## Schülerseminar

# Bewegungssimulation mit dem Computer

http://sim.mathematik.uni-halle.de/~arnold/courses/Schueler05


Institut für Numerische Mathematik

- Prof. Dr. M. Arnold ( martin.arnold@mathematik.uni-halle.de )

Georg-Cantor-Haus (Heide Süd), Theodor-Lieser-Str. 5, Raum 221

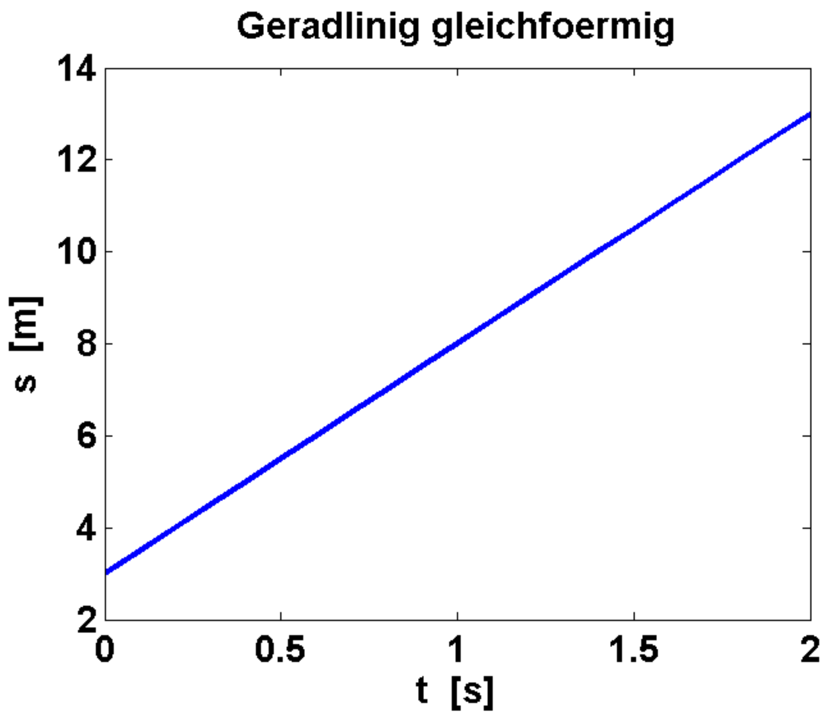
URL: http://www.mathematik.uni-halle.de/~arnold/

# Beispiel 1.2: Geradlinig gleichförmige Bewegung

see beispiel0102.m

```
% how to use
help beispiel0102
```

```
% -> physikalische Parameter
s0 = 3.0;                  % Anfangslage
v  = 5.0;                  % Geschwindigkeit

% -> Berechnung der Funktionswerte
t  = 0:0.1:2;              % Zeit
s  = s0 + v*t;             % Weg

% -> graphische Darstellung
plot ( t, s );
xlabel ( 't  [s]' );
ylabel ( 's  [m]' );
title ( 'Geradlinig gleichfoermig' );
```

