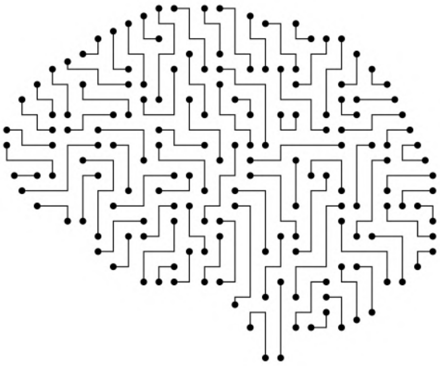
TESIS DOCTORAL / DOKTOREGO TESIA

Patient-specific modelling of cortical spreading depression applied to migraine studies

Autora / Egilea:
Julia M. Kroos

Director / Zuzendaria:
Dr. Luca Gerardo-Giorda



eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

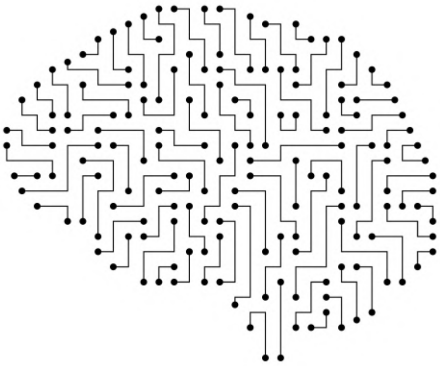
Bilbao, 2019

DOCTORAL THESIS

Patient-specific modelling of cortical spreading depression applied to migraine studies

Author:
Julia M. Kroos

Supervisor:
Dr. Luca Gerardo-Giorda



basque center for applied mathematics

Bilbao, 2019

“If the human brain were so simple that we could understand it,
we would be so simple that we couldn’t.”

Emerson M. Pugh, in *The Biological Origin of Human Values* (1978)

This research was carried out at the Basque Center for Applied Mathematics (BCAM) within the Mathematical Modelling in Biosciences (MMB) group. The research was supported by the Basque Government through the BERC 2014-2017 program, and by the Spanish Ministry of Economics and Competitiveness MINECO through the BCAM Severo Ochoa excellence accreditation SEV-2013-0323 and SEV-2017-0718 and the Spanish "Plan Estatal de Investigación Desarrollo e Innovación Orientada a los Retos de la Sociedad" under Grant BELEMET - Brain ELEctro-METabolic modeling and numerical approximation (MTM2015-69992-R).

Resumen corto

La migraña es un trastorno neurológico muy común. Un tercio de los pacientes que sufren migraña experimentan lo que se denomina aura, una serie de alteraciones sensoriales que preceden al típico dolor de cabeza unilateral. Diversos estudios apuntan a la existencia de una correlación entre el aura visual y la depresión cortical propagada (DCP), una onda de despolarización que tiene su origen en el córtex visual para propagarse, a continuación, por todo el córtex hacia las zonas periféricas. La complejidad y la elevada especificidad de las características del córtex cerebral sugieren que la geometría podría tener un impacto significativo en la propagación de la DCP. En esta tesis hemos combinado dos modelos existentes: un modelo neurológico pormenorizado para el componente electrofisiológico de la DCP y un modelo de reacción-difusión que tiene en consideración la difusión del potasio, el impulsor de la propagación de la DCP. Durante el proceso, hemos integrado dos aspectos de la DCP que tienen lugar en diferentes escalas de tiempo: la dinámica electrofisiológica seguiría un patrón temporal del orden de milisegundos, mientras que la dinámica del potasio extracelular que acciona las funciones de propagación de la DCP se mediría en una escala de minutos. Como resultado, obtendremos un modelo multiescalar EDP-EDO. Asimismo, hemos incorporado los datos específicos del paciente en el modelo DCP: (i) la geometría cerebral específica de un paciente obtenida a través de resonancia magnética, y (ii) los tensores de conductividad personalizados obtenidos a través de *diffusion tensor images*. A fin de estudiar el papel que desempeña la geometría en la propagación de la DCP, hemos definido las cantidades de interés (CdI) relacionadas con la geometría y las que dependen de la DCP y las hemos evaluado en dos casos prácticos. Si bien la geometría no parece tener un impacto significativo en la propagación de la DCP, algunas CdI han resultado ser unas candidatas muy prometedoras para facilitar la clasificación de individuos sanos y pacientes con migraña. Finalmente, para justificar la carencia de datos experimentales para la validación y selección de los parámetros del modelo, hemos aplicado diversas técnicas de cuantificación de la incertidumbre al modelo DCP y hemos analizado el impacto de las diversas elecciones de parámetros en el resultado del modelo.

Laburpena

Migraina ohiko gaixotasun neurologiko bat da, eta hiru pazientetik batek migraina-aura, hots, buruko minaren aurretik sortzen den ikusmenaren asaldura, jasaten du. Hainbat ikerketaren arabera, aura korrelazioan egon liteke *cortical spreading depression* (CSD) delakoarekin, garun azal bisualean sortzen den eta gune periferikoetara zabaltzen den despolarizazio uhinarekin alegia. Garun azalaren egitura konplexua eta pazientearen araberakoak diren ezaugarriak direla eta, bere geometriak CSDaren hedapenean zeresan handia izan lezakeela uste da. Tesi honetan ezagunak diren bi eredu bateratuko ditugu: bata, CSDaren elektrofisiologia zehazki deskribatzen duen eredua; bestea, CSDaren garapenean eragina duen potasioaren hedapena aurreikusten duen erreakzio-difusio eredua. Biak konbinatzean, denbora-eskala desberdinetan gertatzen diren CSDaren bi ezaugarri hartuko ditugu kontuan; izan ere, dinamika elektrofisiologikoa milisegundoen eskalan neurtzen da; CSDa eragiten duen zelulaz kanpoko potasioaren dinamika, ordea, minututan. Ondorioz, eskala anitzeko eredu diferentziala lortuko dugu. Halaber, pazientearen araberako datuak kontuan hartuko ditugu ereduan. Alde batetik, pazientearen burmuinaren geometria behatuko dugu erresonantzia magnetikoaren bidez, eta bestetik, difusio-tentsoreen bidezko irudien bidez lortutako konduktibitate-tentsore pertsonalizatuak erregistratuko ditugu. Geometriak CSDaren hedapenean duen eragina aztertzeko, ezaugarri geometrikoak dituzten eta CSDaren menpekoak diren zenbait kantitate definituko ditugu *quantities of interest* (QoI) izenpean eta aurreko bi kasuetan neurtuko ditugu. Dirudienez, geometriak ez du eragin handirik CSDaren hedapenean, baina hala ere zenbait QoI erabilgarriak izan litezke banako osasuntsuak eta migrainadun banakoak bereizteko. Amaitzeko, ereduaren parametroak egoki aukeratzeko eta datu esperimentalen falta konpontzeko, ziurgabetasun-kuantifikazio teknikak erabiliko ditugu ereduarekin batera, eta parametroen aukera desberdinek ereduan duten eragina aztertuko dugu.

Abstract

Migraine is a common neurological disorder and one-third of migraine patients suffer from migraine aura, a perceptual disturbance preceding the typically unilateral headache. Cortical spreading depression (CSD), a depolarisation wave that originates in the visual cortex and propagates across the cortex to the peripheral areas, has been suggested as a correlate of visual aura by several studies. The complex and highly individual-specific characteristics of the brain cortex suggest that the geometry might have a significant impact on CSD propagation. In this thesis, we combine two existing models, a detailed neurological model for the electrophysiological component of CSD and a reaction-diffusion model accounting for the potassium diffusion, the driving force of CSD propagation. In the process, we integrate two aspects of CSD that occur at different time scales: the electrophysiological dynamics features a temporal scale in the order of milliseconds, while the extracellular potassium dynamics that triggers CSD propagation features is on the scale of minutes. As a result we obtain a multi-scale PDE-ODE model. In addition, we incorporate patient-specific data in the CSD model: (i) a patient-specific brain geometry obtained from magnetic resonance imaging, and (ii) personalised conductivity tensors derived from diffusion tensor imaging data. To study the role of the geometry in CSD propagation, we define geometric and CSD-dependent quantities of interest (QoI) that we evaluate in two case studies. Even though the geometry does not seem to have a major impact on the CSD propagation, some QoI are promising candidates to aid in the classification of healthy individuals and migraine patients. Finally, to account for the lack of experimental data for validation and selection of the model parameters, we apply different techniques of uncertainty quantification to the CSD model and analyse the impact of various parameter choices on the model outcome.

Resumen

La migraña es una de las principales causas mundiales de incapacidad en la población actual. Un tercio de los pacientes que sufren migraña padecen también lo que se denomina aura de migraña, una serie de trastornos sensoriales que anteceden al típico dolor de cabeza. Diversos estudios apuntan a la existencia de una correlación entre el aura visual y la depresión cortical propagada (DCP), una onda de despolarización que tiene su origen en el córtex visual para propagarse a continuación por todo el córtex hacia las zonas periféricas (Hadjikhani et al., 2001; Richter y Lehmkühler, 2008). Tras más de 70 años de investigación en el campo de la migraña y de la DCP, el origen y las causas de este fenómeno siguen sin estar completamente claros.

