

Periodische Strukturen im invarianten Maß von stochastischen Hodgkin-Huxley-Prozessen

von

Sandra Martina Kozuschek

Masterarbeit in Mathematik
vorgelegt dem Fachbereich Physik, Mathematik und Informatik (FB 08)
der Johannes Gutenberg-Universität Mainz
am 8. August 2019

Ich versichere, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Mainz, den 8. August 2019

Sandra Martina Kozuschek

Institut für Mathematik
Staudingerweg 9
Johannes Gutenberg-Universität
D-55099 Mainz

Inhaltsverzeichnis

| | |
|---------------------------------------------------------------------------------------------|-----------|
| Einleitung | 6 |
| 1. Grundlagen | 8 |
| 1.1. Das Hodgkin-Huxley-Modell | 8 |
| 1.2. Biologische Bedeutung | 10 |
| 1.3. Mathematische Eigenschaften | 14 |
| 1.4. Simulation | 18 |
| 2. Ergebnisse | 23 |
| 2.1. Beispiel mit regelmäßigen Spikes | 24 |
| 2.1.1. Abbildung von $X = (V, n, m, h, \xi)$ in Abhängigkeit von t | 24 |
| 2.1.2. Projektionen von X auf zweidimensionale Ebenen . . | 25 |
| 2.1.3. Theorie zu Kerndichteschätzern | 27 |
| 2.1.4. Kerndichteschätzer für ψ_s zu 26 Zeitpunkten im Torus | 33 |
| 2.1.5. Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig | 37 |
| 2.2. Beispiel mit unregelmäßigen Spikes | 39 |
| 2.2.1. Abbildung von $X = (V, n, m, h, \xi)$ in Abhängigkeit von t | 39 |
| 2.2.2. Projektionen von X auf zweidimensionale Ebenen . . | 40 |
| 2.2.3. Kerndichteschätzer für ψ_s zu 26 Zeitpunkten im Torus | 42 |
| 2.2.4. Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig | 45 |
| 2.3. Beispiel mit Dirichletkern als Input | 47 |
| 2.3.1. Abbildung von $X = (V, n, m, h, \xi)$ in Abhängigkeit von t | 47 |
| 2.3.2. Projektionen von X auf zweidimensionale Ebenen . . | 50 |
| 2.3.3. Kerndichteschätzer für ψ_s zu 26 Zeitpunkten im Torus | 52 |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| 2.3.4. Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig | 57 |
| 2.4. Bezug zum deterministischen Hodgkin-Huxley-Modell | 59 |
| 2.4.1. Beispiel mit regelmäßigen Spikes | 60 |
| 2.4.2. Beispiel mit Dirichletkern als Input | 63 |
| Zusammenfassung und Ausblick | 66 |
| A. Quellcodes | 68 |
| A.1. Ratingfunktionen | 68 |
| A.2. Mögliche Inputfunktionen | 69 |
| A.3. Driftfunktionen | 70 |
| A.4. Simulation des stochastischen Hodgkin-Huxley-Modells und Darstellung der Lösung in Abhängigkeit von t | 71 |
| A.5. Simulation des stochastischen Hodgkin-Huxley-Modells und Darstellung der Lösung in Form von Projektionen auf zweidi- mensionale Ebenen | 73 |
| A.6. Simulation des stochastischen Hodgkin-Huxley-Modells und Dichteschätzung zu einer festen Toruszeit | 76 |
| A.7. Simulation des stochastischen Hodgkin-Huxley-Modells und Dichteschätzung zu vielen Toruszeiten gleichzeitig | 79 |
| A.8. Deterministische Inputfunktion | 83 |
| A.9. Simulation des deterministischen Hodgkin-Huxley-Modells und Darstellung der Lösung in Abhängigkeit von t | 84 |
| A.10. Simulation des deterministischen Hodgkin-Huxley-Modells und Darstellung der Lösung in Form von Projektionen auf zweidi- mensionale Ebenen | 85 |
| B. Literaturverzeichnis | 89 |

Abbildungsverzeichnis

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1.1. Aufbau eines Neurons | 11 |
| 1.2. Zustände von bewachten Kanälen | 12 |
| 1.3. Spike im Lösungsprozess des stochastischen Hodgkin-Huxley-Modells mit 10-periodischer Sinusfunktion als Input, $a = 12$, $\tau = 0,5$, $\gamma = 0,5$ bei drei beobachteten Perioden | 18 |
| 2.1. Regelmäßige Spikes: Werte von V , (n, m, h) und ξ in Abhängigkeit von t für die ersten 30 Periodizitätsintervalle des Inputs | 25 |
| 2.2. Regelmäßige Spikes: Projektionen von $X = (V, n, m, h, \xi)$ auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h) für die letzten fünf Periodizitätsintervalle des Outputs | 26 |
| 2.3. Regelmäßige Spikes: Kerndichteschätzer für ψ_s zur Toruszeit s_j , $j = 0, 1, \dots, 25$, Projektion auf (V, n) | 36 |
| 2.4. Regelmäßige Spikes: Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig, Projektionen auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h) | 38 |
| 2.5. Unregelmäßige Spikes: Werte von V , (n, m, h) und ξ in Abhängigkeit von t für die ersten 30 Periodizitätsintervalle des Inputs | 40 |
| 2.6. Unregelmäßige Spikes: Projektionen von $X = (V, n, m, h, \xi)$ auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h) für die letzten fünf Periodizitätsintervalle des Outputs | 41 |
| 2.7. Unregelmäßige Spikes: Kerndichteschätzer für ψ_s zur Toruszeit s_j , $j = 0, 1, \dots, 25$, Projektion auf (V, n) | 45 |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.8. Unregelmäßige Spikes: Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig, Projektionen auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h) | 46 |
| 2.9. Dirichletkern mit $a = 6$ als Input: Werte von V , (n, m, h) und ξ in Abhängigkeit von t für die ersten 20 Periodizitätsintervalle des Inputs | 47 |
| 2.10. Dirichletkern mit $a = 4$ als Input: Werte von V , (n, m, h) und ξ in Abhängigkeit von t für die ersten 20 Periodizitätsintervalle des Inputs | 49 |
| 2.11. Dirichletkern mit $a = 7$ als Input: Werte von V , (n, m, h) und ξ in Abhängigkeit von t für die ersten 20 Periodizitätsintervalle des Inputs | 49 |
| 2.12. Dirichletkern mit $a = 6$ als Input: Projektionen von $X = (V, n, m, h, \xi)$ auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h) für die letzten fünf Periodizitätsintervalle des Outputs | 50 |
| 2.13. Dirichletkern mit $a \in \{4, 5, 6, 7\}$ als Input: Projektionen von X auf (n, m) für die letzten 95 Periodizitätsintervalle des Inputs | 51 |
| 2.14. Dirichletkern mit $a = 6$ als Input: Kerndichteschätzer für ψ_s zur Toruszeit s_j , $j = 0, 1, \dots, 25$, Projektion auf (V, n) | 55 |
| 2.15. Dirichletkern mit $a \in \{4, 5, 6, 7\}$ als Input: Kerndichteschätzer für ψ_s zur Toruszeit s_4 , Projektion auf (V, n) | 56 |
| 2.16. Dirichletkern mit $a = 6$ als Input: Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig, Projektionen auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h) | 58 |
| 2.17. Deterministisches Modell mit regelmäßigen Spikes: Werte von V , (n, m, h) und aR' in Abhängigkeit von t für die ersten 30 Periodizitätsintervalle des Inputs | 60 |
| 2.18. Regelmäßige Spikes: Vergleich des stochastischen und des deterministischen Modells anhand von Projektionen von X auf (V, n) , (n, m) und (m, h) für die letzten 50 Periodizitätsintervalle des Outputs | 62 |

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.19. Deterministisches Modell mit Dirichletkern als Input: Werte von V , (n, m, h) und aR' in Abhängigkeit von t für die ersten 20 Periodizitätsintervalle des Inputs | 63 |
| 2.20. Dirichletkern als Input: Vergleich des stochastischen und des deterministischen Modells anhand von Projektionen von X auf (V, n) , (n, m) und (m, h) für die letzten 50 Periodizitätsintervalle des Outputs | 65 |

Einleitung

Das 1952 von Alan L. Hodgkin und Andrew F. Huxley entwickelte Hodgkin-Huxley-Modell ist bis heute eines der berühmtesten und meistgenutzten Modelle zur Simulation von Neuronen. Es beschreibt das Membranpotential und das Öffnen beziehungsweise Schließen von Ionenkanälen in der Zellmembran durch ein vierdimensionales System von gewöhnlichen Differentialgleichungen in Abhängigkeit von einem deterministischen Strominput. Für das stochastische Hodgkin-Huxley-Modell wird dieser Input durch einen sogenannten rauschenden Input ersetzt, der durch eine fünfte (stochastische) Differentialgleichung modelliert wird.

In dieser Arbeit wird das stochastische Hodgkin-Huxley-Modell mit einem T -periodischen Inputsignal mit Amplitudenfaktor $a > 0$ betrachtet. Bezeichnet man den Lösungsprozess mit $X = (X_t)_{t \geq 0}$ und die zugehörige Übergangshalbgruppe mit $P_{s,t}(x, dy)$, $0 \leq s < t < \infty$, so besitzt die zeitlich homogene Markovkette $(X_{kT})_{k \in \mathbb{N}_0}$ ein eindeutiges invariantes Wahrscheinlichkeitsmaß μ . Es ist jedoch nicht bekannt, wie μ konkret aussieht. Ziel dieser Arbeit ist es nun, den Prozess X zu simulieren und die zeitliche Dynamik $[0, T] \ni s \mapsto [\mu P_{0,s}]$, die sich in Abhängigkeit von a verändert, zu veranschaulichen.

Den theoretischen Hintergrund für diese Herangehensweise haben Reinhard Höpfner, Eva Löcherbach und Michèle Thieullen in den beiden 2016 veröffentlichten Artikeln „*Ergodicity and limit theorems for degenerate diffusions with time periodic drift. Application to a stochastic Hodgkin-Huxley model.*“ und „*Ergodicity for a stochastic Hodgkin-Huxley model driven by Ornstein-Uhlenbeck type input*“ (siehe [10] und [11]) geliefert.

Im ersten Kapitel dieser Arbeit werden zunächst das deterministische und das stochastische Hodgkin-Huxley-Modell eingeführt und deren biologische Bedeutung beschrieben. Anschließend werden mathematische Eigenschaften

des stochastischen Modells angegeben, die für das nächste Kapitel benötigt werden. Außerdem werden die Verfahren zur Simulation der Modelle erklärt, die den Ergebnissen aus Kapitel 2 zugrunde liegen.

Im zweiten Kapitel wird dann die zu untersuchende zeitliche Dynamik anhand von drei Beispielen anschaulich gemacht. Dafür werden verschiedene grafische Darstellungsarten zum Einsatz kommen und der Einfluss der einzelnen Parameter des stochastischen Hodgkin-Huxley-Modells soll deutlich werden. Zum Abschluss des Kapitels wird ein Zusammenhang zwischen dem stochastischen und dem deterministischen Modell thematisiert.

1. Grundlagen

1.1. Das Hodgkin-Huxley-Modell

Das deterministische Hodgkin-Huxley-Modell ist eines der wichtigsten Modelle der Neurowissenschaften. Es geht zurück auf Experimente, die Alan L. Hodgkin und Andrew F. Huxley an der Zellmembran des Riesenaxons des Tintenfischs durchgeführt haben und deren Ergebnisse sie 1952 in [7] veröffentlicht haben. Eine modernere Einführung in dieses Thema bietet das Buch „*Dynamical Systems in Neuroscience*“ von Eugene M. Izhikevich (siehe [16]). Da wir das Hodgkin-Huxley-Modell zunächst nur mathematisch betrachten wollen, basiert dieser Abschnitt auf [10, Section 2.3] und [3, Section 1.1].

Definition 1.1. Die Funktionen $\alpha_n, \alpha_m, \alpha_h, \beta_n, \beta_m: \mathbb{R} \rightarrow \mathbb{R}$ und $\beta_h: \mathbb{R} \rightarrow (0, 1)$ gegeben durch

$$\begin{aligned} \alpha_n(V) &:= \begin{cases} \frac{10-V}{100(\exp(\frac{10-V}{10})-1)}, & \text{falls } V \neq 10 \\ \frac{1}{10}, & \text{sonst} \end{cases}, & \beta_n(V) &:= \frac{1}{8} \exp\left(\frac{-V}{80}\right), \\ \alpha_m(V) &:= \begin{cases} \frac{25-V}{10(\exp(\frac{25-V}{10})-1)}, & \text{falls } V \neq 25 \\ 1, & \text{sonst} \end{cases}, & \beta_m(V) &:= 4 \exp\left(\frac{-V}{18}\right), \\ \alpha_h(V) &:= \frac{7}{100} \exp\left(\frac{-V}{20}\right), & \beta_h(V) &:= \frac{1}{\exp\left(\frac{30-V}{10}\right) + 1} \end{aligned}$$

heißen *Ratingfunktionen*.

Definition 1.2. Die Konstanten

$$E_K = -12, \quad E_{Na} = 120, \quad E_L = 10,6$$

heißen *Gleichgewichtspotentiale* und die Konstanten

$$g_K = 36, \quad g_{Na} = 120, \quad g_L = 0,3$$

sind die *maximalen Leitfähigkeiten*.

Definition 1.3. Sei $I: \mathbb{R}_+ \rightarrow \mathbb{R}$, $t \mapsto I(t)$ eine vorgegebene Inputfunktion. Das *deterministische Hodgkin-Huxley-Modell* ist das folgende vierdimensionale Differentialgleichungssystem:

$$\begin{cases} dV_t = I(t) dt - [g_K n_t^4 (V_t - E_K) + g_{Na} m_t^3 h_t (V_t - E_{Na}) + g_L (V_t - E_L)] dt \\ dn_t = [\alpha_n(V_t)(1 - n_t) - \beta_n(V_t)n_t] dt \\ dm_t = [\alpha_m(V_t)(1 - m_t) - \beta_m(V_t)m_t] dt \\ dh_t = [\alpha_h(V_t)(1 - h_t) - \beta_h(V_t)h_t] dt \end{cases}$$

Der Prozess $V = (V_t)_{t \geq 0}$ heißt *Membranpotential* und die Funktionen $n = (n_t)_{t \geq 0}$, $m = (m_t)_{t \geq 0}$ und $h = (h_t)_{t \geq 0}$ heißen *Gatingvariablen*.

Definition 1.4. Sei $S: \mathbb{R}_+ \rightarrow \mathbb{R}$, $t \mapsto S(t)$ eine deterministische, unendlich oft differenzierbare, 1-periodische Inputfunktion. Außerdem seien $T > 0$ eine vorgegebene Periodizität, $a > 0$ ein Amplitudenfaktor, $\tau > 0$ und $\gamma > 0$ Konstanten und $W = (W_t)_{t \geq 0}$ eine eindimensionale Brown'sche Bewegung. Das *stochastische Hodgkin-Huxley-Modell* ist das folgende fünfdimensionale Differentialgleichungssystem:

$$\begin{cases} dV_t = d\xi_t - [g_K n_t^4 (V_t - E_K) + g_{Na} m_t^3 h_t (V_t - E_{Na}) + g_L (V_t - E_L)] dt \\ dn_t = [\alpha_n(V_t)(1 - n_t) - \beta_n(V_t)n_t] dt \\ dm_t = [\alpha_m(V_t)(1 - m_t) - \beta_m(V_t)m_t] dt \\ dh_t = [\alpha_h(V_t)(1 - h_t) - \beta_h(V_t)h_t] dt \\ d\xi_t = \left[aS\left(\frac{t}{T}\right) - \xi_t \right] \tau dt + \gamma \sqrt{\tau} dW_t \end{cases} \quad (1.1)$$

Der stochastische Prozess $\xi = (\xi_t)_{t \geq 0}$ heißt *rauschender Input*.

Während im deterministischen Hodgkin-Huxley-Modell nur gewöhnliche Differentialgleichungen erster Ordnung vorkommen, handelt es sich bei der ersten

und fünften Gleichung im stochastischen Hodgkin-Huxley-Modell um stochastische Differentialgleichungen. Die fünfte Gleichung befindet sich dabei in Ornstein-Uhlenbeck-Form. Die Konstante τ steuert, mit welcher Geschwindigkeit der Prozess ξ der T -periodischen Inputfunktion $aS(\frac{\cdot}{T})$ folgt, und γ ist verantwortlich für die Streuung von ξ . Somit ersetzen im stochastischen Modell die Inkremente von ξ den deterministischen Input I aus der ersten Gleichung des deterministischen Modells und das durch die Brown'sche Bewegung verursachte Rauschen wirkt sich deshalb auf das Membranpotential aus.

Man könnte die fünfte Gleichung im stochastischen Modell auch deutlich allgemeiner formulieren, aber wir haben uns hier auf die Variante des stochastischen Hodgkin-Huxley-Modells beschränkt, die im weiteren Verlauf dieser Arbeit relevant sein wird.

1.2. Biologische Bedeutung

Dieser Abschnitt soll einen kurzen Einblick in die neurologische Bedeutung des Hodgkin-Huxley-Modells bieten, für Details zu diesem Thema wird jedoch auf [16, Chapter 2] verwiesen.

Um das Modell besser zu verstehen, lohnt es sich, sich zunächst mit dem Aufbau von Neuronen und den dort ablaufenden Ionenströmen zu beschäftigen. Ein Neuron besteht aus Dendriten, Soma, Axon und Synapsen (siehe Abbildung 1.1). Die Dendriten sind dabei mit den Synapsen von mehr als 10000 anderen Neuronen verbunden, von denen sie Signale erhalten, welche im Soma gesammelt werden und über das Axon und die Synapsen wiederum an viele andere Neuronen weitergegeben werden.

In den Nervenzellen breitet sich elektrische Aktivität anhand von Ionenströmen durch die Zellmembran aus. Dies geschieht mit Hilfe von großen Proteinmolekülen in der Zellmembran, die Kanäle für die Ionen bilden. Der Auslöser für diese Ionenströme sind die unterschiedlichen Konzentrationen der Ionen im Inneren und außerhalb der Zelle. Die Konzentration von K^+ -Ionen ist beispielsweise im Inneren der Zelle höher als außerhalb. Deshalb verlassen zunächst viele K^+ -Ionen die Zelle, wodurch sich positive Ladung

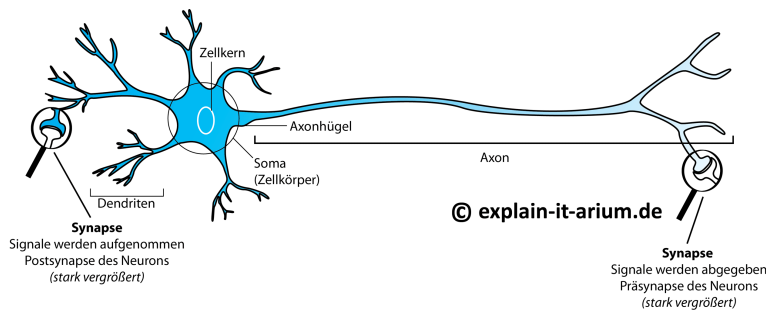


Abbildung 1.1.: Aufbau eines Neurons (von [21])

außerhalb der Zelle ansammelt und negative Ladung im Inneren zurückbleibt. Es entsteht eine elektrische Spannung durch die Membran hindurch, die Membranpotential genannt wird. Diese Spannung verursacht jedoch einen nach innen gerichteten Ionenfluss, denn die K^+ -Ionen werden von der negativen Ladung im Inneren der Zelle angezogen, wodurch irgendwann ein Gleichgewicht der K^+ -Ionenströme entsteht. Das dazugehörige Membranpotential wird Gleichgewichtspotential genannt. Dieses Gleichgewichtspotential hängt von der jeweiligen Ionenart ab und kann durch die Nernst-Gleichung (siehe [16, Equation (2.1)])

$$E_{\text{Ion}} = \frac{RT}{zF} \ln \left(\frac{[\text{Ion}]_{\text{out}}}{[\text{Ion}]_{\text{in}}} \right)$$

bestimmt werden, wobei $[\text{Ion}]_{\text{in}}$ und $[\text{Ion}]_{\text{out}}$ die Konzentrationen der Ionen im Inneren und außerhalb der Zelle sind, R die universelle Gaskonstante, T die Temperatur in Kelvin, F die Faraday-Konstante und z die Wertigkeit des Ions. Mit der Nernst-Gleichung lassen sich die Gleichgewichtspotentiale der wichtigsten Ionenarten (K^+ , Na^+ , Ca^{2+} und Cl^-), die an vielen Strömen durch die Zellmembran beteiligt sind, berechnen.

Der Nettostrom I_{Ion} einer Ionenart ist gegeben durch

$$I_{\text{Ion}} = g_{\text{Ion}} (V - E_{\text{Ion}}),$$

wobei V das Membranpotential, E_{Ion} das Gleichgewichtspotential und $g_{\text{Ion}} > 0$ die Leitfähigkeit der Zellmembran für die jeweilige Ionenart ist. Der Nettostrom

ist dementsprechend null, wenn das Membranpotential dem Gleichgewichtspotential entspricht.

Wenn in den Nervenzellen nur eine Art von Ionen wirken würde, würde sich das Membranpotential schnell dem Gleichgewichtspotential annähern. Da jedoch unterschiedliche Ionenströme stattfinden, die sich möglicherweise gegenseitig behindern, dauert es länger, bis sich ein Zustand einstellt, in dem sich alle nach innen und nach außen gerichteten Ströme gegenseitig ausgleichen. Dieser Zustand wird Ruhepotential genannt und in ihm ist der Nettostrom durch die Zellmembran null.

Die Leitfähigkeit der Proteinmoleküle in der Zellmembran, die als Kanäle für Ionen dienen, wird durch „Wächterpartikel“ kontrolliert. Diese Wächter entscheiden in Abhängigkeit vom Membranpotential, ob Kanäle offen oder geschlossen sind. Wenn man nun eine große Menge von Kanälen für eine einzelne Ionenart betrachtet, kann man den Nettostrom durch

$$I = gp(V - E)$$

beschreiben, wobei g die maximale Leitfähigkeit und E das Gleichgewichtspotential ist und p den Anteil der offenen Kanäle angibt. Es gibt Wächter, die dafür zuständig sind, Kanäle zu öffnen (activate) und solche, deren Aufgabe es ist, Kanäle zu schließen (inactivate) (siehe Abbildung 1.2).

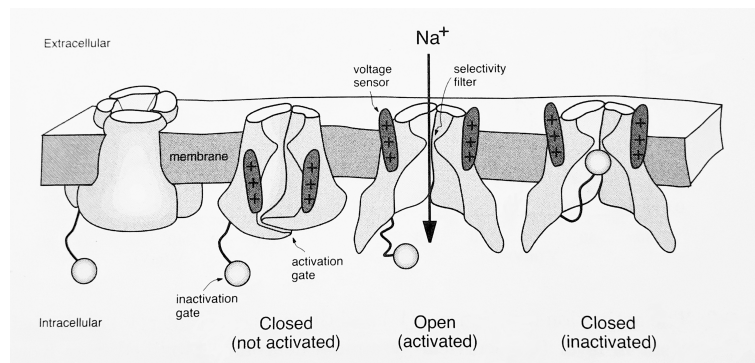


Abbildung 1.2.: Zustände von bewachten Kanälen (von [16, Figure 2.8])

Die Wahrscheinlichkeit, dass ein Aktivierungswächter (activation gate) einen Na^+ -Kanal offen lässt, wird durch die Variable m notiert. Handelt es sich um einen K^+ -Kanal, wird die Variable n verwendet. Die Wahrscheinlichkeit, dass ein Inaktivierungswächter (inactivation gate) einen beliebigen Ionenkanal offen lässt, wird durch die Variable h notiert. Der Anteil der offenen Na^+ -Kanäle ist damit

$$p = m^x h^y,$$

wobei x die Anzahl der Aktivierungswächter und y die Anzahl der Inaktivierungswächter pro Kanal ist. Ersetzt man m durch n , so erhält man den Anteil der offenen K^+ -Kanäle.

Durch Experimente fanden Alan L. Hodgkin und Andrew F. Huxley heraus, dass es im Riesenaxon des Tintenfischs drei wichtige Ionenströme gibt: den K^+ -Strom $I_K(t) = g_K n_t^4 (V_t - E_K)$ mit vier Aktivierungswächtern, den Na^+ -Strom $I_{Na}(t) = g_{Na} m_t^3 h_t (V_t - E_{Na})$ mit drei Aktivierungswächtern und einem Inaktivierungswächter und den sogenannten Leckstrom $I_L(t) = g_L (V_t - E_L)$, der hauptsächlich aus Cl^- -Ionen besteht. Diese drei Ionenströme findet man in der ersten Gleichung des deterministischen Hodgkin-Huxley-Modells aus Definition 1.3 wieder.

Die Parameter in den Ratingfunktionen aus Definition 1.1 passen zu einem um $+65$ mV verschobenen Membranpotential, sodass das Ruhepotential bei ungefähr $V = 0$ liegt. Die Ratingfunktionen beschreiben die Übergangsraten zwischen offenen und geschlossenen Zuständen der Ionenkanäle, da die Gatingvariablen n , m und h durch sie definiert sind.

Die Gleichgewichtspotentiale aus Definition 1.2 erhält man, wenn man experimentell ermittelte Ionenkonzentrationen in die Nernst-Gleichung einsetzt und die erhaltenen Gleichgewichtspotentiale um $+65$ mV verschiebt. Die maximalen Leitfähigkeiten aus Definition 1.2 sind ebenfalls experimentell ermittelt.

Die Einheiten der physikalischen Größen (Membranpotential in mV, Zeit in ms, Gleichgewichtspotentiale in mV, maximale Leitfähigkeiten in mS/cm^2 , Input in $\mu A/cm^2$) werden im weiteren Verlauf dieser Arbeit nicht erwähnt, da wir uns für die mathematischen Eigenschaften des Modells interessieren und sie dafür von geringer Bedeutung sind.

Die Motivation für das stochastische Hodgkin-Huxley-Modell aus Definition 1.4 ist, dass jedes Neuron zu einem aktiven Netzwerk gehört und seinen Input von vielen anderen Neuronen über einen stark verzweigten dendritischen Baum mit komplexer topologischer Struktur bekommt (siehe [11, Chapter 1]). Um das nachzubilden, wird der dendritische Input durch die Zuwächse des (von den ersten vier Gleichungen von (1.1)) unabhängigen, zeitlich inhomogenen stochastischen Prozesses ξ modelliert und es wird untersucht, wie sich ein externes, durch die Brown'sche Bewegung verursachtes Rauschen auf das Verhalten des gesamten Systems auswirkt.