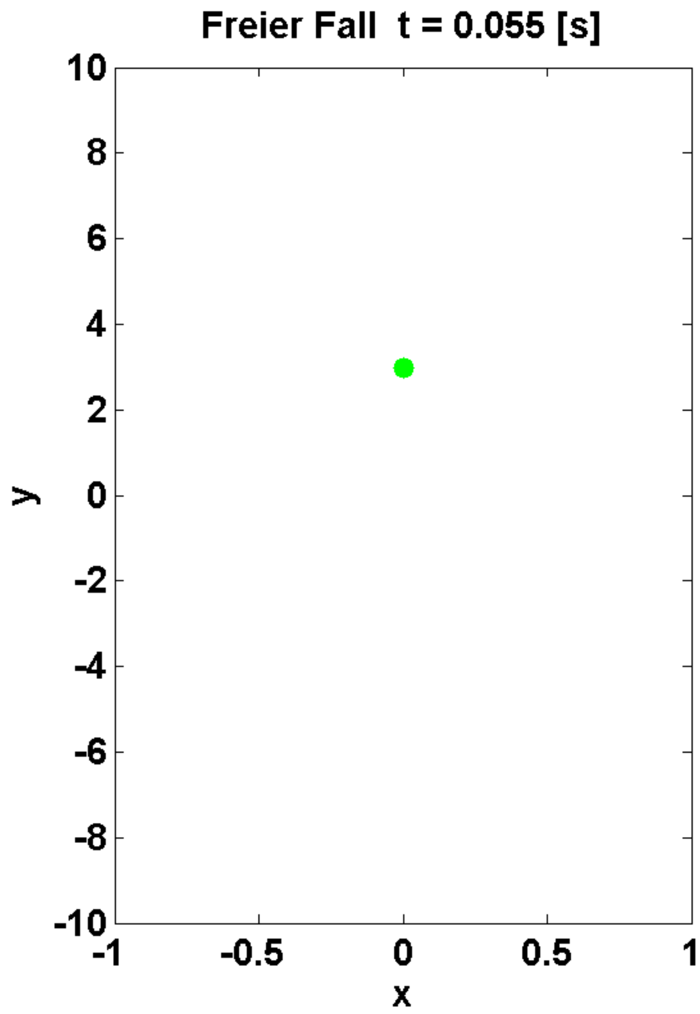


# Beispiel 1.3: Newtons Apfel

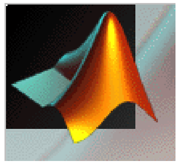
see beispiel0103.m

```
% how to use
help beispiel0103
```

```
% -> physikalische Parameter
t0 = 0.0;                   % Anfangszeit
y0 = 3.0;                   % Anfangshoehe
g  = 9.81;                  % Fallbeschleunigung

% -> Initialisierung der graphischen Darstellung
ymin = -10.0;   ymax = 10.0;
h = plot ( t0, y0, '.g', 'MarkerSize', 24 );
xlabel ( 'x' );   ylabel ( 'y' );
title ( 'Freier Fall' );
axis ( [ -1 1 ymin ymax ] );

% -> Berechnung der Ergebnisse und Animation
t = t0;   y = y0;           % Initialisierung
while y>ymin,
    t = t + 0.005;          % Zeitschritt von 5 ms
    y = y0 - g*t^2/2;       % neue Hoehe
    set ( h, 'YData', y );
    title ( sprintf('Freier Fall  t = %4.3f [s]',t) );
    drawnow;
end;
```



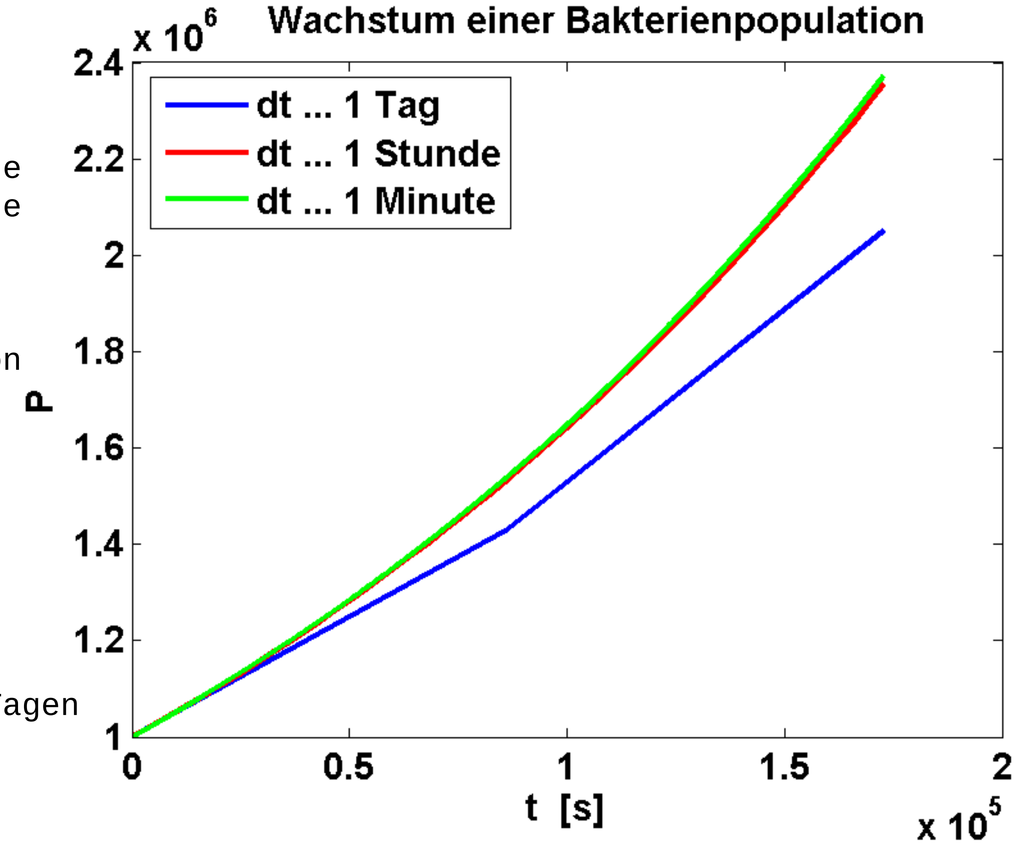
# Beispiel 1.4: Wachstum einer Bakterienpopulation



