Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Information Technology
FH Zentralschweiz

ODEs - Exercise II

Prof. Dr. Josef F. Bürgler

I.BA_IMATH, Semesterweek 13

The solution of the exercises should be presented in a clear and concise manner. Numerical results should be accurate to 4 digits. The exercises are accepted if You solve 75% of the exercises adequately.

Problem 1: Reducing the Order

If $x(t)$ represents the height of a free falling body at time t, then the (normalized) equation of the motion can be described by the second order ODE $\ddot{x} = -1 + (\dot{x})^2$.

Write this ODE as a system of first order ODEs using the initial conditions (IC) $x(0) = 1$ und $\dot{x}(0) = 0$.

Solution: Use the new variables y_1 and y_2 defined by

$$\begin{aligned} y_1(t) &= x(t) \\ y_2(t) &= \dot{x}(t) \end{aligned}$$

Then we find the following system of first order ODEs

$$\begin{aligned} \dot{y}_1(t) &= y_2(t) \\ \dot{y}_2(t) &= -1 + y_2(t)^2 \end{aligned}$$

where the initial conditions are $\mathbf{y}(0) = (y_1(0), y_2(0))^T = \mathbf{y}_0 = (1, 0)^T$

Problem 2: Numerical Solution of a System of first order ODEs

Solve the previously derived system of first order ODEs in the interval $[0, 2]$. Estimate the time t when the body hits the ground ($x(t) = 0$).

Solution: The following Matlab/Octave-commands compute the numerical solution and plot the result (position and velocity) dependent on the time t.

```
>> pkg load odepkg
>> freefall = @(t,y) [y(2); -1+y(2)^2];
>> y0 = [1; 0];
>> global h0 = 0;
```

```
>> opts = odeset("Events", @gstop, "RelTol", 1e-6, "AbsTol", 1e-6);
>> [t,y,te,ye,ie] = ode45(freefall, [0 2], y0, opts);
>> plot(t,y(:,1),'g',t,y(:,2),'r')
```

where the file `gstop.m` contains the following lines:

```
function [val,isterm,dir] = gstop(t,y,y0)
  global h0;
  val = y(1,1)-h0;
  isterm = 1;
  dir = 0;
endfunction
```

The plot-command can be replaced by `plot(t,y)` in which case the y-component is drawn versus time t. Figure 1 shows the position Ort $x(t)$ (upper, blue curve) und the velocity $\dot{x}(t)$ (lower, green curve) of the freely falling body if air drag is taken into account.

