Herleitung des Euler-Verfahrens

$$y' = f(x_n, y_n) = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} \approx \frac{\Delta y}{\Delta x} = \frac{y_{n+1} - y_n}{x_{n+1} - x_n}$$

$$= \frac{y_{n+1} - y_n}{h}$$

Steigung an der Stelle (x_n, y_n)

Schrittweite

$$f(x_n, y_n) \approx \frac{y_{n+1} - y_n}{h}$$

$$\Leftrightarrow \quad \boxed{y_{n+1} \approx y_n + h \cdot f(x_n, y_n)}$$

explizites Euler-Verfahren (Vorwärtsverfahren)

ÜBUNG: Numerische Lösung von DGLn mit Hilfe des Euler-Verfahrens

1. Geben Sie ein C-Programm zur Lösung der folgenden DGL 1. Ordnung mit Hilfe des Euler-Verfahrens an:

$$y'(x) = -\frac{x}{y(x)}$$

Anm.: x ist die unabh. Variable

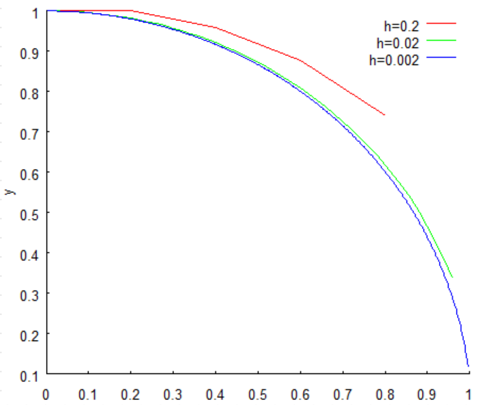
Startwert : y(0)=1

① $y'(x) = \underbrace{-\frac{x}{y(x)}}_{f(x,y)}$ $y(0) = 1$

② $y_{n+1} = y_n + h \cdot \left(-\frac{x_n}{y_n}\right)$

$x_{n+1} = x_n + h$

```
/*****************************************************************/
#include <math.h>
#include <chplot.h>

#define  h  0.02        // Schrittweite
#define  x_End  1


double Ableitung(double x, double y){
   return -x/y;
}


int main(){

int steps = x_End/h;
int i;
double x[steps], y[steps];
class CPlot plot;


// Initialisierung
x[0]=0;
y[0]=1;    // Startwert


for(i=0; i<steps-1; i++){
    y[i+1] = y[i]+h*Ableitung(x[i],y[i]);
    x[i+1] = x[i]+h;
}


plot.data2D(x,y);
plot.plotting();

}
```



ÜBUNG: Numerische Lösung von DGLn mit Hilfe des Euler-Verfahrens

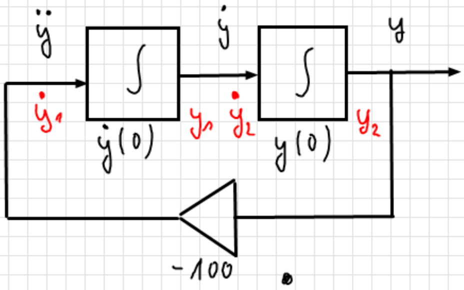
2. Geben Sie ein C-Programm zur Lösung der folgenden DGL 2. Ordnung mit Hilfe des Euler-Verfahrens an:

$$\ddot{y}(t) = -100 \cdot y(t)$$ Anm.: t ist die unabh. Variable

Startwerte: y(0)=0 , $\dot{y}$(0)=10

① $\ddot{y}(t) = -100 \cdot y(t)$ $y(0) = 0$, $\dot{y}(0) = 10$

② Analogrechner-Schaltbild



③ Zwei DGLn 1. Ordnung :

$$\dot{y}_1 = -100\, y_2$$
$$\dot{y}_2 = y_1$$

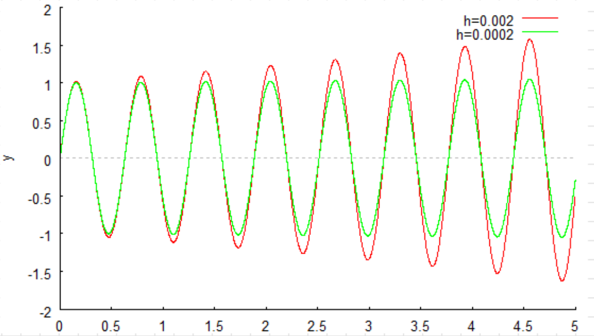
④ Iterationsgleichungen

$$y_{1_{n+1}} = y_{1_n} + h \cdot (-100 \cdot y_{2_n})$$
$$y_{2_{n+1}} = y_{2_n} + h \cdot (y_{1_n})$$
$$t_{n+1} = t_n + h$$