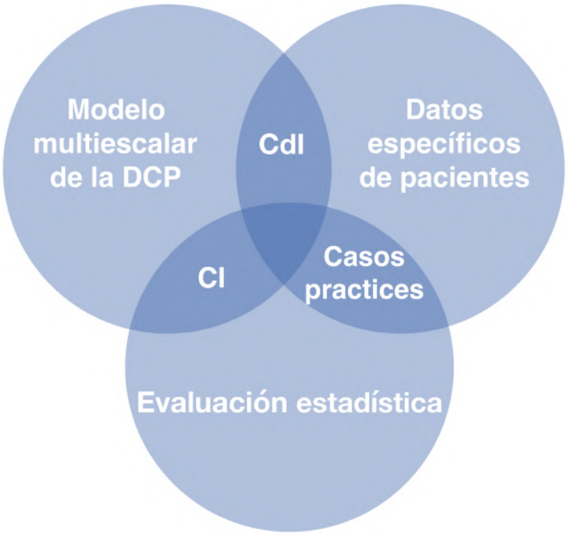
El paso de una onda DCP desencadena diversos cambios en la respuesta vascular, en el flujo sanguíneo y en el metabolismo de la energía. Los cambios drásticos de las concentraciones iónicas en el interior y el exterior de las neuronas corticales provocan un breve periodo de una intensa actividad, seguido de una reducción abrupta del potencial de membrana que silencia la actividad neuronal durante un periodo corto de tiempo (Pietrobon y Moskowitz, 2014). La DCP se caracteriza por un aumento significativo del potasio extracelular y la hipótesis más extendida apunta a la difusión del potasio extracelular como su principal catalizador (Zandt, Haken, y Putten, 2013; Zandt et al., 2015).

Algunos estudios recientes han comenzado a considerar el efecto de las geometrías corticales en la propagación de la DCP, con diverso nivel de detalle. Pocci et al., 2010 simularon la propagación de una onda en un conducto en 2D que presentaba una curva, demostrando que las curvas cerradas podrían bloquear la propagación de la onda. Siguiendo un enfoque de modelización personalizado, Dahlem et al., 2015 emplearon la curvatura de Gauss del córtex para identificar objetivos potenciales para la neuromodulación. Estos estudios ya aluden al hecho de que las dobleces y los pliegues en el córtex podrían influir en la propagación de la onda de despolarización. En consecuencia, la complejidad y la elevada especificidad de las características del córtex cerebral podrían desempeñar un papel significativo en la propagación de la DCP.

Motivación y objetivos principales

Los avances en tecnología han permitido un acceso cada vez mayor a una enorme cantidad de datos. Especialmente en el sector médico, en el que el análisis y la interpretación de los datos de los pacientes individuales podrían ayudarnos a entender las causas y la dinámica de una enfermedad en particular. Los modelos matemáticos pueden proporcionarnos un conocimiento significativo y profundo sobre el origen de una enfermedad y su progresión. Asimismo, estos modelos nos pueden ofrecer un análisis del contexto y una evaluación de los riesgos de las estrategias de tratamiento posibles, que irán desde un tratamiento personalizado y un diseño inteligente de los medicamentos, hasta un cribado de la población y una minería de datos electrónicos sobre la salud. Durante la investigación de la migraña con aura, las resonancias magnéticas que incluyeron datos de *diffusion-weighted imaging* (DWI) o *diffusion tensor imaging* (DTI) ofrecieron información detallada sobre el córtex y la estructura del tejido subyacente. En esta tesis proponemos un modelo matemático detallado para la propagación de la DCP en combinación con información específica de cada paciente sobre la geometría de su cerebro y las características de su córtex cerebral.

El objetivo de la presente tesis consiste en estudiar los rasgos que desempeñan un papel significativo en la propagación de la DCP e investigar el origen de la migraña simulando ondas DCP en las geometrías cerebrales de individuos sanos y de pacientes con migraña. A la hora de abordar este estudio, hemos combinado diversas perspectivas: el modelo DCP multiescalar, los datos específicos de pacientes y evaluaciones estadísticas de diversos tipos. Las características de la DCP son inherentemente multiescalares, ya que la dinámica del potasio extracelular que desencadena la propagación de la DCP tiene lugar en una escala de minutos, mientras que la dinámica electrofisiológica se desarrolla en una escala temporal mucho más rápida. En consecuencia, hemos introducido un modelo multiescalar EDP-EDO para la concentración de potasio extracelular y hemos combinado este modelo con información específica del paciente. Con el fin de realizar una comparativa entre la propagación de la DCP en individuos sanos y en pacientes con migraña, hemos introducido cantidades de interés (CdI) que sintetizan la propagación de la DCP en las geometrías corticales. A continuación, hemos evaluado la propagación de la DCP en diversos conjuntos de datos de pacientes. En el primer caso práctico comparamos la propagación de un frente de onda simulado con la secuencia de síntomas que experimentan los pacientes con migraña con aura crónica durante un ataque de migraña. Más adelante, en un análisis de grupo, simulamos la propagación de la DCP y calculamos las CdI con el objetivo de hallar unos clasificadores que distinguieran de manera significativa los diversos grupos de pacientes. Como los principales parámetros del modelo se desconocen y simplemente se adecuaron para reproducir algunas de las características principales de la propagación de la DCP, hemos empleado la cuantificación de la incertidumbre (CI) para desarrollar un análisis de sensibilidad de los resultados del modelo y hemos estudiado más en profundidad el efecto de realizar una elección específica de parámetros. Por lo tanto, con este trabajo, nuestro objetivo radica en combinar diversas perspectivas (tal y como se muestra en el esquema que se incluye más abajo) a fin de proporcionar un estudio integral de la migraña y la DCP.



Principales logros

En esta tesis hemos identificado cuatro logros principales. En primer lugar, hemos combinado dos modelos existentes para desarrollar un modelo multiescalar EDP-EDO que unifica las características de la DCP en distintas escalas de tiempo. Este modelo proporciona un conocimiento profundo sobre la conexión existente entre la dinámica electrofisiológica y el carácter propagador de las ondas de la DCP. El segundo logro radica en la inclusión de información específica de cada paciente en el modelo arriba mencionado con la geometría cortical obtenida a través de resonancia magnética y con los datos DWI o datos DTI. Con el fin de describir la geometría y la propagación de la onda de la DCP en el córtex, hemos introducido diversas CdI dirigidas a comparar la propagación de la DCP en pacientes con migraña y en individuos sanos. El último logro principal es la evaluación del modelo de la DCP en dos conjuntos de datos distintos y con diversas finalidades: comparar la propagación de la DCP simulada con los síntomas que experimentan los pacientes y calcular e investigar las CdI para poder hallar diferencias significativas entre los grupos de pacientes.

Según nos consta, esta es la primera vez que las ondas de la DCP se han modelado en una geometría completa del cerebro. El uso de datos específicos del córtex cerebral de cada paciente, obtenidos a través de una resonancia magnética en combinación con las propiedades de difusión de los datos DTI, mejora la precisión y la función predictiva de este método. Se trata de un importante paso hacia la obtención de respuestas sobre preguntas vitales dentro del campo de la medicina personalizada.

Descripción del contenido de la tesis

Esta tesis se estructura en siete capítulos fundamentales. En el primer capítulo describiremos de manera detallada la DCP. Nos centraremos en su fenomenología y en su carácter propagador y destacaremos las redistribuciones iónicas implicadas en la generación y la propagación de la onda DCP. A escala microscópica, la DCP provoca una distorsión drástica en la homeostasis cerebral, seguida por una onda de depresión de la actividad neuronal. A escala macroscópica, la onda de despolarización se propaga, casi con total seguridad, debido a la difusión del potasio en el espacio extracelular (Kraio y Nicholson, 1978; Enger et al., 2015; Zandt et al., 2015). La fuerte conexión existente entre la migraña con aura visual y la DCP nos lleva a la hipótesis de que la DCP sería el desencadenante de los trastornos sensoriales observados (Lauritzen, 1994). También mencionamos de manera sucinta las subdivisiones del córtex cerebral humano en regiones, tanto por su estructura histológica (atlas Desikan-Killiany) como por sus roles funcionales en cuanto a sensación, conocimiento y comportamiento (áreas de Brodmann).

La dinámica de la DCP y su propagación están relacionadas con los flujos de iones a lo largo de la membrana de las neuronas y la difusión del potasio extracelular que puede describirse a través de diversos modelos computacionales. El Capítulo 2 se centra en la presentación de los modelos fenomenológicos y aquellos electrofisiológicos más minuciosos de la excitación neuronal que nos servirán como base para los modelos de la DCP que presentaremos en el siguiente capítulo. El primer apartado repasa los modelos matemáticos que plasman la actividad eléctrica, empezando por el típico modelo de Hodgkin-Huxley para continuar con los de Fitzhugh-Nagumo e Izhikevich. A lo largo del segundo apartado describiremos el modelo electrofisiológico, propuesto inicialmente por Cressman et al., 2009 y por Barreto y Cressman, 2011, y posteriormente ampliado por Wei, Ullah, y Schiff, 2014 al tener en consideración los metabolitos cerebrales. Este modelo permite captar la actividad electrofisiológica del cerebro y los diversos patrones de activación, dependiendo de la concentración extracelular de potasio y de la disponibilidad de oxígeno, que actúa como un estimador de la energía disponible.