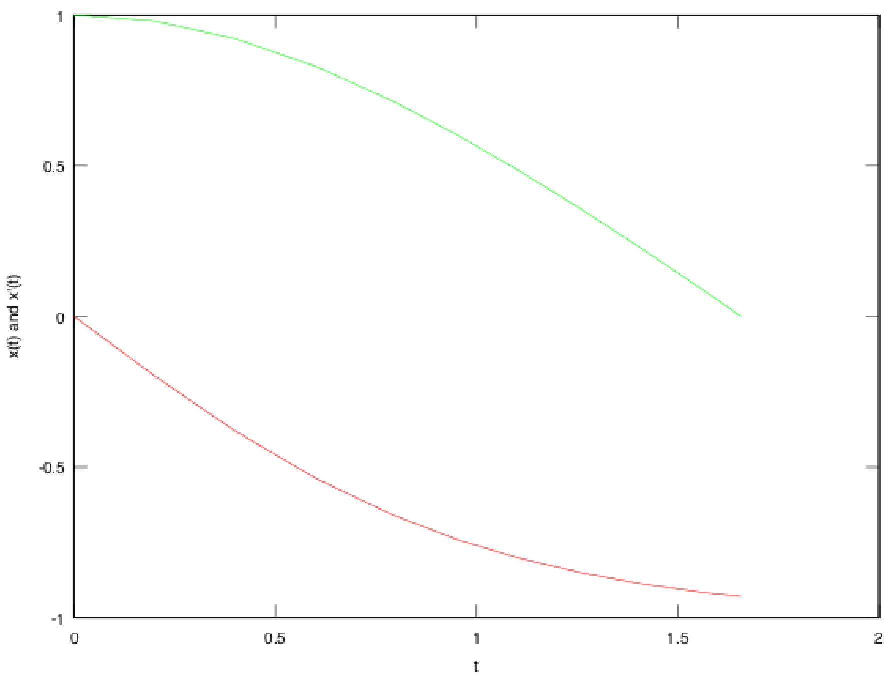

Abbildung 1: Position $x(t)$ and velocity $\dot{x}(t)$ versus time t.

Using `te` (1.6570) and `ye` (0.0000, -0.92906) we know that the ground is hit at $t = 1.6570$ and the velocity is then -0.9906.

Problem 3: Numerical Solution of the predator-prey problem

The following modified Lotke-Volterra predator-prey model prevents the unlimited growth of rabbits.

$$\dot{r} = 2\left(1 - \frac{r}{R}\right) r - \alpha r f, \qquad \text{IC} \quad r(0) = r_0$$
$$\dot{f} = -f + \alpha r f, \qquad \text{IC} \quad f(0) = f_0$$

Solve the system for $R = 400$, $r_0 = 300$, $f_0 = 150$, and $\alpha = 0.01$ in the interval $[0, 50]$. Use the Matlab/Octave procedure `ode45` (Runge-Kutta 4 or 5). Produce the following four figures

1. Number of foxes ($f(t)$) and rabbits ($r(t)$) versus time for the old model, i.e. without the factor r/R
2. same for the new model.
3. Number of foxes versus the number of rabbits in the old model.
4. same for the new model.

Solution: Using the following Matlab/Octave-commands

```
>> global R;
>> R=400;
>> r0=300;
>> f0=150;
>> alpha=0.01;
>> PredPrey01=@(t,x) [2*(1-x(1)/R)*x(1)-alpha*x(1)*x(2); ...
                      -x(2)+alpha*x(1)*x(2)];
>> x0=[r0; f0];
>> [t,x]=ode45(PredPrey01,[0 20], x0);
>> plot(t,x);
>> title('FÃ¼chse␣u.␣Hasen␣versus␣Zeit');
>> xlabel('Zeit␣t');
>> [t,xEe]=ForwardEuler(PredPrey01,[0 20], x0, 100);
>> hold on;
>> plot(t,xEe(:,1),'-r');
>> [t,xEi]=BackwardEuler(PredPrey01,[0 20], x0, 100);
>> plot(t,xEi(:,1),'-m');
```

we find the fox- (green) and rabbitpopulation (blue) depicted in Figure 2.

To compare `ode45` with Euler explicit (red) and implicit (magenta) the population of the rabbits is drawn for these methods. Notice that the rabbit population oscillates longer and stronger if using the Euler explicit method (red curve) then if using Euler implicit. This has to do with the stability of the method.

Obviously the solution converges to $(r_\infty, f_\infty)^T = (100, 150)^T$ for large t.

If the system approaches a stationary state, the solution does not change any more and therefore the derivates are becoming zero, i.e we have $\dot{r}(t) = \dot{f}(t) \equiv 0$.

Therefore the Lotke-Volterra ODEs reduces to the system of (nonlinear) equations

$$\begin{aligned} 2\left(1 - \frac{r}{R}\right) r - \alpha r f &= 0 \\ -f + \alpha r f &= 0 \end{aligned}$$

The solution is easy to find: $(150, 100)^T$.