see beispiel0104.m

```
% how to use
help beispiel0104
```

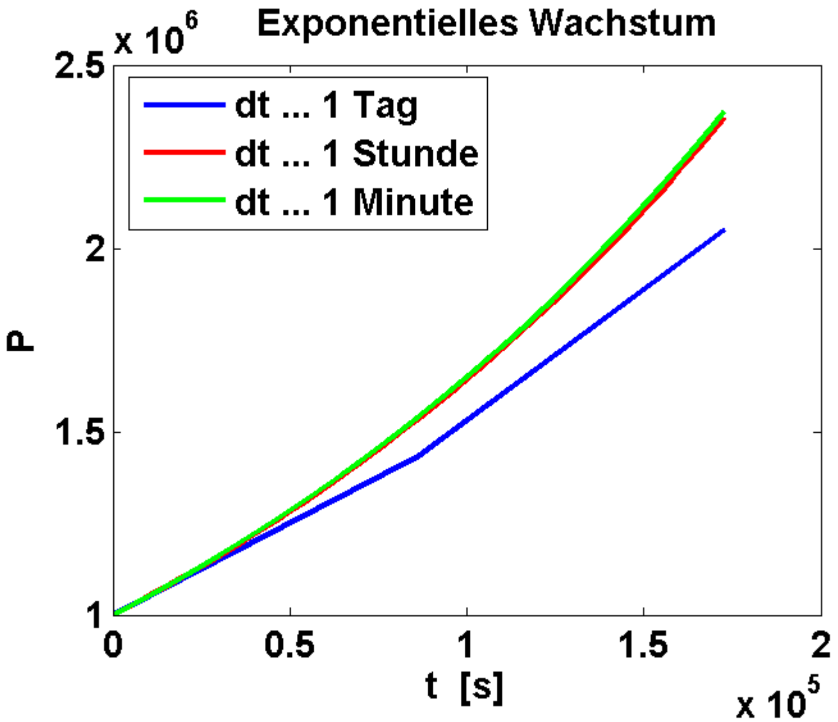
```
function [ t, P ] = beispiel0104 ( dt );
% Parameter:
%   dt      (input)  : Zeitschritt
%   t       (output) : Vektor der Zeitpunkte
%   P       (output) : Vektor der Ergebnisse

% -> Modellparameter
t0 = 0.0;                   % Anfangszeit
P0 = 1.0e+6;                % Anfangspopulation
al = 5.0e-6;                % Wachstumsrate

% -> Initialisierung der Ergebnisdaten
t = [ t0 ];   P = [ P0 ];

% -> Berechnung der Ergebnisse
tn = t0;   Pn = P0;
while tn+dt<=2*(24*60*60),
                            % Abbruch nach 3 Tagen
    tn = tn + dt;           % neuer Zeitpunkt
    Pn = Pn + dt*al*Pn;     % neue Population
    t = [ t; tn ];   P = [ P; Pn ];
end;
```

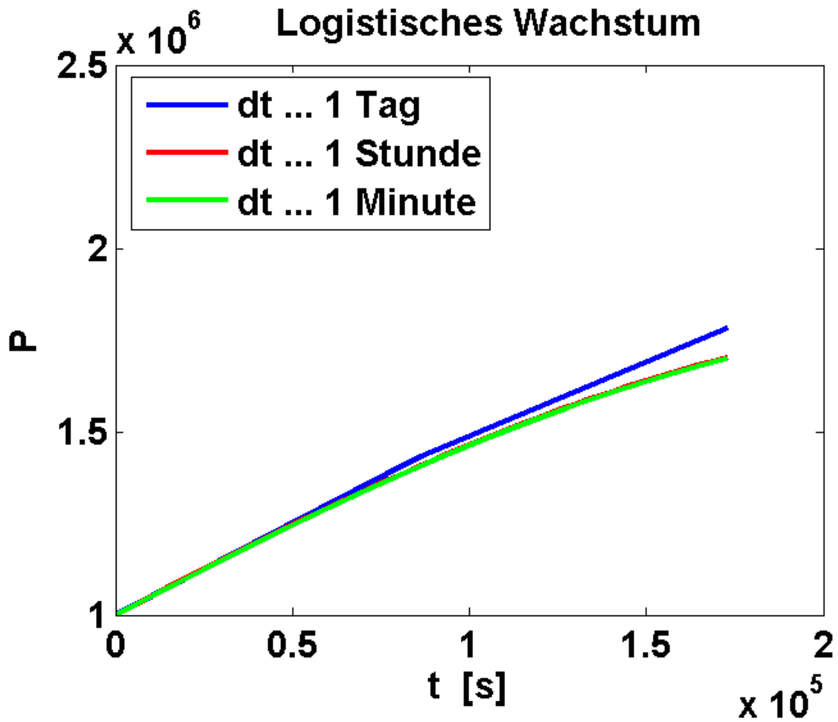



# Aufgabe 2: Wachstum einer Bakterienpopulation

$$P(t+\Delta_t) = P(t) + \Delta_t \cdot \alpha P(t)$$
$$P(0) = 1.0_{\mathsf{E}} + 6$$
$$\alpha = 5.0_{\mathsf{E}} - 6$$