En el Capítulo 3 deducimos un modelo multiescalar EDP-EDO que no solo tiene en consideración los cambios iónicos y sus correspondientes circunstancias electrofisiológicas, sino también el carácter propagador de las ondas DCP. Este modelo se construye a partir de una combinación de dos modelos existentes: el modelo electrofisiológico propuesto por Wei, Ullah, y Schiff, 2014 y un modelo reacción-difusión presentado por Rogers y McCulloch, 1994. Al acoplar ambos modelos, podemos aprovechar la reacción específica del modelo electrofisiológico en diversas concentraciones de potasio con el fin de reproducir el comportamiento neuronal característico durante una DCP. Una concentración baja de potasio, de unos 5 mM provoca una frecuencia de 12 Hz que coincide con la actividad neuronal en descanso. Un aumento de la concentración de potasio hasta los 64 mM dará como resultado un comportamiento parecido al de la DCP. El acople del sistema se realiza de manera unidireccional: la electrofisiología se rige por la difusión de potasio, si bien el componente electrofisiológico no responde a la parte propagadora del modelo. Este modelo se aproxima en el tiempo mediante diferencias finitas y en el espacio mediante elementos finitos. La última parte del capítulo está dedicada a ilustrar esta interacción a dos niveles mediante diversos ejemplos de representaciones del tejido neuronal en una y dos dimensiones.

El cuarto capítulo muestra cómo enriquecer el modelo con información específica del paciente sobre la geometría de su córtex cerebral con datos de difusión que muestren la microestructura del tejido subyacente. Como el modelo multiescalar previamente presentado resulta muy costoso a nivel computacional, en este caso hemos restringido nuestros cálculos a la simulación de la onda de potasio extracelular. Sin embargo, al estar ambas partes del modelo acopladas de manera unidireccional, la electrofisiología podrá reproducirse fácilmente a través de los resultados de la onda de potasio que se propaga por el córtex cerebral a costa de unos gastos computacionales más elevados. Debido a la definición del tensor de difusión en cada elemento de la geometría de la malla en el ajuste del método numérico de elementos finitos, los datos 3D de DTI tendrán que transformarse en datos de difusión 2D en la superficie cortical. También comentamos brevemente diversos métodos numéricos dirigidos a reducir el coste computacional para resolver el sistema lineal resultante de la discretización de los elementos finitos del modelo distribuido en la malla cortical. Por último, hemos simulado la propagación de la DCP en una geometría cerebral real obtenida a través de los datos de la resonancia magnética e incorporando los datos DTI.

El Capítulo 5 aborda la presentación de una técnica de suavizado y de las CdI geométricas y dependientes de la DCP. Las CdI geométricas incluyen las mediciones de la curvatura y el índice de regularidad de la superficie, una proporción adimensional entre el volumen determinado de la geometría y el volumen de una esfera con el mismo tamaño de superficie. A fin de corregir los artefactos de la reconstrucción cerebral y hacer que la geometría cortical resulte más realista y más adecuada para las valoraciones de la curvatura y las simulaciones de la propagación de la DCP, hemos aplicado la técnica de suavizado de Taubin antes de estimar la curvatura y el índice de regularidad de la superficie. Para las aproximaciones de la curvatura media y de la curvatura de Gauss, hemos recurrido a los métodos presentados por Magid, Soldea, y Rivlin, 2007. Las CdI dependientes de la DCP pueden calcularse mediante el postprocesamiento de los datos obtenidos a través de la simulación de la propagación de la DCP por todo el córtex para permitir establecer comparaciones (especialmente entre sujetos) y relacionar los resultados de la simulación con los síntomas observados a nivel experimental. El capítulo finaliza con un resumen de las CdI resultantes para la geometría específica del paciente y para la propagación de la DCP basada en los datos DTI que se observaron en el Capítulo 4.

En el Capítulo 6 aplicamos el modelo DCP para estudiar dos conjuntos de datos distintos. El primer caso práctico de pacientes con migraña con aura crónica aspira a establecer una correlación entre la propagación del frente de onda DCP simulado y los síntomas que experimentan los pacientes durante un ataque de migraña. El segundo caso práctico compara diversos grupos de pacientes en cuanto a la propagación de la DCP y a los CdI que se presentaron el capítulo anterior. Con estas aplicaciones, buscamos investigar el impacto de la geometría en la propagación de la DCP y su relación con la migraña (con aura). De este modo, podremos encontrar posibles clasificadores que puedan ayudarnos a distinguir entre cerebros sanos y cerebros con migraña identificando las regiones específicas que se hallan implicadas en la propagación de la DCP.

En los capítulos anteriores, los parámetros del modelo se ajustaron manualmente a fin de que se correspondieran con las dos características principales de la DCP: el tiempo total de propagación a través del córtex y la duración de una elevada concentración de potasio. En el Capítulo 7 aplicamos la cuantificación de la incertidumbre para tener en cuenta las incertidumbres en la elección de parámetro. Considerando los parámetros independientes en el espacio como variables aleatorias, aplicamos el método probabilístico de colocación, una técnica según la cual las incertidumbres en los cálculos pueden corresponderse con las incertidumbres en los parámetros empleando únicamente una cantidad reducida de simulaciones. Para poder investigar el impacto de algunas distribuciones de parámetros, normalmente asumimos que los parámetros se distribuyen de manera uniforme, pero también comentamos el caso de una distribución beta. A través de un análisis de sensibilidad basado en el enfoque del análisis de varianza (ANOVA), hemos estudiado la contribución de cada uno de los parámetros a la varianza de la solución modelo y dos medidas específicas del problema: la concentración máxima de potasio y la duración de una elevada concentración de potasio tras el paso de una onda DCP. El parámetro dependiente del espacio del término de difusión en el modelo puede, a su vez, observarse como un proceso aleatorio. Si bien abordar los procesos aleatorios en el espacio es más complejo que en el caso de otros parámetros previamente considerados, ofrecemos una breve idea de lo que podría hacerse siguiendo esta dirección y cómo podría emplearse la información deducida en la práctica.

Por último, el Apéndice A ofrece un listado de las contribuciones a diversas publicaciones y el código desarrollado, y el Apéndice B proporciona información detallada sobre los conceptos matemáticos que se emplearon en los capítulos anteriores. El Apéndice C incluye una guía introductoria sobre cómo extraer la geometría cortical y los tensores de difusión de los datos DTI, y el Apéndice D contiene resultados adicionales de las simulaciones de la DCP para los diversos pacientes que incluídos en el Capítulo 6.

Summary

Migraine is a leading cause of disability worldwide in the present-day population. One-third of migraine patients suffers from migraine aura, perceptual disturbances preceding a typical headache. Several studies suggest cortical spreading depression (CSD), a depolarisation wave that originates in the visual cortex and propagates across the cortex to the peripheral areas, to be a correlate of migraine aura (Hadjikhani et al., 2001; Richter and Lehmenkühler, 2008). Despite more than 70 years of research in the field of migraine and CSD, the origin and causes of this phenomenon are not entirely understood yet.

The passage of a CSD wave triggers various changes in the vascular response, blood flow and energy metabolism. Massive changes in ionic concentrations inside and outside of cortical neurons cause a brief period of intense activity followed by a drastic reduction of the membrane potential that silences the neuronal activity for a short period (Pietrobon and Moskowitz, 2014). A significant increase in extracellular potassium characterises CSD and the most accepted hypothesis points to extracellular potassium diffusion as its driving force (Zandt, Haken, and Putten, 2013; Zandt et al., 2015).

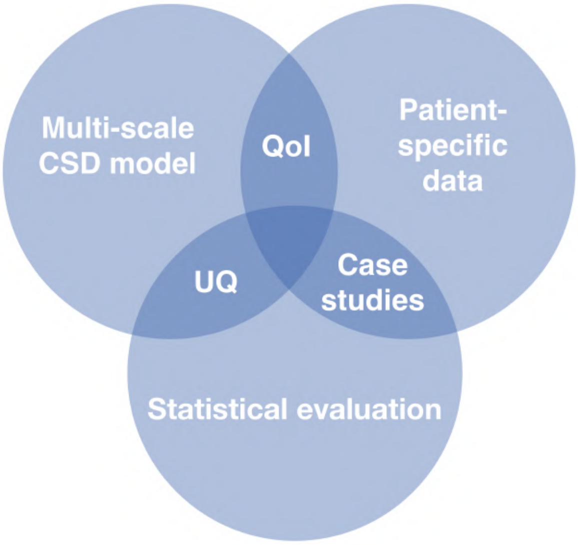
Some recent studies started to consider the effects of cortical geometries on CSD propagation, at different levels of detail. Pocci et al., 2010 simulated a wave propagating in a 2D duct containing a bend, showing that sharp bends can block the wave propagation. In a personalised modelling approach, Dahlem et al., 2015 used the Gaussian curvature of the cortex to identify potential targets for neuromodulation. These studies already hint at the fact that folds and creases of the cortex could influence the propagation of the depolarisation wave. Hence, the complex, highly individual-specific characteristics of the brain cortex might play a significant role in CSD propagation.

Motivation and main objectives

Advancements in technology have created increasing access to vast amounts of data. Especially in the medical sector, the analysis and interpretation of individual patient data can help to understand the causes and dynamics of a particular disease. Mathematical models can provide both significant insights into disease origin and progression as well as scenario analysis and risk assessment of potential treatment strategies ranging from personalised therapy and intelligent drug design to population screening and electronic health data-mining. In the investigation of migraine with aura, magnetic resonance imaging (MRI) scans including diffusion-weighted (DWI) or diffusion tensor imaging (DTI) data provide detailed information about the cortex and its underlying tissue structure. In this thesis we propose a detailed mathematical model for CSD propagation and combine it with patient-specific information on the brain geometry and diffusion characteristics of the cerebral cortex.