Fortsetzung :

```
/****************************************************************/
#include <math.h>
#include <chplot.h>

#define  h  0.002        // Schrittweite
#define  x_End  1


// Ableitungsfunktionen
double Ableitung1(double y2){
                return -100*y2;
}

double Ableitung2(double y1){
                return y1;
}



int main(){

        int         steps=x_End/h;
        int         i;
        double      x[steps], y1[steps], y2[steps];
        class CPlot plot;

        // Initialisierung
        x[0]  =  0;
        y1[0] = 10;
        y2[0] =  0;

        // Euler-Iteration
        for(i=0;  i<steps-1;  i++){
                y1[i+1]=y1[i]  + h*Ableitung1(y2[i]);
                y2[i+1]=y2[i]  + h*Ableitung2(y1[i]);
                x[i+1]=x[i]+h;
        }

        plot.data2D(x, y2);
        plot.plotting();
}
```

Anm.: x ist die Zeit t



ÜBUNG: Numerische Lösung mit Runge-Kutta (2. Ordnung)

1. Geben Sie ein C-Programm zur Lösung der folgenden DGL 1. Ordnung mit Hilfe des RK2-Verfahrens an:

$$y'(x) = -\frac{x}{y(x)}$$

Anm.: x ist die unabh. Variable

Startwert : y(0)=1

① $y'(x) = -\frac{x}{y(x)}$ $\qquad y(0) = 1$

② Iterationsgleichungen

$$K_1 = h \cdot \left(-\frac{(x_n)}{(y_n)}\right)$$

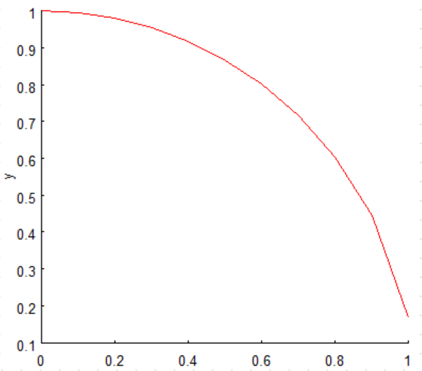
$$K_2 = h \cdot \left(-\frac{(x_n + \frac{h}{2})}{(y_n + \frac{k_1}{2})}\right)$$

$$y_{n+1} = y_n + k_2$$

$$x_{n+1} = x_n + h$$

Fortsetzung :

```
/*****************************************************************/
#include <math.h>
#include <chplot.h>

#define  h  0.1       // Schrittweite
#define  x_End  1

// Ableitungsfunktion
double Ableitung(double x, double y){
   return -x/y;
}

int main(){

int    steps = x_End/h;
int    i;
double k1, k2;
double x[steps], y[steps];
class  CPlot plot;

// Init
x[0]=0;
y[0]=1;

// RK2-Iteration
for(i=0;i<steps-1;i++){
   k1 = h * Ableitung(x[i], y[i]);
   k2 = h * Ableitung(x[i]+h/2 , y[i]+k1/2);
   y[i+1] = y[i] + k2;
   x[i+1] = x[i] + h;
}

plot .....

}
```



ÜBUNG: Numerische Lösung mit Runge-Kutta (2. Ordnung)