1.3. Mathematische Eigenschaften

Wir wollen nun gewisse Eigenschaften des Lösungsprozesses des stochastischen Hodgkin-Huxley-Modells untersuchen, aber dafür muss zuerst sichergestellt werden, dass eine Lösung existiert.

Proposition 1.5. Sei $E_5 := \mathbb{R} \times [0, 1]^3 \times \mathbb{R}$. Für jeden Startpunkt $x \in E_5$ hat das Differentialgleichungssystem (1.1) eine eindeutige, starke, nicht explodierende Lösung $X = (X_t)_{t \geq 0} := (V_t, n_t, m_t, h_t, \xi_t)_{t \geq 0}$ mit Werten in E_5 .

Beweis. siehe [11, Proposition 1]

Als eindeutige Lösung einer stochastischen Differentialgleichung ist X ein starker Markovprozess und wir interessieren uns dafür, ob es eine invariante Verteilung gibt und wie diese aussehen könnte. Allerdings ist X zeitlich inhomogen, da der Drift in der ersten und fünften Gleichung von (1.1) abhängig von t ist, und in diesem Fall ist es deutlich komplizierter, den Begriff „invariante Verteilung“ zu definieren. Deshalb wird X modifiziert, um einen homogenen Markovprozess zu erhalten.

Sei $P_{s,t}(x, dy)$, $0 \leq s < t < \infty$, die zu X gehörige Übergangshalbgruppe auf $(E_5, \mathcal{B}(E_5))$. Da der Input $S(\frac{\cdot}{T})$ aus Definition 1.4, und somit der Drift von Gleichung (1.1), T -periodisch in der Zeit ist, gilt

$$P_{s,t}(x, dy) = P_{s+kT, t+kT}(x, dy), \quad k \in \mathbb{N}_0. \quad (1.2)$$

Daraus lässt sich das folgende Lemma ableiten (siehe [8, Lemma 2.1]):

- Lemma 1.6.** 1. Die Gitterkette $X^G = (X_k^G)_{k \in \mathbb{N}_0} := (X_{kT})_{k \in \mathbb{N}_0}$ ist ein E_5 -wertiger, zeitlich homogener Markovprozess in diskreter Zeit mit 1-Schritt-Übergangskern $P_{0,T}$.
2. Sei $i_T(t) := t \bmod T$ und $\mathbb{T} := [0, T]$. Der 6-dimensionale Prozess $\overline{X} = (\overline{X}_t)_{t \geq 0}$ mit $\overline{X}_t := (i_T(t), X_t)$ ist ein zeitlich homogener Markovprozess in stetiger Zeit mit Werten in $\overline{E}_5 := \mathbb{T} \times E_5$.

Für zeitlich homogene Markovprozesse ist die wichtige Eigenschaft der Harris-Rekurrenz (nach [19, Definition 3.5] bzw. [1, Section 1.1]) definiert:

- Definition 1.7.** 1. Sei $Y = (Y_k)_{k \in \mathbb{N}_0}$ eine homogene Markovkette mit Werten in einem polnischen Raum (E, \mathcal{E}) und Startverteilung P_x . Y ist *Harris-rekurrent*, wenn es ein σ -endliches Maß φ auf (E, \mathcal{E}) gibt, sodass für alle $F \in \mathcal{E}$ mit $\varphi(F) > 0$ gilt, dass

$$\limsup_{k \rightarrow \infty} \mathbb{1}_F(Y_k) = 1 \quad P_x\text{-fast sicher für alle } x \in E.$$

2. Sei $Y = (Y_t)_{t \geq 0}$ ein zeitlich homogener, starker Markovprozess mit Verteilungen $(P_x)_{x \in E}$ und Werten in einem polnischen Raum (E, \mathcal{E}) . Y ist *Harris-rekurrent*, wenn es ein σ -endliches Maß φ auf (E, \mathcal{E}) gibt, sodass für alle $F \in \mathcal{E}$ mit $\varphi(F) > 0$ gilt, dass

$$\int_0^\infty \mathbb{1}_F(Y_s) \, ds = \infty \quad P_x\text{-fast sicher für alle } x \in E.$$

Satz 1.8. Sei Y Harris-rekurrent (in diskreter oder in stetiger Zeit). Dann gibt es ein σ -endliches invariantes Maß π auf (E, \mathcal{E}) und φ aus Definition 1.7 kann durch π ersetzt werden. π ist eindeutig bestimmt bis auf Multiplikation mit einer Konstanten.

Beweis. siehe [19, Corollary 5.2] bzw. [1, Sections 2.4-2.5]

Definition 1.9. Ist Y Harris-rekurrent und $\pi(E) < \infty$, so heißt Y *positiv Harris-rekurrent*.

Bemerkung 1.10. Wenn Y positiv Harris-rekurrent ist, kann π durch Normierung zu einem invarianten Wahrscheinlichkeitsmaß gemacht werden.

Satz 1.11. Die Gitterkette X^G und der zeitstetige Prozess \bar{X} sind positiv Harris-rekurrent.

Beweis. siehe [10, Theorem 1 und 2]

Wegen Satz 1.11 weiß man nun, dass X^G und \bar{X} eindeutige invariante Verteilungen μ und $\bar{\mu}$ besitzen.

Proposition 1.12. Die folgenden Aussagen sind äquivalent:

1. μ ist die invariante Verteilung von X^G .
2. Für jedes $s \in \mathbb{T}$ ist das Maß $[\mu P_{0,s}]$ mit

$$[\mu P_{0,s}](A) = \int_{E_5} P_{0,s}(y, A) \mu(dy), \quad A \in \mathcal{B}(E_5)$$

die invariante Verteilung der Markovkette $(X_{kT+s})_{k \in \mathbb{N}_0}$.

3. Das Maß $\bar{\mu}$ mit

$$\bar{\mu}(A) = \frac{1}{T} \int_0^T (\delta_s \otimes [\mu P_{0,s}])(A) ds, \quad A \in \mathcal{B}(\bar{E}_5)$$

ist die invariante Verteilung von \bar{X} .

Beweis. siehe [11, Proposition 5]

Es ist nicht bekannt, wie μ , $[\mu P_{0,s}]$ und $\bar{\mu}$ konkret aussehen. Um darüber einen Eindruck zu gewinnen, wird uns später das folgende starke Gesetz der großen Zahlen (aus [10, Corollary 1]), das aus der Harris-Rekurrenz von X^G und \bar{X} folgt, behilflich sein:

Korollar 1.13. Sei Q_x die Verteilung von $(X_t)_{t \geq 0}$ bei Start in $x \in E_5$ als Wahrscheinlichkeitsmaß auf dem Raum der Pfade $C([0, \infty), E_5)$. Seien $F, \tilde{F}: E_5 \rightarrow [0, \infty)$ $\mathcal{B}(E_5)$ -messbare Funktionen. Dann gilt

$$\frac{1}{n} \sum_{k=0}^{n-1} F(X_{kT}) \xrightarrow{n \rightarrow \infty} \mu(F) := \int_{E_5} F(y) \mu(dy)$$

Q_x -fast sicher für alle $x \in E_5$. Außerdem gilt für festes $s \in \mathbb{T}$

$$\frac{1}{n} \sum_{k=0}^{n-1} \tilde{F}(X_{kT+s}) \xrightarrow{n \rightarrow \infty} [\mu P_{0,s}](\tilde{F}) := \int_{E_5} \tilde{F}(y) [\mu P_{0,s}](dy)$$

Q_x -fast sicher für alle $x \in E_5$.

Sei $G: \overline{E_5} \rightarrow [0, \infty)$ eine $\mathcal{B}(\overline{E_5})$ -messbare Funktion. Dann gilt

$$\frac{1}{t} \int_0^t G(i_T(s), X_s) ds \xrightarrow{t \rightarrow \infty} \frac{1}{T} \int_0^T \int_{E_5} G(s, y) [\mu P_{0,s}](dy) ds$$

Q_x -fast sicher für alle $x \in E_5$.

Betrachtet man Simulationen von Hodgkin-Huxley-Lösungsprozessen, fallen dabei schnell die sogenannten *Aktionspotentiale* oder *Spikes* auf. Dabei handelt es sich um lokale Maxima des Membranpotentials V (siehe Abbildung 1.3). Es ist jedoch nicht möglich, nur anhand der Werte von V zu bestimmen, wann ein Spike beginnt und endet, denn es gibt beispielsweise keinen festen V -Schwellenwert dafür. Um das Auftreten eines Spikes zu definieren, benötigt man deshalb die Werte der Gatingvariablen m und h .

Definition 1.14. Der Lösungsprozess X befindet sich zur Zeit $t \geq 0$ *innerhalb eines Spikes*, wenn

$$X_t \in F_{sp} := \{x = (v, n, m, h, \xi) \in E_5 \mid m > h\}$$

und *zwischen zwei Spikes*, wenn

$$X_t \in F_b := \{x = (v, n, m, h, \xi) \in E_5 \mid m < h\}.$$

Satz 1.15. Für jeden Startpunkt $x \in E_5$ gilt:

1. $Q_x(\text{Es gibt unendlich viele Spikes}) = 1$
2. $Q_x(\text{Es gibt unendlich viele Intervalle } [kT, (k+1)T), \text{ in denen keine Spikes auftreten}) = 1$

Beweis. siehe [10, Theorem 3]

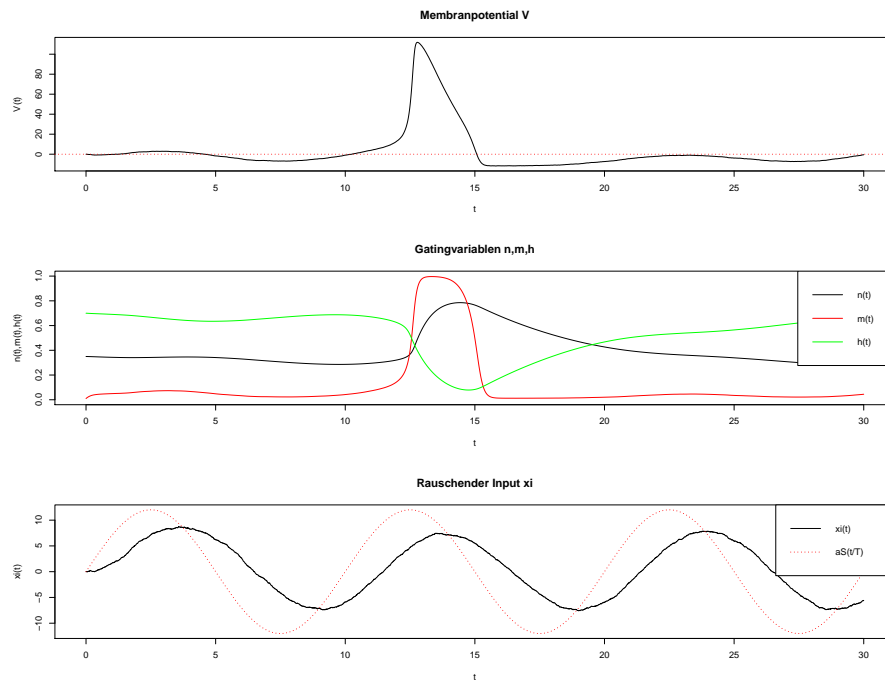


Abbildung 1.3.: Spike im Lösungsprozess des stochastischen Hodgkin-Huxley-Modells mit 10-periodischer Sinusfunktion als Input, $a = 12$, $\tau = 0,5$, $\gamma = 0,5$ bei drei beobachteten Perioden

1.4. Simulation

Zur Simulation des stochastischen und des deterministischen Hodgkin-Huxley-Modells wurde die Programmiersprache *R* (<https://www.r-project.org/>) verwendet, denn *R* ist auf stochastische Problemstellungen ausgerichtet und bietet außerdem viele Möglichkeiten zur Erzeugung von Grafiken. Die Programmcodes aus Anhang A wurden deshalb mit Hilfe der grafischen Benutzeroberfläche RStudio (<https://www.rstudio.com/>) geschrieben. Zusätzlich zu der Aufführung im Anhang findet man alle Codes auf der beigelegten CD. Dort befinden sich auch die Abbildungen aus dieser Arbeit und einige zusätzliche Bilder.

Um das Hodgkin-Huxley-Modell simulieren zu können, werden sowohl bei der stochastischen als auch bei der deterministischen Variante als Erstes die Ratingfunktionen aus Definition 1.1 implementiert (siehe A.1).

Danach werden in A.2 mögliche 1-periodische, unendlich oft differenzierbare Inputfunktionen für S aus Definition 1.4 erstellt. Die einfachste Wahl ist dabei

$$S(t) = \sin(2\pi t), \quad t \geq 0,$$

aber wir wollen auch sogenannte Dirichletkerne als Input betrachten. Als n -ten *Dirichletkern* bezeichnet man (nach [6, Abschnitt X.2]) die Funktion

$$D_n(t) := \sum_{k=-n}^n e^{ikt} = 1 + 2 \sum_{k=1}^n \cos(kt).$$

Wir verwenden

$$S(t) = D_n(2\pi t), \quad t \geq 0$$

mit $n \in \{2, 3, 4\}$ als möglichen Input.

Zur Simulation des stochastischen Hodgkin-Huxley-Modells wird das Euler-Maruyama-Schema verwendet (analog zu der Beschreibung in [2, Abschnitt 2.2]). Dies ist ein Näherungsverfahren für die Lösung einer stochastischen Differentialgleichung

$$dX_t = b(t, X_t)dt + \sigma(t, X_t) dW_t, \quad X_0 = x_0$$

mit Startwert $x_0 \in \mathbb{R}^d$, Drift $b: \mathbb{R}_+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, Diffusion $\sigma: \mathbb{R}_+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ und einer eindimensionalen Brown'schen Bewegung $(W_t)_{t \geq 0}$. Für eine festzulegende Schrittweite $\Delta > 0$ und $t_{\text{end}} = n\Delta$ mit $n \in \mathbb{N}$ wird die Approximation von $(X_t)_{t \in [0, t_{\text{end}}]}$ durch

$$\begin{aligned} X_0 &= x_0 \\ X_{(i+1)\Delta} &= X_{i\Delta} + b(i\Delta, X_{i\Delta}) \Delta + \sigma(i\Delta, X_{i\Delta}) \zeta_i, \quad i = 0, \dots, n-1 \end{aligned} \tag{1.3}$$

berechnet, wobei $\zeta_i := W_{(i+1)\Delta} - W_{i\Delta}$ unabhängige, normalverteilte Zufalls-

variablen mit Erwartungswert 0 und Varianz Δ sind.

Dieses Näherungsverfahren verwenden wir, um das stochastische Hodgkin-Huxley-Modell auf einem ganzzahligen Vielfachen der Periodizität T der Inputfunktion $S(\frac{\cdot}{T})$ zu simulieren. Für die Bestimmung der Schrittweite Δ werden ein oder mehrere Periodizitätsintervalle der Länge T in E Abschnitte unterteilt (z.B. $E = 2500$).

Um b und σ aus Gleichung (1.1) abzulesen, sei zunächst

$$F(V_t, n_t, m_t, h_t) := g_K n_t^4 (V_t - E_K) + g_{Na} m_t^3 h_t (V_t - E_{Na}) + g_L (V_t - E_L). \quad (1.4)$$

Die Implementierung von F wird in A.3, Zeile 3-19 angegeben.

Der Driftkoeffizient b von (1.1) ist nun für $X = (X_t)_{t \geq 0} = (V_t, n_t, m_t, h_t, \xi_t)_{t \geq 0}$ gegeben durch

$$b(t, X_t) = \begin{pmatrix} b_V(t, X_t) \\ b_n(V_t, n_t) \\ b_m(V_t, m_t) \\ b_h(V_t, h_t) \\ b_\xi(t, \xi_t) \end{pmatrix},$$

wobei

$$\begin{aligned} b_V(t, X_t) &:= \left[aS\left(\frac{t}{T}\right) - \xi_t \right] \tau - F(V_t, n_t, m_t, h_t), \\ b_j(V_t, j_t) &:= \alpha_j(V_t)(1 - j_t) - \beta_j(V_t)j_t, \quad j \in \{n, m, h\}, \\ b_\xi(t, \xi_t) &:= \left[aS\left(\frac{t}{T}\right) - \xi_t \right] \tau. \end{aligned}$$

Der Diffusionskoeffizient σ von (1.1) lautet

$$\sigma(X_t) = \begin{pmatrix} \gamma\sqrt{\tau} \\ 0 \\ 0 \\ 0 \\ \gamma\sqrt{\tau} \end{pmatrix}.$$

Die Implementierung der Funktionen b_j und b_ξ findet man in A.3, Zeile 21-43.

Aufgrund des Zusammenhangs $b_V = b_\xi - F$ wird b_V zusammen mit σ direkt in das Euler-Maruyama-Schema aufgenommen. Dort (siehe A.4, Zeile 40-53) werden schließlich die Werte von $(\xi_t)_{t \geq 0}$, $(V_t)_{t \geq 0}$ und $(n_t, m_t, h_t)_{t \geq 0}$ durch

$$\begin{aligned}\xi_{(i+1)\Delta} &= \xi_{i\Delta} + b_\xi(i\Delta, \xi_{i\Delta})\Delta + \gamma\sqrt{\tau}\zeta_i, \\ V_{(i+1)\Delta} &= V_{i\Delta} + b_V(i\Delta, X_{i\Delta})\Delta + \gamma\sqrt{\tau}\zeta_i, \\ j_{(i+1)\Delta} &= j_{i\Delta} + b_j(V_{i\Delta}, j_{i\Delta})\Delta, \quad j \in \{n, m, h\}\end{aligned}$$

näherungsweise berechnet.

Als Startwert für die Simulation wurde

$$x_0 = \begin{pmatrix} V_0 \\ n_0 \\ m_0 \\ h_0 \\ \xi_0 \end{pmatrix} := \begin{pmatrix} 0 \\ 0,35 \\ 0,01 \\ 0,7 \\ 0 \end{pmatrix} \quad (1.5)$$

verwendet. Bei V_0 handelt es sich dabei um das Ruhepotential und n_0 , m_0 und h_0 sind die Werte, die die Gatingvariablen außerhalb eines Spikes in etwa annehmen. Die Startwerte verlieren jedoch bei hinreichend langem Lauf der Simulation an Bedeutung.

Die Simulation des deterministischen Hodgkin-Huxley-Modells, das wir in Abschnitt 2.4 genauer untersuchen werden, verläuft größtenteils analog zur Simulation des stochastischen Modells.

Bei der speziellen Version des deterministischen Modells, mit der wir uns beschäftigen werden, wird die Inputfunktion S aus dem stochastischen Modell modifiziert und dann als Input für das deterministische Modell verwendet. Die Implementierung dieser Modifikation findet man in A.8.

Da das deterministische Hodgkin-Huxley-Modell aus gewöhnlichen Differentialgleichungen besteht, wird für die Simulation das explizite Euler-Verfahren (nach [5, Abschnitt 74]) verwendet. Dies ist ein numerisches Lösungsverfahren

für Anfangswertprobleme

$$y' = f(t, y), \quad y(0) = y_0$$

mit $f: \mathbb{R}_+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ und Startwert $y_0 \in \mathbb{R}^d$. Für eine Schrittweite $\Delta > 0$ und $t_{\text{end}} = n\Delta$ mit $n \in \mathbb{N}$ wird durch

$$\begin{aligned} y(0) &= y_0 \\ y((i+1)\Delta) &= y(i\Delta) + f(i\Delta, y(i\Delta))\Delta, \quad i = 0, \dots, n-1 \end{aligned} \tag{1.6}$$

eine Näherung an $y(t)$, $t \in [0, t_{\text{end}}]$ berechnet. Beim Vergleich von (1.3) und (1.6) ist erkennbar, dass das explizite Euler-Verfahren die Basis für das zuvor verwendete Euler-Maruyama-Schema bildet.

Die Implementierung des Verfahrens findet man in A.9, Zeile 35-45. Dabei wurden das Ende des Beobachtungszeitraums t_{end} , die Schrittweite Δ und die Startwerte (V_0, n_0, m_0, h_0) genauso gewählt wie bei der Simulation des stochastischen Modells.

2. Ergebnisse

Das Ziel dieses Kapitels, das den Hauptteil der Arbeit darstellt, ist es, sich empirisch einen Eindruck von den bereits in Abschnitt 1.3 eingeführten invarianten Verteilungen μ , $[\mu P_{0,s}]$ und $\bar{\mu}$ zu verschaffen. Insbesondere soll dabei die zeitliche Dynamik

$$\mathbb{T} \ni s \mapsto [\mu P_{0,s}]$$

anschaulich gemacht werden. Die Zeit wird in dem von uns betrachteten Modell demnach als Torus $\mathbb{T} = [0, T]$ mit einer gewissen Anzahl von Durchläufen aufgefasst.

Da $[\mu P_{0,s}]$ ein Maß auf dem fünfdimensionalen Raum E_5 ist, bietet es sich an, Projektionen von $X = (V, n, m, h, \xi)$ auf zweidimensionale Räume zu verwenden. Wir betrachten beispielsweise

$$\begin{aligned} \psi: E_5 &\rightarrow \mathbb{R} \times [0, 1] \\ (V, n, m, h, \xi) &\mapsto (V, n) \end{aligned}$$

oder

$$\begin{aligned} \psi: E_5 &\rightarrow [0, 1]^2 \\ (V, n, m, h, \xi) &\mapsto (n, m) \end{aligned}$$

und bezeichnen mit $[\mu P_{0,s}]|_{\mathbb{R}^2}$ das Bildmaß von $[\mu P_{0,s}]$ unter der Projektion ψ . Mit der Notation λ^d für das Lebesgue-Maß auf \mathbb{R}^d besitzt $[\mu P_{0,s}]|_{\mathbb{R}^2}$ eine Dichte

$$\psi_s := \frac{d([\mu P_{0,s}]|_{\mathbb{R}^2})}{d\lambda^2}, \quad s \in \mathbb{T} \tag{2.1}$$

bezüglich des Lebesgue-Maßes auf \mathbb{R}^2 , was wir in Unterabschnitt 2.1.3 zeigen werden.

Das Ziel ist nun, ψ_s für $s \in \mathbb{T}$ anhand von drei Beispielen auf verschiedene Arten darzustellen. Dabei soll deutlich werden, welche Auswirkung der Amplitudenfaktor a aus (1.1) auf periodische Strukturen in ψ_s hat, aber auch der Einfluss von unterschiedlichen Inputsignalen S , Periodizitäten T und verschiedenen Werten von τ und γ wird betrachtet werden.

2.1. Beispiel mit regelmäßigen Spikes

Im ersten Beispiel wählen wir $a = 25$ als Amplitudenfaktor, die Sinusfunktion mit Periode $T = 5$ als Inputsignal, $\tau = 0,75$ als Diffusionsgeschwindigkeit und $\gamma = 0,5$ als Streuung. Mit diesen Werten von τ und γ befindet man sich in einer biologisch realistischen Situation, da die Werte als unrealistisch gelten, wenn sie deutlich größer als 1 sind.

2.1.1. Abbildung von $X = (V, n, m, h, \xi)$ in Abhängigkeit von t

Für Abbildung 2.1 wurde das Modell über 30 Perioden des Inputs simuliert. Die Werte von V , (n, m, h) und ξ werden in Abhängigkeit von der Zeit t dargestellt. Gemeinsam mit ξ wird der deterministische Input $aS(\frac{\cdot}{T})$ eingezeichnet (für die Implementierung siehe A.4, Zeile 56-71).

In dieser ersten Darstellung sieht man, dass die Spikes offenbar mit hoher Regelmäßigkeit auftreten. Wegen Satz 1.15 wissen wir zwar, dass es beim stochastischen Hodgkin-Huxley-Modell im langen Lauf keine völlige Regelmäßigkeit geben wird, sondern immer wieder einzelne Spikes ausfallen werden, aber dies ist anhand der ersten 30 Periodizitätsintervalle nicht erkennbar. Auch bei Simulation der ersten 300 Intervalle entsteht noch ein regelmäßiges Bild. In Abbildung 2.1 sieht man außerdem, dass drei Intervalle des Inputs einem Periodizitätsintervall von V bzw. (n, m, h) entsprechen. Somit gibt es eine (aus der Abbildung abgelesene) Periodizität des Outputs von $T_{\text{out}} = 15$.

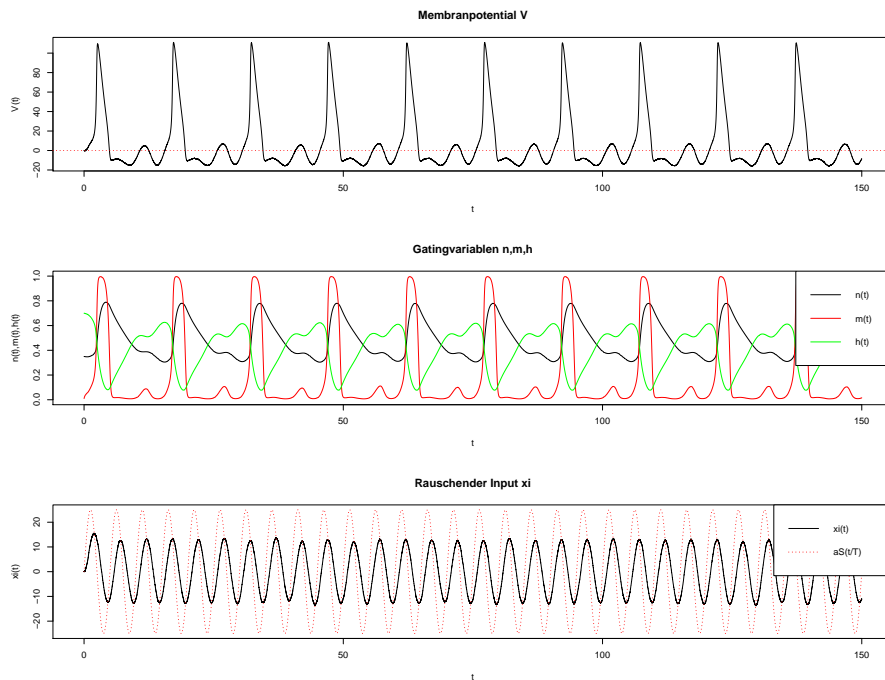


Abbildung 2.1.: Regelmäßige Spikes: Werte von V , (n, m, h) und ξ in Abhängigkeit von t für die ersten 30 Periodizitätsintervalle des Inputs

2.1.2. Projektionen von X auf zweidimensionale Ebenen

Die anhand von Abbildung 2.1 beobachteten Eigenschaften wurden verwendet, um Abbildung 2.2 zu erstellen. Dabei handelt es sich um Projektionen der fünfdimensionalen Lösung $X = (V, n, m, h, \xi)$ auf die Komponenten (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h) . In jeder der Einzelabbildungen sieht man sich überlagernde Orbits, wobei für einen Orbitdurchlauf drei Perioden des Inputs benötigt werden, da $T_{\text{in}} := T = 5$ und $T_{\text{out}} = 15$ ist. Die Simulation erfolgte über zehn Perioden des Outputs, sodass derselbe Zeithorizont wie in der vorherigen Abbildung verwendet wurde. Die schwarzen Linien in Abbildung 2.2 zeichnen die letzten fünf Orbits nach. Auf dem letzten Orbit wurden 26 zeitlich äquidistante Punkte eingezeichnet, wobei der Startpunkt blau mar-

kiert wurde. Anhand dieser nummerierten Punkte ist es möglich, in jeder der Einzelabbildungen die Flussrichtung und -geschwindigkeit nachzuvollziehen. Den Code zur Erstellung dieser Plots findet man in A.5, Zeile 59-108. Damit in Abbildung 2.1 und 2.2 dieselben, für das Euler-Maruyama-Schema benötigten, Zufallszahlen verwendet werden, wurde der Befehl `set.seed(1202)` in beide Programme eingebaut. Dadurch wird sichergestellt, dass die beiden Abbildungen unterschiedliche Darstellungen desselben Lösungsprozesses X sind.