The objective of this thesis is to study features that play a significant role in CSD propagation and investigate the origin of migraine by simulating CSD waves on brain geometries of healthy individuals and migraine patients. To approach this study, we merge different perspectives: a multi-scale CSD model, patient-specific data and statistical evaluations

of different kinds. The characteristics of CSD are inherently multi-scale as the extracellular potassium dynamics that triggers CSD propagation occurs on the scale of minutes, whereas the electrophysiological dynamics features a much faster temporal scale. Thus, we introduce a multi-scale PDE-ODE model for the extracellular potassium concentration and combine this model with patient-specific information. In order to compare the CSD propagation between healthy individuals and migraine patients, we introduce quantities of interest (QoI) that summarise the CSD propagation on cortical geometries. Then we evaluate CSD propagation on different patient data sets. In the first case study we compare the simulated wavefront propagation with the sequence of symptoms experienced by chronic migraine patients with aura during a migraine attack; later, in a group analysis, we simulate CSD propagation and compute QoI with the purpose of finding classifiers that significantly distinguish the different patient groups. As the major model parameters are unknown and were simply tuned to reproduce some main characteristics of CSD propagation, we use uncertainty quantification (UQ) to perform a sensitivity analysis of the model outputs and study more in-depth the effect of making a particular parameter choice. Therefore, with this work, our aim was to merge different perspectives (as illustrated in the diagram below) to provide an integral study of migraine and CSD.



Main achievements

In this thesis we identify four main achievements. First, we combine two existing models to develop a multi-scale PDE-ODE model that unifies CSD characteristics on different temporal scales. This model provides a deeper understanding of the connection between the electrophysiological dynamics and the propagative character of CSD waves. The second accomplishment is the inclusion of patient-specific information in the aforementioned model with the cortical geometry obtained from MRI and with the diffusion tensor from DWI or DTI data. In order to describe the geometry and CSD wave propagation on the cortex, we introduce various QoI that aim at comparing CSD propagation between migraine patients and healthy subjects. The last main achievement is the evaluation of the CSD model on two different data sets with different purposes: to compare the simulated CSD propagation to the symptoms experienced by patients and to compute and investigate QoI to find significant differences between the patient groups.

To the best of our knowledge, this is the first time that CSD waves were modelled on a whole brain geometry. Utilising patient-specific data of the brain cortex from MRI data in combination with the diffusion properties from DTI data further improves the accuracy and the predictive role of this approach. This is an important step towards addressing vital questions in the field of personalised medicine.

Description of the thesis content

This thesis is structured in seven main chapters. In the first chapter we discuss CSD in detail. We focus on its phenomenology and propagative behaviour and highlight the ionic rearrangements involved in the generation and propagation of the CSD wave. At the microscopic scale, CSD causes a drastic distortion in brain homeostasis followed by a wave of depression of neuronal activity. At the macroscopic scale, the depolarisation wave propagates, most probably due to the diffusion of potassium in the extracellular space (Enger et al., 2015; Kraio and Nicholson, 1978; Zandt et al., 2015). The strong connection between migraine visual aura and CSD leads to the hypothesis that CSD is triggering the observed perceptual disturbances (Lauritzen, 1994). We briefly mention subdivisions of the human cerebral cortex into regions based either on the histological structure (Desikan-Killiany atlas) or their functional roles in sensation, cognition, and behaviour (Brodmann areas).

The dynamics of CSD and its propagation involve ion fluxes across the membrane of neurons and the diffusion of extracellular potassium that can be described by different computational models. Chapter 2 is dedicated to the introduction of phenomenological and more detailed electrophysiological models of neuronal excitation that will serve as the basis for the CSD models we introduce in the next chapter. The first section reviews mathematical models capturing electrical activity, starting from the classical Hodgkin–Huxley model and continuing with the Fitzhugh–Nagumo and the Izhikevich model. In the second section we describe the electrophysiological model, originally proposed by Cressman et al., 2009 and Barreto and Cressman, 2011 that was extended in Wei, Ullah, and Schiff, 2014 taking into account brain metabolites. This model is able to capture brain electrophysiological activity and different firing patterns depending on the extracellular potassium concentration and the availability of oxygen that serves as an estimate for the available energy.

In Chapter 3 we derive a multi-scale PDE-ODE model which takes into account not only the ionic changes and their corresponding electrophysiological details but also the propagative character of CSD waves. The model is built upon a combination of two existing models: the electrophysiological model proposed by Wei, Ullah, and Schiff, 2014 and a reaction-diffusion model introduced by Rogers and McCulloch, 1994. Coupling these two models, we exploit the specific reaction of the electrophysiological model to different potassium concentrations in order to reproduce the characteristic neuronal behaviour during a CSD. A low potassium concentration of about 5 mM triggers a spiking frequency of 12 Hz which coincides with the neuronal activity at rest. Increasing the potassium concentration up to 64 mM results in CSD-like behaviour. The coupling of the system is unidirectional: the electrophysiology is ruled by the potassium diffusion, but there is no feedback from the electrophysiological component to the propagative part of the model. This model is approximated in time by finite differences and in space by finite elements. The last part of the chapter is devoted to examples in one- and two-dimensional representations of neuronal tissue to illustrate the interaction at these two levels.

The fourth chapter shows how to enrich the model with patient-specific information on the cerebral cortex geometry with diffusion data representing the microstructure of the underlying tissue. As the previously introduced multi-scale model is computationally expensive, here we restrict our computations to the simulation of the extracellular potassium wave. However, as the two parts of the model are coupled unidirectionally, the electrophysiology could be easily reproduced from the results of the propagating potassium wave on the cerebral cortex at the expenses of higher computational costs. Due to the definition of the diffusion tensor on each element of the mesh geometry in the finite elements numerical approximation setting, the 3D DTI data has to be transformed to 2D diffusion data on the cortical surface. We briefly discuss different numerical methods to reduce the computational costs of solving the linear system resulting from the finite element discretisation of the distributed model on the cortical mesh. Finally, we simulate the CSD propagation on a real brain geometry derived from MRI data and incorporate diffusion data from DTI imaging.

Chapter 5 is devoted to the introduction of a smoothing technique and of geometric and CSD-dependent QoI. The geometric QoI include curvature measures and the surface regularity index, a dimensionless ratio between the given volume of the geometry and the volume of a sphere with the same surface size. To correct artefacts from the brain reconstruction and make the cortical geometry more realistic and more suited for curvature estimations and simulations of CSD propagation, we apply Taubin smoothing before estimating the curvature and the surface regularity index. For the discrete Gaussian and mean curvature approximations, we resort to methods introduced by Magid, Soldea, and Rivlin, 2007. The CSD-dependent QoI can be computed by post-processing the results from the simulation of CSD propagation across the whole cortex to allow for comparisons (especially between subjects) and relate simulation results to experimentally observed symptoms. We conclude the chapter with an overview of the resulting QoIs for a patient-specific brain geometry and CSD propagation based on the DTI-data discussed in Chapter 4.

In Chapter 6 we apply the CSD model to study two different data sets. The first case study of chronic migraine patients with aura aims to correlate the simulated CSD wavefront propagation to the symptoms experienced by the patients during a migraine attack. The second case study compares different patient groups in terms of CSD propagation and the QoI introduced in the previous chapter. With these applications, we want to investigate the impact of the geometry on the CSD propagation and its correlation to migraine (with aura) and find potential classifiers that could help to distinguish between migraine brains and healthy brains by identifying specific regions involved in the CSD propagation.

In the previous chapters the model parameters were tuned manually to match two major features of CSD, the total propagation time across the cortex and the duration of high potassium concentration. In Chapter 7 we apply uncertainty quantification to account for the uncertainties in the parameter choice. Considering the spatially independent parameters as random variables, we apply the probabilistic collocation method, a technique by which uncertainties in the computations can be related to parameter uncertainties using only a small number of simulations. To investigate the impact of certain parameter distributions, we typically assume the parameters to be uniformly distributed but also comment on the solution statistics in the case of an underlying beta-distribution. With a sensitivity analysis based on the analysis of variance (ANOVA) approach, we study the contribution of each parameter to the variance of the model solution and two problem-specific measures: the maximum potassium concentration and the duration of high potassium concentration following the passage of the CSD wave. The space-dependent parameter of the diffusion term in the model can, in turn, be seen as a random process. Although coping with random processes in space is more complex than for the other parameters previously considered, we provide a brief idea of what could be done in this direction and how the inferred information could be used in practice.

Finally, Appendix A provides a list of contributions to publications and developed code, and Appendix B gives detailed information on the mathematical concepts used in the previous chapters. Appendix C provides an introductory guide on how to extract the cortical geometry and diffusion tensors from MRI and DTI data, and Appendix D contains additional results of CSD simulations for the different patients in Chapter 6.

Contents

Chapter 1

Biological introduction

"In order to understand the universe, we need to connect observations into comprehensive theories. Earlier traditions usually formulated their theories in terms of stories. Modern science uses mathematics."