Problem 4: Numerical Solution of the Two-Body Problem

The two-body problem describes the motion of two bodies, i.e. the earth and a satellite under the influence of gravitational forces. We use the coordinates (u, v) in coordinate system, whos origin rests in the larger

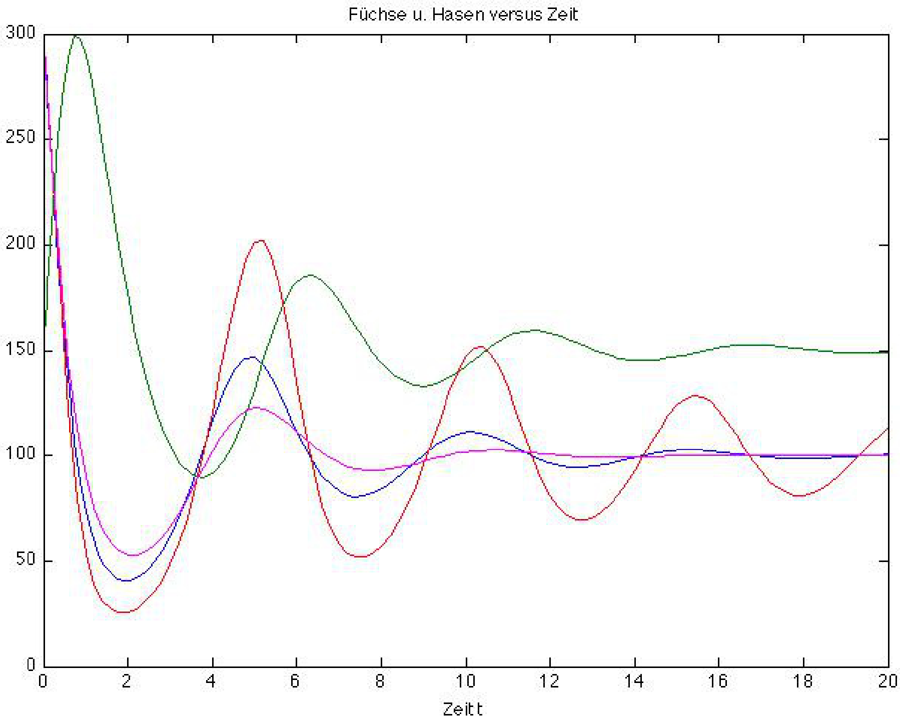

Abbildung 2: Population of rabbits, $r(t)$, and foxes, $f(t)$, versus time t.

of the two bodies (the earth):

$$\ddot{u}(t) = -\frac{u(t)}{(r(t))^3}$$
$$\ddot{v}(t) = -\frac{v(t)}{(r(t))^3}$$

where $r(t) = \sqrt{(u(t))^2 + (v(t))^2}$.

Deduce from this the corresponding system of first order ODEs.

The Matlab/Octave-function, which represents the right hand side of first order system

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}(x), x)$$

should have the following form:

```
function ydot = twobody(t,y,y0)
r    = sqrt(y(1)^2 + y(2)^2);
ydot = [y(3); y(4); -y(1)/r^3; -y(2)/r^3];
```

A more compact form would be:

```
ydot = @(t,y) [y(3:4); -y(1:2)/norm(y(1:2))^3];
```

Compute and plot the orbit of the satellite as follows:

```
>> global y0;
>> y0 = [1; 0; 0; 0.3];
>> [t,y,te,ye,ie] = ode45(@twobody, [0 2*pi], y0);
>> plot(y(:,1),y(:,2),'-',0,0,'ro')
>> axis equal
>> axis([-0.2 1.1 -0.4 .4])
>> axis equal
>> xlabel('u(t)')
>> ylabel('v(t)')
```

Use function calls like `orbit(2.0e-3)` and `orbit(1.0e-6)`.