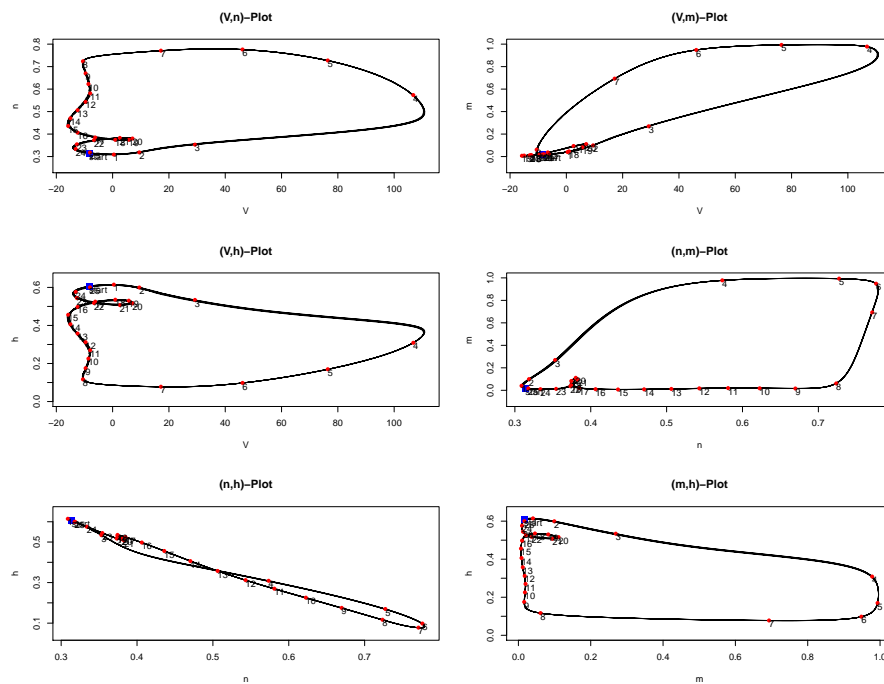


Abbildung 2.2.: Regelmäßige Spikes: Projektionen von $X = (V, n, m, h, \xi)$ auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h) für die letzten fünf Periodizitätsintervalle des Outputs

Abbildung 2.2 verdeutlicht noch einmal, dass in diesem Beispiel die Spikes nahezu regelmäßig auftreten, denn die Orbits überlagern sich so stark, dass es so aussieht, als würden sie nur eine einzige dünne Linie bilden. Außerdem kann man anhand der eingezeichneten Punkte einen ersten Eindruck von der Dichte

ψ_s bekommen: Für viele der $s \in \mathbb{T}$ wird ψ_s dort große Werte annehmen, wo in den jeweiligen Einzelabbildungen viele Punkte liegen, und nur für wenige $s \in \mathbb{T}$ wird ψ_s Gewicht auf Gegenden legen, in denen wenige Punkte sind.

2.1.3. Theorie zu Kerndichteschätzern

Um eine genauere Vorstellung von ψ_s für $s \in \mathbb{T}$ zu bekommen, werden wir im weiteren Verlauf Kerndichteschätzer verwenden. Dafür liefern wir zunächst eine kurze theoretische Einführung in dieses Thema, die (bis zu der Formulierung von Satz 2.3) auf [20, Section 1.2] und [17, Section 6.2] basiert.

Seien X_1, \dots, X_n unabhängige, identisch verteilte, \mathbb{R}^d -wertige Zufallsvariablen mit Dichte p bezüglich des Lebesgue-Maßes. Ziel ist es, einen Schätzer für p finden. Sei dazu $G: \mathbb{R}^d \rightarrow [0, \infty)$ eine messbare Funktion mit $\int_{\mathbb{R}^d} G(u) du = 1$, die als *Kern* bezeichnet wird. Außerdem sei $h_n > 0$ die sogenannte *Bandbreite*, die von n abhängig ist. Der *Kerndichteschätzer* ist die Funktion

$$p_n^*(x) := \frac{1}{nh_n^d} \sum_{k=1}^n G\left(\frac{X_k - x}{h_n}\right), \quad x \in \mathbb{R}^d.$$

Aus den Eigenschaften von G folgt, dass p_n^* bei festen X_1, \dots, X_n eine Dichte ist.

Die Wahl der optimalen Bandbreite h_n ist ein komplexeres Problem. Wählt man eine zu kleine Bandbreite, werden einzelne Beobachtungen zu stark gewichtet und der Graph von p_n^* sieht verrauscht aus. Ist h_n jedoch zu groß, so wird der Graph zu stark geglättet und ist dadurch wenig aussagekräftig. Um eine passende Bandbreite zu finden und Aussagen über die Qualität des Kerndichteschätzers zu formulieren, werden die folgenden beiden Definitionen benötigt.

Definition 2.1. Seien $\beta, L > 0$ und sei $l = \lfloor \beta \rfloor$, wobei $\lfloor \beta \rfloor$ für die größte ganze Zahl steht, die strikt kleiner ist als β . Die *Hölder-Klasse* $\Sigma(\beta, L)$ auf \mathbb{R}^d ist die Menge aller l -mal differenzierbaren Funktionen $f: \mathbb{R}^d \rightarrow \mathbb{R}$, für deren partielle Ableitungen $f^{(l)}$ der Ordnung l gilt, dass

$$\left| f^{(l)}(x) - f^{(l)}(x') \right| \leq L \|x - x'\|^{\beta-l} \quad \text{für alle } x, x' \in \mathbb{R}^d,$$

wobei $\|\cdot\|$ hier und im weiteren Verlauf dieses Unterabschnitts für die Euklid-Norm steht.

Mit $\mathcal{P}(\beta, L)$ wird die Menge aller Lebesgue-Dichten in der Hölder-Klasse $\Sigma(\beta, L)$ auf \mathbb{R}^d bezeichnet.

Definition 2.2. Sei $l \in \mathbb{N}$. Ein Kern $G: \mathbb{R}^d \rightarrow [0, \infty)$ ist von *Ordnung* l , wenn für alle $j = 1, \dots, l$ und alle $j_1, \dots, j_d \in \mathbb{N}_0$ mit $j_1 + \dots + j_d = j$ gilt, dass

$$\int_{\mathbb{R}} \dots \int_{\mathbb{R}} G(u_1, \dots, u_d) u_1^{j_1} \dots u_d^{j_d} du_1 \dots du_d = 0.$$

Satz 2.3. Sei G ein Kern von Ordnung $l = \lfloor \beta \rfloor$, für den

$$\int_{\mathbb{R}^d} \|u\|^\beta G(u) du < \infty \quad \text{und} \quad \int_{\mathbb{R}^d} G^2(u) du < \infty \quad (2.2)$$

gelte. Außerdem sei $\alpha > 0$ eine Konstante und $h_n = \alpha n^{-\frac{1}{2\beta+d}}$ die Bandbreite. Dann erfüllt der Kerndichteschätzer p_n^* für jedes $n \in \mathbb{N}$

$$\sup_{x \in \mathbb{R}^d} \sup_{p \in \mathcal{P}(\beta, L)} E_p[(p_n^*(x) - p(x))^2] \leq C n^{-\frac{2\beta}{2\beta+d}},$$

wobei die Konstante $C > 0$ nur von β, L, α und G abhängt.

Der Kerndichteschätzer p_n^* besitzt bei Wahl der Bandbreite aus dem Satz folglich eine Konvergenzrate von $n^{-\frac{\beta}{2\beta+d}}$.

Beweis. Für den Fall $d = 1$ wird Satz 2.3 in [20, Theorem 1.1] bewiesen. Außerdem wird der eindimensionale Fall auch in [15, Theorem 4.1 auf Seite 235] behandelt. In [15, Remark 5.5 auf Seite 240] wird zudem schon angemerkt, dass sich die in dieser Quelle zuvor gemachten Aussagen zur Konvergenz von Kerndichteschätzern auf den allgemeinen Fall $d \in \mathbb{N}$ übertragen lassen. Das wird in [13] genauer ausgeführt. Im dortigen Theorem 3 wird unter der zusätzlichen Forderung, dass $\sup_{p \in \mathcal{P}(\beta, L)} \|p\|_{L^2(\lambda^d)} = M$ für ein $M < \infty$ ist, gezeigt, dass der Kerndichteschätzer bei geeigneter Wahl der Bandbreite die behauptete Konvergenzrate bei Verwendung der L^2 -Norm besitzt.

Bemerkung 2.4. Die Aussage von Satz 2.3 lässt sich verallgemeinern, indem

man die sogenannte anisotropische Nikolskii-Klasse (nach [4, Definition 1]) einführt.

Sei dafür $\bar{\beta} = (\beta_1, \dots, \beta_d)$ mit $\beta_i > 0$ und $L > 0$. Eine hinreichend oft differenzierbare Lebesgue-Dichte $f: \mathbb{R}^d \rightarrow \mathbb{R}$ gehört zu der *anisotropischen Nikolskii-Klasse* $N(\bar{\beta}, L)$, wenn gilt:

- (i) $\|D_i^{\lfloor \beta_i \rfloor} f\|_{L^2(\lambda^d)} \leq L$ für alle $i = 1, \dots, d$.
- (ii) Für alle $i = 1, \dots, d$ und $z \in \mathbb{R}$ gilt

$$\left(\int_{\mathbb{R}^d} \left| D_i^{\lfloor \beta_i \rfloor} f(x_1, \dots, x_i + z, \dots, x_d) - D_i^{\lfloor \beta_i \rfloor} f(x_1, \dots, x_i, \dots, x_d) \right|^2 dx \right)^{\frac{1}{2}} \leq L |z|^{\beta_i - \lfloor \beta_i \rfloor}.$$

Dabei steht $D_i^l f$ für die l -te partielle Ableitung von f nach x_i .

Im Gegensatz zu der Hölder-Klasse aus Definition 2.1 haben die Dichten in der Nikolskii-Klasse folglich möglicherweise unterschiedliche Glattheitsordnungen in den einzelnen Komponenten. Deshalb ist es naheliegend, verschiedene Bandbreiten $h_{n,i}$ für die Komponenten $i = 1, \dots, d$ zu verwenden. Dafür führen wir die Notation $\beta := \left(\sum_{i=1}^d \frac{1}{\beta_i} \right)^{-1}$ und den Kerndichteschätzer

$$p_n^{**}(x) := \frac{1}{n \prod_{i=1}^d h_{n,i}} \sum_{k=1}^n G\left(\frac{X_k - x}{h_n}\right), \quad x \in \mathbb{R}^d$$

ein, wobei die Division in $G(\cdot)$ komponentenweise zu verstehen ist. Laut [14, Table 1] beziehungsweise [4, Theorem 4] besitzt p_n^{**} bei Wahl der Bandbreiten $h_{n,i} = n^{-\frac{\beta}{\beta_i(2\beta+1)}}$ eine Konvergenzrate von $n^{-\frac{\beta}{2\beta+1}}$.

Um sich davon zu überzeugen, dass die Aussage von Satz 2.3 ein Spezialfall hiervon ist, wählt man $\beta_i = \beta_1$ für alle $i = 1, \dots, d$. Damit ist $\beta = \left(\sum_{i=1}^d \frac{1}{\beta_1} \right)^{-1} = \frac{\beta_1}{d}$ und für die Konvergenzrate gilt deshalb $n^{-\frac{\beta}{2\beta+1}} = n^{-\frac{\beta_1/d}{2(\beta_1/d)+1}} = n^{-\frac{\beta_1}{2\beta_1+d}}$, was der Konvergenzrate aus Satz 2.3 entspricht. Es lässt sich ebenfalls überprüfen, dass die optimale Bandbreite bei gleichen Glattheitsordnungen in allen Komponenten mit der aus dem Satz übereinstimmt.

Nun sollen die bisher allgemein formulierten Ergebnisse auf unsere konkrete

Situation übertragen werden. In unserem Fall ist $p = \psi_s$ die zu schätzende Dichte und als Eingangsdaten verwenden wir X_{kT+s} für $k = 0, 1, \dots, K-1$, das heißt Beobachtungen zu einer festen Toruszeit $s \in \mathbb{T}$ in K betrachteten Perioden des Inputs. Man kann zwar nicht davon ausgehen, dass diese Eingangsdaten unabhängig, identisch verteilt sind, aber man weiß, dass der Prozess $(X_{kT+s})_{k \in \mathbb{N}_0}$ Harris-rekurrent ist. Auf einen solchen Prozess kann die Technik des „Nummelin splitting“ angewendet werden (siehe [9, Section 6] oder die Originalquelle [18]). Dabei wird künstlich ein sogenanntes rekurrentes Atom eingeführt und die Austrittszeiten aus diesem Atom zerlegen den Verlauf des Prozesses in unabhängige, identisch verteilte Lebenszyklen. Dadurch wird ein Zusammenhang zwischen Harris-rekurrenten Markovprozessen und unabhängigen, identisch verteilten Zufallsvariablen geschaffen, der vermuten lässt, dass sich die bisherigen Aussagen zu Kerndichteschätzern auf unsere Situation übertragen lassen. Insbesondere für den Beweis von Satz 2.3 dürfte in unserem Fall das starke Gesetz der großen Zahlen aus Korollar 1.13 eine wichtige Rolle spielen.

Um einen Schätzer für ψ_s zu konstruieren, verwenden wir als Kern den zweidimensionalen Gaußkern, was bedeutet, dass $G(u_1, u_2) = \varphi(u_1)\varphi(u_2)$ ist, wobei φ die Dichte der eindimensionalen Standardnormalverteilung ist. Es lässt sich überprüfen, dass G von Ordnung $l = 1$ ist. Unser Kerndichteschätzer lautet

$$p_K^*(x) := \frac{1}{Kh_K^2} \sum_{k=0}^{K-1} G\left(\frac{\psi(X_{kT+s}) - x}{h_K}\right), \quad x \in \psi(E_5) \subset \mathbb{R}^2.$$

Dabei beschränken wir uns hier und in den folgenden theoretischen Ausführungen auf die Situation, in der in allen Komponenten $i = 1, \dots, d$ dieselbe Bandbreite verwendet wird, obwohl wir später in der Praxis verschiedene Bandbreiten in den einzelnen Komponenten benutzen wollen.

Unser Ziel ist nun, zu zeigen, dass $\beta = 1$ die Glattheitsordnung von ψ_s ist. Dafür müssen wir zuerst herausfinden, wie ψ_s konkret aussieht. Damit würde sich dann auch bestätigen, dass ψ_s existiert, denn diese Frage ist momentan noch offen.

Aus [11, Corollary 1] ist bekannt, dass die Verteilung $P_{0,s}(x, \cdot)$ mit $x \in E_5$ eine Lebesgue-Dichte $p_{0,s}(x, \cdot)$ besitzt. Somit ist für $A \in \mathcal{B}(E_5)$

$$\begin{aligned} [\mu P_{0,s}](A) &= \int_{E_5} \mu(dx) P_{0,s}(x, A) \\ &= \int_{E_5} \mu(dx) \int_A p_{0,s}(x, y) dy \\ &= \int_A \left(\int_{E_5} \mu(dx) p_{0,s}(x, y) \right) dy. \end{aligned}$$

Das bedeutet, dass der Prozess X bei Start zur Zeit 0 in μ zu Zeitpunkten $kT + s$ die Dichte $\int_{E_5} \mu(dx) p_{0,s}(x, \cdot)$ hat.

Wir betrachten jetzt ohne Beschränkung der Allgemeinheit nur die Projektion

$$\begin{aligned} \psi: E_5 &\rightarrow \mathbb{R} \times [0, 1] \\ (V, n, m, h, \xi) &\mapsto (V, n) \end{aligned}$$

und definieren für $\zeta = (\zeta_1, \zeta_2) \in \mathbb{R} \times [0, 1]$

$$\psi_s((\zeta_1, \zeta_2)) := \int_{[0,1]} d\zeta_3 \int_{[0,1]} d\zeta_4 \int_{\mathbb{R}} d\zeta_5 \left(\int_{E_5} \mu(dx) p_{0,s}(x, (\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5)) \right). \quad (2.3)$$

Dabei handelt es sich tatsächlich um die in (2.1) eingeführte Dichte ψ_s von $[\mu P_{0,s}]|_{\mathbb{R}^2}$ bezüglich des zweidimensionalen Lebesgue-Maßes, denn für $B \in \mathcal{B}(\mathbb{R} \times [0, 1])$ gilt

$$\begin{aligned} \int_B \psi_s((\zeta_1, \zeta_2)) \lambda^2(d(\zeta_1, \zeta_2)) &= [\mu P_{0,s}](B \times [0, 1]^2 \times \mathbb{R}) \\ &= ([\mu P_{0,s}] \circ \psi^{-1})(B) \\ &= [\mu P_{0,s}]|_{\mathbb{R}^2}(B). \end{aligned}$$

Jetzt wollen wir die Glattheitsordnung von ψ_s untersuchen. Laut [12, Theorem 1] ist $p_{0,s}(x, y)$ bei festem $x \in E_5$ unendlich oft differenzierbar und damit lokal Lipschitz-stetig in $y = y_0$ für alle $y_0 \in \text{int}(E_5)$. Außerdem zeigen die letzten drei Zeilen des Beweises von [12, Theorem 1], dass die Stetigkeit von $p_{0,s}(x, y)$ in $y = y_0$ gleichmäßig in $x \in E_5$ für alle $y_0 \in \text{int}(E_5)$ ist. Deshalb

gibt es für jedes Kompaktum $\hat{K} \subset E_5$ eine Konstante $L = L(\hat{K}) > 0$, sodass für alle $y, y' \in \hat{K}$

$$\sup_{x \in E_5} |p_{0,s}(x, y) - p_{0,s}(x, y')| \leq L \|y - y'\|$$

gilt. Da μ ein Wahrscheinlichkeitsmaß auf E_5 ist, gilt folglich auch

$$\left| \int_{E_5} \mu(dx) p_{0,s}(x, y) - \int_{E_5} \mu(dx) p_{0,s}(x, y') \right| \leq L \|y - y'\| \quad (2.4)$$

für alle $y, y' \in \hat{K}$. Allerdings ist nicht gewährleistet, dass ψ_s ebenfalls eine Lipschitz-Eigenschaft besitzt, weil in (2.3) eine Integration über \mathbb{R} eingeht. Das Maß $[\mu P_{0,s}]$ ist jedoch als einzelnes Wahrscheinlichkeitsmaß auf dem polnischen Raum $E_5 = \mathbb{R} \times [0, 1]^3 \times \mathbb{R}$ straff. Das heißt, zu jedem $\varepsilon > 0$ existiert ein $C = C(\varepsilon) < \infty$, sodass

$$[\mu P_{0,s}](E_5 \setminus ([-C, C] \times [0, 1]^3 \times [-C, C])) < \varepsilon$$

ist. Dies motiviert uns dazu,

$$\tilde{\psi}_s((\zeta_1, \zeta_2)) := \int_{[0,1]} d\zeta_3 \int_{[0,1]} d\zeta_4 \int_{[-C,C]} d\zeta_5 \left(\int_{E_5} \mu(dx) p_{0,s}(x, (\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5)) \right) \quad (2.5)$$

als Approximation an $\psi_s((\zeta_1, \zeta_2))$ für $(\zeta_1, \zeta_2) \in [-C, C] \times [0, 1]$ einzuführen. Wegen des Satzes von der monotonen Konvergenz nähert sich $\tilde{\psi}_s$ für $C \rightarrow \infty$ an ψ_s an. Da in (2.5) nur über Kompakta integriert wird, und wegen (2.4) gilt für alle $\zeta = (\zeta_1, \zeta_2), \zeta' = (\zeta'_1, \zeta'_2) \in [-C, C] \times [0, 1]$

$$\begin{aligned} & \left| \tilde{\psi}_s(\zeta) - \tilde{\psi}_s(\zeta') \right| \\ & \leq \int_{[0,1]} d\zeta_3 \int_{[0,1]} d\zeta_4 \int_{[-C,C]} d\zeta_5 \\ & \quad \left| \int_{E_5} \mu(dx) p_{0,s}(x, (\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5)) - \int_{E_5} \mu(dx) p_{0,s}(x, (\zeta'_1, \zeta'_2, \zeta_3, \zeta_4, \zeta_5)) \right| \\ & \leq \tilde{L} \|\zeta - \zeta'\| \end{aligned}$$

mit einer Konstanten $\tilde{L} > 0$. Somit ist $\tilde{\psi}_s$ Lipschitz-stetig.

Heuristisch gesehen dürfte in unserer praktischen Anwendungssituation für hinreichend große C fast kein Unterschied zwischen ψ_s und $\tilde{\psi}_s$ zu sehen sein. Außerdem werden die im weiteren Verlauf der Arbeit verwendeten Kerndichteschätzer auch nur auf großen Kompakta ausgewertet. Deshalb behandeln wir die Dichte ψ_s so, als ob sie Lipschitz-stetig wäre, was bedeutet, dass wir den Unterschied zwischen ψ_s und $\tilde{\psi}_s$ und zwischen $[-C, C] \times [0, 1]$ und $\mathbb{R} \times [0, 1]$ für große C vernachlässigen.

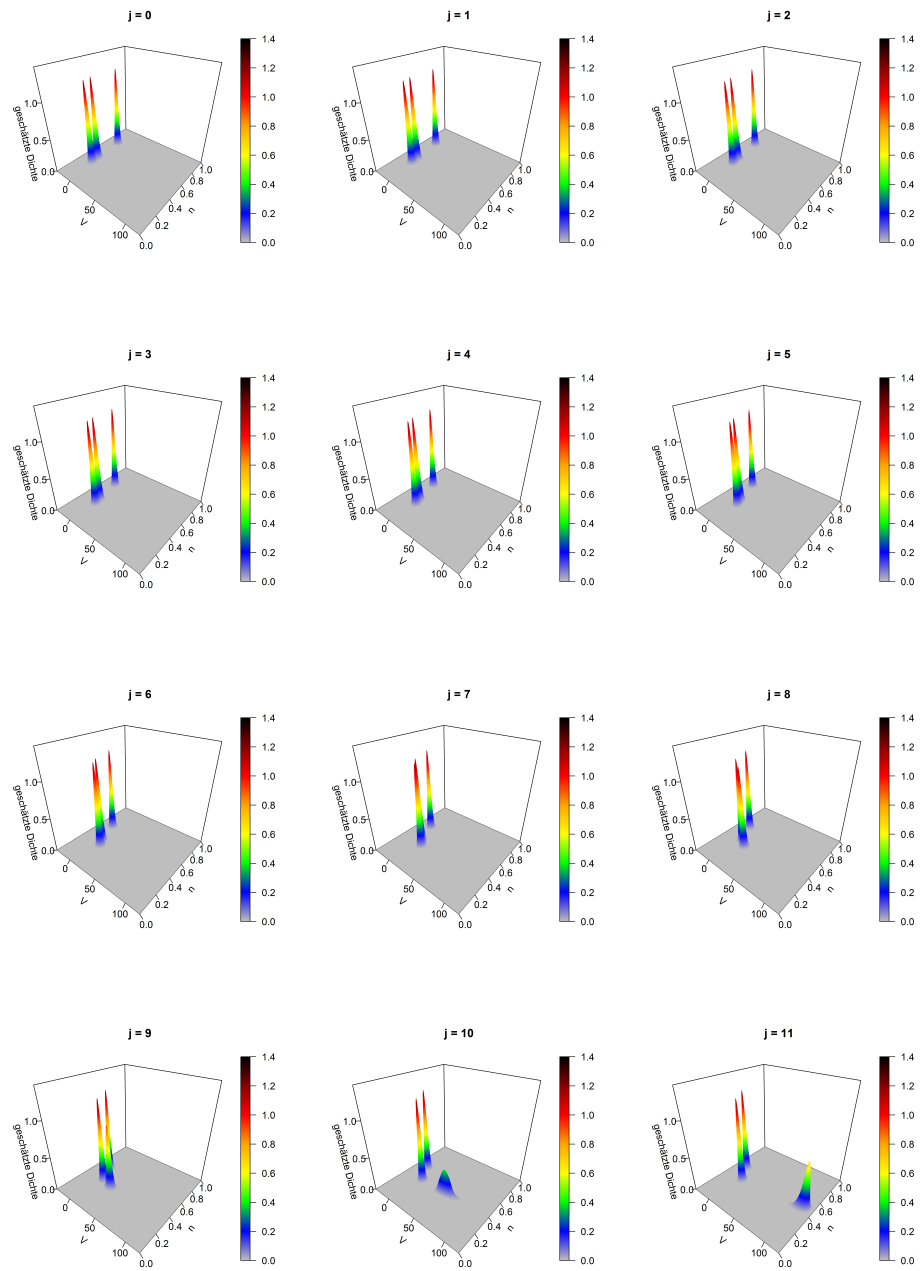
Folglich nehmen wir an, dass $\beta = 1$ die Glattheitsordnung von ψ_s ist. Damit werden in Satz 2.3 keine Anforderungen an die Ordnung des Kerns gestellt und der Gauß-Kern kann benutzt werden, da er die Voraussetzungen (2.2) erfüllt. Wir wissen deshalb jetzt, dass unser Kerndichteschätzer p_K^* bei Wahl der Bandbreite $h_K = K^{-\frac{1}{4}}$ eine Konvergenzrate von $K^{-\frac{1}{4}}$ hat.

2.1.4. Kerndichteschätzer für ψ_s zu 26 Zeitpunkten im Torus

Für die Erstellung von Abbildung 2.3 beschränken wir uns auf 26 äquidistante Zeitpunkte s im Torus \mathbb{T} , das heißt $s \in S := \{s_j = \frac{jT}{25} \mid j = 0, 1, \dots, 25\}$. Wie in Abbildung 2.2 gibt es dabei einen Startpunkt s_0 und 25 weitere Punkte. Diese 26 Punkte reichen aus, um die zeitliche Dynamik von ψ_s nachzuvollziehen.

