

Übungen zur Vorlesung  
**Mathematische Aspekte der Neuronenmodellierung und Simulation**  
[http://conan.iwr.uni-heidelberg.de/teaching/numsimneuro\\_ss2012](http://conan.iwr.uni-heidelberg.de/teaching/numsimneuro_ss2012)

Dr. S. Lang, D. Popović

Abgabe: 23. Mai 2012 in der Übung

---

**Übung 5 Lösen von linearen Gleichungssystemen mit Octave (5 Punkte)**

Octave besitzt einige built-in Integratoren zum Lösen von Differentialgleichungs-Systemen. Einer der bekanntesten ist *lsode* von Alan C. Hindmarsh.

Machen Sie sich per Hilfe-Funktion oder anderer Dokumentation mit der Funktion `lsode` vertraut und lösen Sie anschließend das folgende System gewöhnlicher Differential-Gleichungen:

$$\begin{aligned}\partial_t x_1 &= 5 \cdot (x_2 - x_1 \cdot x_2 + x_1 - 10^{-6} \cdot x_1^2) \\ \partial_t x_2 &= (x_3 - x_1 \cdot x_2 - x_2)/5 \\ \partial_t x_3 &= 0.1 \cdot (x_1 - x_3)\end{aligned}$$

mit den Startwerten  $(x_1(0), x_2(0), x_3(0))^t = (0.1, 3, 0.3)^T$ . Der Aufruf von `lsode` geschieht über `y = lsode ("lsode_example1", x0, t)`; wobei `x0` der Startwert und `t` ein Zeitintervall ist. Der Parameter `lsode_example1` ist der Name einer Funktion, die die Differentialgleichung implementiert. Am Besten wird diese Funktion in einer Datei gleichen Namens gespeichert (also `lsode_example1.m`). Die Datei kann für dieses Beispiel dann wie folgt aussehen:

```
1 1 function xdot = lsode_example1 (x, t)
2 2
3 3     xdot = zeros (3,1);
4 4
5 8     xdot(1) = x(1)*x(2);
6 8     xdot(2) = ...
7 8     xdot(3) = ...
8 11
9 12 endfunction
```

Die Funktion `lsode_example1` berechnet das Differentialgleichungs-System und gibt die Lösung des Systems in einem Vektor zurück.

Plotten Sie alle drei Komponenten der Lösung unseres Systems im Bereich  $x = [0, 500]$  mit einem Spacing von 2. Bei ungefähr  $t = 0, 125, 250$  und  $t = 365$  sollten Sie Peaks der Lösung erkennen. Mit Hilfe des `logspace`-Befehls können Sie den Plotbereich um die Peaks herum verfeinern. Exportieren Sie einen Plot der Lösung, falls möglich, mit angepasstem Bereich (ansonsten mit dem uniformen).

## Übung 6 Integrate-and-Fire-Neuron mit Backward Euler und Crank-Nicolson (10 Punkte)

In dieser Aufgabe wird das *Leaky Integrate and Fire*-Modell,

$$\begin{aligned}\tau_m \partial_t v(t) &= -(v(t) - v_L) + R \cdot I(t), \\ v(t) &= v_{reset}, & \text{falls } v(t) > v_{threshold}, \\ v(0) &= v_0,\end{aligned}$$

mit zwei neuen numerischen Verfahren, *Backward Euler* (BE) und *Crank-Nicolson* (CN), numerisch gelöst. Bisher hatten wir das *Forward Euler*-Verfahren (FE) verwendet. Zur Bedeutung der Parameter im Modell siehe das erste Übungsblatt.

Für eine lineare ODE

$$\begin{aligned}\partial_t v &= a \cdot v + b(t) \\ v(0) &= v_0,\end{aligned}$$

$a \in \mathbb{R}$  eine Konstante, hat das Backward Euler-Verfahren die Form

$$\begin{aligned}k_n^{-1} (v_n - v_{n-1}) &= a \cdot v_n + b(t_n) \\ v(0) &= v_0,\end{aligned}$$

wobei  $v_n$  das numerisch berechnete Potential zum Zeitpunkt  $t_n$  und  $\kappa_n := t_n - t_{n-1}$  die Zeitschrittweite beim Schritt von  $t_{n-1}$  nach  $t_n$  bezeichnen, und das Crank-Nicolson-Verfahren lautet

$$\begin{aligned}k_n^{-1} (v_n - v_{n-1}) &= \frac{1}{2} a \cdot (v_n + v_{n-1}) + \frac{1}{2} (b(t_n) + b(t_{n-1})) \\ v(0) &= v_0.\end{aligned}$$

Das Leaky-integrate-and-fire-Modell ist zwar nicht-linear (durch das Rücksetzen des Potentials beim Spiken), dennoch können die Verfahren stückweise für die Inter-Spike-Intervalle verwendet werden, wenn der Spike explizit gehandelt wird – wie bisher beim Forward Euler-Verfahren.

Für den Fall eines konstanten angelegten Stroms kann eine analytische Lösung angegeben werden:

$$v(t) = v_{steady} + (v_0 - v_{steady}) \cdot \exp\{-t/\tau_m\}$$

mit der Gleichgewichtslage  $v_{steady} = v_L + R \cdot I = v_{rest} + R \cdot I$ . Die Zeit  $t$  in der analytischen Lösung läuft ab dem letzten Rücksetz-Zeitpunkt (zu dem ein Spike getriggert wurde) und  $v_0$  ist der initiale Potential-Wert zu dieser Zeit. Mit Hilfe der analytischen Lösung können wir das Konvergenzverhalten der Verfahren analysieren, also wie schnell sich der Fehler  $|e| = |v - v^h|$  zu einem bestimmten Zeitpunkt bei Verkleinerung der Zeitschrittweite reduziert.

