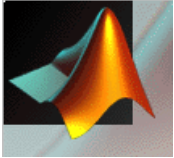
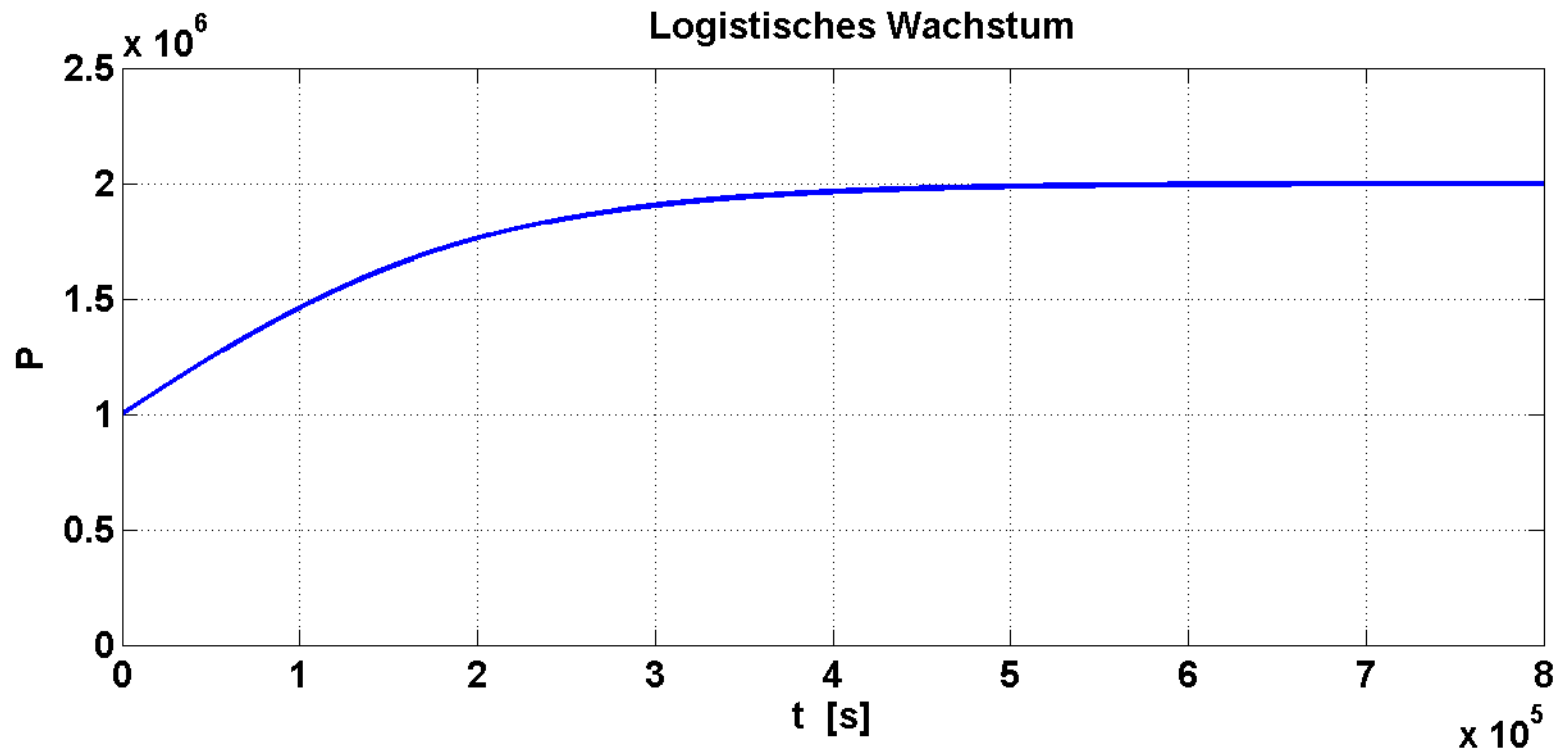


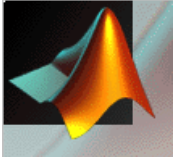
4. Oktober, Aufgabe 2d : Logistisches Wachstum