Um Speicherplatz zu sparen, ist das Programm in A.6, das zur Erstellung von Abbildung 2.3 verwendet wurde, so eingerichtet, dass nach Simulation des Prozesses X über ein Periodizitätsintervall $[kT, (k+1)T]$ nur jeweils die 26 Werte X_{kT+s_j} , $s_j \in S$ abgespeichert bleiben. Alle anderen berechneten Werte können im nächsten Schritt wieder überschrieben werden. Der letzte gespeicherte Wert $X_{kT+s_{25}} = X_{(k+1)T+s_0}$ liefert den Startpunkt für die Simulation von X über das nächste Periodizitätsintervall $[(k+1)T, (k+2)T]$. In den ersten zehn Perioden des Inputs werden allerdings noch keine Daten gespeichert, da es in der Anfangsphase noch zu Phänomenen kommen kann, die auf die Wahl der Startwerte (1.5) zurückzuführen sind.

In Abbildung 2.3 betrachten wir die Projektion von X auf (V, n) . Für die Berechnung von zweidimensionalen Kerndichteschätzern wird in R die Funktion



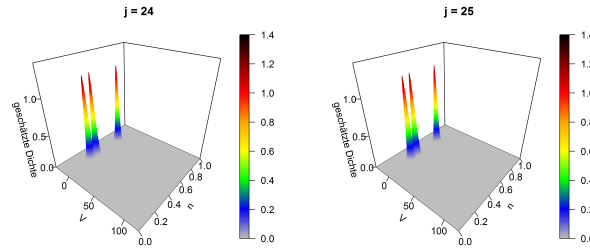


Abbildung 2.3.: Regelmäßige Spikes: Kerndichteschätzer für ψ_s zur Toruszeit s_j , $j = 0, 1, \dots, 25$, Projektion auf (V, n)

`kde2d` im Paket `MASS` zur Verfügung gestellt. Diese Funktion verwendet als Kern den Gaußkern. Als Inputdaten wurden die Werte V_{kT+s_j} und n_{kT+s_j} für $k = 10, \dots, K - 1$ zu einer festen Toruszeit $s_j \in S$ verwendet (siehe A.6, Zeile 81-85). Außerdem wurde festgelegt, dass der Schätzer auf einem Gitter mit jeweils 500 Punkten in Richtung V und n berechnet werden soll.

Die benutzte Bandbreite wurde empirisch bestimmt. Da die Werte von V etwa im Bereich $[-30, 120]$ liegen, n aber nur Werte im Intervall $[0, 1]$ annimmt, ist es sinnvoll, nicht die gleiche Bandbreite für V und n zu verwenden. Nach Ausprobieren verschiedener Werte für $h = (h_V, h_n)$ wurde $h_V = 10$ und $h_n = 0,07$ gewählt, da bei dieser Wahl übersichtliche Bilder entstehen, die aber auch nicht zu stark geglättet aussehen.

Der Output von `kde2d` liefert die Koordinaten der verwendeten Gitterpunkte und den errechneten Schätzwert für die Dichte an jedem einzelnen Gitterpunkt. Diese Daten wurden benutzt, um den Kerndichteschätzer mit der Funktion `persp3D` aus dem Paket `plot3D` zeichnen zu lassen.

Für jedes Bild aus Abbildung 2.3 wurde der Prozess X über $K = 750$ Perioden des Inputs und somit über 250 Perioden des Outputs simuliert.

Abbildung 2.3 stellt die zirkuläre zeitliche Dynamik von ψ_s für $s \in S$ dar. Man sieht auf jedem der Bilder drei Berge, von denen einer einmal den Orbit aus dem ersten Bild von Abbildung 2.2 durchläuft, während sich die anderen beiden deutlich weniger bewegen. Da in der Simulation der letzte gespeicherte Wert in einem Periodizitätsintervall als Startpunkt für die Simulation des nächsten

Periodizitätsintervalls genutzt wurde, entspricht das Bild zur Toruszeit s_{25} dem Bild zur Toruszeit s_0 . Die drei Berge verdeutlichen noch einmal, dass in diesem Beispiel drei Perioden des Inputs einer Periode des Outputs entsprechen. Dabei stehen die zwei Berge, die sich weniger bewegen, für die Inputperioden, in denen es zu keinem Spike kommt, und der Berg, der auf dem Orbit wandert, läuft die Werte von V und n während eines Spikes ab. Man erkennt anhand der 26 Bilder, wo der Berg sich auf dem Orbit schnell bewegt und in welchen Gegenden er länger verweilt. Die Abbildung belegt außerdem noch einmal, wie regelmäßig die Spikes in diesem Beispiel auftreten, denn man sieht auf allen Bildern nur die drei separaten Berge ohne Abweichungen in Form von kleineren Hügeln zwischen ihnen.

2.1.5. Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig

Für Abbildung 2.4 wurde der Prozess $X = (V, n, m, h, \xi)$ sechsmal über 110 Perioden des Inputs simuliert. Im Gegensatz zur vorherigen grafischen Darstellungsmöglichkeit gibt es in A.7 keine Einschränkung auf einzelne Toruszeitpunkte, sondern es bleiben $E = 1000$ Werte von X pro Periodizitätsintervall des Inputs gespeichert. Alle Werte aus den letzten 100 Intervallen werden dann für die Erstellung der Plots verwendet, in denen jeweils auf zwei Komponenten von X projiziert wird. Der Kerndichteschätzer wird wieder auf einem Gitter aus 500×500 Punkten mit Hilfe der Funktion `kde2d` berechnet. Die dabei benutzten, empirisch bestimmten Bandbreiten lauten $h_V = 10$, $h_n = 0,07$, $h_m = 0,08$ und $h_h = 0,06$. Um die von `kde2d` ermittelten Daten grafisch darzustellen, wird dieses Mal die Funktion `filled.contour` aus dem Paket `graphics` verwendet. Diese erstellt ein zweidimensionales Konturdiagramm, in dem die Werte der geschätzten Dichte durch Farben symbolisiert werden. Jedes der Bilder in Abbildung 2.4 hat eine eigene Legende, die anzeigt, welche Farbe für welche Dichtewerte steht. Es wurde dieses Mal darauf verzichtet, den Kerndichteschätzer dreidimensional darzustellen, da es schwierig ist, einen Blickwinkel einzustellen, bei dem der komplette Plot gut zu erkennen ist und die Gebirge nicht gegenseitig die Sicht aufeinander behindern. Außerdem

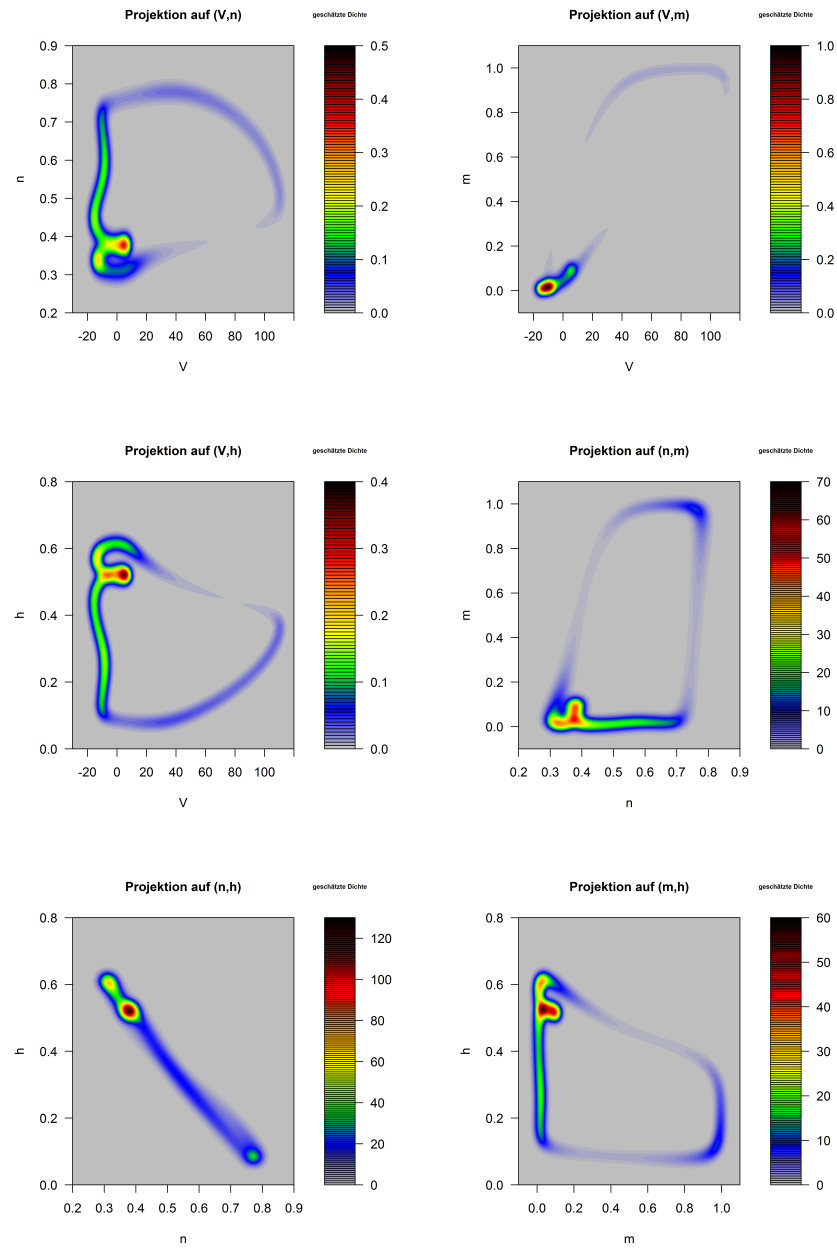


Abbildung 2.4.: Regelmäßige Spikes: Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig, Projektionen auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h)

erkennt man im Konturdiagramm die Orbits aus Abbildung 2.2 besser wieder als im 3D-Plot. Die Codes für die Erstellung der 3D-Plots findet man aber trotzdem auskommentiert in A.7 (siehe zum Beispiel Zeile 76-78).

Abbildung 2.4 unterstützt die Ergebnisse, die wir bisher über die Dynamik von ψ_s , $s \in \mathbb{T}$ gewonnen haben. Die roten Stellen in den Orbits zeigen beispielsweise, wohin ψ_s für viele s das meiste Gewicht legt. Betrachtet man die Projektion auf (V, n) , so sieht man, dass die rote Stelle im Orbit genau die ist, an der sich die beiden weniger beweglichen Berge in Abbildung 2.3 permanent aufgehalten haben. Die hellblauen Stellen in den Orbits sind wiederum die Bereiche, in denen sich ψ_s mit großer Geschwindigkeit bewegt. In der Projektion auf (V, m) ist die Geschwindigkeit stellenweise so hoch, dass der Orbit nicht durchgängig gezeichnet wurde. Wenn man das mit dem (V, m) -Plot aus Abbildung 2.2 vergleicht, sieht man, dass in diesen Bereichen keiner der 26 auf dem letzten Orbit eingezeichneten Punkte liegt.

2.2. Beispiel mit unregelmäßigen Spikes

Für das nächste Beispiel wählen wir $a = 14$ als Amplitudenfaktor und übernehmen die übrigen Parameter aus dem vorherigen Beispiel (5-periodische Sinusfunktion als Inputsignal, $\tau = 0,75$ als Diffusionsgeschwindigkeit und $\gamma = 0,5$ als Streuung).

2.2.1. Abbildung von $X = (V, n, m, h, \xi)$ in Abhängigkeit von t

Für Abbildung 2.5 wurde das Modell wieder über 30 Perioden des Inputs simuliert. Man sieht, dass die Spikes dieses Mal unregelmäßiger auftreten als bei der Wahl von $a = 25$. Weitere Simulationen haben gezeigt, dass bei einer Wahl von immer kleineren Amplitudenfaktoren immer seltener Spikes zu beobachten sind. Bei Werten von a , die kleiner sind als ungefähr 9, kommt es in den ersten 300 Periodizitätsintervallen des Inputs fast nie zu Spikes. In Abbildung 2.5 ist jedoch auch zu erkennen, dass es für $a = 14$ noch Intervalle gibt, in denen die Spikes annähernd regelmäßig auftreten. Beispielsweise für $t \in [95, 125]$ kommt es auf einem Zeitintervall der Länge 30 zu zwei Spikes.

Deshalb gehen wir auch in diesem Beispiel von einer Periodizität des Outputs von $T_{\text{out}} = 15$ aus.

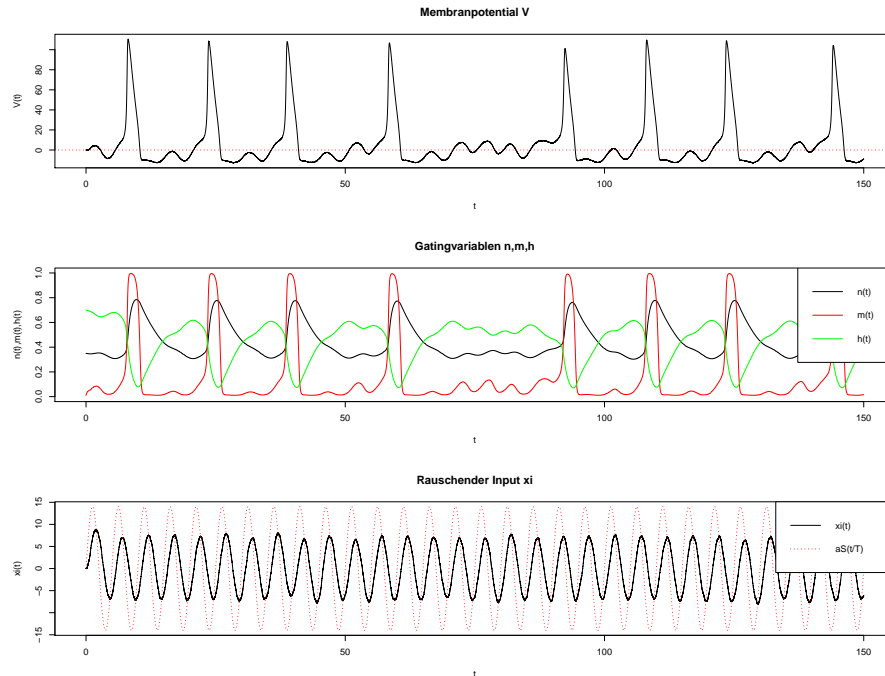


Abbildung 2.5.: Unregelmäßige Spikes: Werte von V , (n, m, h) und ξ in Abhängigkeit von t für die ersten 30 Periodizitätsintervalle des Inputs

2.2.2. Projektionen von X auf zweidimensionale Ebenen

Um Abbildung 2.6 zu erstellen, wurde X über 10 Zeitintervalle der Länge 15 simuliert, da $T_{\text{out}} = 15$ ist. Somit wurde wieder derselbe Zeithorizont wie in der vorherigen Abbildung verwendet. Gezeichnet wurde der Verlauf von je zwei Komponenten von X in den letzten fünf Zeitintervallen und auf dem letzten Orbit wurden wieder 26 zeitlich äquidistante Punkte markiert.

Die Orbits in Abbildung 2.6 zeigen noch einmal, dass die Spikes in diesem Beispiel nicht so regelmäßig auftreten wie im vorherigen. In Abbildung 2.2

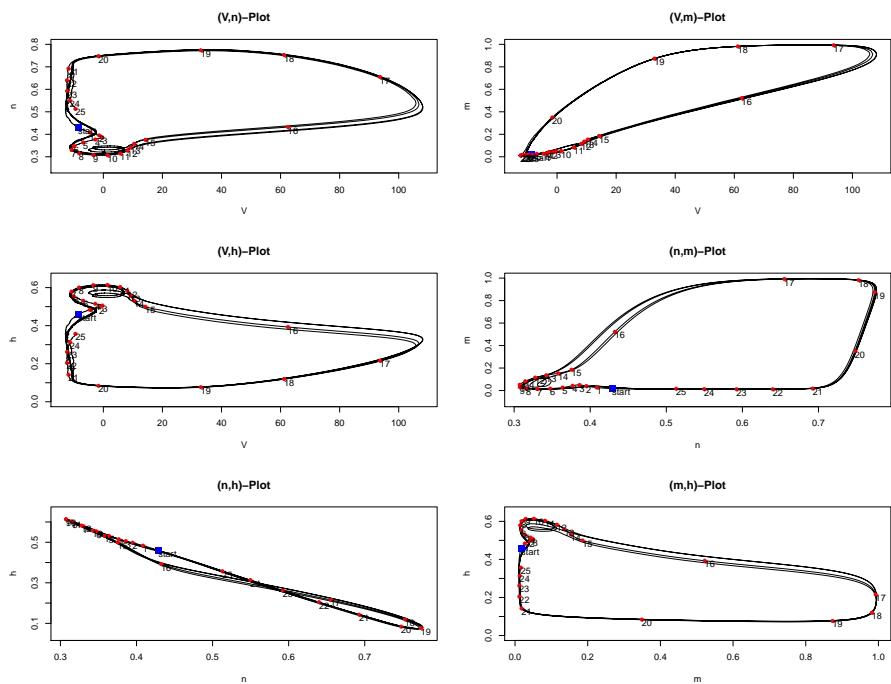


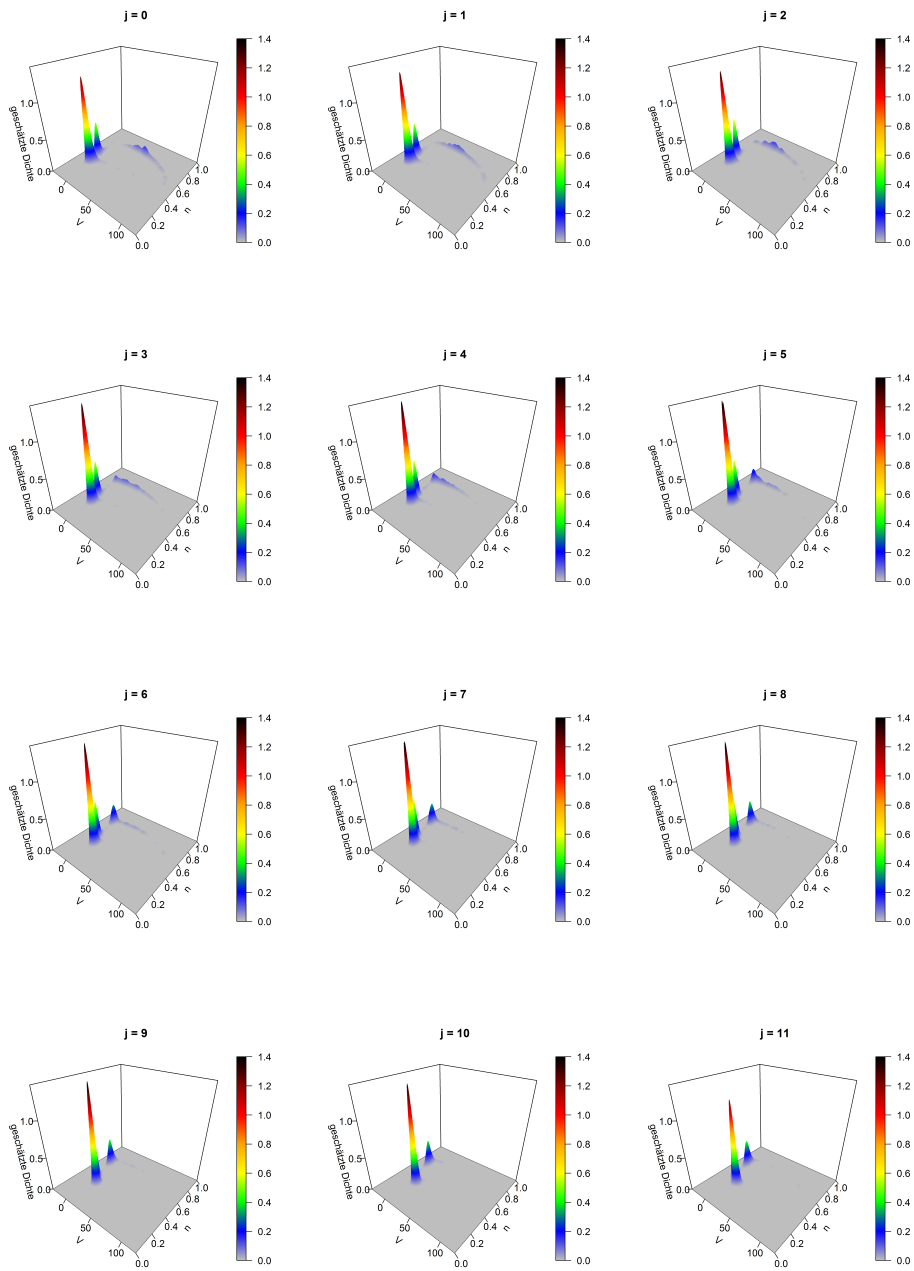
Abbildung 2.6.: Unregelmäßige Spikes: Projektionen von $X = (V, n, m, h, \xi)$ auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h) für die letzten fünf Periodizitätsintervalle des Outputs

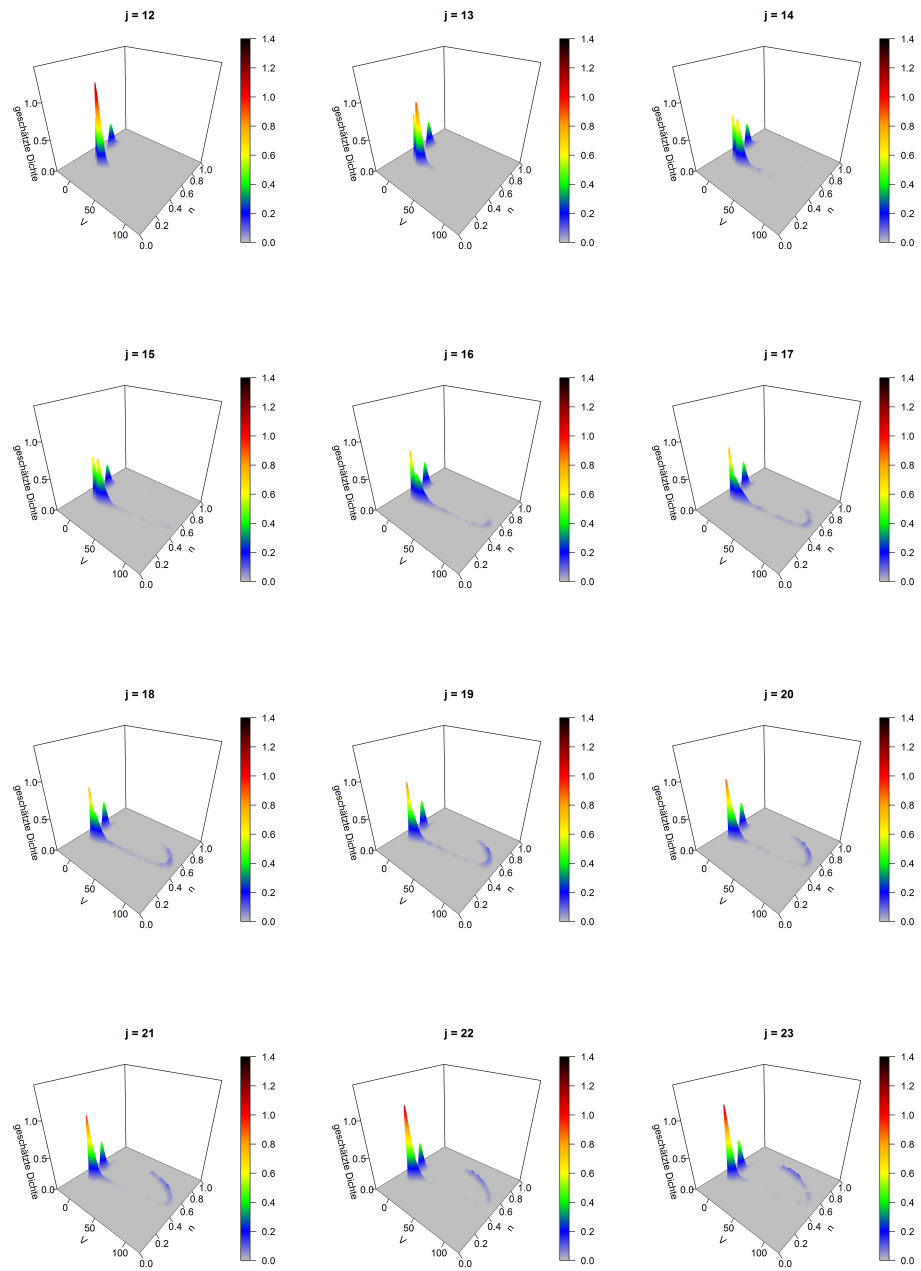
bildeten die Orbits in allen Bildern fast nur eine einzelne dünne Linie. Hier sieht man an mehreren Stellen separate Linien oder eine breite Linie. Betrachtet man zum Beispiel den (V, n) -Plot genauer, so erkennt man mehrere einzelne Linien, die den großen Orbit bilden, und eine Linie, die sich kreisförmig in etwa im Bereich von $V = 0$ und $n = 0,33$ bewegt. Die Linien des großen Orbits stehen dabei für die letzten vier Periodizitätsintervalle des Outputs, da X in diesen gespiket hat, und die kreisende Linie entspricht dem Zeitintervall $[75, 90]$, in dem es keine Spikes gab (vgl. Abbildung 2.5). Wie in Abbildung 2.2 ist es auch hier möglich, anhand der eingezeichneten Punkte einen ersten Eindruck von ψ_s , $s \in \mathbb{T}$ zu bekommen.

2.2.3. Kerndichteschätzer für ψ_s zu 26 Zeitpunkten im Torus

Für die Erstellung von Abbildung 2.7 wurde der Prozess X wieder 26-mal über $K = 750$ Perioden des Inputs simuliert. Es wurden dieselben Funktionen in R wie für Abbildung 2.3 verwendet und auch die Wahl von Parametern wie zum Beispiel der Bandbreite ist unverändert geblieben. Um eine einfache Vergleichbarkeit von Abbildung 2.3 und 2.7 zu ermöglichen, wurden auch die gleichen Grenzen der drei Achsen in beiden Abbildungen festgelegt.