Yuval Noah Harari, in *Sapiens: A Brief History of Humankind* (2015)

Migraine is a prevailing disease in the present-day population with about 15% of the global population suffering from it. The prevalence in women is 18.79%, whereas only 10.68% of men are affected (Vos et al., 2012). Typical symptoms of migraine include unilateral severe headache, nausea and photophobia, and one-third of the migraine patients additionally experiences a migraine aura preceding the typical headache (Hadjikhani et al., 2001; Richter and Lehmenkühler, 2008). An aura is a transient focal neurological phenomenon causing perceptual disturbances that can occur in the form of visual and auditorial changes like hallucinations or déjà-vus.

Several studies and experiments suggest that cortical spreading depression (CSD) underlays visual aura (Hadjikhani et al., 2001; Richter and Lehmenkühler, 2008). CSD is a depolarisation wave propagating across the cortex in the course of 20 minutes. This wave causes a drastic failure of brain homeostasis followed by a wave of depression of neuronal activity.

The CSD wave was first described and measured in mice and rabbits by Aristides Leão (Leão, 1944), and despite more than 70 years of research in this field, the underlying mechanisms of its origin and its propagation are not yet clearly understood. Previous results are controversial due to a lack of experimental evidence and the complexity of the phenomenon. Hence, a deeper understanding of the biophysical and molecular principles underlying the initiation and propagation of CSD is essential for a deeper understanding of this disease which can help to construct new interventional approaches.

The wave of excitement, that serves as the basis of the CSD, is caused by successively conducting spikes of dendritic activity, also called *action potentials*. Action potentials are transient, sudden and large changes in the membrane potential of excitable cells that are governed by the opening and closing of ion channels. Depending on the type of excitable cell, the duration and shape of the action potential can vary tremendously. Neuronal activity is measured in terms of the firing rate, that is, the number of action potentials produced in one second (Hz).

During CSD propagation, neurons undergo a brief period of intense firing, followed by a drastic reduction of the membrane potential, which silences the neuronal activity for a short period (of 1 to 3 minutes), and then a slow recovery of the firing activity in which neurons get back to their normal frequency (Sawat-Pokam et al., 2016). CSD is characterised by a significant increase in both extracellular potassium and glutamate, as well as a rise in the intracellular sodium and calcium concentrations, and the two most accepted hypotheses suggest that CSD propagation is due to diffusion of either potassium or glutamate in the extracellular space (Zandt et al., 2015).

Some recent studies (Dahlem et al., 2015; Pocci et al., 2010) started to consider the effects of cortical geometries on CSD propagation, at different levels of detail. Folds and creases of the cortex influence the propagation of the depolarisation waves (and can stop the migraine in different positions depending on the patient).

This chapter is dedicated to the phenomenology of CSD, its characteristics and the correlation with migraine. We shortly describe in the following sections the biological compartments and processes involved in neuronal excitation. For these descriptions, we follow mainly the work by Sadava et al., 2006. Subsequently, we explain the compartments of the cerebral cortex and the mechanisms of CSD in more detail, drawing their connection to migraine and migraine with aura.

1.1 Generation and propagation of action potentials

The cell membrane of neurons acts as a capacitor separating different types of ions. The main ions involved in the electrophysiology of action potentials are sodium (Na^+), potassium (K^+), calcium (Ca^{2+}) and chloride (Cl^-). At rest, the concentration of Na^+ is higher in the extracellular fluid, while the concentration of K^+ is higher inside the cell. Large organic anions inside the cell are responsible for the overall negative charge across the membrane. At rest, neurons exhibit a membrane potential between -60 mV and -70 mV.

Ion specific channels and ion pumps in the cell membrane govern the distribution and changes of the ion concentration in the intracellular and extracellular space. The sodium-potassium pump actively exchanges K^+ and Na^+ ions in order to keep the concentration of K^+ inside the cell higher when the neuron is at rest. This pump uses adenosine triphosphate (ATP) as an energy source to move the ions against the electrochemical gradient in the cell. Ion channels are selective passive transporters – allowing only some types of ions to pass through – and the direction and magnitude of the movement of ions depend on the electrochemical gradient.

When an action potential is generated, a depolarising stimulus causes the voltage-gated Na^+ channels to open. Driven by the electrochemical gradient, this allows Na^+ ions to enter the cell and thus depolarise the membrane potential. If the threshold of -50 mV is reached, more Na^+ channels open causing a flush of Na^+ ions inside the neuron and the membrane potential to rise up to 50 mV. At the peak of the action potential, Na^+ channels start to close and at the same time voltage-gated (but time-delayed) K^+ channels open. This leads to an efflux of K^+ to the extracellular space and the repolarisation of the membrane potential towards the resting potential of -60 mV. Many voltage-gated K^+ channels do not close immediately when the membrane returns to its resting potential. This can lead to a hyperpolarisation of the membrane potential to values that are even lower than the original resting potential. Subsequently, the sodium-potassium pump restores the resting balance. After the excitation, the voltage-gated channels undergo a refractory period (of 1-2 milliseconds in which they do not react to any further incoming stimulus). A typical action potential in the course of time with the corresponding ion channel activities is shown in Figure 1.1.

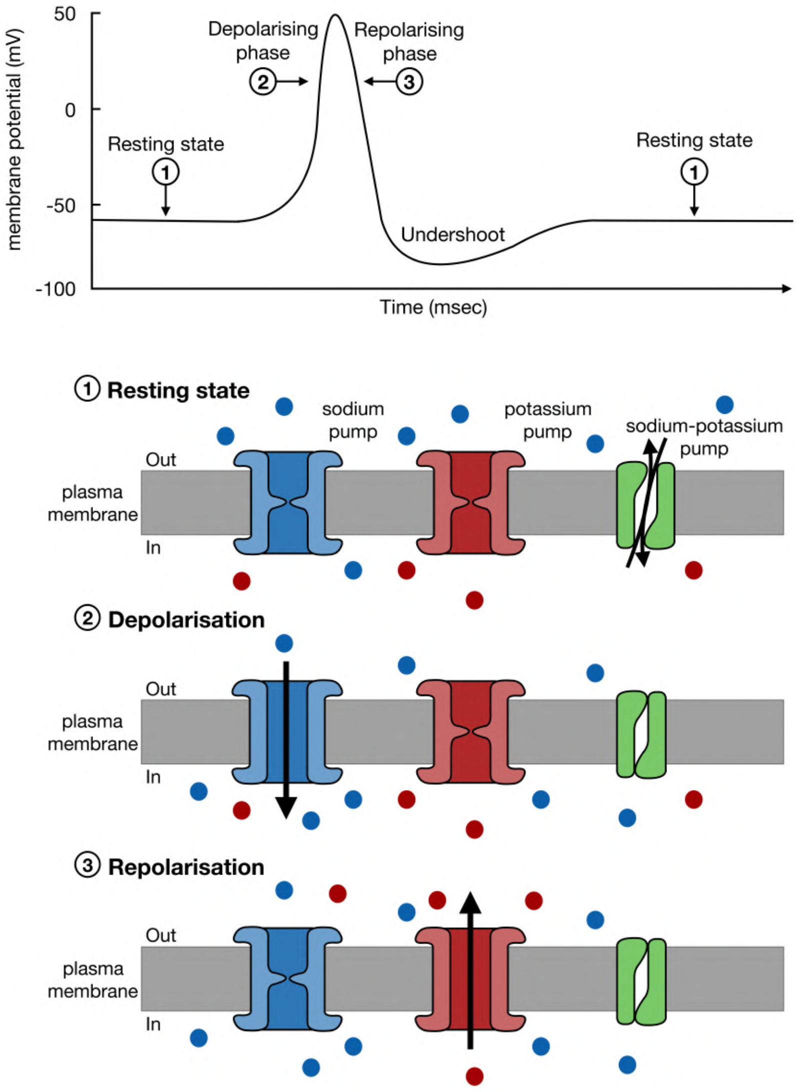


FIGURE 1.1: The changes in the membrane potential, the dynamics of the voltage-gated ion channels and the ion fluxes across the membrane in the different phases of an action potential generation.

Action potentials travel along axons without losing signal strength. As a consequence of the refractory period of the voltage-gated Na^+ channels, this movement is unidirectional. In vertebrates, the conductance of axons is 200 times faster than in invertebrates due to the myelination of the axon. Myelin sheaths – outgrowths of a type of glial cells – are wrapped around the axon leaving regularly spaced gaps, called nodes of Ranvier. The speed of conduction increases because instead of generating action potentials all along the axon, they are only expressed in the nodes of Ranvier; this process is called saltatory conduction.

1.1.1 Transmission of action potentials

Neurons transmit action potentials to neighbouring cells via the space between the presynaptic and the postsynaptic membrane, also called *synapse*. In electrical synapses, the action potential is directly passed from the presynaptic to the postsynaptic cell. Chemical synapses transmit action potentials with the help of neurotransmitters and allow more complex dynamics.

Action potentials arriving in the presynaptic axon terminal cause voltage-gated Ca^{2+} channels to open. As the concentration of Ca^{2+} is higher in the extracellular space, Ca^{2+} ions enter the axon terminal and trigger a release of vesicles containing neurotransmitters. These vesicles fuse with the membrane of the axon terminal releasing the neurotransmitters into the synapse. The neurotransmitters diffuse across the synapse and bind to receptors on the postsynaptic membrane. These receptors activate chemically gated channels that allow Na^{+}, K^{+} and Ca^{2+} to pass, but because of the electrochemical gradient, there is an influx of Na^{+} ions into the cell depolarising the membrane. Sufficient depolarisation leads to an activation of the voltage-gated Na^{+} channels and a generation of an action potential in the postsynaptic cell. The neurotransmitters in the synapse are broken down by specific enzymes, and the neurotransmitters and vesicles are recycled for further use in the presynaptic axon terminal.