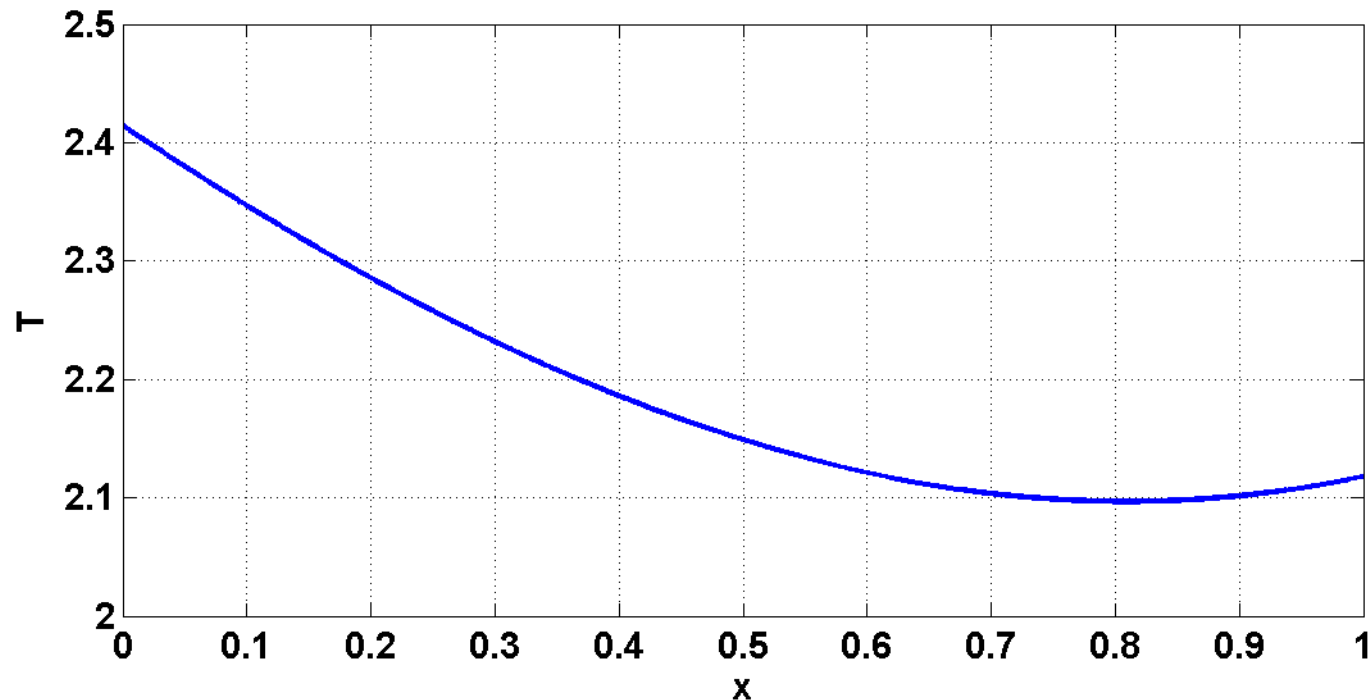
```
>> alpha = 5.0e-12; P0 = 1.0e+6; Pmax = 2.0e+6;  
>> t = 0:100:8.0e+5;  
>> P = Pmax ./ ( 1 + (Pmax/P0-1) * exp(-alpha*Pmax*t) );  
>> plot ( t, P ); xlabel ( 't [s]' ); ylabel ( 'P' );  
>> axis ( [ 0 8.0e+5 0 2.5e+6 ] ); grid;  
>> title ( 'Logistisches Wachstum' );
```



Beispiel 2.16 : Fermatsches Prinzip



```
>> a = 2.0;    b = 1.0;    l = 1.0;    v1 = 2.0;    v2 = 1.0;
>> T = sqrt(a^2+x.^2)/v1 + sqrt(b^2+(l-x).^2)/v2;
>> plot(x,T);  xlabel ( 'x' );  ylabel ( 'T' );
>> axis ( [ 0 1 2 2.5 ] );  grid;
```