Anhand der 26 Bilder von Abbildung 2.7 lässt sich wieder die zeitliche Dynamik von ψ_s nachvollziehen. Im Gegensatz zum vorherigen Beispiel sieht man auf den einzelnen Bildern dieses Mal jedoch unterschiedlich hohe Berge, die in manchen Bildern miteinander verschmelzen oder kleinere Hügel zwischen ihnen aufweisen. Der Grund dafür ist das unregelmäßige Auftreten der Spikes in diesem Beispiel. Dies bedeutet aber auch, dass allgemein weniger Spikes stattfinden und es deshalb mehr ruhende Masse gibt, was auf den Bildern zu den Toruszeiten s_j , $j = 3, \dots, 10$ besonders gut zu erkennen ist. Dort liegt das Maximum von ψ_{s_j} nämlich jeweils bei ungefähr 1,4, während das Maximum der geschätzten Dichte im vorherigen Beispiel nie größer als circa 1,2 war. Auch in diesem Beispiel ist eine Wanderung von Masse durch den Orbit aus dem ersten Bild von Abbildung 2.6 erkennbar. Die wandernde Masse hat hier allerdings nicht die Form eines Berges, sondern eher die eines hügeligen Gebirgszuges. Trotz aller Unregelmäßigkeit sieht man in jedem Bild von Abbildung 2.7, dass die Masse dreigeteilt ist. Dies bestätigt noch einmal unsere Beobachtung, dass in Zeitintervallen, in denen die Spikes annähernd regelmäßig auftreten, drei Perioden des Inputs einer Periode des Outputs entsprechen. In manchen Bildern (zum Beispiel für $j = 9, 10, 11$) sieht es zwar so aus, als wäre die Masse zweigeteilt, aber das liegt nur daran, dass bei dem gewählten Blickwinkel der große Berg die Sicht auf den kleineren verstellt.





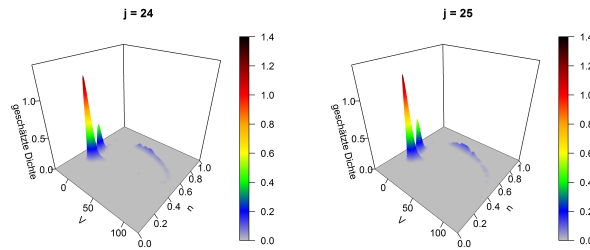


Abbildung 2.7.: Unregelmäßige Spikes: Kerndichteschätzer für ψ_s zur Toruszeit s_j , $j = 0, 1, \dots, 25$, Projektion auf (V, n)

2.2.4. Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig

Abbildung 2.8 wurde auf die gleiche Art erstellt wie Abbildung 2.4. Das bedeutet, der Prozess wurde über 110 Perioden des Inputs simuliert und für die grafische Darstellung wurden die dieselben Funktionen mit denselben Parametern verwendet. Um die beiden Abbildungen vergleichen zu können, wurden in jedem der sechs Bilder jeweils die gleichen Grenzen für die beiden Achsen und die Farbskala gewählt.

Die Erkenntnisse aus Abbildung 2.8 werden vor allem durch den Vergleich mit Abbildung 2.4 deutlich. Da die Spikes im aktuellen Beispiel nicht periodisch auftreten, sind die Orbits in Abbildung 2.8 nicht so klar definiert wie in Abbildung 2.4, sondern sehen unschärfer aus. Das erkennt man gut, wenn man die beiden Projektionen auf (V, h) miteinander vergleicht: In Abbildung 2.4 gibt es einen präzisen dunkelroten Punkt, auf den ψ_s für viele $s \in \mathbb{T}$ Gewicht legt, aber in Abbildung 2.8 gibt es einen größeren gelb-orangen Bereich, auf den sich die Masse verteilt. Eine weitere Beobachtung, die wir beispielsweise durch die Projektionen auf (V, h) oder (m, h) machen, ist, dass die Orbitlinien in Abbildung 2.8 blasser sind und häufiger abbrechen als in Abbildung 2.4. Dies verdeutlicht noch einmal, dass es in diesem Beispiel im zeitlichen Verlauf von ψ_s eine größere Ortsunschärfe gibt als im vorherigen Beispiel, was durch die höhere Zufallsabhängigkeit beim Auftreten der Spikes zu erklären ist.

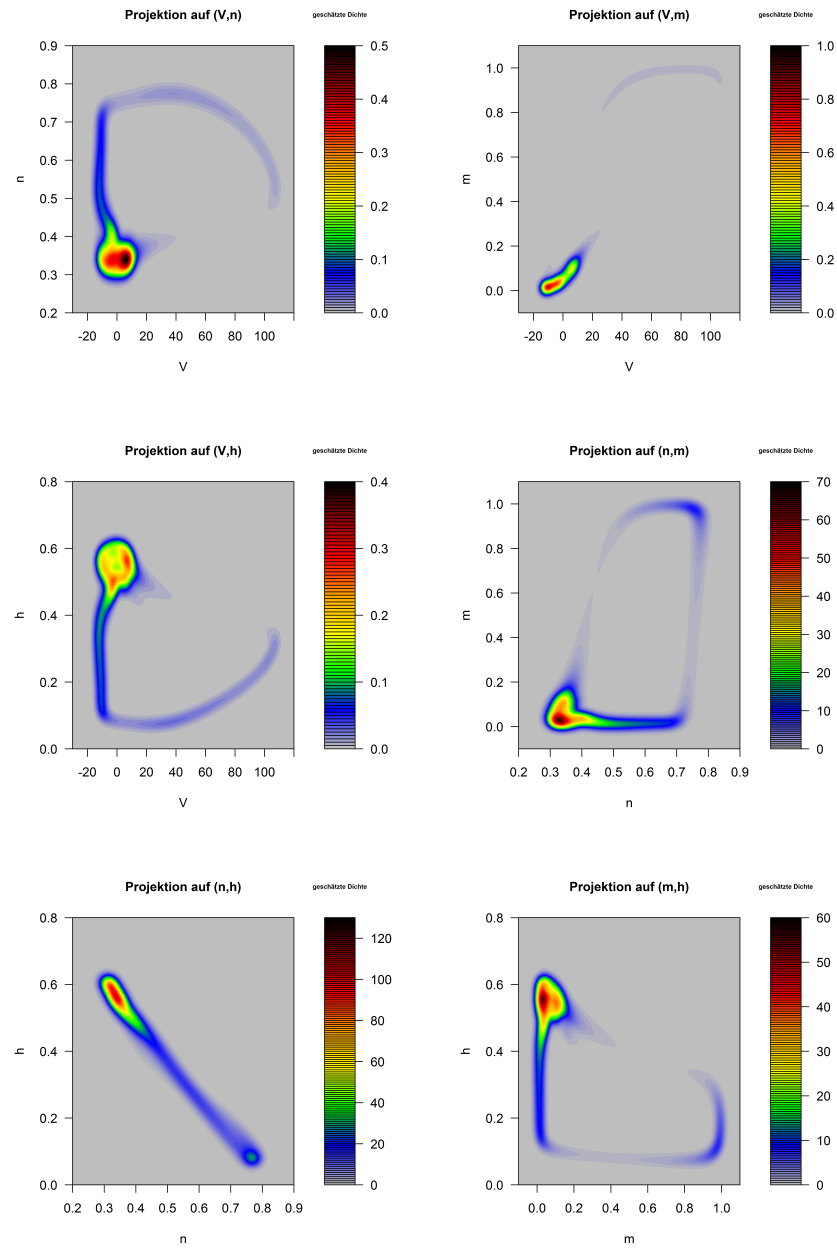


Abbildung 2.8.: Unregelmäßige Spikes: Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig, Projektionen auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h)

2.3. Beispiel mit Dirichletkern als Input

Im letzten Beispiel verwenden wir den vierten Dirichletkern $D_4(\frac{2\pi}{T}\cdot)$ mit Periodizität $T = 10$ als Inputfunktion. Als Amplitudenfaktor wählen wir hauptsächlich $a = 6$, aber vergleichen auch teilweise das Verhalten der Lösung X für $a \in \{4, 5, 6, 7\}$. Für die Diffusionsgeschwindigkeit wird wie in den vorherigen Beispielen $\tau = 0,75$ festgelegt, aber wir untersuchen dieses Mal, welche Auswirkungen eine deutlich höhere Streuung von $\gamma = 3$ hat.

2.3.1. Abbildung von $X = (V, n, m, h, \xi)$ in Abhängigkeit von t

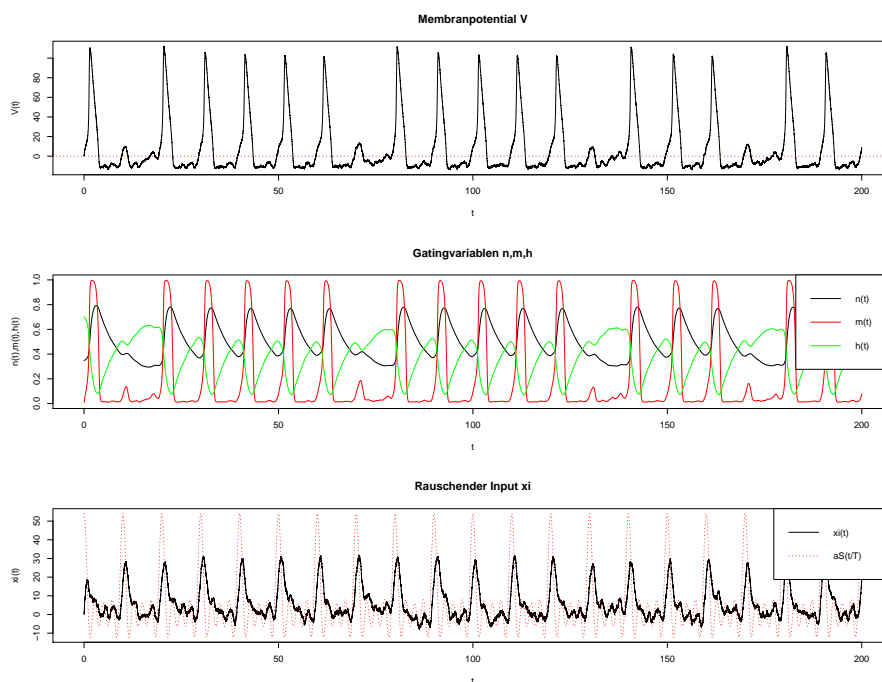


Abbildung 2.9.: Dirichletkern mit $a = 6$ als Input: Werte von V , (n, m, h) und ξ in Abhängigkeit von t für die ersten 20 Periodizitätsintervalle des Inputs

Für Abbildung 2.9 wurde der Prozess X über 20 Periodizitätsintervalle des Inputs mit $a = 6$ simuliert. Man sieht, dass es in diesem Beispiel Abschnitte

gibt, in denen es in jedem Periodizitätsintervall des Inputs zu einem Spike kommt. Deshalb gehen wir von einer Periodizität des Outputs von $T_{\text{out}} = 10$ aus. Wenn man den gesamten simulierten Verlauf betrachtet, sieht man aber auch, dass immer wieder Spikes ausfallen.

Außerdem ist im dritten Bild von Abbildung 2.9 das durch $\gamma = 3$ verursachte, starke Rauschen von ξ erkennbar. Dieses wirkt sich auch auf das Verhalten von V zwischen den Spikes aus. Im Unterschied zu den Abbildungen 2.1 und 2.5 sieht der Graph von V an diesen Stellen unscharf aus.

Zum Vergleich der periodischen Strukturen in Abhängigkeit von a wird der Verlauf von X in Abbildung 2.10 bei Wahl von $a = 4$ und in Abbildung 2.11 bei Wahl von $a = 7$ dargestellt. Man erkennt sofort, dass die Spikes in beiden Fällen nahezu regelmäßig auftreten, aber die Periodizitäten des Outputs unterscheiden sich: Für $a = 4$ ist $T_{\text{out}} = 20$, aber für $a = 7$ ist $T_{\text{out}} = 10$. Weitere Simulationen haben gezeigt, dass die Spikes für Werte von $a < 4$ langsam seltener und unregelmäßiger werden, aber auch bei sehr kleinen Amplitudenfaktoren (zum Beispiel $a = 0,1$) kommt es noch immer wieder zu einzelnen Spikes, die durch den hohen Wert von γ ausgelöst werden. Bei der Wahl von einem beliebig großen $a > 7$ bleibt die Outputperiodizität von 10 bestehen. Es ist deshalb besonders interessant, Amplitudenfaktoren wie $a = 5$ oder $a = 6$ zu betrachten, denn für $a \in (4, 7)$ kommt es zum Übergang von $T_{\text{out}} = 20$ zu $T_{\text{out}} = 10$. Während wir uns im schon betrachteten Fall $a = 6$ auf $T_{\text{out}} = 10$ festgelegt haben, gilt bei $a = 5$ eher noch $T_{\text{out}} = 20$. Alle Festlegungen der Outputperiodizitäten beruhen nur auf empirischen Beobachtungen, werden aber durch die noch folgenden Darstellungen bestätigt werden.

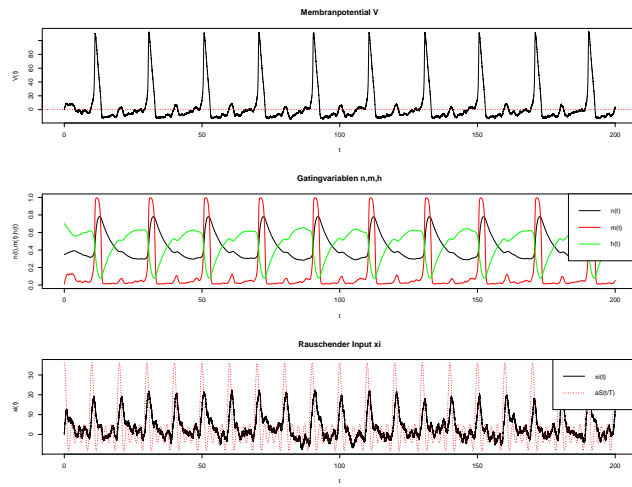


Abbildung 2.10.: Dirichletkern mit $a = 4$ als Input: Werte von V , (n, m, h) und ξ in Abhängigkeit von t für die ersten 20 Periodizitätsintervalle des Inputs

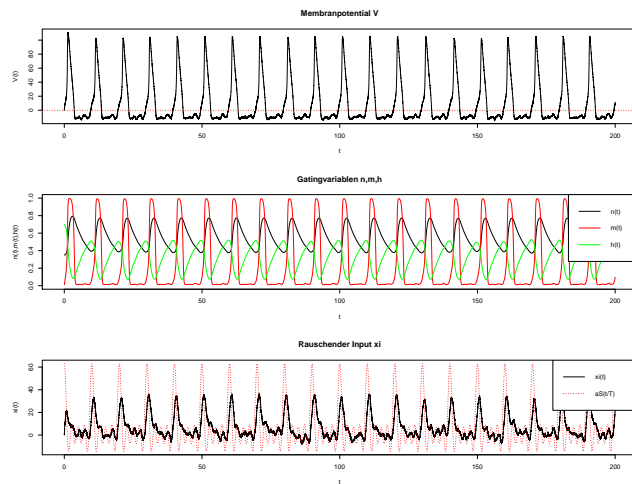


Abbildung 2.11.: Dirichletkern mit $a = 7$ als Input: Werte von V , (n, m, h) und ξ in Abhängigkeit von t für die ersten 20 Periodizitätsintervalle des Inputs

2.3.2. Projektionen von X auf zweidimensionale Ebenen

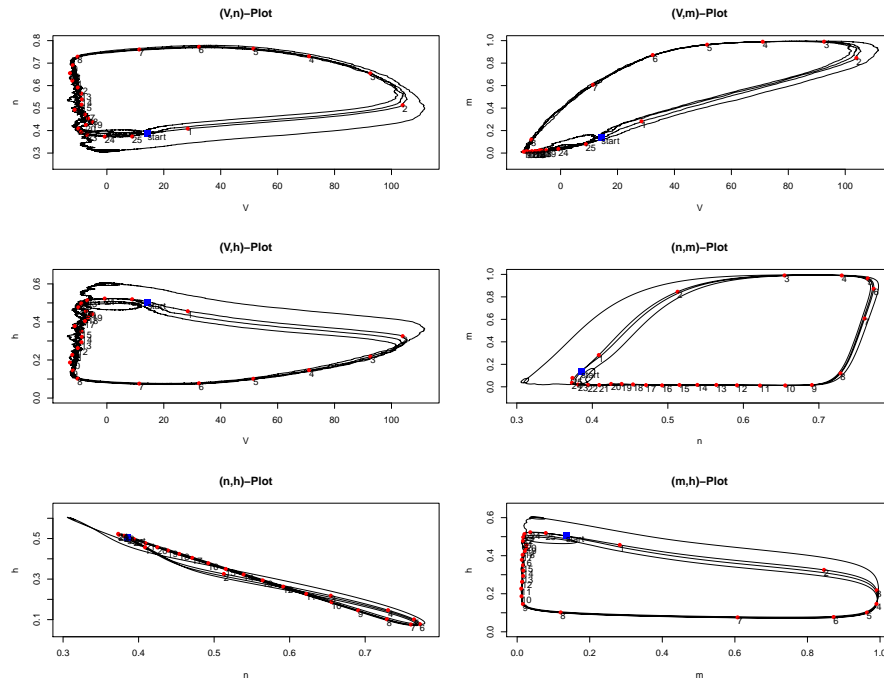


Abbildung 2.12.: Dirichletkern mit $a = 6$ als Input: Projektionen von $X = (V, n, m, h, \xi)$ auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h) für die letzten fünf Periodizitätsintervalle des Outputs

Für die Erstellung von Abbildung 2.12 wurde X bei einer Wahl von $a = 6$ über 20 Zeitintervalle der Länge 10 simuliert, da wir in diesem Fall von $T_{\text{out}} = 10$ ausgehen. Gezeichnet wurde der Verlauf von $X = (X_t)_{t \geq 0}$ in den letzten fünf Zeitintervallen, das heißt für $t \in [150, 200]$. Wegen Abbildung 2.9 wissen wir, dass es in diesem Zeitraum vier Spikes gab, und deshalb sind in mehreren Bildern von Abbildung 2.12 vier separate Orbitlinien zu erkennen. Außerdem macht sich auch in dieser Darstellungsart der Einfluss von $\gamma = 3$ bemerkbar. Insbesondere auf der linken Seite im (V, n) - und (V, h) -Plot zittern die Linien deutlich stärker als in den vorherigen Beispielen.

In einigen Bildern von Abbildung 2.12 lässt sich eine Zweiteilung der Orbits

erahnen. Im (V, n) -, (V, h) -, (n, m) - und (m, h) -Plot bilden jeweils drei Orbitlinien ein Bündel und die vierte verläuft mit größerem Abstand zu ihnen. Um diese Beobachtung genauer zu untersuchen, wurde Abbildung 2.13 erstellt. Für $a \in \{4, 5, 6, 7\}$ wurde dafür der Prozess X jeweils über 100 Perioden des Inputs simuliert (unabhängig von den zuvor festgestellten Periodizitäten des Outputs). Gezeichnet wurde dann der Verlauf von (n, m) in den letzten 95 Periodizitätsintervallen, denn bei der Projektion auf diese beiden Komponenten ist das zu untersuchende Phänomen besonders deutlich zu erkennen. Um die vier Bilder gut vergleichen zu können, wurden die Grenzen der Achsen

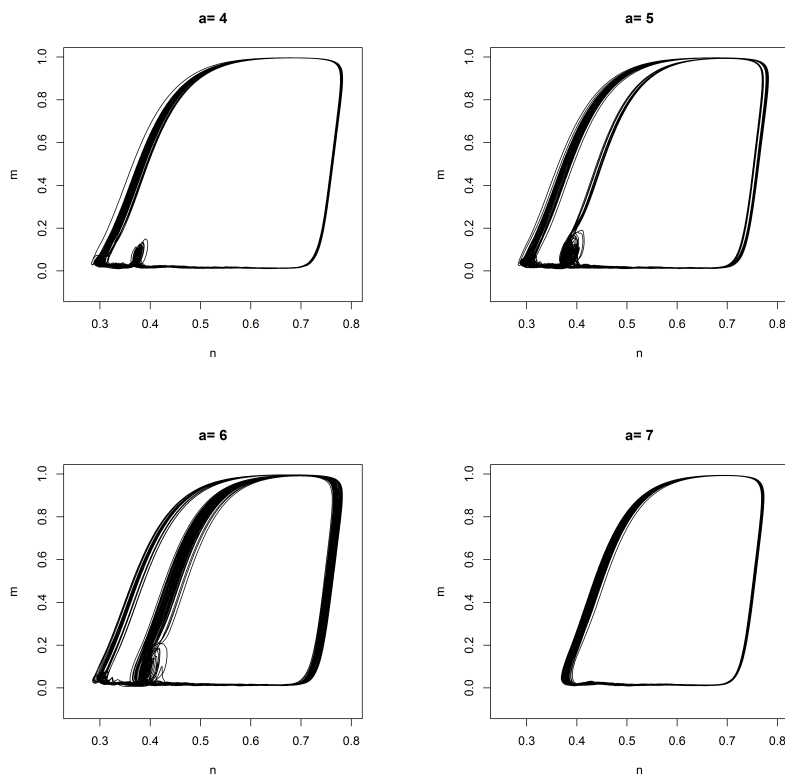


Abbildung 2.13.: Dirichletkern mit $a \in \{4, 5, 6, 7\}$ als Input: Projektionen von X auf (n, m) für die letzten 95 Periodizitätsintervalle des Inputs

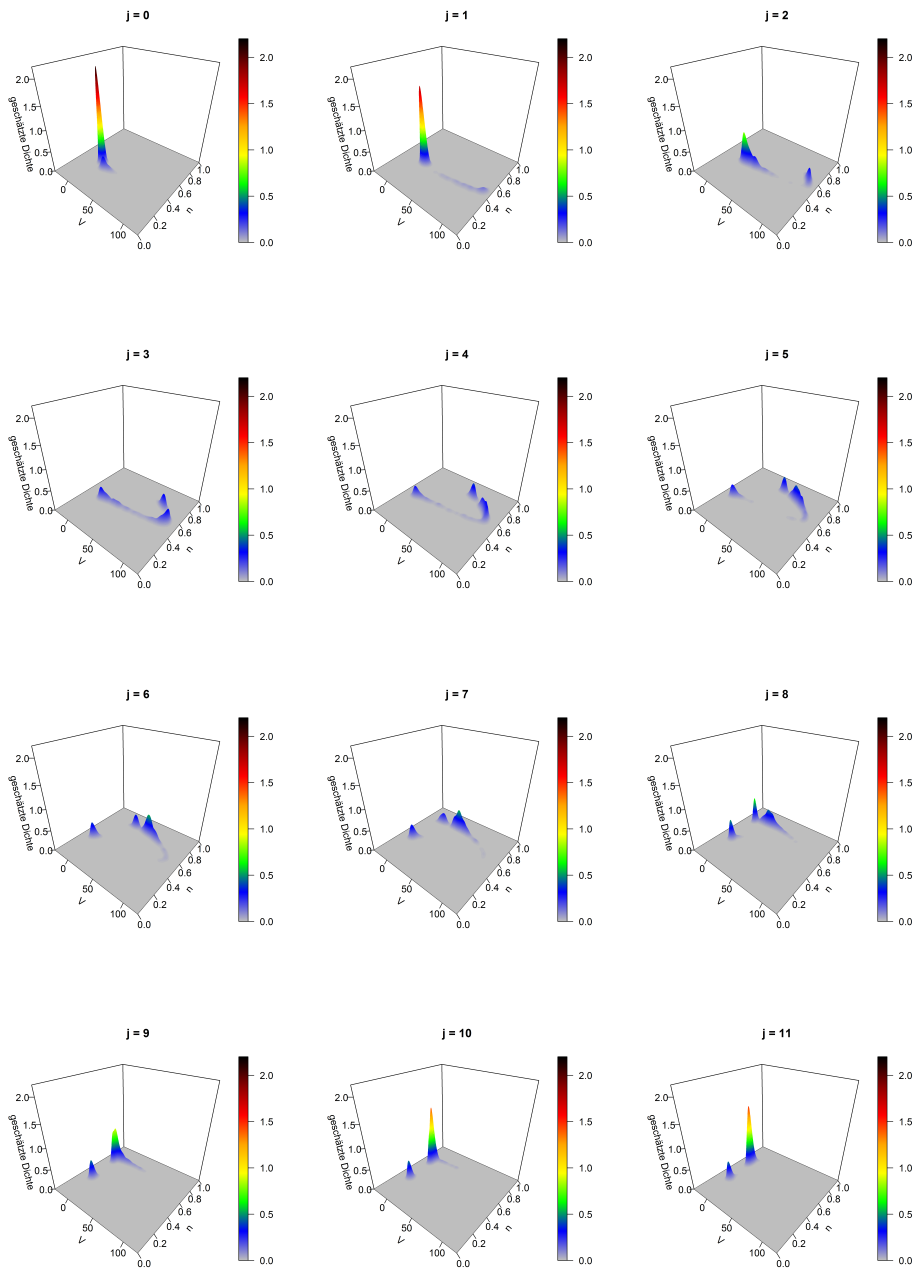
einheitlich festgelegt. Außerdem wurde darauf verzichtet, die roten Punkte auf dem letzten Orbit einzuzeichnen, damit der Fokus auf dem Verlauf der Orbits liegt und die Bilder nicht zu unübersichtlich werden.