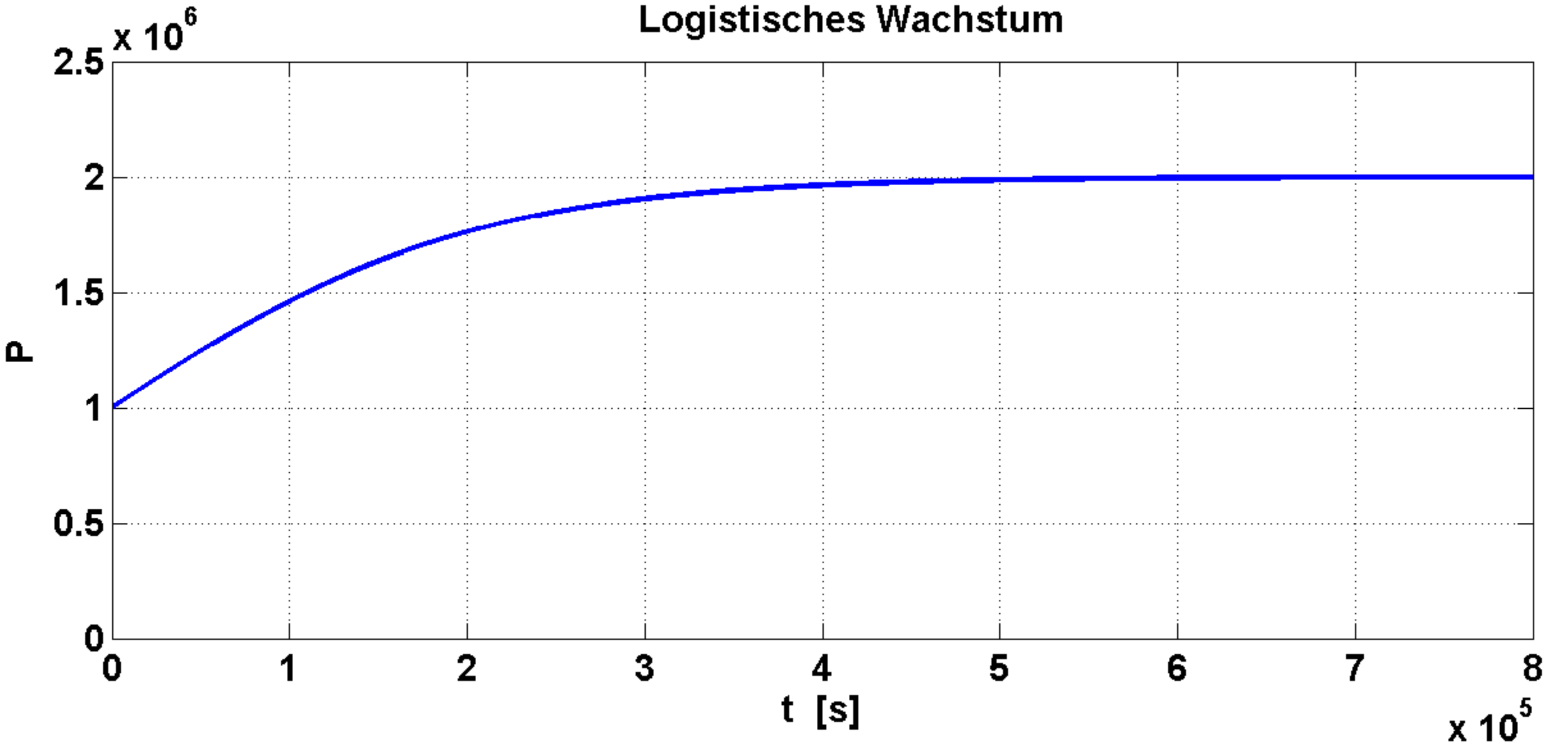
$$P(t+\Delta_t) = P(t) + \Delta_t \cdot \alpha P(t)(P_{\mathsf{max}} - P(t))$$
$$P(0) = 1.0_{\mathsf{E}} + 6$$
$$\alpha = 5.0_{\mathsf{E}} - 12,\ P_{\mathsf{max}} = 2.0_{\mathsf{E}} + 6$$



