravitaatiovakio = 1.

Kirjoita kumpaakin massaa koskeva liikeyhtälö, voit rajoittua tasotapaukseen. Tällöin saat 8:n yhtälön systeemin.

Funktio voisi alkaa vaikka näin:

```
function Yp=kakskpl(t,Y)
% Kahden kappaleen ongelman diffyhtsys. Gravitaatiovakio = 1
global m1 m2
R1=Y(1:2);
V1=Y(3:4);
R2=Y(5:6);
V2=Y(7:8);
r=norm(R2-R1);
...
```

Teht. 4

Tässä eräs tapa, jolla voisi edetä. Tehdään ensin kaikkein yksinkertaisin ei-triviaali (Ei-Euler) tapaus: 2. kertaluvun Taylorin menetelmän askel.

```
taystep2:=proc(f,t0,y0,h)
local y1,y2,t,tay2;
y1:=f(t,y(t));
y2:=diff(y1,t);
y2:=subs(diff(y(t),t)=y1,y2);
tay2:=y0+h*y1+h^2/2*y2;
subs([t=t0,y(t)=y0],tay2);
end:
```

Testausta:

```
# Esim:
 f:=(t,y)->y*t;
 dsolve({diff(y(t),t)=t*y(t),y(0)=1},y(t));
 ytarkka:=subs(%,y(t));
 Y[0]:=1: h:=0.1: T:=seq(k*h,k=0..10);
 for k to 10 do
   Y[k]:=taystep2(f,T[k],Y[k-1],h):
  end do:T[0]:=0;
 pisteet:=seq([T[k],Y[k-1]],k=1..10);
 plot([[pisteet],ytarkka],t=0..1);
```

Jatko sujuu samaan tapaan. Helppo on kirjoittaa yleinen n:nnen asteen Taylor.

Ongelmaksi tulee laskettaessa suuriksi paisuvat lausekkeet (ongelmasta riippuen), jolloin aritmetiikkaa tulee paljon. Tätä on kyllä mahdollista vähentää mm. seuraavilla välineillä.
`horner` (jos päädytään korkea-asteisen polynoin laskentaan).
`codegen[optimize]`, joka muodostaa sopivia apumuuttjia, joiden avulla vältytään laskemaan samoja lausekkeita monta kertaa.
Olkoon laskettu h^4:n kerroin c_4, jonka lausekkeesta tulee pitkä ja korkea-asteinen polynomi:

```
c4:=coeff(yuusi,h,4);
with(codegen):
horner(numer(c4),y0); # Jos osoittaja on korkea-asteinen polynomi.
opti:=optimize(%);
nimet:=map(lhs,[opti]);
arvot:=map(rhs,[opti]);
zip((a,b)->a=b,nimet,arvot); # Muodostetaan yhtälöt.
assign(%);                   # = - merkit := - merkeiksi.
```

(Näissä harjoituksissa kyllä riittää tehdä ilman optimointia,)