The synapses between neurons can be excitatory or inhibitory depending on the neurotransmitters. Excitatory postsynaptic potentials depolarise the postsynaptic membrane, while inhibitory postsynaptic potentials cause hyperpolarisation. Every postsynaptic neuron has a variety of different receptors for neurotransmitters, and each neurotransmitter has multiple receptor types. These receptors and channels are ion selective, and the type of ions passing these channels determines if the presynaptic action potential triggers an excitatory or inhibitory action potential.

Neurons are connected to many axon terminals by branching fibres (dendrites) and that way, a lot of incoming information arrives at the neurons' cell body. Whether or not a postsynaptic neuron conducts the action potential does not only depend on the strength of the input stimulus of the presynaptic neuron but rather on the sum of all input stimuli. The neuron integrates the information that arrives in the dendrites in the axon hillock, a specialised part of the cell body of a neuron that connects to the axon. Only if the combination of the excitatory and inhibitory potentials reaches the threshold, an action potential is generated and passed on along the axon.

1.1.2 Neurotransmitters in the central nervous system

More than 50 different neurotransmitters have been discovered, varying from simple amino acids to peptides and gasses. The main neurotransmitters in the central nervous system are glutamate (or glutamic acid) and γ-aminobutyric acid.

The excitatory neurotransmitter glutamate can bind to a variety of different receptors, and the activation of glutamate receptors always brings about an influx of Na^{+} ions and a depolarisation. Depending on the type of receptor, glutamate can trigger either a rapid inflow of Na^{+} ions into the postsynaptic cell or a slower and longer-lasting inflow that is coupled to an influx of Ca^{2+} ions. Here, Ca^{2+} ions act as a second messenger and can cause different long-term effects. Free glutamate is rapidly removed from the synapse by transporters in the neuronal and glial membrane. In addition, glutamate acts as a precursor for the synthesis of γ-aminobutyric acid and this reaction is catalysed by glutamate decarboxylase.

The primary inhibitory neurotransmitter in the central nervous system is γ-aminobutyric acid, whose binding to the various receptors can lead to either an influx of Cl^- ions into the cell or an efflux of K^+ ions. For both receptor types, this results in a negative change in the membrane potential and thus a hyperpolarisation.

1.1.3 Energy metabolism

Most actions taking place in organisms require energy. For example, after the generation of an action potential, the sodium-potassium pump requires energy in the form of ATP in order to transport the Na^+ ions and K^+ ions against the electrochemical gradient across the membrane. Depending on the absence or presence of oxygen, different metabolic pathways transform glucose into energy. During this conversion the free energy is stored in the form of ATP, nicotinamide adenine dinucleotide (NAD^+) or flavin adenine dinucleotide (FAD). These coenzymes act as electron carriers and can release energy for cellular work after oxidation.

There are three main processes involved in the energy generation from glucose: glycolysis, cellular respiration, and fermentation. Glycolysis takes place in the cell's cytosol and each molecule of glucose is converted into two molecules of pyruvate. During the ten enzyme-catalysed reactions of glycolysis, four molecules of ATP and two molecules of NAD^+ are gained. These reactions do not need any oxygen. Another way of energy generation happens in the citric acid cycle. This is a pathway of eight reactions in the mitochondrial matrix in which each pyruvate molecule is converted into three molecules of CO_2. The free energy released from these reactions is stored in the form of ATP and of the electron carriers NAD^+ and FAD.

The citric acid cycle does not directly require oxygen but oxygen is necessary to oxidise the electron carriers in order to provide a sufficient supply of excitable electron carriers. In the absence of oxygen, pyruvate is reduced in the cytosol. There are several different fermentation ways but the most frequently occurring one is the lactic acid fermentation. This fermentation takes place especially in regions of high activity and energy consumption with a low oxygen supply. Pyruvate is reduced to lactate and the energy is stored in the form of NAD^+. In total, the yield of cellular respiration is 32 molecules of ATP for every molecule of glucose in the presence of oxygen. Under anaerobic conditions, the glycolysis via fermentation only produces two molecules of ATP per molecule of glucose.

1.1.4 Astrocytes and glutamatergic synapses

The energy metabolism of central nervous system cells is closely linked to their synaptic activity. Cell respiration and fermentation provide the neurons with the energy necessary to trigger action potentials and restore the resting potential. Astrocytes are characteristic star-shaped glial cells in the brain and spinal cord, in which the glucose metabolism is coupled to the glutamate uptake from the synapse and to the transfer of lactate and glutamine to the neuron. Astrocytes in the central nervous system are predominately positioned between the synapse of neurons and capillaries. This makes them the ideal connectors between synaptic activity and energy metabolism. Astrocytic end-feet are wrapped around the capillaries providing a constant supply with glucose and oxygen.

Astrocytes rapidly take up the glutamate from the synapse that was released by the presynaptic neuron. This uptake is driven by the electrochemical gradient of Na^+ ions, which acts as a co-transporter for glutamate. In astrocytes, the glutamate is converted to glutamine in a catalysed reaction that requires ATP. The glutamine is then released into the extracellular space, taken up by the neurons and converted to glutamate to restock the neurotransmitter pool. The co-transport of Na^+ ions into the astrocyte increases the intracellular Na^+ concentration, which leads to the activation of the sodium-potassium pump. This stimulates the glucose uptake and triggers aerobic glycolysis. In aerobic glycolysis, glucose is transformed into lactate in the presence of sufficient oxygen. The lactate is transferred to the neuron where it is converted to pyruvate. This pyruvate is – with the help of pyruvate dehydrogenase – converted to acetyl coenzyme A and enters the citric acid cycle converting energy. The main processes involved in the action potential transmission of a glutamatergic synapse are shown in Figure 1.2.

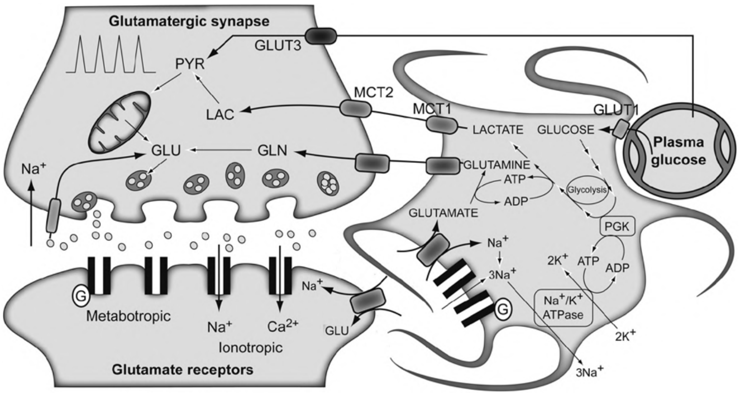


FIGURE 1.2: Schematic representation of a glutamatergic synapse and the recycling pathway of its neurotransmitters via a neighbouring astrocyte (Rodrigues, Valette, and Bouzier-Sore, 2013).

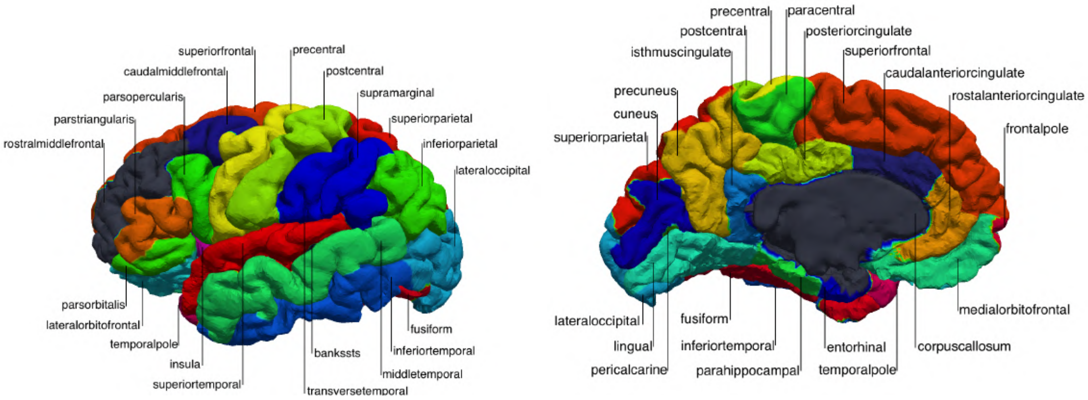
Due to the reduced sodium permeability (compared to neurons) and due to a high density of potassium channels, astrocytes clear the extracellular potassium that is released by active neurons. During an activity, there is a build up of potassium, which has been suggested to be cleared by the astrocytes (Walz, 2000).

1.2 Cerebral cortex and its compartments

The nervous system allows us to experience our environment and react to it accordingly. More than 10^{11} neurons and glial cells process information in the form of electrical impulses and pass them on through more than 10^{14} synapses.

