

HMDGL – Beispiel zur Programmierung in C

```
/* ----- */
/* Ein 100 µF-Kondensator wird über einen 1000 Ω-Widerstand an eine */
/* Spannung von 10 V angeschlossen. Für den Aufladestrom i(t) gilt: */
/* */
/* di/dt = -i / (0.1 sec)      Startwert: i(0 sec) = 0.01 A */
/* */
/* Es wird der Stromverlauf i(t) im Zeitintervall 0...1 sec ermittelt, */
/* die Ausgabe der berechneten Werte erfolgt auf dem Bildschirm. */
/* ----- */
#include <stdio.h>
#include "hmdgl.h"

/* ----- */
/* Die rechte Seite der Differentialgleichung di/dt = -i / (0.1 sec) wird */
/* berechnet, die Parameter "t" und "pdata" werden dazu nicht benötigt. */
/* ----- */
void sys(double didt[], double t, const double i[], void *pdata)
{
    didt[0] = -i[0] / 0.1;
}

/* ----- */
/* Nach jedem Integrationsschritt wird die Funktion <step> aufgerufen. */
/* ----- */
void step(double t, const double i[], void *pdata)
{
    /* Aktuelle "Simulationszeit" und Stromstärke ausgeben */
    printf("t = %6.3f sec, i = %6.4f mA\n", t, 1000.0 * i[0]);
}

/* ----- */
/* Die Funktion <init> legt die Anfangswerte fest, hier: i(0 sec) = 0.01 A */
/* ----- */
void init(double i[], void *pdata)
{
    i[0] = 0.01;
}

/* ----- */
/* Hauptprogramm mit Aufruf des numerischen Lösungsverfahrens. */
/* ----- */
int main(void)
{
    size_t num_steps = 1000; /* Anzahl Integrationsschritte */
    size_t dim       = 1;    /* Eine DGL erster Ordnung */
    double t_begin   = 0;    /* Startzeitpunkt der Simulation */
    double t_end     = 1.0;  /* Endzeitpunkt der Simulation */

    dgl_rk4(t_begin, t_end, num_steps, dim, sys, step, init, 0);
    // dgl_euler(t_begin, t_end, num_steps, dim, sys, step, init, 0);
    // dgl_heun(t_begin, t_end, num_steps, dim, sys, step, init, 0);

    return 0;
}
```