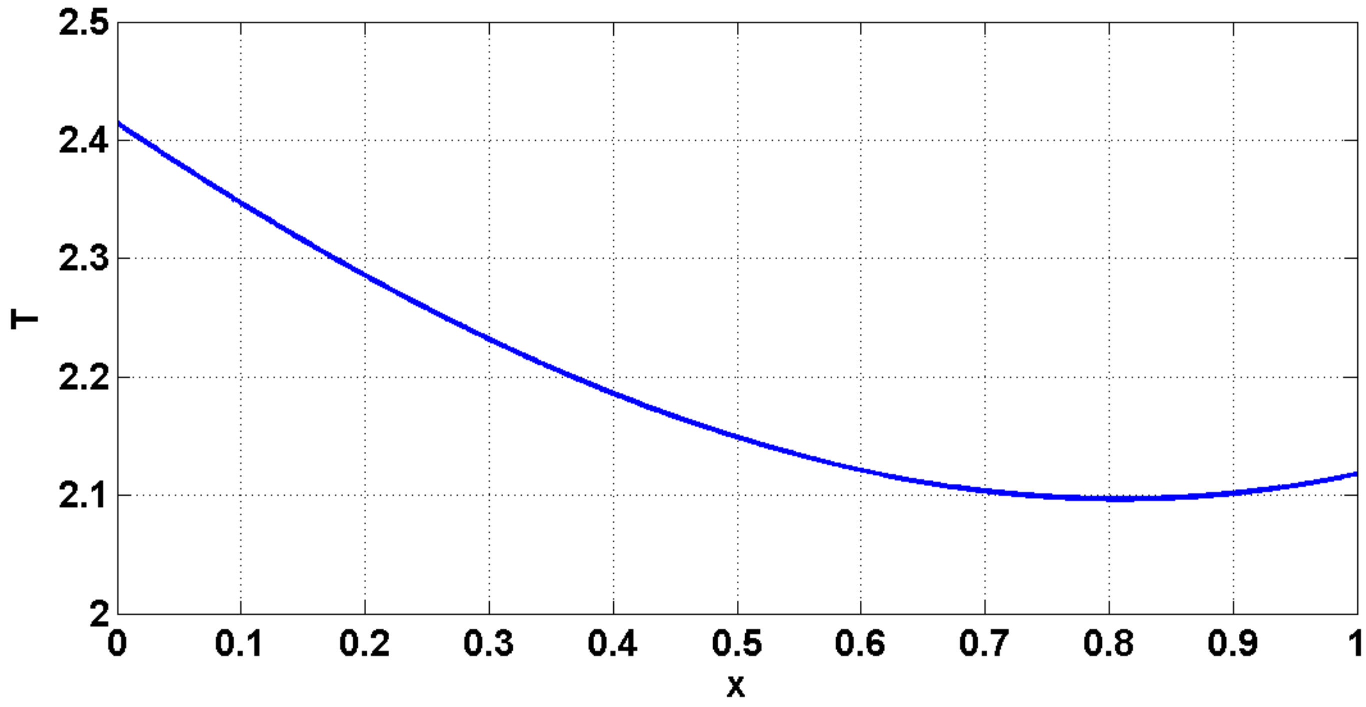
# 4. Oktober, Aufgabe 2d : Logistisches Wachstum

```
>> 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
%   Schuelerseminars "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;             % Anfangshoehe
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)

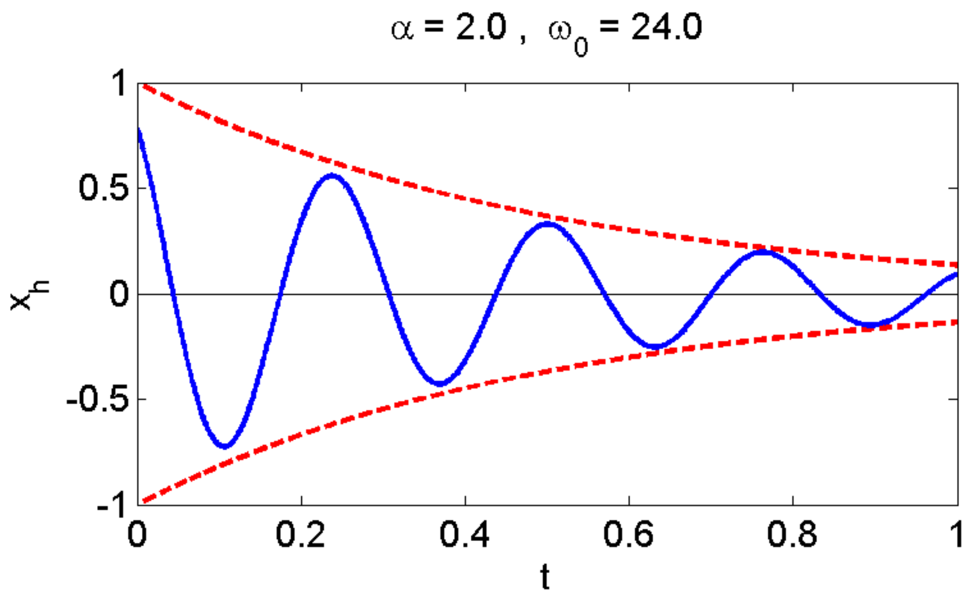
```
% -> Berechnung der Ergebnisse
tn = t0;    yn = y0;    vn = v0;
for istep=1:nstep,
    [ h, hp ] = hfcn ( tn, vx, len, ihfcn, hdat );
    f  = - k * ( yn - h ) - d * ( vn - hp );
    tn = tn + dt;        % neuer Zeitpunkt
    yb = yn + dt*vn;
    vb = vn + dt*f/m;
    [ h, hp ] = hfcn ( tn, vx, len,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