The cerebrum (lat. brain) is the principal part of the brain in mammals, located in the front area of the skull and consisting of two hemispheres, left and right, separated by a fissure. It makes up the largest part of the brain and is the centre of the nervous system playing a vital role in memory, language, attention, consciousness, and perception. The *cerebral cortex* is a thin layer (2 - 4 mm) of grey matter that overlays the cerebrum and is folded in larger mammals, thus allowing for an increased surface despite the limiting size of the skull. The ridges of the folding are called *gyri*, and the fissures are termed *sulci*. The grey matter contains neuronal cell bodies, dendrites, axons, glial cells, synapses and capillaries. In contrast to that, white matter is mainly composed of long myelinated axons that are responsible for the lighter colouring of this area.

Different cortical regions can be described based on the histological structure and their functional roles in sensation, cognition, and behaviour. A widely used parcellation in automated labelling systems is the Desikan-Killiany atlas, dividing each hemisphere into 34 cortical regions of interest (ROI) (Desikan et al., 2006). This subdivision of the cortex into gyral-based neuroanatomical regions is reliable and serves for morphological as well as functional studies and is shown in Figure 1.3. Brodmann introduced the first map to subdivide the human cortex based on the neuronal organisation, and many of the areas identified have been found to correlate to diverse cortical functions (Brodmann, 2006). For example, Brodmann area 17 is the primary visual cortex and the left Brodmann areas 44 and 45 coincide with Broca's speech and language areas (Broca, 1861).



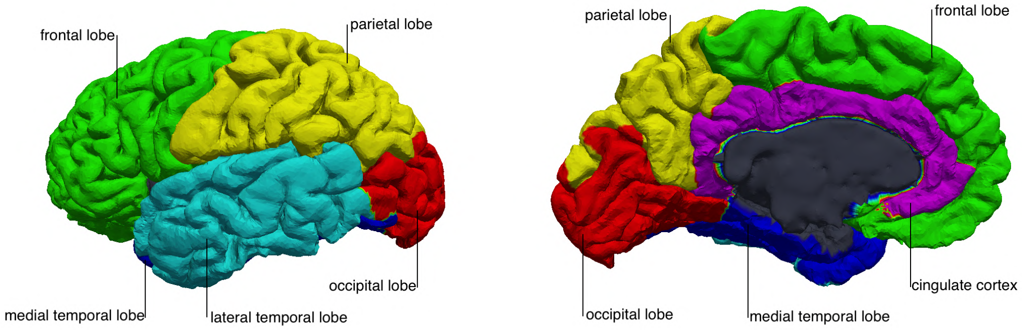
(a) lateral view (b) medial view

FIGURE 1.3: The 34 regions of interest of the cerebral cortex, in the lateral (a) and medial (b) view of the left hemisphere.

A more general classification of the cerebral cortex is based on gross topographical conventions into six lobes: medial lobe, lateral temporal lobe, occipital lobe, parietal lobe, frontal lobe, and cingulate cortex. The lobes of the left hemisphere are shown in Figure 1.4. Originally this classification was purely anatomical, but recent discoveries have shown recently that the different lobes also correspond to different brain functions (Ribas, 2010).

1.3 Cortical spreading depression

CSD is a propagating wave of depolarisation of neuronal and glial cells in the grey matter originating in the visual cortex (located in the occipital lobe), spreading across the cortex, and followed by a wave of suppression of spontaneous activity. The outstanding duration of depolarisation during the CSD (lasting up to a minute or more) and its extremely slow propagation (about 3 mm/min), are in stark contrast to the action potential features (depolarisation lasting milliseconds and propagation velocity of several m/sec) thus distinguishing the CSD from the normal neuronal activity.



(a) lateral view (b) medial view

FIGURE 1.4: The lobes of the brain, in the lateral (a) and medial (b) view of the left hemisphere.

The depolarisation wave of CSD causes various and tremendous changes in the vascular response, blood flow, and energy metabolism. During this excessive depolarisation, there are drastic fluxes inside and outside the cell, and vast quantities of the neurotransmitter glutamate are released and can be observed in the extracellular space. Thus, CSD can be seen as a reaction-diffusion process, where the reaction includes processes like the release of K^+ and glutamate, the pump activities and the tissue recovery, while the diffusion of K^+ and glutamate enable CSD propagation.

The first scientist to describe the CSD was Aristides Leão observing the basic phenomenon of the propagating depolarisation waves triggered in the cortex of rabbits, pigeons, and cats (Leão, 1944, 1947). His findings gave rise to the hypothesis that CSD is the underlying cause of migraine aura. In the past, CSD has been widely studied in animal models, but only lately the occurrence in the human brain started to be investigated. A lot of these recent observations and investigations support this hypothesis, but the electrophysiological features of CSD have not been observed in a migraine patient yet. The main reason for this is that non-invasive electroencephalography techniques are not sensitive enough to detect the weak signal of a CSD.

However, in injured brains CSD can be observed after strokes as peri-infarct depolarisation or during global hypoxia, enlarging the damaged tissue area (Zandt, Haken, and Putten, 2013). Under normal metabolic conditions, CSD does not cause cell death or long-term damage but increases the bioenergetic burden on the tissue (Pietrobon and Moskowitz, 2014).

1.3.1 Phenomenology of CSD

The complex phenomenon of CSD evolves in different stages and can be subdivided into an early, main and late phase (Pietrobon and Moskowitz, 2014). These phases are characterised by massive rearrangements of ions across the plasma membrane and are associated with changes in the pH-value as well as neuronal cell swelling in response to osmolarity changes.

The onset of the propagation wavefront triggers complex dynamics leading to abrupt changes in the neuronal behaviour. First neurons undergo a brief period of intense activity, featuring a firing rate that is 10-20 times higher than the one at rest, typically between 8 and 12 Hz (Brötzner et al., 2014; Hyder et al., 2013). This period is followed by a membrane hyperpolarisation silencing the neuronal activity for a couple of minutes, after which the neurons slowly recover their spiking activity and return to their normal spiking frequency at rest (Pietrobon and Moskowitz, 2014; Sawat-Pokam et al., 2016; Zandt et al., 2015).

In the early phase, the membrane potential is depolarised for a few seconds. This rapid change in the membrane voltage is conjoined with a swift increase of extracellular K^+ to 30-60 mM from a baseline value of 2.7-3.5 mM, a fast decrease in extracellular Na^+ and Cl^- to 50-70 mM from a baseline value of 140-150 mM and in Ca^{2+} to 0.2-0.8 mM from 1-1.5 mM (Pietrobon and Moskowitz, 2014). This is followed by the main phase, 15-20 seconds in which the entire plasma membrane is completely depolarised. The decrease in extracellular Na^+ is greater than the one in extracellular K^+, but the electroneutrality is maintained by the efflux of organic anions, like glutamate and aspartate. The electrophysiological features and ionic changes during CSD are shown in Figure 1.5.

In both neuron and astrocyte, CSD causes a large increase in the intracellular Ca^{2+} concentration. However, the rise in the neuronal Ca^{2+} precedes the one in the astrocyte and is not affected by the latter. During CSD, the astrocyte depolarisation is caused by the increase in extracellular K^+ and is mostly of passive nature.

The influx of Na^+, Cl^- and water ions cause neuronal swelling of about 40-70%, while astrocytes only reveal passive swelling in response to a CSD-induced higher extracellular concentration of K^+ (Somjen, 2001).

The cerebral blood flow and brain metabolism are closely connected; the brain blood flow increases with a rising demand for oxygen and glucose. As ATP is essential to restore the ionic gradients and repolarise the membrane potential after an action potential, CSD is associated with high energy metabolism and consequently a large transient rise in the cerebral blood flow. A potential source of ATP are the astrocytes that release ATP in correspondence to the intracellular Ca^{2+} signalling. In addition, CSD stimulates anaerobic glycolysis which results in a high production of lactate mainly generated in astrocytes.

1.3.2 Mechanisms of CSD

Experimentally, CSD can be triggered in healthy brain tissue of animals by the local release of K^+ or glutamate (Pietrobon and Moskowitz, 2014). Similar to the neuronal action potential, the CSD propagates in an all-or-nothing manner, once initiated and independent of the stimulus strength or type. Experimental data, as well as computational models, support the concept that increasing the K^+ concentration above a threshold is a key event triggering CSD, as originally proposed by Grafstein, 1956.