2. Geben Sie ein C-Programm zur Lösung der folgenden DGL 2. Ordnung mit Hilfe des RK2-Verfahrens an:

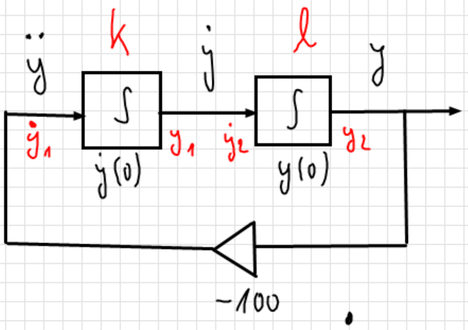
$$\ddot{y}(t) = -100 \cdot y(t)$$

Anm.: t ist die unabh. Variable

Startwerte: y(0)=0 , $\dot{y}$(0)=10

① $\ddot{y} = -100 \cdot y \qquad y(0) = 0 \, , \; \dot{y}(0) = 10$

② Analogrechner-Schaltbild



③ Zwei DGLn 1. Ordnung: $\dot{y}_1 = -100 \cdot y_2$

$\dot{y}_2 = y_1$

④ RK 2. Ordnung

$k_1 = h \cdot (-100 \cdot y_{2n})$

$\ell_1 = h \cdot (y_{1n})$

k_1, k_2 : Integrator 1 $(\dot{y}_1, y_1)$

ℓ_1, ℓ_2 : Integrator 2 $(\dot{y}_2, y_2)$

beachte

$k_2 = h \cdot \left[-100 \cdot \left(y_{2n} + \frac{\ell_1}{2} \right) \right]$

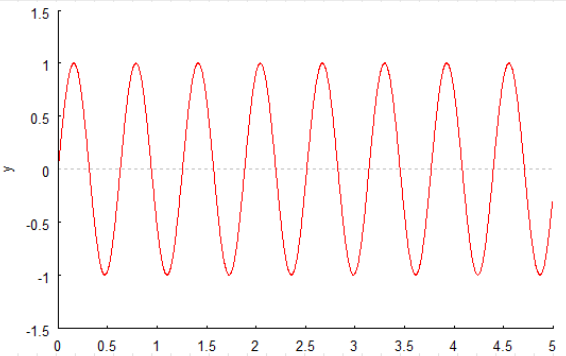
$\ell_2 = h \cdot \left(y_{1n} + \frac{k_1}{2} \right)$

$y_{1_{n+1}} = y_{1_n} + k_2 \qquad y_{2_{n+1}} = y_{2n} + \ell_2 \qquad t_{n+1} = t_n + h$

Fortsetzung:

```
/******************************************************************/
#include <math.h>
#include <chplot.h>

#define  h  0.002        // Schrittweite
#define  x_End  1

// Ableitungsfunktionen
double Ableitung1(double y2){
   return -100*y2;
}

double Ableitung2(double y1){
   return y1;
}


int main(){

....
....

for(...){
   k1 = h*Ableitung1(y2[i]);
   l1 = h*Ableitung2(y1[i]);

   k2 = h*Ableitung1(y2[i]+l1/2);
   l2 = h*Ableitung2(y1[i]+k1/2);

   y1[i+1]=y1[i]+k2;
   y2[i+1]=y2[i]+l2;

   x[i+1] = x[i]+h;

}
```

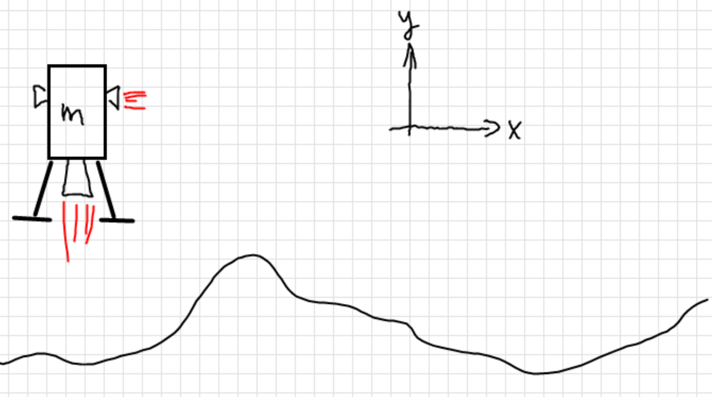
Anm.: x ist die Zeit t



Moonlander

- 3 Tasten
- Konstanter Schub oder Ø

$\pm F_x,\ F_y$



Physik: $\boxed{F = m \cdot a}$ $\boxed{m(t) = m_L + m_T(t)}$

Schub/Grav. $\ddot{x}$ bzw. $\ddot{y}$

Treibstoffmasse

Leermasse "Lander"