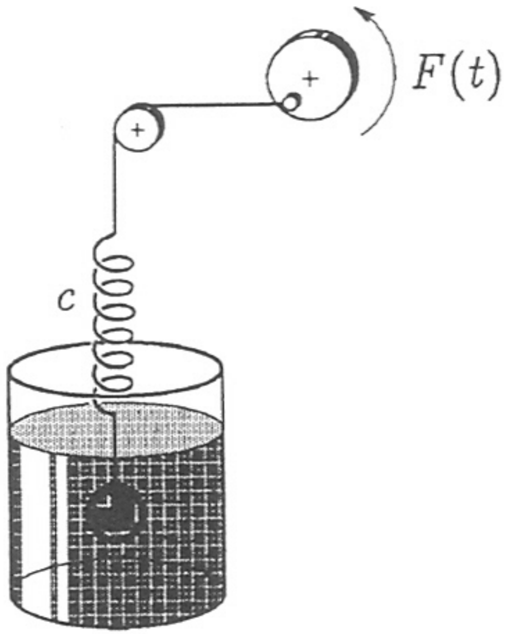
# Bemerkung 3.9b: Lineare Schwingungen

Aufgabenstellung

$$\ddot{x}(t) + 2\alpha\dot{x}(t) + \omega_0^2 x(t) = 0$$

Periodischer Fall ( $\alpha^2 - \omega_0^2 < 0$ ): $\quad x_h(t) = C\mathrm{e}^{-\alpha t}\cos(\omega_1 t - \delta)$



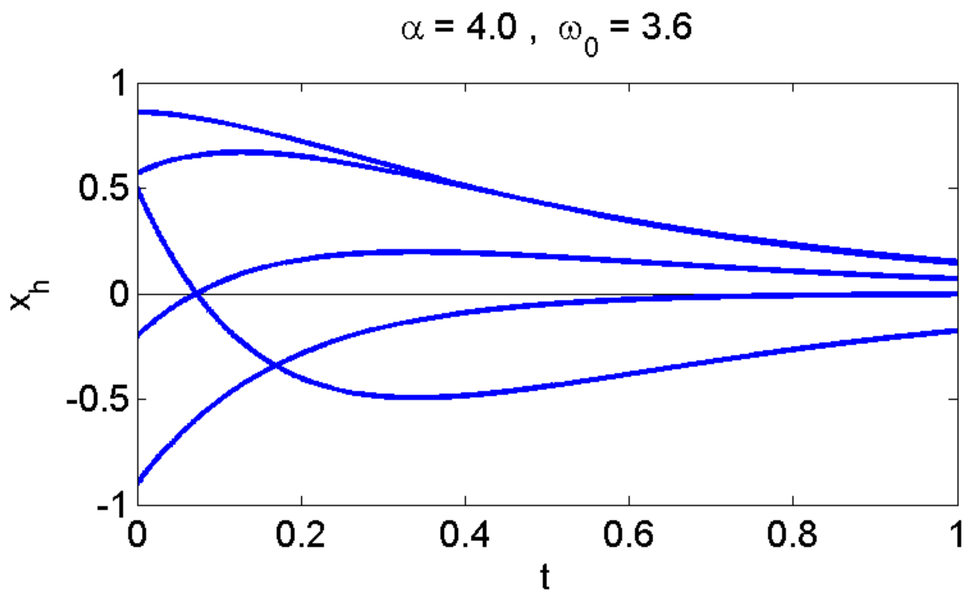
# Bemerkung 3.9b: Lineare Schwingungen (II)