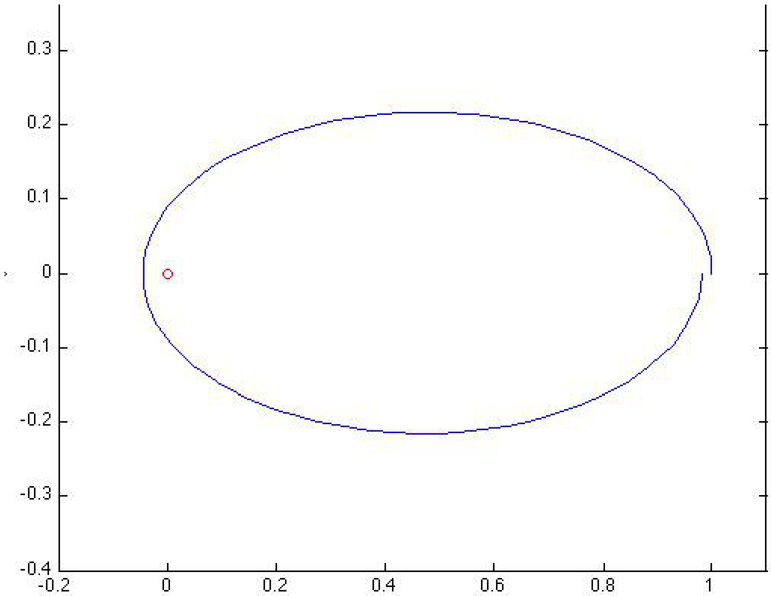

Abbildung 3: For `reltol=2.e-3` the orbit is not closed.

Solution: Using the substitution

$$
\begin{aligned}
y_1(t) &= u(t)\\
y_2(t) &= v(t)\\
y_3(t) &= \dot{u}(t)\\
y_4(t) &= \dot{v}(t)
\end{aligned}
$$

the system of first order ODEs reads:

$$
\begin{aligned}
\dot{y}_1(t) &= y_3(t)\\
\dot{y}_2(t) &= y_4(t)\\
\dot{y}_3(t) &= -\frac{y_1(t)}{r(t)^3}\\
\dot{y}_4(t) &= -\frac{y_2(t)}{r(t)^3}
\end{aligned}
$$

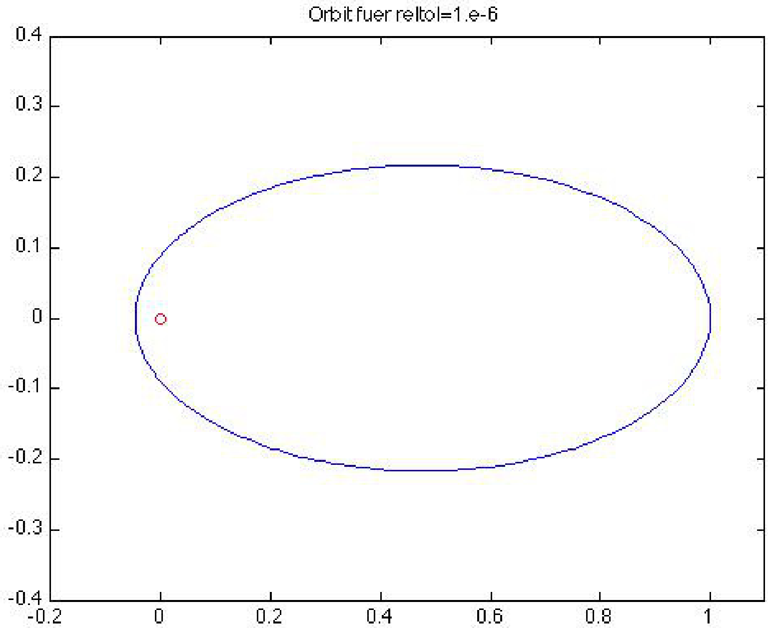

Abbildung 4: For `reltol=1.e-6` the orbit is, or seems to be closed.

where $r(t) = \sqrt{y_1(t)^2 + y_2(t)^2}$. The initial conditions (ICs) are $\mathbf{y}(0) = (1,0,0,0.3)^T$.

The orbit of this inaccurate numerical computation is shown in figure 3. Only if we increase the accuracy, the orbit will be closed as shown in figure 4. From physics, we know, that this is the case (in the absence of any frictional forces).