Ursache Wirkung

$$\Longrightarrow \quad F_x = m \cdot \ddot{x}$$

$$F_y - m \cdot g_M = m \cdot \ddot{y}$$

$$\Longrightarrow \quad \ddot{x} = \frac{F_x}{m} \cdot T_x$$

$$\ddot{y} = \frac{F_y}{m} \cdot T_y - g$$

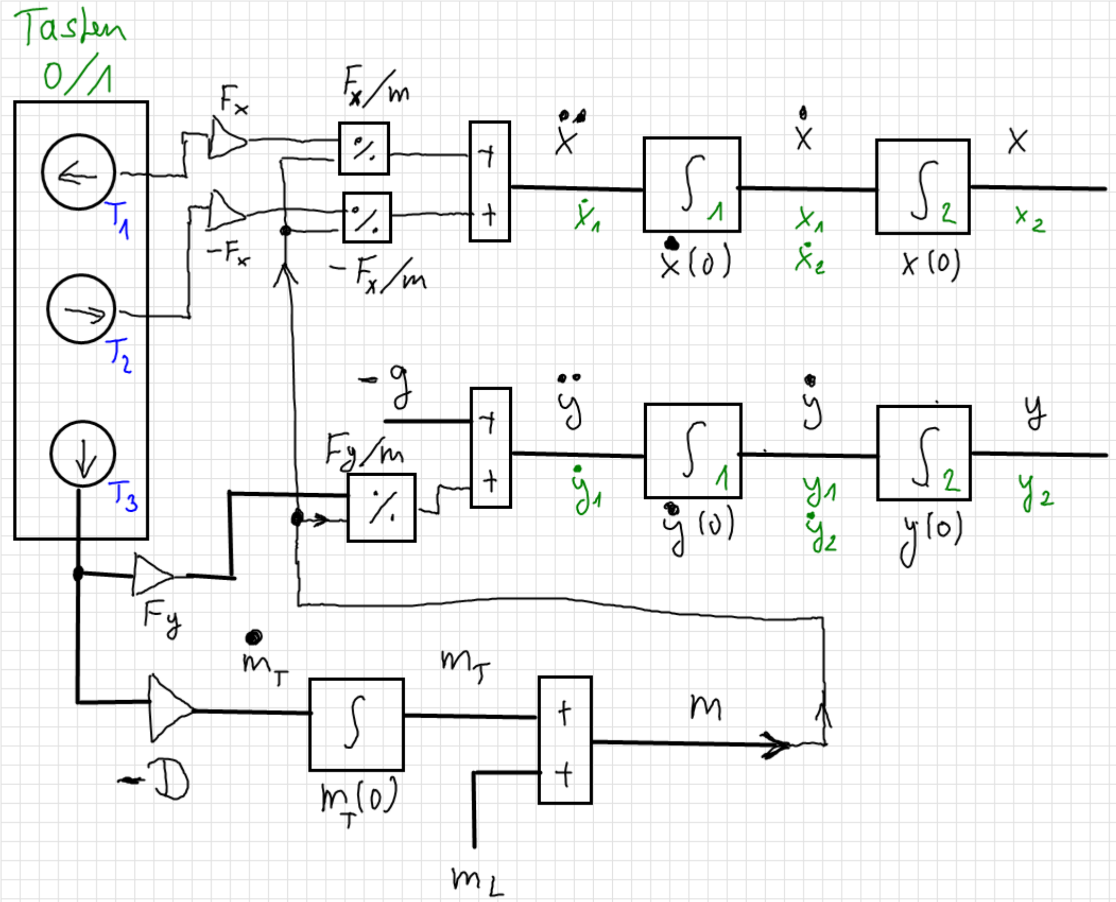
$$\dot{m}_T = -D \cdot T_y$$

Durchsatz in kg/s

T_x : Schubtasten x
Wert: 0 oder 1

T_y : Schubtaste y
Wert: 0 oder 1

Anm.: Treibstoffverbrand nur bei y-Schub (Vereinfachung)



Umwandeln in DGL'n 1. Ordnung

$$\dot{x}_1 = T_1 \cdot \frac{F_x}{m} - T_2 \cdot \frac{F_x}{m} \qquad \dot{x}_2 = x_1$$

$$\dot{y}_1 = T_3 \cdot \frac{F_y}{m} - g \qquad \dot{y}_2 = y_1$$

$$\dot{m}_T = -D \cdot T_3$$

$$m = m_L + m_T$$

$$\boxed{\begin{array}{ll} \dot{x}_1 = T_1 \cdot \frac{F_x}{m} - T_2 \cdot \frac{F_x}{m} & \dot{x}_2 = x_1 \\ \dot{y}_1 = T_3 \cdot \frac{F_y}{m} - g & \dot{y}_2 = y_1 \\ \dot{m}_T = -D \cdot T_3 & \\ m = m_L + m_T & \end{array}}$$

Iterationsgleichungen

$$x_{1_{n+1}} = x_{1_n} + h \cdot \left[T_1 \cdot \frac{F_x}{m_n} - T_2 \cdot \frac{F_x}{m_n} \right]$$

$$x_{2_{n+1}} = x_{2_n} + h \cdot \left[x_{1_n} \right]$$

$$y_{1_{n+1}} = y_{1_n} + h \cdot \left[T_3 \cdot \frac{F_y}{m_n} - g \right]$$

$$y_{2_{n+1}} = y_{2_n} + h \cdot \left[y_{1_n} \right]$$

$$m_{T_{n+1}} = m_{T_n} + h \cdot \left[-D \cdot T_3 \right]$$

$$m_{n+1} = m_L + m_{T_{n+1}}$$

$$t_{n+1} = t_n + h$$