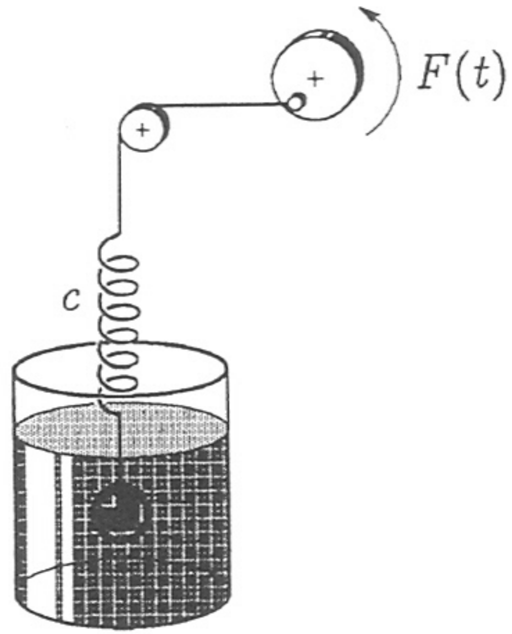
Aufgabenstellung

$$\boxed{\ddot{x}(t) + 2\alpha\dot{x}(t) + \omega_0^2 x(t) = 0}$$

$$\beta := \sqrt{\alpha^2 - \omega_0^2}$$

Aperiodischer Fall ( $\alpha^2 - \omega_0^2 > 0$ ): $\quad x_h(t) = c_1 \mathrm{e}^{(-\alpha+\beta)t} + c_2 \mathrm{e}^{(-\alpha-\beta)t}$



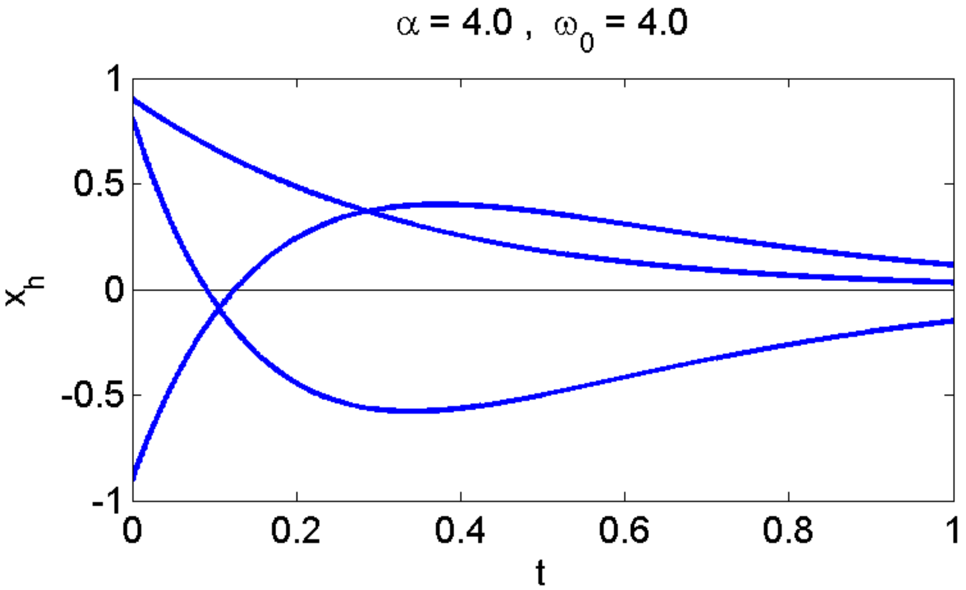



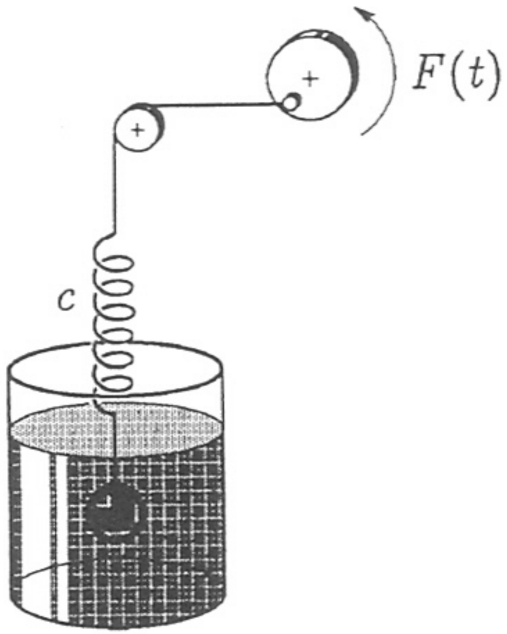
# Bemerkung 3.9b: Lineare Schwingungen (III)