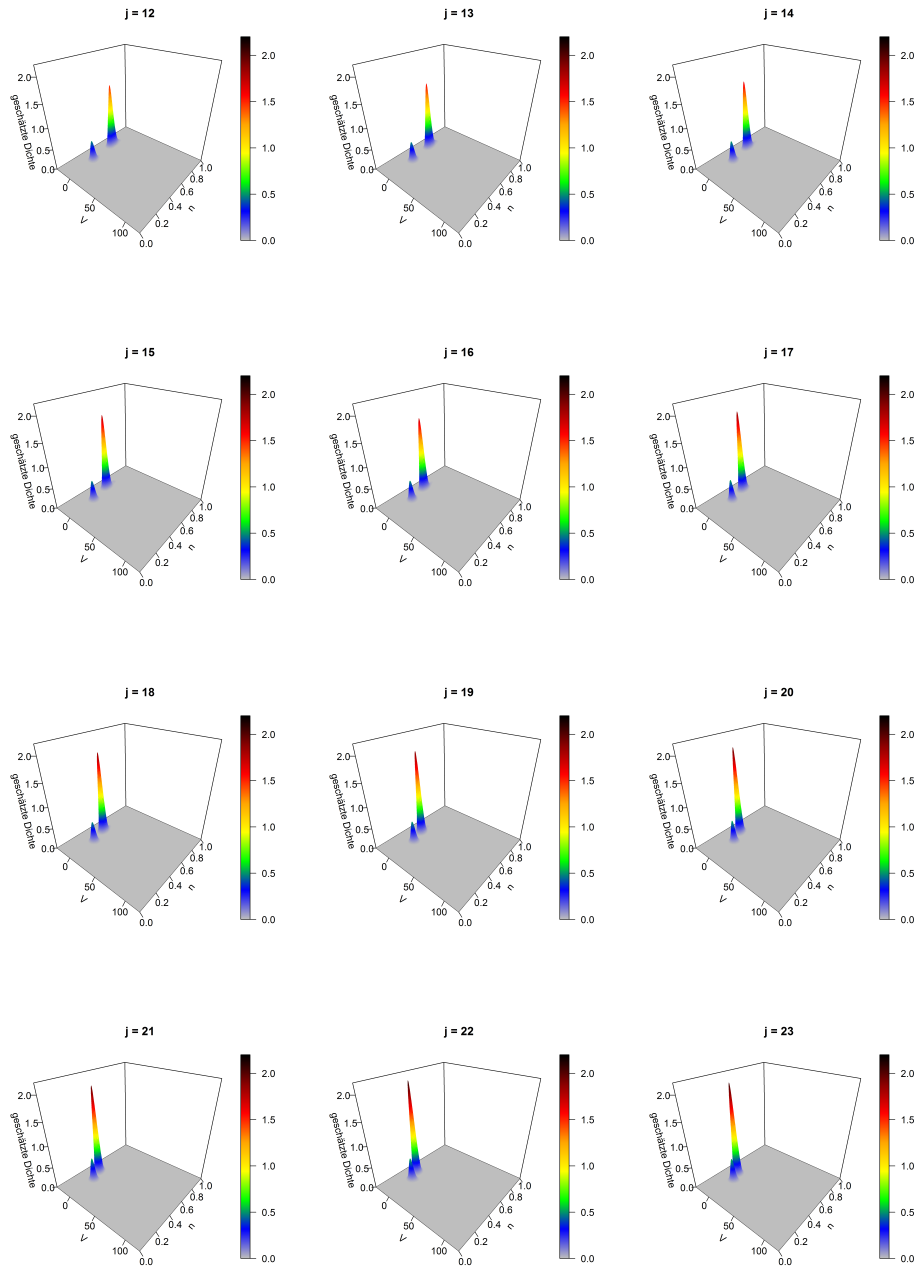
In Abbildung 2.13 ist nun zu erkennen, wie sich die Orbits von (n, m) mit wachsendem a verändern. Für $a = 4$ sieht man auf der linken Seite neben dem breiten Streifen einen kleinen Zipfel. Dieser gehört zu den Periodizitätsintervallen des Inputs, in denen es keinen Spike gab, denn bei Betrachtung von Abbildung 2.10 sieht man, dass es zwischen zwei Spikes immer zu einer kleinen Erhöhung von m kommt. Im Gegensatz dazu trifft der breite Streifen im Bild für $a = 7$ (Abbildung 2.13) genau dort auf, wo im Bild für $a = 4$ der kleine Zipfel war. Das liegt daran, dass es für $a = 7$ in jedem Periodizitätsintervall des Inputs zu einem Spike kommt und es somit kein Zwischenverhalten gibt. Die übrigen beiden Bilder, in denen die Zweiteilung der Orbits deutlich zu erkennen ist, zeigen, dass man sich für $a = 5$ und $a = 6$ in einer Übergangsphase befindet. Der Prozess X zeigt für solche Werte von a teilweise das gleiche periodische Verhalten wie bei $a = 4$, teilweise aber auch ein Verhalten wie bei $a = 7$. Abbildung 2.13 liefert somit wichtige Erkenntnisse über ψ_s , $s \in \mathbb{T}$. Wir wissen nun in Abhängigkeit von a , auf welchem Orbit sich die von ψ_s hingelegte Masse in der Projektion auf (n, m) für $s \in \mathbb{T}$ bewegen wird.

2.3.3. Kerndichteschätzer für ψ_s zu 26 Zeitpunkten im Torus

Um die einzelnen Bilder von Abbildung 2.14 zu erstellen, wurde X jeweils über $K = 500$ Periodizitätsintervalle des Inputs mit $a = 6$ simuliert. Durch diese Wahl von K wurde sichergestellt, dass auch in diesem Beispiel über mindestens 250 Periodizitätsintervalle des Outputs simuliert wurde. Für die Berechnung und Zeichnung des Kerndichteschätzers wurden wieder dieselben Funktionen in R mit denselben Parametern wie in den vorherigen Beispielen verwendet und wir betrachten wieder nur die Projektion auf (V, n) .

Man erkennt auf den Bildern für die Toruszeiten s_j , $j = 0$ und $j = 9, \dots, 25$, dass sich die bewegliche Masse jeweils in zwei separate, unterschiedlich hohe Berge aufteilt. Der größere Berg bewegt sich dabei im Laufe der Zeit mehr als der kleine Hügel, denn dieser bleibt stets ungefähr an der gleichen Stelle.





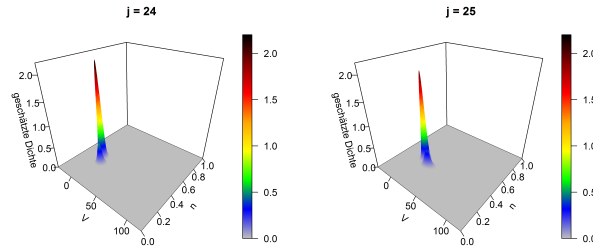


Abbildung 2.14.: Dirichletkern mit $a = 6$ als Input: Kerndichteschätzer für ψ_s zur Toruszeit s_j , $j = 0, 1, \dots, 25$, Projektion auf (V, n)

Deshalb steht der große Berg für die Inputperioden, in denen X gespiket hat, und der kleine Hügel für die, in denen der Spike ausgefallen ist. Der Höhenunterschied der beiden Massen verdeutlicht noch einmal, dass das periodische Verhalten im Fall $a = 6$ näher an das Verhalten im Fall $a = 7$ als im Fall $a = 4$ herankommt.

Betrachtet man die Bilder zu den Toruszeiten s_j , $j = 1, \dots, 8$, so erkennt man, dass die bewegliche Masse hier von Bild zu Bild unterschiedlich aussieht und allgemein hügeliger und zusammenhängender als in den anderen Bildern ist. Die Masse legt in diesen Toruszeiten die größte Strecke auf dem Orbit aus dem ersten Bild von Abbildung 2.12 zurück, was bedeutet, dass sie sich mit hoher Geschwindigkeit bewegt. Allerdings erkennt man auch in diesen Bildern den kleinen, unbeweglichen Hügel, der wie zuvor beschrieben für die Inputperioden ohne Spike steht. Dass ψ_s in diesen Toruszeiten so chaotisch aussieht, zeigt, dass die Spikes in diesem Beispiel zwar häufig auftreten, aber nicht jedes Mal exakt gleich aussehen.

Zur Toruszeit s_4 ist ein besonders interessantes Bild entstanden, da die Masse dort auf große Teile des Orbits verteilt wurde. Deshalb vergleichen wir das Verhalten zu dieser Zeit mit dem Verhalten bei Wahl von $a = 4, 5, 7$ (siehe Abbildung 2.15). Für diese Abbildung wurde der Prozess X für jedes $a \in \{4, 5, 6, 7\}$ über 500 Inputperioden simuliert, und gezeichnet wurde nur der Kerndichteschätzer für ψ_{s_4} . Auf dem Bild für $a = 4$ sieht man zwei separate Hügel, während die Masse auf dem Bild für $a = 7$ einen einzelnen

Berg bildet. Dies passt zu unserer Beobachtung, dass es für $a = 7$ in jedem Periodizitätsintervall des Inputs zu einem Spike kommt, aber für $a = 4$ nur in jedem zweiten. Auch in dieser Darstellung wird deutlich, dass man sich für $a = 5$ und $a = 6$ in einer Übergangsphase der Outputperiodizitäten befindet. Im Bild für $a = 5$ ist die Zweiteilung der Masse noch gut zu erkennen und man sieht, dass sich die beiden Hügel ungefähr an den gleichen Stellen befinden wie im Bild für $a = 4$. Im Bild für $a = 6$ kommt dann noch ein dritter Schwerpunkt von Masse hinzu. Weitere Simulationen, die hier nicht dargestellt sind, haben gezeigt, dass dieser mit wachsendem $a \in [6, 7]$ immer größer wird, während die beiden ursprünglichen Hügel immer kleiner werden. Dadurch ist für $a = 7$ schlussendlich nur noch ein einzelner Berg zu sehen.

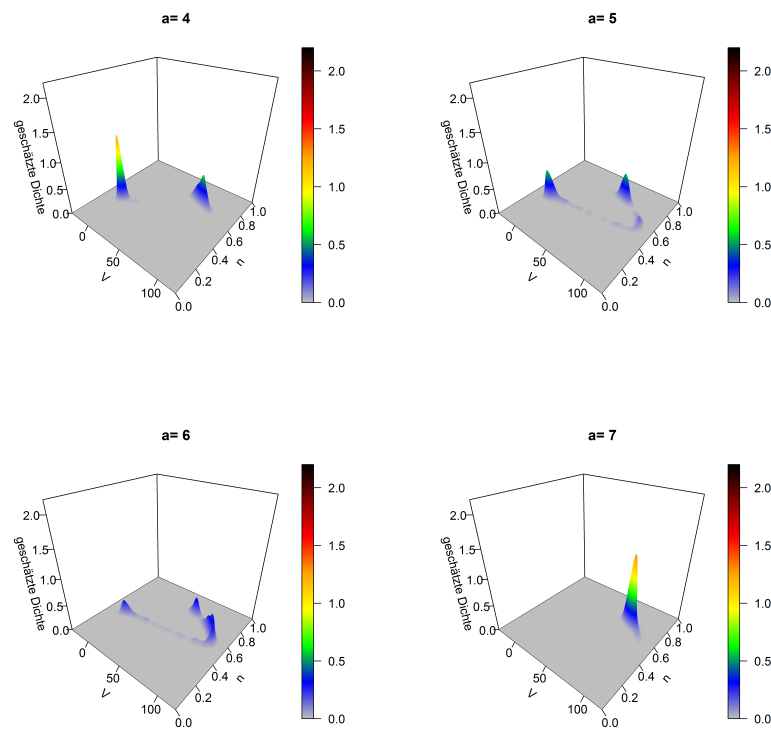


Abbildung 2.15.: Dirichletkern mit $a \in \{4, 5, 6, 7\}$ als Input: Kerndichteschätzer für ψ_s zur Toruszeit s_4 , Projektion auf (V, n)

2.3.4. Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig

Für die Erstellung der einzelnen Bilder von Abbildung 2.16 wurde der Prozess X jeweils über 110 Periodizitätsintervalle des Inputs mit $a = 6$ simuliert. Unsere bisherigen Erkenntnisse über das zeitliche Verhalten von ψ_s werden durch diese Darstellung noch einmal bestätigt. Wie bereits durch Abbildung 2.12 und 2.13 untersucht wurde, ist auch hier in den Projektionen auf (V, n) , (V, h) , (n, m) und (m, h) eine Zweiteilung der Orbits erkennbar, wobei immer ein Teil der Orbits stärker ausgeprägt ist als der andere. Man sieht auch, dass die Masse in den meisten Bildern von Abbildung 2.16 weniger auf einzelne Punkte konzentriert ist, denn es gibt dort größere dunkelrote Bereiche. Der Grund dafür ist wieder das durch die Wahl von a bedingte unregelmäßige Auftreten und unterschiedliche Aussehen der Spikes. Außerdem sind die Orbitlinien in dieser Abbildung breiter als beispielsweise in Abbildung 2.4. Dies lässt sich unter anderem durch den Einfluss des großen Wertes von γ erklären, der zu einem stärkeren Rauschen des Lösungsprozesses führt.

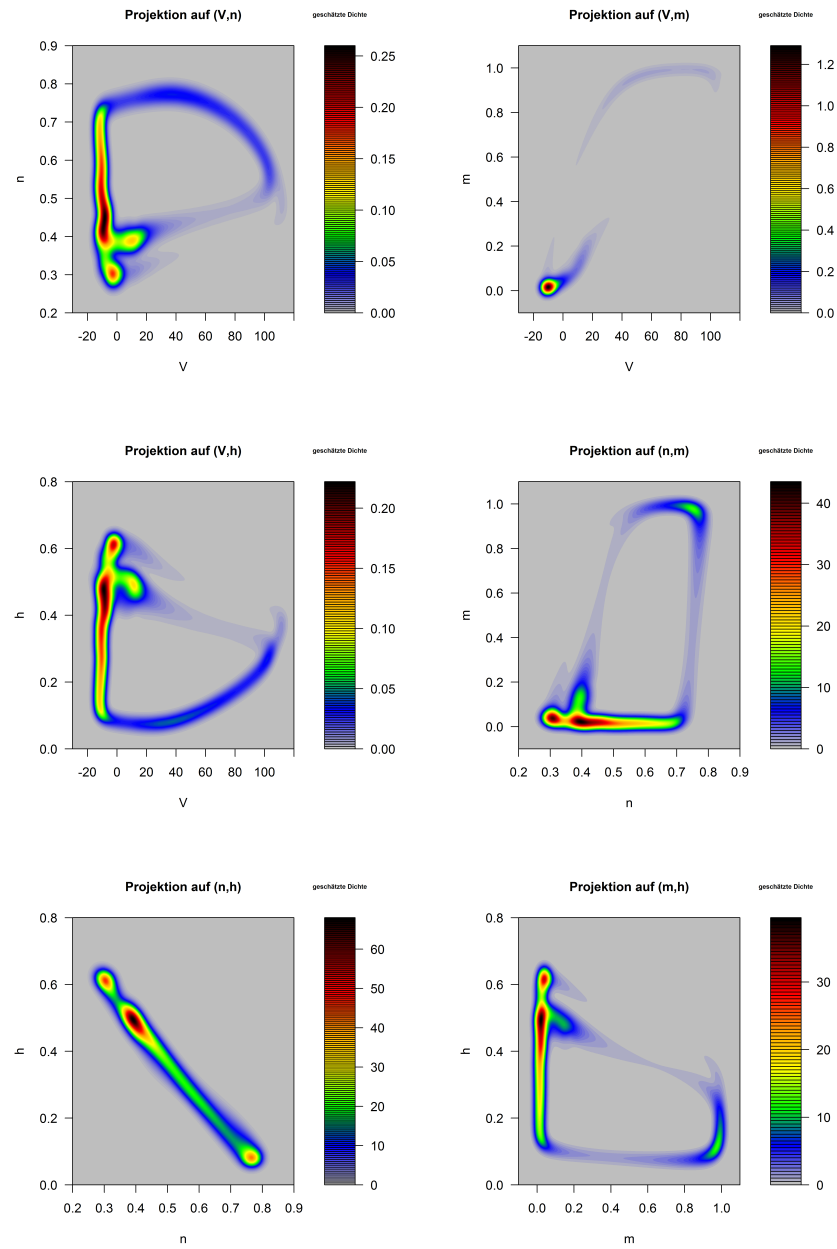


Abbildung 2.16.: Dirichletkern mit $a = 6$ als Input: Kerndichteschätzer für ψ_s zu vielen Toruszeitpunkten gleichzeitig, Projektionen auf (V, n) , (V, m) , (V, h) , (n, m) , (n, h) und (m, h)

2.4. Bezug zum deterministischen Hodgkin-Huxley-Modell

Dieser letzte Abschnitt, der als Ergänzung zu den in Abschnitt 2.1 und 2.3 betrachteten Beispielen zu verstehen ist, führt uns vom stochastischen Hodgkin-Huxley-Modell zurück zum deterministischen Modell.

Sei dafür a der Amplitudenfaktor, S die 1-periodische Inputfunktion, T die Periodizität, τ die Diffusionsgeschwindigkeit und γ die Streuung aus dem stochastischen Hodgkin-Huxley-Modell aus Definition 1.4, wobei S auf ganz \mathbb{R} fortgesetzt wird. Mit einigen von diesen Parametern wird eine ebenfalls T -periodische Funktion

$$R(t) := \int_{-\infty}^t \tau e^{-\tau(t-s)} S\left(\frac{s}{T}\right) ds, \quad t \geq 0 \quad (2.6)$$

definiert.

Sei μ_5 das Bildmaß der invarianten Verteilung μ von der Gitterkette X^G unter einer Projektion auf \mathbb{R} , das heißt auf die fünfte Komponente von E_5 . Für den Lösungsprozess ξ der fünften Gleichung von (1.1) bei Start zur Zeit $t = 0$ in der invarianten Verteilung μ_5 gilt laut [11, Remark 1], dass ξ_t normalverteilt mit Mittelwert $aR(t)$ und Varianz $\frac{a^2\gamma^2}{2}$ ist. Das heißt in anderen Worten, dass $aR(t)$ der Mittelwert des stationär verlaufenden Prozesses ξ ist. Bei Wahl einer verschwindend kleinen Streuung γ ist es deshalb naheliegend, $d\xi_t$ in der ersten Gleichung von (1.1) durch $aR'(t) dt$ zu ersetzen und somit aR' als Inputsignal für das deterministische Hodgkin-Huxley-Modell zu verwenden. Das bedeutet, dass das Modell

$$\begin{cases} dV_t = aR'(t) dt - F(V_t, n_t, m_t, h_t) dt \\ dn_t = [\alpha_n(V_t)(1 - n_t) - \beta_n(V_t)n_t] dt \\ dm_t = [\alpha_m(V_t)(1 - m_t) - \beta_m(V_t)m_t] dt \\ dh_t = [\alpha_h(V_t)(1 - h_t) - \beta_h(V_t)h_t] dt \end{cases} \quad (2.7)$$

betrachtet wird, wobei F wie in (1.4) definiert ist.

Nun sollen anhand der beiden Beispiele aus Abschnitt 2.1 und 2.3 die Lö-

sungsprozesse von (1.1) und (2.7) verglichen werden und es soll deutlich werden, welcher Zusammenhang zwischen Orbits des stochastischen und des deterministischen Modells besteht.

2.4.1. Beispiel mit regelmäßigen Spikes

Zuerst wählen wir wie in Abschnitt 2.1 $a = 25$ als Amplitudenfaktor, die 1-periodische Sinusfunktion für S , $T = 5$ als Periodizität und $\tau = 0,75$ als Diffusionsgeschwindigkeit.

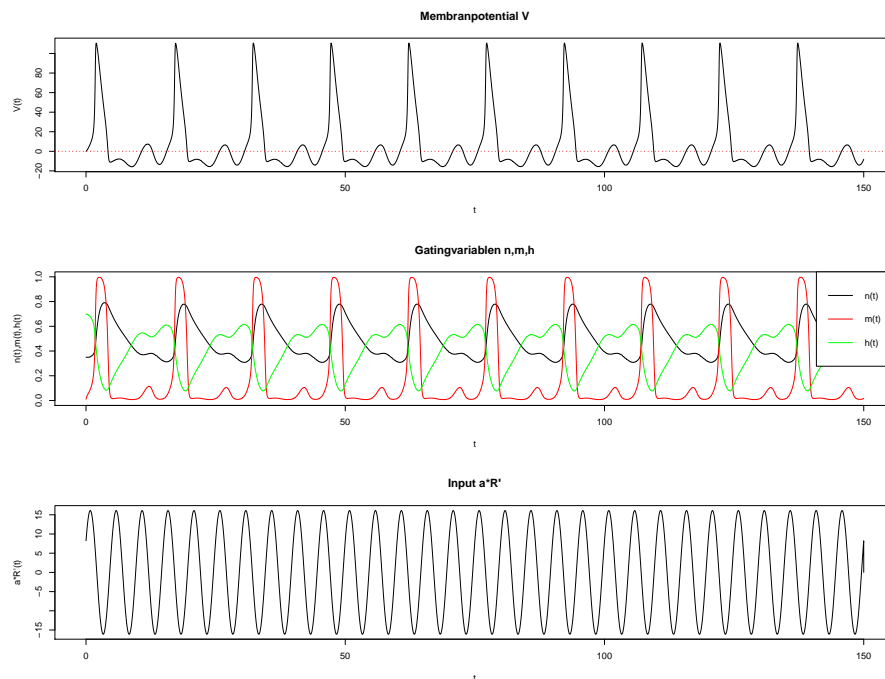


Abbildung 2.17.: Deterministisches Modell mit regelmäßigen Spikes: Werte von V , (n, m, h) und aR' in Abhängigkeit von t für die ersten 30 Periodizitätsintervalle des Inputs

Um einen Eindruck von dem Lösungsprozess von (2.7) zu bekommen, wurde das Modell zunächst über 30 Perioden des Inputs simuliert. In Abbildung 2.17 werden die Werte von V , (n, m, h) und vom Input aR' in Abhängigkeit von t

dargestellt. Den Code für die Erstellung der Plots findet man in A.9, Zeile 48-60.

Man erkennt an der Abbildung sofort, dass die Spikes regelmäßig auftreten und dass eine große Ähnlichkeit zu Abbildung 2.1 besteht. Dementsprechend gibt es auch im deterministischen Fall eine Periodizität des Outputs von $T_{\text{out}} = 15$. Ein Unterschied zum Verhalten im stochastischen Fall wird jedoch deutlich, wenn man das Modell mit anderen Werten von a simuliert. Während das stochastische Modell (mit $\gamma = 0,5$) schon bei Werten von ungefähr $a = 9,5$ erste Spikes produziert hat, treten im deterministischen Modell für Werte von $a < 11,93$ in den ersten 100 Periodizitätsintervallen des Inputs keine Spikes auf. Dies lässt sich durch den Einfluss von γ im stochastischen Modell erklären, denn γ unterstützt das Auslösen der Spikes.

Um die Lösungsprozesse von (1.1) und (2.7) noch besser vergleichen zu können, wurde Abbildung 2.18 erstellt. Für die drei Bilder auf der linken Seite wurde das stochastische Hodgkin-Huxley-Modell jeweils über 60 Perioden des Outputs simuliert und gezeichnet wurde der Verlauf der Komponenten (V, n) , (n, m) und (m, h) in den letzten 50 Periodizitätsintervallen des Outputs. Dabei wurde wie in Abschnitt 2.1 $\gamma = 0,5$ gesetzt. Es wurde im Gegensatz zu vorherigen Abbildungen dieser Art darauf verzichtet, die Punkte auf dem letzten Orbit einzuzichnen, damit der Verlauf der Orbits bestmöglich zu erkennen ist. Für die drei Bilder auf der rechten Seite wurde das deterministische Modell jeweils über 60 Perioden des Outputs simuliert und die Erstellung der Bilder erfolgte wieder auf die eben beschriebene Art. Den Code hierzu findet man in A.10.

In Abbildung 2.18 ist deutlich zu erkennen, wie ähnlich in diesem Beispiel die Orbits des stochastischen und des deterministischen Modells verlaufen. In Bezug auf Form und Position der Orbits sind in allen drei Projektionen kaum Unterschiede zu erkennen. Das Einzige, das die Plots auf der rechten und linken Seite unterscheidet, ist, dass die Orbitlinien in den stochastischen Plots etwas breiter sind. Dies ist durch das von γ verursachte Rauschen zu erklären, aber da in diesem Beispiel ein relativ kleiner Wert von γ gewählt wurde, ist nur ein schwaches Rauschen vorhanden.

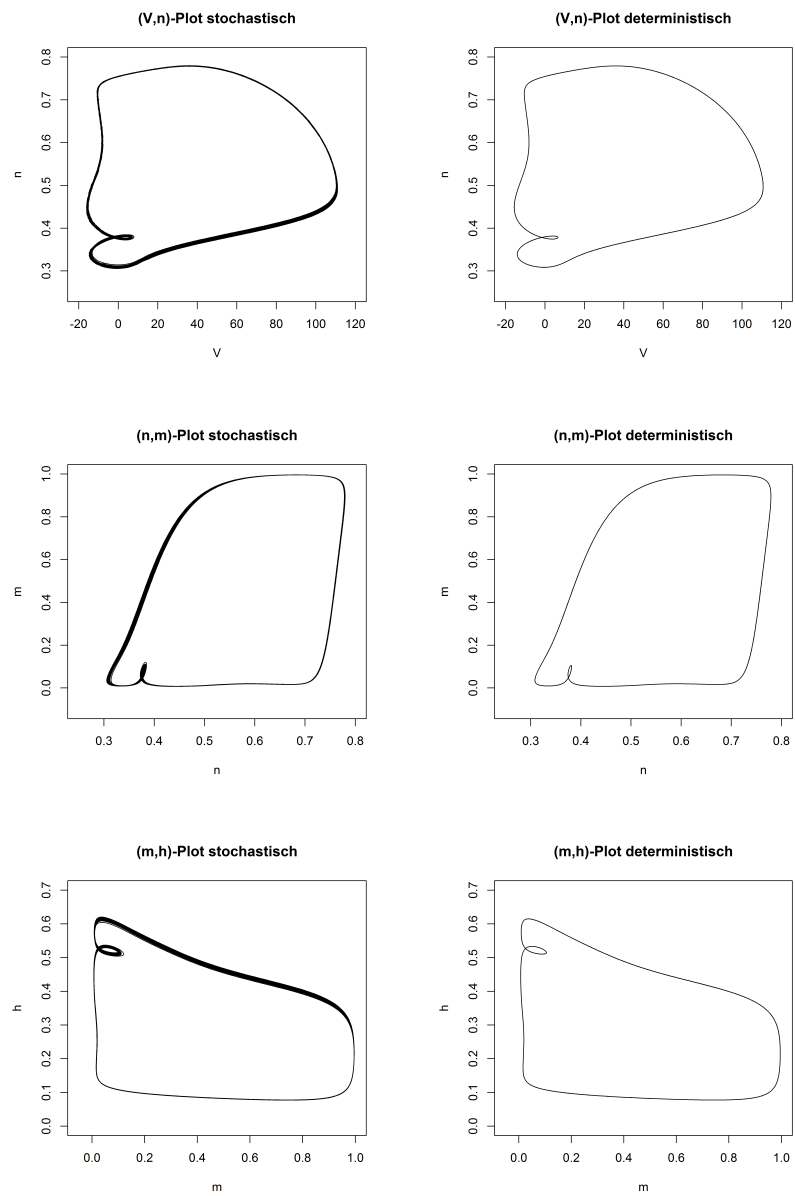


Abbildung 2.18.: Regelmäßige Spikes: Vergleich des stochastischen und des deterministischen Modells anhand von Projektionen von X auf (V, n) , (n, m) und (m, h) für die letzten 50 Periodizitätsintervalle des Outputs

2.4.2. Beispiel mit Dirichletkern als Input

Als Nächstes betrachten wir das Beispiel aus Abschnitt 2.3, das heißt $a = 6$, $S(\cdot) = D_4(\frac{2\pi}{T}\cdot)$, $T = 10$ und $\tau = 0,75$. Der Lösungsprozess von (2.7) bei Wahl dieser Parameter ist in Abbildung 2.19 dargestellt, für die das Modell über 20 Perioden des Inputs simuliert wurde.

