Hochschule München, Fak. 03, SoSe 14 **Numerische Verfahren (Warendorf, Wolfsteiner)**

Aufgabe: Differentialgleichung / Euler-Verfahren

Zur numerischen Lösung gewöhnlicher Differentialgleichungen greift man im Regelfall auf bestehende Funktionen zurück, die eine Approximation höherer Ordnung mit Schrittweiten-kontrolle durchführen (z.B. Runge-Kutta-Verfahren: `ode23`, `ode45`). Beispielhaft sei hier die Differentialgleichung

$$\dot{x} = x\,, \text{ mit der Anfangsbedingung } x(t=0)=1 \qquad (1)$$

genannt, deren Lösung die Exponentialfunktion $x(t)=e^t$ ergibt. Die numerische Lösung kann in Matlab z.B. mit `ode45` folgendermaßen bestimmt werden:

```
function exponentialfunktion_ode23
[t,x]=ode45(@odefun,[0 3],1);
plot(t,x,'-o',t,exp(t),'-'); grid on;
legend('numerische Lösung','analytische Lösung')
end
%
function x_p=odefun(t,x)
x_p=x;
end
```

Aufgaben

1. Programmieren Sie das Euler-Verfahren für die Differentialgleichung (1)!
2. Schreiben Sie eine Funktion `ode_euler`, die die gleiche Übergabeliste wie die `ode`-Routinen aus Matlab hat. Speichern Sie diese Funktion in einer separaten Datei `ode_euler.m`.
3. Testen Sie diese Funktion für die Differentialgleichungen (1) und (2)!

   DGL: $\dot{x} = -t \cdot x$ mit der Anfangsbedingung $x(1)=2$ (2)