1. Implementieren Sie das Forward Euler-, Backward Euler- und Crank-Nicolson-Verfahren für das Modell mit Octave und den Parametern  $R = 10$ ,  $\tau_m = 10$ ,  $v_L = -65$ ,  $v_{threshold} = -50$  und  $v_{reset} = -65$ . Der Startwert des Potentials sei  $v(0) = v_0 = v_L$ . Der Strom soll konstant sein,  $I(t) = I_0 = 2.0$ .

Messen Sie für alle drei Verfahren die Konvergenzrate, in dem Sie den Fehler zu einem festen Zeitpunkt betrachten (absolute Differenz der analytischen Lösung und des numerischen Potentials) und gleichzeitig die Zeitschrittweite  $\kappa$  halbieren, verwenden Sie  $\kappa = 0.1, 0.05, 0.025, 0.0125, 0.00625, 0.003125$ . Passen Sie auf, dass Sie immer zum selben Zeitpunkt auswerten, den Messzeitpunkt dürfen Sie aber beliebig wählen!

2. Exportieren Sie einen doppelt-logarithmische Plot „Fehler“ über „Zeitschrittweite“, in dem Sie die Werte aller drei Verfahren eintragen. Können Sie in den Plots die Konvergenzrate ablesen? Welche Konvergenzrate ergibt sich für das jeweilige Verfahren?

## Übung 7 Integrate-and-Fire Neuron mit Spike Frequency Adaption (10 Punkte)

In dieser Übung wird das Leaky Integrate-and-Fire-Modell um einen  $K^+$ -Kanal erweitert. Dieser erzeugt einen hyperpolarisierenden Strom, der das Spiken verzögert, und somit die Spike-Frequenz adaptieren kann. Daher wird dieses Modell IFA abgekürzt, „Integrate-and-Fire mit Adaption“. In physikalischer Schreibweise lautet es:

$$\begin{aligned} C_m \partial_t v(t) &= -G_L(v - v_{rest}) - g(v - E_K) + I/A, \\ \tau_a \partial_t g(t) &= -g \\ v(0) &= v_0, \quad g(0) = g_0, \\ \text{if } v \geq v_{thresh} &: \begin{cases} v \longrightarrow v_{reset}, \\ g \longrightarrow g + \Delta g. \end{cases} \end{aligned}$$

Hierbei ist  $I/A$  die Eingangstromdichte für eine Membran mit Gesamt-Oberfläche  $A$  [ $cm^2$ ],  $C_m$  die spezifische Membran-Kapazität (Einheit [ $\mu F/cm^2$ ]),  $G_L$  die spezifische Membran-Leitfähigkeit (Einheit [ $1/(k\Omega cm^2)$ ]),  $g$  die Variable des Kalium-Kanals (hier eine spezifische Leitfähigkeit in [ $1/(k\Omega cm^2)$ ]),  $E_K$  die Kalium-Batterie in [ $mV$ ],  $\tau_a$  die Zeitkonstante des Kalium-Kanals in [ $ms$ ] und  $\Delta g$  ein Offset für die Kalium-Variable. Zu jeder Zeit, an dem ein Spike erzeugt wird, erhöht der Offset die Kalium-Leitfähigkeit. Der hemmende Strom akkumuliert sich so auf und verzögert das Spiking.

Man kann nun mit dem spezifischen Membran-Widerstand  $R_m = 1/G_L$  multiplizieren,  $g$  reskalieren und erhält mit den Startbedingungen  $v(0) = v_0$  und  $g(0) = g_0$ :

$$\begin{aligned} \tau_m \partial_t v(t) &= -(v - v_{rest}) - g(v - E_K) + R I, \\ \tau_a \partial_t g(t) &= -g \\ \text{if } v \geq v_{thresh} &: \begin{cases} v \longrightarrow v_{reset}, \\ g \longrightarrow g + \frac{\Delta g}{G_L}. \end{cases} \end{aligned}$$

mit der Membran-Zeitkonstante  $\tau_m = C_m \cdot R_m$  in [ $ms$ ] und dem gesamten Membran-Widerstand  $R = R_m/A$  in [ $M\Omega$ ]. Der Strom ist nun der gesamte in die Zelle applizierte Strom in [ $nA$ ].

Implementieren Sie das (reskalierte) Modell mit Hilfe von *Octave* und einem Verfahren Ihrer Wahl. Verwenden Sie die Parameter  $t = 1000 ms$ ,  $dt = 0.1 ms$ ,  $v_{rest} = -65 mV$ ,  $R = 10.0$ ,  $v_{reset} = -65 mV$ ,  $v_{thresh} = -50.0 mV$ ,  $\tau = 15 ms$ . Startwerte sind  $v_0 = v_{rest}$  und  $g_0 = 0.0$ . Der Kalium-Kanal hat eine Batterie von  $E_K = -85 mV$  und von  $t = 50 ms$  bis  $t = 200 ms$  wird ein Strom von  $I = 4 nA$  appliziert.

Erzeugen Sie dann Plots des Potentials und der Kalium-Variablen  $g$  über der Zeit  $t$  für die Parameter-Kombinationen  $\{\tau_a, \Delta g/G_L\} = \{100, 0.1\}$ ,  $\{100, 0.5\}$ ,  $\{500, 0.1\}$ ,  $\{500, 0.5\}$  und versuchen Sie knapp zu interpretieren, wie sich das Ändern der Zeitkonstante  $\tau_a$  des Kalium-Kanals und des Kalium Offsets  $\Delta g$  qualitativ auswirken.