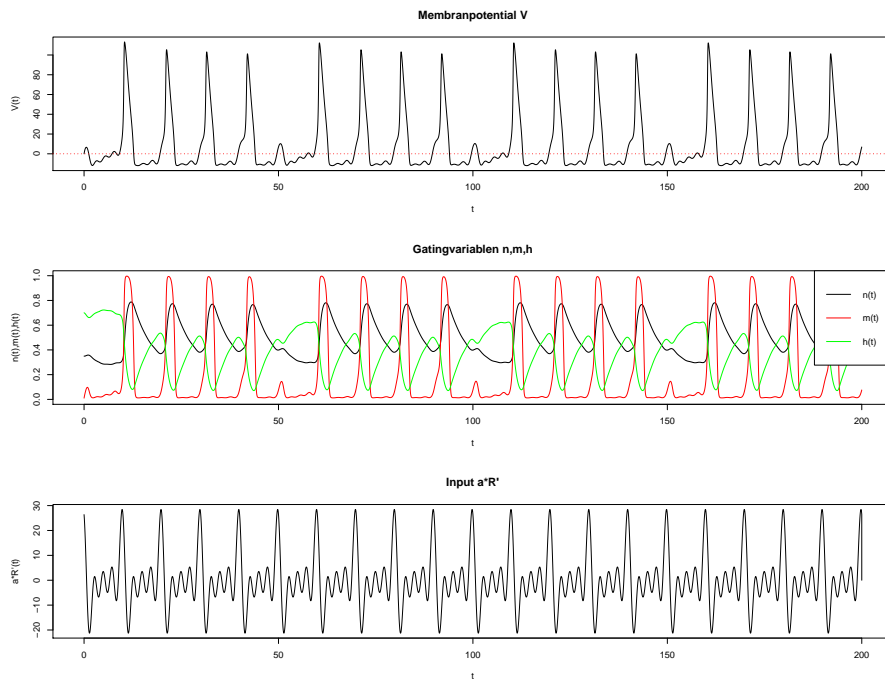


Abbildung 2.19.: Deterministisches Modell mit Dirichletkern als Input: Werte von V , (n, m, h) und aR' in Abhängigkeit von t für die ersten 20 Periodizitätsintervalle des Inputs

Man erkennt an der Abbildung, dass es regelmäßig in einem Periodizitätsintervall des Inputs zu einem Spike kommt, aber auch, dass jeder fünfte Spike ausfällt. Außerdem werden die maximalen Werte von V innerhalb einer Vierergruppe von Spikes mit jedem Spike kleiner. Diese beiden Beobachtungen bestätigen sich auch bei Simulation des Prozesses über 100 Perioden des Inputs. Vergleicht man Abbildung 2.19 mit Abbildung 2.9, so fällt auf, dass

dort zwar auch immer wieder einzelne Spikes ausfallen, jedoch ohne eine bestimmte Regelmäßigkeit. Wenn wir in beiden Abbildungen nur Intervalle betrachten, in denen die Spikes regelmäßig auftreten, können wir aber trotzdem im stochastischen und im deterministischen Fall von $T_{\text{out}} = 10$ ausgehen.

Um die beobachteten Gemeinsamkeiten und Unterschiede weiter zu untersuchen, wurde Abbildung 2.20 erstellt. Dafür wurden das stochastische und das deterministische Hodgkin-Huxley-Modell wieder jeweils dreimal über 60 Perioden des Outputs simuliert und die Projektionen des jeweiligen Lösungsprozesses auf (V, n) , (n, m) und (m, h) wurden für die letzten 50 Periodizitätsintervalle des Outputs dargestellt. Wie in Abschnitt 2.3 wurde für die Simulation des stochastischen Modells $\gamma = 3$ gewählt.

Man sieht anhand von Abbildung 2.20, dass sich die Orbits des stochastischen und des deterministischen Modells auch in diesem Beispiel in Bezug auf Lage und Form ähneln. Allerdings sind die Orbitlinien in den stochastischen Plots dieses Mal deutlich breiter als in den deterministischen, was an dem hohen Wert von γ liegt. Dieser Wert von $\gamma = 3$ verursacht auch das Zittern, das auf der linken Seite im stochastischen (V, n) -Plot gut zu erkennen ist. In allen drei stochastischen Plots gibt es wieder die Zweiteilung der Orbits, die bereits in Abschnitt 2.3 thematisiert wurde. Die deterministischen Plots sind im Vergleich dazu feiner ausgeprägt, denn man erkennt in ihnen jeweils vier separate Orbitlinien. Diese stehen für die vier aufeinanderfolgenden, unterschiedlich hohen Spikes, die in Abbildung 2.19 sichtbar sind. Das fünfte Periodizitätsintervall des Inputs, in dem der Spike immer ausfällt, zeigt sich durch einen kleinen ellipsenförmigen Bogen ebenfalls in jedem der deterministischen Plots von Abbildung 2.20.

Insgesamt haben wir in diesem Abschnitt gesehen, dass eine große Ähnlichkeit zwischen den Orbits des stochastischen Modells (1.1) und des deterministischen Modells (2.7) besteht, wenn γ eher klein ist und a so gewählt ist, dass es in beiden Modellen mit hoher Regelmäßigkeit zu Spikes kommt. Für größere Werte von γ und bei unregelmäßigem Spike-Verhalten nimmt die Ähnlichkeit ab, aber sie ist noch immer zu erkennen.

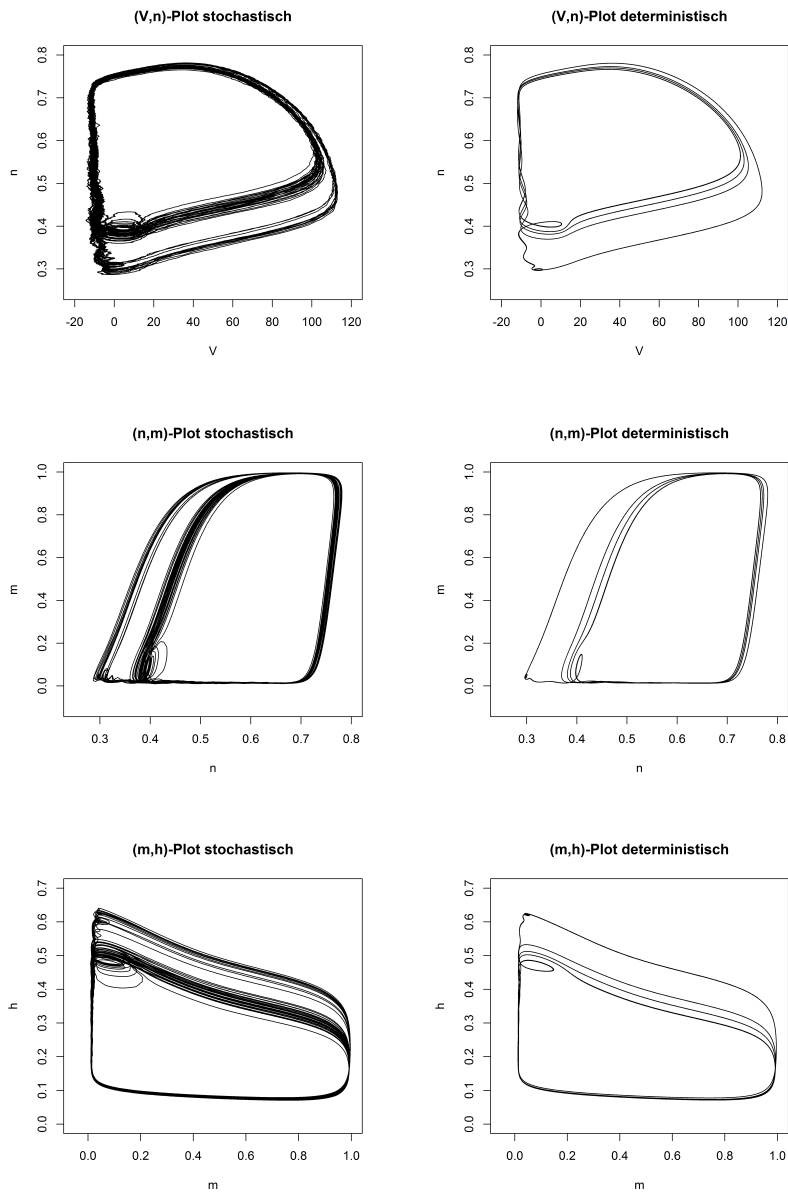


Abbildung 2.20.: Dirichletkern als Input: Vergleich des stochastischen und des deterministischen Modells anhand von Projektionen von X auf (V, n) , (n, m) und (m, h) für die letzten 50 Periodizitätsintervalle des Outputs

Zusammenfassung und Ausblick

Ziel dieser Arbeit war es, die invariante Verteilung $[\mu P_{0,s}]$ von Lösungsprozessen X des stochastischen Hodgkin-Huxley-Modells zu verschiedenen Toruszeitpunkten s zu veranschaulichen. Dafür wurden die Prozesse zunächst simuliert, auf zweidimensionale Ebenen projiziert und in Form von Orbits mit darauf liegenden Punktwolken dargestellt. Auf diese Art konnte man einen ersten Eindruck von der zeitlichen Dynamik des invarianten Maßes bekommen, aber noch aussagekräftiger war die Darstellung des Kerndichteschätzers für die Dichte von $[\mu P_{0,s}]$. Als Eingangsdaten für den Schätzer wurden dabei sowohl die Werte von X zu bestimmten Toruszeiten s als auch alle bei der Simulation des Prozesses ermittelten Werte verwendet.

Wir haben für verschiedene Inputsignale gesehen, dass sich mit wachsendem Amplitudenfaktor a die Periodizitäten der Hodgkin-Huxley-Lösungsprozesse verändern und dass die Spikes immer regelmäßiger auftreten. Dieses Verhalten wird auch durch das invariante Maß beziehungsweise durch seine Dichte nachgebildet. Außerdem ist deutlich geworden, dass die Diffusionsstreuung γ Einfluss auf das Aussehen dieser Dichte hat, denn mit wachsendem γ werden die zugrunde liegenden Orbits unschärfer. Im letzten Abschnitt der Arbeit haben wir Ähnlichkeiten aufgedeckt zwischen den Orbits des stochastischen Hodgkin-Huxley-Modells und den Orbits des deterministischen Modells mit einer bestimmten Inputfunktion.

Der Fokus lag in dieser Arbeit eher auf der Veranschaulichung der Dichte des invarianten Maßes und auf den empirischen Folgerungen aus den entstandenen Bildern und weniger auf dem theoretischen Hintergrund der verwendeten Techniken. Die Ausführung einiger Hintergründe hätte den Rahmen dieser Arbeit überstiegen. Es wäre insbesondere interessant und wichtig gewesen, einen formalen Beweis dafür auszuarbeiten, dass man die Aussagen über

die Bandbreite und Konvergenz von Kerndichteschätzern von unabhängigen, identisch verteilten Zufallsvariablen auf Harris-rekurrente Markovprozesse übertragen kann.

A. Quellcodes

A.1. Ratingfunktionen

```
1 # Bestimmung der Ratingfunktionen aus Definition 1.1
2 # Eingabe: V = Membranpotential
3
4 alpha_n <- function(V){
5   if(V!=10){
6     r <- (0.1-0.01*V)/(exp(1-0.1*V)-1)
7   }
8   else{
9     r <- 1/10
10  }
11  return(r)
12 }
13
14 alpha_m <- function(V){
15   if(V!=25){
16     r <- (2.5-0.1*V)/(exp(2.5-0.1*V)-1)
17   }
18   else{
19     r <- 1
20   }
21   return(r)
22 }
23
24 alpha_h <- function(V){
25   r <- 0.07*exp(-V/20)
26   return(r)
27 }
28
29 beta_n <- function(V){
```

```

30 r <- 0.125*exp(-V/80)
31 return(r)
32 }
33
34 beta_m <- function(V){
35   r <- 4*exp(-V/18)
36   return(r)
37 }
38
39 beta_h <- function(V){
40   r <- 1/(exp(3-0.1*V)+1)
41   return(r)
42 }

```

A.2. Mögliche Inputfunktionen

```

1 # Mögliche Inputfunktionen S aus Definition 1.4
2 # Eingabe: time = Vektor von Zeitpunkten
3
4 # 1-periodischer Sinus
5 sin_1p <- function(time){
6   r <- sin(2*pi*time)
7   return(r)
8 }
9
10 # 2. Dirichlet-Kern
11 D_2 <- function(time){
12   r <- 1 + 2*colSums(cos(2*pi*(1:2)%*%t(time)))
13   return(r)
14 }
15
16 # 3. Dirichlet-Kern
17 D_3 <- function(time){
18   r <- 1 + 2*colSums(cos(2*pi*(1:3)%*%t(time)))
19   return(r)
20 }
21

```

```

22 # 4. Dirichlet-Kern
23 D_4 <- function(time){
24   r <- 1 + 2*colSums(cos(2*pi*(1:4)%*%t(time)))
25   return(r)
26 }

```

A.3. Driftfunktionen

```

1 # Bestimmung der Driftfunktionen des stochastischen und
2 # deterministischen Hodgkin-Huxley-Modells
3 F <- function(V,j){
4   # Berechnet den Wert der Funktion F aus Gleichung (1.4)
5   # und wird später für die Bestimmung des Drifts b_V von
6   # V verwendet.
7   # Eingabe: V = Membranpotential
8   #           j=(n,m,h) = Gating-Variablen
9
10  E_K <- -12           # Gleichgewichtspotentiale
11  E_Na <- 120
12  E_L <- 10.6
13  g_K <- 36           # maximale Leitfähigkeit
14  g_Na <- 120
15  g_L <- 0.3
16  r <- g_K*j[1]^4*(V-E_K) + g_Na*j[2]^3*j[3]*(V-E_Na) +
17      g_L*(V-E_L)
18  return(r)
19 }
20
21 b_j <- function(V,j){
22   # Berechnet den Drift b_j von j=(n,m,h).
23   # Eingabe: V = Membranpotential
24   #           j=(n,m,h) = Gating-Variablen
25
26   r <- c(0,0,0)
27   r[1] <- alpha_n(V)*(1-j[1])-beta_n(V)*j[1]
28   r[2] <- alpha_m(V)*(1-j[2])-beta_m(V)*j[2]
29   r[3] <- alpha_h(V)*(1-j[3])-beta_h(V)*j[3]

```

```

30   return(r)
31 }
32
33 # entfällt beim deterministischen Modell:
34 b_xi <- function(a,S,t,T,xi,tau){
35   # Berechnet den Drift b_xi von xi.
36   # Eingabe: a = Amplitudenfaktor
37   #          S = 1-periodische Inputfunktion
38   #          t = Zeitpunkt
39   #          T = Periodizität des Inputs
40   #          xi = rauschender Input
41
42   r <- (a*S(t/T)-xi)*tau
43   return(r)
44 }

```

A.4. Simulation des stochastischen Hodgkin-Huxley-Modells und Darstellung der Lösung in Abhängigkeit von t

```

1  hodgkin_huxley_1 <-
2  function(a,S,T_in,T_out,tau,gamma,K,out=FALSE){
3    # Bestimmt mit dem Euler-Maruyama-Schema (näherungsweise) eine
4    # Lösung des stochastischen Hodgkin-Huxley-Modells und stellt
5    # diese als Plot von V, (n,m,h) und xi in Abhängigkeit von t
6    # dar.
7    # Eingabe: a = Amplitudenfaktor
8    #          S = 1-periodische Inputfunktion
9    #          T_in = Periodizität des Inputs
10   #          T_out = Periodizität des Outputs (nicht nötig, wenn
11   #                  out=FALSE)
12   #          tau = Geschwindigkeit der Diffusion
13   #          gamma = Streuung der Diffusion
14   #          K = Anzahl der beobachteten Perioden des Inputs bzw.
15   #              Outputs
16   #          out: gibt an, ob Perioden des Inputs oder Outputs

```

```

17 # betrachtet werden sollen
18
19 if (out==TRUE){ # Festlegung, ob Perioden des Inputs
20 # oder Outputs betrachtet werden
21   T <- T_out
22 }
23 else{
24   T <- T_in
25 }
26 t_end <- K*T # Beobachtungszeitraum
27 E <- 2500 # Einteilung eines Periodizitäts-
28 # intervalls des Inputs bzw. Outputs
29 Delta <- T/E # Schrittweite
30 num <- t_end/Delta # Anzahl der Schritte
31 set.seed(1202) # um für verschiedene Darstellungs-
32 # arten dieselben Zufallszahlen zu
33 # zu verwenden
34 zeta <- rnorm(num,0,sqrt(Delta))
35 start <- c(0,0.35,0.01,0.7,0) # Startvektor für
36 # (V,n,m,h,xi)
37 t <- seq(0,t_end,by=Delta)
38
39 # Euler-Maruyama-Schema für xi, V und j=(n,m,h):
40 xi <- numeric(num+1)
41 xi[1] <- start[5]
42 V <- numeric(num+1)
43 V[1] <- start[1]
44 j <- matrix(0, nrow=3, ncol=num+1)
45 j[,1] <- start[2:4]
46 for(s in 1:num){
47   xi[s+1] <- xi[s] + b_xi(a,S,t[s],T_in,xi[s],tau)*Delta +
48     gamma*sqrt(tau)*zeta[s]
49   V[s+1] <- V[s] + (b_xi(a,S,t[s],T_in,xi[s],tau)-
50     F(V[s],j[,s]))*Delta +
51     gamma*sqrt(tau)*zeta[s]
52   j[,s+1] <- j[,s] + b_j(V[s],j[,s])*Delta
53 }
54
55 # für Plots von V, (n,m,h) und xi in Abhängigkeit von t:
56 par(mfrow=c(3,1))

```

```

57 plot(t,V,type="l",col="black",main="Membranpotential V",
58       xlab="t",ylab="V(t)")
59 abline(h=0,col="red",lty=3)
60 plot(t,j[1,],type="l",ylim=c(0,1),col="black",
61       main="Gatingvariablen n,m,h",xlab="t",
62       ylab="n(t),m(t),h(t)")
63 points(t,j[2,],type="l",ylim=c(0,1),col="red")
64 points(t,j[3,],type="l",ylim=c(0,1),col="green")
65 legend("topright",legend=c("n(t)","m(t)","h(t)"),
66        col=c("black","red","green"),lty=1,cex=1)
67 plot(t,xi,type="l",col="black",main="Rauschender Input xi",
68       xlab="t",ylab="xi(t)",ylim=range(xi,a*S(t/T_in)))
69 lines(t,a*S(t/T_in),col="red",lwd=1,lty=3)
70 legend("topright",legend=c("xi(t)","aS(t/T)"),
71        col=c("black","red"),lty=c(1,3),cex=1)
72 }

```

A.5. Simulation des stochastischen Hodgkin-Huxley-Modells und Darstellung der Lösung in Form von Projektionen auf zweidimensionale Ebenen

```

1 hodgkin_huxley_2 <-
2   function(a,S,T_in,T_out,tau,gamma,K,out=TRUE){
3     # Bestimmt mit dem Euler-Maruyama-Schema (näherungsweise) eine
4     # Lösung des stochastischen Hodgkin-Huxley-Modells und stellt
5     # diese als Projektionen auf (V,n), (V,m), (V,h), (n,m), (n,h)
6     # und (m,h) dar.
7     # Eingabe: a = Amplitudenfaktor
8     #           S = 1-periodische Inputfunktion
9     #           T_in = Periodizität des Inputs
10    #           T_out = Periodizität des Outputs (nicht nötig, wenn
11    #                  out=FALSE)
12    #           tau = Geschwindigkeit der Diffusion
13    #           gamma = Streuung der Diffusion
14    #           K = Anzahl der beobachteten Perioden des Inputs bzw.
15    #              Outputs

```

```

16 #           out: gibt an, ob Perioden des Inputs oder Outputs
17 #           betrachtet werden sollen
18
19 if (out==TRUE){           # Festlegung, ob Perioden des Inputs
20                           # oder Outputs betrachtet werden
21     T <- T_out
22 }
23 else{
24     T <- T_in
25 }
26 t_end <- K*T              # Beobachtungszeitraum
27 E <- 2500                 # Einteilung eines Periodizitäts-
28                           # intervalls des Inputs bzw. Outputs
29                           # (sollte ganzzahliges Vielfaches von
30                           # 25 sein)
31 Delta <- T/E              # Schrittweite
32 num <- t_end/Delta        # Anzahl der Schritte
33 set.seed(1202)           # um für verschiedene Darstellungs-
34                           # arten dieselben Zufallszahlen zu
35                           # zu verwenden
36 zeta <- rnorm(num,0,sqrt(Delta))
37 start <- c(0,0.35,0.01,0.7,0)   # Startvektor für
38                                   # (V,n,m,h,xi)
39 t <- seq(0,t_end,by=Delta)
40
41 # Euler-Maruyama-Schema für xi, V und j=(n,m,h):
42 xi <- numeric(num+1)
43 xi[1] <- start[5]
44 V <- numeric(num+1)
45 V[1] <- start[1]
46 j <- matrix(0, nrow=3, ncol=num+1)
47 j[,1] <- start[2:4]
48 for(s in 1:num){
49     xi[s+1] <- xi[s] + b_xi(a,S,t[s],T_in,xi[s],tau)*Delta +
50                   gamma*sqrt(tau)*zeta[s]
51     V[s+1] <- V[s] + (b_xi(a,S,t[s],T_in,xi[s],tau)-
52                     F(V[s],j[,s]))*Delta +
53                   gamma*sqrt(tau)*zeta[s]
54     j[,s+1] <- j[,s] + b_j(V[s],j[,s])*Delta
55 }

```

```

56
57 # Für das letzte Periodizitätsintervall sollen 26 Werte
58 # abgespeichert werden:
59 V_2 <- V[seq(from=(K-1)*E+1,to=K*E+1,length=26)]
60 n <- j[1,seq(from=(K-1)*E+1,to=K*E+1,length=26)]
61 m <- j[2,seq(from=(K-1)*E+1,to=K*E+1,length=26)]
62 h <- j[3,seq(from=(K-1)*E+1,to=K*E+1,length=26)]
63
64 # für grafische Darstellung:
65 par(mfrow=c(3,2))
66 b <- 5 # Anzahl der Orbits, die gezeichnet
67 # werden sollen
68 # für (V,n)-Plot:
69 plot(V[(length(V)-E*b):length(V)],
70      j[1,(length(V)-E*b):length(V)],type="l",main="(V,n)-Plot",
71      xlab="V",ylab="n",ylim=c(0.25,0.8))
72 points(V_2[1],n[1],type="p",col="blue",pch=15,cex=1.7)
73 points(V_2[2:26],n[2:26],type="p",col="red",pch=16)
74 text(V_2,n,labels=c("start",1:25),adj=c(0,1))
75 # für (V,m)-Plot:
76 plot(V[(length(V)-E*b):length(V)],
77      j[2,(length(V)-E*b):length(V)],type="l",main="(V,m)-Plot",
78      xlab="V",ylab="m",ylim=c(-0.1,1))
79 points(V_2[1],m[1],type="p",col="blue",pch=15,cex=1.7)
80 points(V_2[2:26],m[2:26],type="p",col="red",pch=16)
81 text(V_2,m,labels=c("start",1:25),adj=c(0,1))
82 # für (V,h)-Plot:
83 plot(V[(length(V)-E*b):length(V)],
84      j[3,(length(V)-E*b):length(V)],type="l",main="(V,h)-Plot",
85      xlab="V",ylab="h",ylim=c(0,0.65))
86 points(V_2[1],h[1],type="p",col="blue",pch=15,cex=1.7)
87 points(V_2[2:26],h[2:26],type="p",col="red",pch=16)
88 text(V_2,h,labels=c("start",1:25),adj=c(0,1))
89 # für (n,m)-Plot:
90 plot(j[1,(length(V)-E*b):length(V)],
91      j[2,(length(V)-E*b):length(V)],type="l",main="(n,m)-Plot",
92      xlab="n",ylab="m",ylim=c(-0.1,1))
93 points(n[1],m[1],type="p",col="blue",pch=15,cex=1.7)
94 points(n[2:26],m[2:26],type="p",col="red",pch=16)
95 text(n,m,labels=c("start",1:25),adj=c(0,1))

```

```

96 # für (n,h)-Plot:
97 plot(j[1,(length(V)-E*b):length(V)],
98       j[3,(length(V)-E*b):length(V)],type="l",main="(n,h)-Plot",
99       xlab="n",ylab="h",ylim=c(0.04,0.65))
100 points(n[1],h[1],type="p",col="blue",pch=15,cex=1.7)
101 points(n[2:26],h[2:26],type="p",col="red",pch=16)
102 text(n,h,labels=c("start",1:25),adj=c(0,1))
103 # für (m,h)-Plot:
104 plot(j[2,(length(V)-E*b):length(V)],
105       j[3,(length(V)-E*b):length(V)],type="l",main="(m,h)-Plot",
106       xlab="m",ylab="h",ylim=c(0,0.65))
107 points(m[1],h[1],type="p",col="blue",pch=15,cex=1.7)
108 points(m[2:26],h[2:26],type="p",col="red",pch=16)
109 text(m,h,labels=c("start",1:25),adj=c(0,1))
110 }

```

A.6. Simulation des stochastischen Hodgkin-Huxley-Modells und Dichteschätzung zu einer festen Toruszeit

```

1 hodgkin_huxley_3 <-
2   function(a,S,T_in,T_out,tau,gamma,K,out=FALSE){
3     # Bestimmt mit dem Euler-Maruyama-Schema (näherungsweise) eine
4     # Lösung des stochastischen Hodgkin-Huxley-Modells und stellt
5     # die geschätzte Dichte des invarianten Maßes zu einer festen
6     # Toruszeit durch Projektion auf (V,n) als 3d-Plot dar.
7     # Eingabe: a = Amplitudenfaktor
8     #           S = 1-periodische Inputfunktion
9     #           T_in = Periodizität des Inputs
10    #           T_out = Periodizität des Outputs (nicht nötig, wenn
11    #                  out=FALSE)
12    #           tau = Geschwindigkeit der Diffusion
13    #           gamma = Streuung der Diffusion
14    #           K = Anzahl der beobachteten Perioden des Inputs bzw.
15    #              Outputs
16    #           out: gibt an, ob Perioden des Inputs oder Outputs