Aufgabenstellung

$$\boxed{\ddot{x}(t) + 2\alpha\dot{x}(t) + \omega_0^2 x(t) = 0}$$

Aperiodischer Grenzfall ( $\alpha^2 = \omega_0^2$ ): $\quad x_h(t) = (c_1 + c_2 t)\,\mathrm{e}^{-\alpha t}$



# Bemerkung 3.9b: Lineare Schwingungen (IV)

**Aufgabenstellung**

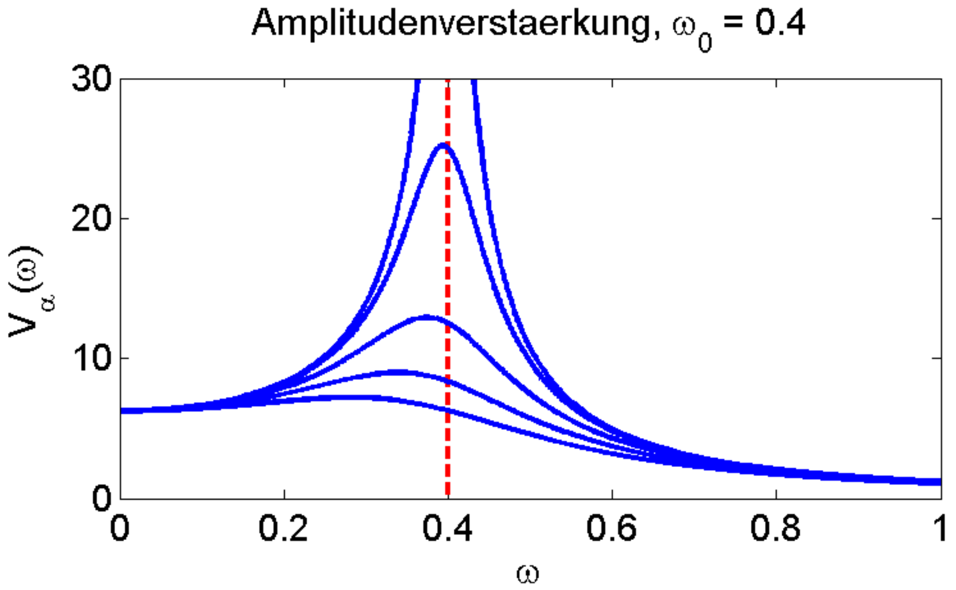
$$\boxed{\ddot{x}(t) + 2\alpha\dot{x}(t) + \omega_0^2 x(t) = A\cos\omega t}$$

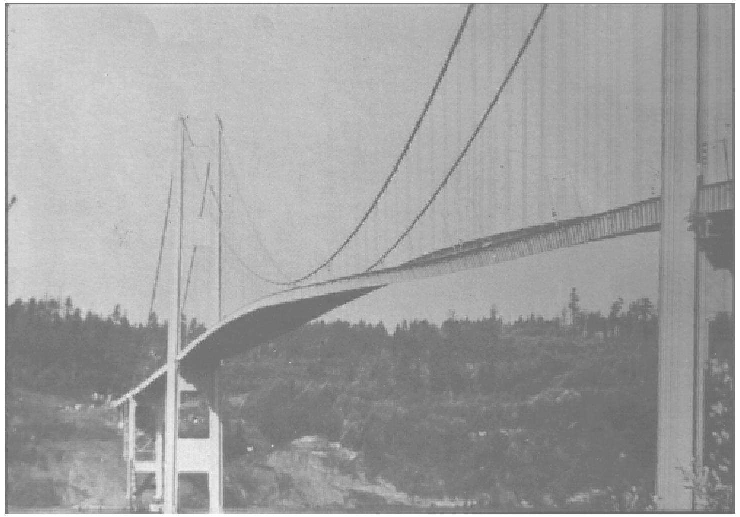
**Partikuläre Lösung**

$$x_0(t) = \frac{A}{\sqrt{(\omega_0^2-\omega^2)^2+4\alpha^2\omega^2}}\cos(\omega t-\varphi), \quad \varphi = \arctan\frac{2\alpha\omega}{\omega_0^2-\omega^2}$$

**Amplitudenverstärkung**

$$V_\alpha(\omega) = \frac{1}{\sqrt{(\omega_0^2-\omega^2)^2+4\alpha^2\omega^2}}$$



http://www.cornelsen.de/physikextra/htdocs/Resonanz.html