4. Oktober, Aufgabe 3 : Bewegungssimulation (I)

```
function [] = aufgabe6 ( dt, ihfcn );
% Aufgabe6
%
% Dieses m-file ist Teil der Arbeitsmaterialien des
% Schülerseminars "Bewegungssimulation mit dem Computer"
% am Fachbereich Mathematik und Informatik der
% Martin-Luther-Universitaet Halle-Wittenberg, Deutschland
%
% Author : Prof. Dr. M. Arnold, martin.arnold@...
% Version of : Oct 4, 2005
%
% Bewegungssimulation
%
% Parameter:
% dt (input) : Zeitschritt
% ihfcn (input) : control flag "function h(x)"
%               =1 .. sinusoidal
%               =2 .. table look-ups
%
% Beispiel:
% aufgabe6 ( 1.0e-3, 1 );
% print -dpng ../png/aufgabe6a.png
%
% -> Fahrweg einlesen
if ihfcn==1,
    hdat = [];
elseif ihfcn==2,
    hdat = load ( 'h.dat' );
end;
%
% -> Modellparameter
t0 = 0.0; % Anfangszeit
te = 10.0; % Endzeit
y0 = 0.0; % Anfangshoeh
v0 = 0.0; % Anfangsgeschwindigkeit
vx = 2.0; % Geschwindigkeit
hmax = 0.1; % Amplitude der Weganregung
len = 2.0; % Wellenlaenge der Weganregung
m = 285.0; % Masse
k = 1.0e+6; % Federsteifigkeit
d = 1.0e+2; % Daempfung
%
% -> Initialisierung der Ergebnisdaten
nstep = round ( (te-t0)/dt );
t = zeros ( 1+nstep, 1 );
y = zeros ( 1+nstep, 1 );
v = zeros ( 1+nstep, 1 );
t(1) = t0; y(1) = y0; v(1) = v0;
```



4. Oktober, Aufgabe 3 : Bewegungssimulation (II)

```
% -> Berechnung der Ergebnisse
tn = t0; yn = y0; vn = v0;
for istep=1:nstep,
    [ h, hp ] = hfcn ( tn, vx, len, ihfcn, hdat );
    f = - k * ( yn - h ) - d * ( vn - hp );
    tn = tn + dt; % neuer Zeitpunkt
    yb = yn + dt*vn;
    vb = vn + dt*f/m;
    [ h, hp ] = hfcn ( tn, vx, len, ihfcn, hdat );
    fb = - k * ( yb - h ) - d * ( vb - hp );
    yn = yn + dt*(vn+vb)/2;
    vn = vn + dt*(f+fb)/(2*m);
    t(istep+1) = tn;
    y(istep+1) = yn;
    v(istep+1) = vn;
end;

% -> Graphische Ausgabe
hvec = zeros ( size(t) );
hpvec = zeros ( size(t) );
for it=1:length(t),
    [ hvec(it), hpvec(it) ] = ...
        hfcn ( t(it), vx, len, ihfcn, hdat );
end;
plot ( t, hvec, 'g', t, y, 'b' );
xlabel ( 't [s]' );
ylabel ( 'y [m]' );
set ( gca, 'XLim', [ t0 te ] );

% -> Fahrweg
function [ h, hp ] = hfcn ( t, vx, len, ihfcn, hdat ),
if ihfcn==1,
    h = 0.1 * sin ( 2*pi*vx*t / len );
    hp = 0.1 * 2 * pi * vx / len * cos ( 2*pi*vx*t / len );
elseif ihfcn==2,
    delx = diff ( hdat(1:2,1) );
    x = vx * t;
    icur = 1 + floor ( x / delx );
    hdel = ( hdat(icur+1,2) - hdat(icur,2) ) / ...
        ( hdat(icur+1,1) - hdat(icur,1) );
    h = hdat(icur,2) + hdel * ( x - hdat(icur,1) );
    hp = hdel * vx;
end;
```