```

```

17 #          betrachtet werden sollen
18
19 if (out==TRUE){          # Festlegung, ob Perioden des Inputs
20                          # oder Outputs betrachtet werden
21   T <- T_out
22 }
23 else{
24   T <- T_in
25 }
26 E <- 1000                # Einteilung eines Periodizitäts-
27                          # intervalls des Inputs bzw. Outputs
28                          # (sollte ganzzahliges Vielfaches von
29                          # 25 sein)
30 Delta <- T/E             # Schrittweite
31 start <- c(0,0.35,0.01,0.7,0) # Startvektor für
32                          # (V,n,m,h,xi)
33
34 # Für jedes Periodizitätsintervall sollen nur 26 Werte von
35 # (V,n,m,h) abgespeichert werden (die Werte von xi werden gar
36 # nicht dauerhaft gespeichert), um Speicherplatz zu sparen:
37 V_2 <- matrix(0,nrow=K-10,ncol=26)
38 n <- matrix(0,nrow=K-10,ncol=26)
39 m <- matrix(0,nrow=K-10,ncol=26)
40 h <- matrix(0,nrow=K-10,ncol=26)
41
42 for(i in 1:K){
43   zeta <- rnorm(E,0,sqrt(Delta))
44   t <- seq(0,T,by=Delta)
45   # Euler-Maruyama-Schema für xi, V und j=(n,m,h):
46   xi <- numeric(E+1)
47   xi[1] <- start[5]
48   V <- numeric(E+1)
49   V[1] <- start[1]
50   j <- matrix(0, nrow=3, ncol=E+1)
51   j[,1] <- start[2:4]
52   for(s in 1:E){
53     xi[s+1] <- xi[s] + b_xi(a,S,t[s],T_in,xi[s],tau)*Delta +
54                   gamma*sqrt(tau)*zeta[s]
55     V[s+1] <- V[s] + (b_xi(a,S,t[s],T_in,xi[s],tau)-
56                     F(V[s],j[,s]))*Delta +

```

```

57         gamma*sqrt(tau)*zeta[s]
58     j[,s+1] <- j[,s] + b_j(V[s],j[,s])*Delta
59 }
60 # In den ersten 10 Perioden sollen noch keine Werte
61 # gespeichert bleiben:
62 if(i<=10){
63     start <- c(V[E+1],j[1,E+1],j[2,E+1],j[3,E+1],xi[E+1])
64     next
65 }
66 V_2[i-10,] <- V[seq(from=1,to=E+1,length=26)]
67 n[i-10,] <- j[1,seq(from=1,to=E+1,length=26)]
68 m[i-10,] <- j[2,seq(from=1,to=E+1,length=26)]
69 h[i-10,] <- j[3,seq(from=1,to=E+1,length=26)]
70 # Der letzte Wert ist der Startpunkt für die Simulation
71 # über das nächste Periodizitätsintervall:
72 start <- c(V_2[i-10,26],n[i-10,26],m[i-10,26],h[i-10,26],
73           xi[E+1])
74 }
75
76 # für grafische Darstellung:
77 par(mfrow=c(1,1))
78 # Definition einer Farbpalette:
79 mycols <- colorRampPalette(c("grey","blue","green","yellow",
80                             "orange","red","darkred","black"))
81 j <- 0 # Toruszeit s_j, j=0,...,25
82 # zweidimensionaler Kerndichteschätzer mit Bandbreite
83 # h=(h_V,h_n) für psi_s zur Toruszeit s_j:
84 k <- kde2d(V_2[,j+1],n[,j+1],n=500,h=c(10,0.07),
85           lims=c(-30,120,0,1))
86 persp3D(k$x,k$y,k$z,theta=40,phi=30,xlab="\n V",ylab="\n n",
87         zlab="\n geschätzte Dichte",main=paste("j =",j),
88         col=mycols(256),ticktype="detailed",xlim=c(-30,120),
89         ylim=c(0,1),zlim=c(0,2.2),clim=c(0,2.2))
90 }

```

A.7. Simulation des stochastischen Hodgkin-Huxley-Modells und Dichteschätzung zu vielen Toruszeiten gleichzeitig

```
1 hodgkin_huxley_4 <-
2   function(a,S,T_in,T_out,tau,gamma,K,out=FALSE){
3     # Bestimmt mit dem Euler-Maruyama-Schema (näherungsweise) eine
4     # Lösung des stochastischen Hodgkin-Huxley-Modells und stellt
5     # die geschätzte Dichte des invarianten Maßes zu vielen
6     # Toruszeiten gleichzeitig, durch Projektion auf (V,n), (V,m),
7     # (V,h), (n,m), (n,h) und (m,h), als 3d-Plot dar.
8     # Eingabe: a = Amplitudenfaktor
9     #           S = 1-periodische Inputfunktion
10    #           T_in = Periodizität des Inputs
11    #           T_out = Periodizität des Outputs (nicht nötig, wenn
12    #                 out=FALSE)
13    #           tau = Geschwindigkeit der Diffusion
14    #           gamma = Streuung der Diffusion
15    #           K = Anzahl der beobachteten Perioden des Inputs bzw.
16    #              Outputs
17    #           out: gibt an, ob Perioden des Inputs oder Outputs
18    #                 betrachtet werden sollen
19
20    if (out==TRUE){          # Festlegung, ob Perioden des Inputs
21                            # oder Outputs betrachtet werden
22      T <- T_out
23    }
24    else{
25      T <- T_in
26    }
27    t_end <- K*T             # Beobachtungszeitraum
28    E <- 1000                # Einteilung eines Periodizitäts-
29                            # intervalls des Inputs bzw. Outputs
30    Delta <- T/E            # Schrittweite
31    num <- t_end/Delta      # Anzahl der Schritte
32    zeta <- rnorm(num,0,sqrt(Delta))
33    start <- c(0,0.35,0.01,0.7,0) # Startvektor für
34                                       # (V,n,m,h,xi)
35    t <- seq(0,t_end,by=Delta)
```

```

36
37 # Euler-Maruyama-Schema für xi, V und j=(n,m,h):
38 xi <- numeric(num+1)
39 xi[1] <- start[5]
40 V <- numeric(num+1)
41 V[1] <- start[1]
42 j <- matrix(0, nrow=3, ncol=num+1)
43 j[,1] <- start[2:4]
44 for(s in 1:num){
45   xi[s+1] <- xi[s] + b_xi(a,S,t[s],T_in,xi[s],tau)*Delta +
46     gamma*sqrt(tau)*zeta[s]
47   V[s+1] <- V[s] + (b_xi(a,S,t[s],T_in,xi[s],tau)-
48     F(V[s],j[,s]))*Delta +
49     gamma*sqrt(tau)*zeta[s]
50   j[,s+1] <- j[,s] + b_j(V[s],j[,s])*Delta
51 }
52
53 # für grafische Darstellung:
54 par(mfrow=c(1,1))
55 # Definition einer Farbpalette:
56 mycols <- colorRampPalette(c("grey", "blue", "green", "yellow",
57   "orange", "red", "darkred", "black"))
58 b <- 100 # Anzahl der Periodizitätsintervalle
59 # des Inputs bzw. Outputs, die
60 # gezeichnet werden sollen
61 # Bei Aufruf der Funktion sollten von den folgenden
62 # Projektionen alle bis auf eine auskommentiert werden:
63
64 # für Projektion auf (V,n):
65 # zweidimensionaler Kerndichteschätzer mit Bandbreite
66 # h=(h_V,h_n) für psi_s zu vielen Toruszeiten gleichzeitig:
67 k <- kde2d(V[(length(V)-E*b):length(V)],
68   j[1,(length(V)-E*b):length(V)],
69   n=500,h=c(10,0.07),lims=c(-30,120,0.2,0.9))
70 filled.contour(k$x,k$y,k$z,color.palette=mycols, xlab="V",
71   ylab="n", nlevels=100,
72   key.title={par(cex.main=0.5);
73     title(main="geschätzte Dichte")},
74   main.title={par(cex.main=1);
75     title(main="Projektion auf (V,n)")}))

```

```

76 # persp3D(k$x,k$y,k$z,theta=40,phi=30,xlab="\n V",ylab="\n n",
77 #         zlab="\n geschätzte Dichte",col=mycols(256),
78 #         main="Projektion auf (V,n)",ticktype="detailed")
79
80 # für Projektion auf (V,m):
81 # zweidimensionaler Kerndichteschätzer mit Bandbreite
82 # h=(h_V,h_m) für psi_s zu vielen Toruszeiten gleichzeitig:
83 k <- kde2d(V[(length(V)-E*b):length(V)],
84           j[2,(length(V)-E*b):length(V)],
85           n=500,h=c(10,0.08),lims=c(-30,120,-0.1,1.1))
86 filled.contour(k$x,k$y,k$z,color.palette=mycols,xlab="V",
87               ylab="m",nlevels=100,
88               key.title={par(cex.main=0.5);
89                           title(main="geschätzte Dichte")},
90               main.title={par(cex.main=1);
91                            title(main="Projektion auf (V,m)")})
92 # persp3D(k$x,k$y,k$z,theta=40,phi=30,xlab="\n V",ylab="\n m",
93 #         zlab="\n geschätzte Dichte",col=mycols(256),
94 #         main="Projektion auf (V,m)",ticktype="detailed")
95
96 # für Projektion auf (V,h):
97 # zweidimensionaler Kerndichteschätzer mit Bandbreite
98 # h=(h_V,h_h) für psi_s zu vielen Toruszeiten gleichzeitig:
99 k <- kde2d(V[(length(V)-E*b):length(V)],
100          j[3,(length(V)-E*b):length(V)],
101          n=500,h=c(10,0.06),lims=c(-30,120,0,0.8))
102 filled.contour(k$x,k$y,k$z,color.palette=mycols,xlab="V",
103               ylab="h",nlevels=100,
104               key.title={par(cex.main=0.5);
105                           title(main="geschätzte Dichte")},
106               main.title={par(cex.main=1);
107                            title(main="Projektion auf (V,h)")})
108 # persp3D(k$x,k$y,k$z,theta=40,phi=30,xlab="\n V",ylab="\n h",
109 #         zlab="\n geschätzte Dichte",col=mycols(256),
110 #         main="Projektion auf (V,h)",ticktype="detailed")
111
112 # für Projektion auf (n,m):
113 # zweidimensionaler Kerndichteschätzer mit Bandbreite
114 # h=(h_n,h_m) für psi_s zu vielen Toruszeiten gleichzeitig:
115 k <- kde2d(j[1,(length(V)-E*b):length(V)],

```

```

116         j [2, (length(V)-E*b) : length(V)],
117         n=500, h=c(0.07, 0.08), lims=c(0.2, 0.9, -0.1, 1.1))
118 filled.contour(k$x, k$y, k$z, color.palette=mycols, xlab="n",
119               ylab="m", nlevels=100,
120               key.title={par(cex.main=0.5);
121                           title(main="geschätzte Dichte")},
122               main.title={par(cex.main=1);
123                           title(main="Projektion auf (n,m)")})
124 # persp3D(k$x, k$y, k$z, theta=40, phi=30, xlab="\n n", ylab="\n m",
125 #         zlab="\n geschätzte Dichte", col=mycols(256),
126 #         main="Projektion auf (n,m)", ticktype="detailed")
127
128 # für Projektion auf (n,h):
129 # zweidimensionaler Kerndichteschätzer mit Bandbreite
130 # h=(h_n, h_h) für psi_s zu vielen Toruszeiten gleichzeitig:
131 k <- kde2d(j [1, (length(V)-E*b) : length(V)],
132            j [3, (length(V)-E*b) : length(V)],
133            n=500, h=c(0.07, 0.06), lims=c(0.2, 0.9, 0, 0.8))
134 filled.contour(k$x, k$y, k$z, color.palette=mycols, xlab="n",
135               ylab="h", nlevels=100,
136               key.title={par(cex.main=0.5);
137                           title(main="geschätzte Dichte")},
138               main.title={par(cex.main=1);
139                           title(main="Projektion auf (n,h)")})
140 # persp3D(k$x, k$y, k$z, theta=40, phi=30, xlab="\n n", ylab="\n h",
141 #         zlab="\n geschätzte Dichte", col=mycols(256),
142 #         main="Projektion auf (n,h)", ticktype="detailed")
143
144 # für Projektion auf (m,h):
145 # zweidimensionaler Kerndichteschätzer mit Bandbreite
146 # h=(h_m, h_h) für psi_s zu vielen Toruszeiten gleichzeitig:
147 k <- kde2d(j [2, (length(V)-E*b) : length(V)],
148            j [3, (length(V)-E*b) : length(V)],
149            n=500, h=c(0.08, 0.06), lims=c(-0.1, 1.1, 0, 0.8))
150 filled.contour(k$x, k$y, k$z, color.palette=mycols, xlab="m",
151               ylab="h", nlevels=100,
152               key.title={par(cex.main=0.5);
153                           title(main="geschätzte Dichte")},
154               main.title={par(cex.main=1);
155                           title(main="Projektion auf (m,h)")})

```

```

156 # persp3D(k$x,k$y,k$z,theta=40,phi=30,xlab="\n m",ylab="\n h",
157 #         zlab="\n geschätzte Dichte",col=mycols(256),
158 #         main="Projektion auf (m,h)",ticktype="detailed")
159 }

```

A.8. Deterministische Inputfunktion

```

1 R_abl <- function(x,tau,S,T){
2   # Bestimmt die Ableitung der Funktion R aus Gleichung (2.6).
3   # R' wird später als Inputfunktion für das deterministische
4   # Hodgkin-Huxley-Modell verwendet.
5   # Eingabe: x = Stelle, an der abgeleitet wird
6   #         tau = Diffusionsgeschwindigkeit aus dem
7   #         stochastischen Hodgkin-Huxley-Modell
8   #         S = 1-periodische Inputfunktion
9   #         T = Periodizität des Inputs
10
11  R <- function(t){ # Definition der Funktion R
12    integrand <- function(s){
13      i <- tau*exp(-tau*(t-s))*S(s/T)
14      return(i)
15    }
16    r <- integrate(integrand, lower=-Inf, upper=t)
17    return(r)
18  }
19  h <- 1e-10 # Bestimmung der Ableitung von R
20  r_abl <- (R(x+h)$value-R(x)$value)/h
21  return(r_abl)
22 }

```

A.9. Simulation des deterministischen Hodgkin-Huxley-Modells und Darstellung der Lösung in Abhängigkeit von t

```

1 hodgkin_huxley_det_1 <-
2   function(a,S,T_in,T_out,tau,K,out=FALSE){
3     # Bestimmt mit dem expliziten Euler-Verfahren (näherungsweise)
4     # eine Lösung des deterministischen Hodgkin-Huxley-Modells mit
5     # Input a*R' und erstellt Plots von V, (n,m,h) und a*R' in
6     # Abhängigkeit von t
7     # Eingabe: a = Amplitudenfaktor
8     #           S = 1-periodische Inputfunktion
9     #           T_in = Periodizität des Inputs
10    #           T_out = Periodizität des Outputs (nicht nötig, wenn
11    #                   out=FALSE)
12    #           tau = Geschwindigkeit der Diffusion aus dem
13    #                   stochastischen Hodgkin-Huxley-Modell
14    #           K = Anzahl der beobachteten Perioden des Inputs bzw.
15    #                   Outputs
16    #           out: gibt an, ob Perioden des Inputs oder Outputs
17    #                   betrachtet werden sollen
18
19    if (out==TRUE){           # Festlegung, ob Perioden des Inputs
20                              # oder Outputs betrachtet werden
21      T <- T_out
22    }
23    else{
24      T <- T_in
25    }
26    t_end <- K*T              # Beobachtungszeitraum
27    E <- 2500                 # Einteilung eines Periodizitäts-
28                              # intervalls des Inputs bzw. Outputs
29    Delta <- T/E              # Schrittweite
30    num <- t_end/Delta         # Anzahl der Schritte
31    start <- c(0,0.35,0.01,0.7) # Startvektor für (V,n,m,h)
32    t <- seq(0,t_end,by=Delta)
33
34    # explizites Euler-Verfahren für V und j=(n,m,h):
35    V <- numeric(num+1)

```

```

36 V[1] <- start[1]
37 j <- matrix(0, nrow=3, ncol=num+1)
38 j[,1] <- start[2:4]
39 input <- numeric(num+1)
40 for(s in 1:num){
41   V[s+1] <- V[s] + (a*R_abl(t[s], tau, S, T_in)-
42                   F(V[s], j[,s]))*Delta
43   j[,s+1] <- j[,s] + b_j(V[s], j[,s])*Delta
44   input[s] <- a*R_abl(t[s], tau, S, T_in)
45 }
46
47 # für Plots von V, (n,m,h) und a*R' in Abhängigkeit von t:
48 par(mfrow=c(3,1))
49 plot(t,V,type="l",col="black",main="Membranpotential V",
50      xlab="t",ylab="V(t)")
51 abline(h=0,col="red",lty=3)
52 plot(t,j[,1],type="l",ylim=c(0,1),col="black",
53      main="Gatingvariablen n,m,h",
54      xlab="t",ylab="n(t),m(t),h(t)")
55 points(t,j[,2],type="l",ylim=c(0,1),col="red")
56 points(t,j[,3],type="l",ylim=c(0,1),col="green")
57 legend("topright",legend=c("n(t)","m(t)","h(t)"),
58       col=c("black","red","green"),lty=1,cex=1)
59 plot(t,input,type="l",col="black",main="Input a*R'",
60      xlab="t",ylab="a*R'(t)")
61 }

```

A.10. Simulation des deterministischen Hodgkin-Huxley-Modells und Darstellung der Lösung in Form von Projektionen auf zweidimensionale Ebenen

```

1 hodgkin_huxley_det_2 <-
2   function(a,S,T_in,T_out,tau,K,out=TRUE){
3     # Bestimmt mit dem expliziten Euler-Verfahren (näherungsweise)

```

```

4 # eine Lösung des deterministischen Hodgkin-Huxley-Modells mit
5 # Input a*R' und stellt diese als Projektionen auf (V,n),
6 # (V,m), (V,h), (n,m), (n,h) und (m,h) dar.
7 # Eingabe: a = Amplitudenfaktor
8 #           S = 1-periodische Inputfunktion
9 #           T_in = Periodizität des Inputs
10 #           T_out = Periodizität des Outputs (nicht nötig, wenn
11 #                out=FALSE)
12 #           tau = Geschwindigkeit der Diffusion aus dem
13 #                stochastischen Hodgkin-Huxley-Modell
14 #           K = Anzahl der beobachteten Perioden des Inputs bzw.
15 #                Outputs
16 #           out: gibt an, ob Perioden des Inputs oder Outputs
17 #                betrachtet werden sollen
18
19 if (out==TRUE){ # Festlegung, ob Perioden des Inputs
20                 # oder Outputs betrachtet werden
21     T <- T_out
22 }
23 else{
24     T <- T_in
25 }
26 t_end <- K*T # Beobachtungszeitraum
27 E <- 2500 # Einteilung eines Periodizitäts-
28           # intervalls des Inputs bzw. Outputs
29           # (sollte ganzzahliges Vielfaches von
30           # 25 sein)
31 Delta <- T/E # Schrittweite
32 num <- t_end/Delta # Anzahl der Schritte
33 start <- c(0,0.35,0.01,0.7) # Startvektor für (V,n,m,h)
34 t <- seq(0,t_end,by=Delta)
35
36 # explizites Euler-Verfahren für V und j=(n,m,h):
37 V <- numeric(num+1)
38 V[1] <- start[1]
39 j <- matrix(0, nrow=3, ncol=num+1)
40 j[,1] <- start[2:4]
41 input <- numeric(num+1)
42 for(s in 1:num){
43     V[s+1] <- V[s] + (a*R_abl(t[s],tau,S,T_in)-

```

```

44         F(V[s], j[,s]))*Delta
45     j[,s+1] <- j[,s] + b_j(V[s], j[,s])*Delta
46     input[s] <- a*R_abl(t[s], tau, S, T_in)
47 }
48
49 # Für das letzte Periodizitätsintervall sollen 26 Werte
50 # abgespeichert werden:
51 V_2 <- V[seq(from=(K-1)*E+1,to=K*E+1,length=26)]
52 n <- j[1,seq(from=(K-1)*E+1,to=K*E+1,length=26)]
53 m <- j[2,seq(from=(K-1)*E+1,to=K*E+1,length=26)]
54 h <- j[3,seq(from=(K-1)*E+1,to=K*E+1,length=26)]
55
56 # für grafische Darstellung:
57 par(mfrow=c(3,2))
58 b <- 50 # Anzahl der Orbits, die gezeichnet
59 # werden sollen
60 # für (V,n)-Plot:
61 plot(V[(length(V)-E*b):length(V)],
62      j[1, (length(V)-E*b):length(V)], type="l", main="(V,n)-Plot",
63      xlab="V", ylab="n", ylim=c(0.25,0.8))
64 # points(V_2[1], n[1], type="p", col="blue", pch=15, cex=1.7)
65 # points(V_2[2:26], n[2:26], type="p", col="red", pch=16)
66 # text(V_2, n, labels=c("start", 1:25), adj=c(0,1))
67 # für (V,m)-Plot:
68 plot(V[(length(V)-E*b):length(V)],
69      j[2, (length(V)-E*b):length(V)], type="l", main="(V,m)-Plot",
70      xlab="V", ylab="m", ylim=c(-0.1,1))
71 # points(V_2[1], m[1], type="p", col="blue", pch=15, cex=1.7)
72 # points(V_2[2:26], m[2:26], type="p", col="red", pch=16)
73 # text(V_2, m, labels=c("start", 1:25), adj=c(0,1))
74 # für (V,h)-Plot:
75 plot(V[(length(V)-E*b):length(V)],
76      j[3, (length(V)-E*b):length(V)], type="l", main="(V,h)-Plot",
77      xlab="V", ylab="h", ylim=c(0,0.65))
78 # points(V_2[1], h[1], type="p", col="blue", pch=15, cex=1.7)
79 # points(V_2[2:26], h[2:26], type="p", col="red", pch=16)
80 # text(V_2, h, labels=c("start", 1:25), adj=c(0,1))
81 # für (n,m)-Plot:
82 plot(j[1, (length(V)-E*b):length(V)],
83      j[2, (length(V)-E*b):length(V)], type="l", main="(n,m)-Plot",

```

```

84     xlab="n",ylab="m",ylim=c(-0.1,1))
85 # points(n[1],m[1],type="p",col="blue",pch=15,cex=1.7)
86 # points(n[2:26],m[2:26],type="p",col="red",pch=16)
87 # text(n,m,labels=c("start",1:25),adj=c(0,1))
88 # für (n,h)-Plot:
89 plot(j[1,(length(V)-E*b):length(V)],
90      j[3,(length(V)-E*b):length(V)],type="l",main="(n,h)-Plot",
91      xlab="n",ylab="h",ylim=c(0.04,0.65))
92 # points(n[1],h[1],type="p",col="blue",pch=15,cex=1.7)
93 # points(n[2:26],h[2:26],type="p",col="red",pch=16)
94 # text(n,h,labels=c("start",1:25),adj=c(0,1))
95 # für (m,h)-Plot:
96 plot(j[2,(length(V)-E*b):length(V)],
97      j[3,(length(V)-E*b):length(V)],type="l",main="(m,h)-Plot",
98      xlab="m",ylab="h",ylim=c(0,0.65))
99 # points(m[1],h[1],type="p",col="blue",pch=15,cex=1.7)
100 # points(m[2:26],h[2:26],type="p",col="red",pch=16)
101 # text(m,h,labels=c("start",1:25),adj=c(0,1))
102 }

```

B. Literaturverzeichnis

- [1] Jacques Azéma, Marie Dufflo, Daniel Revuz: *Mesure invariante des processus de Markov récurrents*, Séminaire de Probabilités III, Lecture Notes in Mathematics 88, Springer (1969)
- [2] Maximilian Dietrich: *Ein Schätzer für die Periodizität eines Signals in einem stochastischen Hodgkin-Huxley-Modell*, Bachelorarbeit Mathematik, Johannes Gutenberg-Universität (2017)
- [3] Kevin Endler: *Periodicities in the Hodgkin-Huxley model and versions of this model with stochastic input*, Masterarbeit Mathematik, Johannes Gutenberg-Universität (2012)
- [4] Alexander Goldenshluger, Oleg Lepski: *Bandwidth selection in kernel density estimation: Oracle inequalities and adaptive minimax optimality*, Ann. Statist. 39(3), 1608-1632 (2011)
- [5] Martin Hanke-Bourgeois: *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*, Vieweg+Teubner (2009)
- [6] Matthias Hieber: *Analysis II*, Springer (2019)
- [7] Alan L. Hodgkin, Andrew F. Huxley: *A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve*, The Journal of Physiology 117, 500-544 (1952)
- [8] Simon Holbach: *Recurrence and parameter estimation for degenerate diffusions with internal variables and randomly perturbed time-inhomogeneous deterministic input*, Doktorarbeit Mathematik, Johannes Gutenberg-Universität (2018)

- [9] Reinhard Höpfner, Eva Löcherbach: Limit theorems for null recurrent Markov processes, *Memoirs AMS* 768 (2003)
- [10] Reinhard Höpfner, Eva Löcherbach, Michèle Thieullen: Ergodicity and limit theorems for degenerate diffusions with time periodic drift. Application to a stochastic Hodgkin-Huxley model., *ESAIM: P&S* 20, 527-554 (2016)
- [11] Reinhard Höpfner, Eva Löcherbach, Michèle Thieullen: Ergodicity for a stochastic Hodgkin–Huxley model driven by Ornstein–Uhlenbeck type input, *Ann. Institut. Henri Poincaré* 52(1), 483-501 (2016)
- [12] Reinhard Höpfner, Eva Löcherbach, Michèle Thieullen: Strongly degenerate time inhomogeneous SDEs: densities and support properties. Application to Hodgkin-Huxley type systems., *Bernoulli* 23(4A), 2587-2616 (2017)
- [13] Ildar Ibragimov, Rafael Hasminskii: An estimate of the density of a distribution, *LOMI* 98, 61-85 (1980)
- [14] Ildar Ibragimov, Rafael Hasminskii: On density estimation in the view of Kolmogorov’s ideas in approximation theory, *Ann. Statist.* 18(3), 999-1010 (1990)
- [15] Ildar Ibragimov, Rafael Hasminskii: *Statistical Estimation. Asymptotic Theory*, Springer (1981)
- [16] Eugene M. Izhikevich: *Dynamical systems in neuroscience: The geometry of excitability and bursting*, MIT Press (2007)
- [17] Yury A. Kutoyants: *Statistical inference for spatial Poisson processes*, Springer (1998)
- [18] Esa Nummelin: A splitting technique for Harris recurrent Markov chains, *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 43, 309-318 (1978)

- [19] Esa Nummelin: General irreducible Markov chains and non-negative operators, Cambridge University Press (1985)
- [20] Alexandre B. Tsybakov: Introduction to nonparametric estimation, Springer (2009)
- [21] <https://i2.wp.com/www.explain-it-arium.de/wp-content/uploads/2016/10/Neuron-beschriftet-1.gif>
(aufgerufen am 28.06.19)