Four hypotheses explaining the CSD propagation on the cortex have been proposed and accepted at different levels. The potassium and glutamate hypotheses state that CSD propagates due to diffusion of potassium or glutamate in the extracellular space. The neuronal gap junction hypothesis suggests that CSD propagates through the opening of neuronal gap junctions. The glial hypothesis states that the CSD is transmitted through glial gap junctions. The most accepted assumption is that of the extracellular potassium

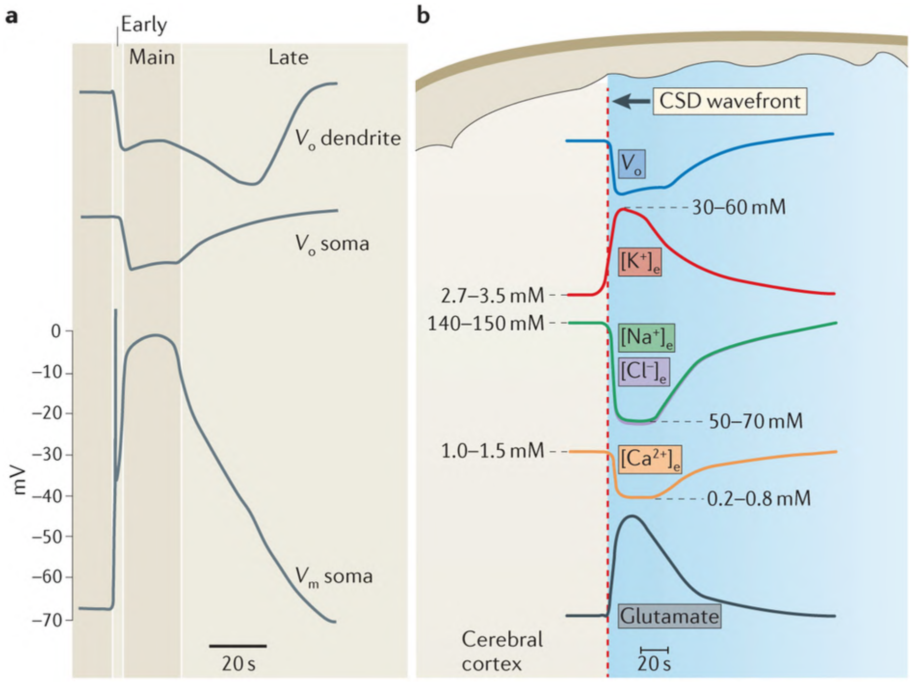


FIGURE 1.5: The electrophysiological features (a) and ionic changes (b) during CSD (Pietrobon and Moskowitz, 2014).

and glutamate diffusion (Zandt et al., 2015), which is also supported by various experimental studies (Enger et al., 2015; Kraio and Nicholson, 1978).

The depolarisation wave propagates at a uniform velocity of 2-5 mm/min (Somjen, 2001 and references therein) and the wave of high excitation is followed by a wave of inhibition, reducing the spontaneous synaptic activities and action potential firings lasting several minutes (Pietrobon and Moskowitz, 2014; Porooshani et al., 2004; Sawat-Pokam et al., 2016). These changes are reversible within 8 to 10 minutes, which coincides with the time frame of recovery from suppression activity measured in electroencephalography (EEG) (Pietrobon and Moskowitz, 2014).

1.4 Migraine

Migraine is a common neurological disorder characterised by an episodic headache, with a cumulative incidence of 43% in women and 18% in men (Steward et al., 2008). Migraine significantly affects the quality of life and imposes a severe public burden on the individual as well as on society. The 2013 Global Burden of Disease Study claims migraine as the sixth highest cause of years lost due to disability worldwide, and the World Health Organization ranks migraine among the world's most disabling medical illnesses (Silberstein, 2004; WHO, 2017). In the US alone, migraine costs the employers about US$ 13 billion annually, and the yearly costs for treatments exceed US$ 1 billion (Silberstein, 2004).

Since 1988, the Headache Classification Committee of the International Headache Society classifies and groups headache disorders and establishes diagnostic criteria to define migraine (Lipton et al., 2004). Migraine can be divided into two main subtypes: migraine with aura and migraine without aura. In general, migraine attacks are characterised by typically unilateral and pulsating moderate to severe headache, lasting between 4 and 72 hours, often accompanied by nausea, phonophobia, and photophobia (migraine without aura) (Pietrobon, 2005). In migraine with aura, the headache is preceded by neurological symptoms, mostly visual but may also involve other senses or cause speech deficits.

In addition to that, migraine has a strong genetic component, with a likely multifactorial polygenetic inheritance (Kors et al., 2004). Familial hemiplegic migraine (FHM), a rare autosomal dominantly inherited subtype of migraine with aura, is the only case in which the causative gene has been identified (Pietrobon, 2005). FHM is characterised by aura symptoms consisting of motor weakness or paralysis, and thus, FHM attacks resemble typical migraine attacks with aura. In general, the genetic load can be seen as an inherent migraine threshold that is influenced by external and internal factors.

Migraine is acutely treated by administering non-steroidal anti-inflammatory drugs, ergot derivatives, or triptans (Ayata et al., 2006). When patients experience more than three or five attacks per month, prophylaxis is recommended in order to reduce the frequency and the intensity of migraine headaches. However, migraine prophylaxis is only partially effective in many patients, and a deeper understanding of molecular, cellular, genetic or physiological drug targets is sought (Ayata et al., 2006).

1.4.1 Migraine with aura

For about 30% of the migraine patients, the migraine attack is preceded by an aura composed of transient focal neurological symptoms such as visual disturbances, paresthesias (abnormal sensations such as tingling or tickling of the skin for no apparent reasons) or language disturbances (Tommaso et al., 2014). The visual disturbances during migraine aura are characterised by a serrated arc of shimmering shapes, the so-called *scotoma*. This scotoma usually appears in the form of a small blind or scintillating spot, increasing in size and drifting across the visual field to one side (Bolay and Moskowitz, 2005).

Analysing his own visual aura, the physiologist Karl S. Lashley published in 1941 his observations of the scotoma and was the first to postulate that the migraine aura results from depressed neuronal activity in the visual cerebral cortex (Lashley, 1941).

1.4.2 Connection with CSD

The considerable connection between migraine visual aura and CSD, lead to the hypothesis that CSD is underlaying migraine aura (Lauritzen, 1994). Recently, using blood oxygenation level-dependent functional magnetic resonance imaging, Hadjikhani et al., 2001 observed various cerebrovascular changes, typical for the CSD, in the cortex of migraine patients while experiencing a visual aura. Another evidence that visual aura results from CSD was obtained with magnetoencephalography (MEG). During a spontaneous visual aura, slow changes of the cortical field were measured that correspond to the potential changes during neuronal depolarisation in CSD (Bowyer et al., 1999, 2001). This demonstration of cerebrovascular and magnetic field correlates of CSD in migraine patients strengthens the conclusion that CSD underlies visual aura.

However, some arguments are questioning the long-standing belief that CSD is underlying migraine aura (Borgdorff, 2018) making the origin of migraine with aura and its connection to the CSD an up-to-date and still controversial topic to investigate.

1.4.3 Migraine and grey matter

Several whole-brain studies identified a common reduction in grey matter volume in migraine patients, in particular in the frontal cortex and the cingulate gyrus (Jia and Yu, 2017). These changes in grey matter may indicate the location and mechanisms of pain processing, but also pinpoint cortical areas that contribute to the propagation of the CSD.

1.5 Summary

Migraine is a common disease in the present-day population, with one-third of the migraine patients suffering from migraine aura, perceptual disturbances preceding the typical headache. CSD, a depolarisation wave that originates in the visual cortex and propagates across the cortex to the peripheral areas, has been suggested as a correlate of the visual aura by several studies.

On the microscopic scale, CSD causes a drastic distortion in the brain homeostasis followed by a wave of depression of neuronal activity. On the macroscopic scale, the depolarisation wave propagates at a uniform velocity, most probably due to the diffusion of potassium or glutamate in the extracellular space.

Up to now, little is known about the origin of this phenomenon and possible curative treatments. However, the complex and highly individual characteristics of the brain cortex suggest that the geometry might have a significant impact in supporting or contrasting the propagation of CSD. Accurate patient-specific computational models are thus fundamental to cope with the high variability in cortical geometries among individuals.

Chapter 2

Overview of models for neuronal dynamics

"Remember that all models are wrong;
the practical question is how wrong do they have to be to not be useful."

Box, G. E. P., and Draper, N. R., in *Empirical Model Building and Response Surfaces, John Wiley & Sons, New York, NY (1987)*

The slow propagative wave of CSD on the cerebral cortex is marked by neuronal depolarisation and redistribution of ions, followed by a short period of suppressed neuronal activity. This reaction-diffusion process consists of a variety of local mechanisms, like active or passive ionic movements and pump activities, and diffusion of potassium or glutamate, causing CSD and enabling it to propagate.

Several mathematical models for CSD, varying from reaction-diffusion models to microscopic models accounting for the cells' connectivity, have been proposed in the past. During spreading depression, the extracellular potassium concentration follows approximately the time-course of depolarised neurons and glial cells (Kraio and Nicholson, 1978). Based on this fact, Tuckwell and Miura, 1978 proposed a model for CSD wave propagation in 1D heterogenous space, including potassium and calcium fluxes, extracellular diffusion and active transport pumps in a Hodgkin–Huxley like system of equations. The numerical results feature the basic qualitative properties of the spreading depression wave and account for the annihilation of two colliding waves. Reggia and Montgomery, 1996 coupled synaptic connectivity and extracellular potassium uptake at a single cell level in a simplified 2D array representation of the cortex. Using reaction-diffusion equations to describe the potassium changes, they projected the simulated cortical activity onto the visual field to mimic the corresponding visual pattern. The potassium wave features irregular patches of highly activated areas on the visual field, which supports the theory of CSD underlying migraine aura. These are just two examples of modelling CSD propagation, but both do not consider the 3D nature of physical space. Even though they feature the main characteristics of spreading depression and link it to the phenomenon of migraine with aura, they lack detailed microscopic and electrophysiological information that could help to understand the underlying mechanisms of migraine.

Computational models for CSD generally feature two major aspects. One is the modelling of microscopic processes, accounting for the electrical activity in the neuronal compartment causing the homeostatic failure and excessive fluctuations in the ionic concentrations. The other component describes the spread throughout the tissue, namely the diffusion leading to the propagation of the homeostatic disturbance (Zandt et al., 2015). The propagation is usually described in terms of simple expressions for diffusion, either through Brownian motion, or any other form of transport through the tissue. The variety of mathematical models accounting for the microscopic interactions, i.e., the excitation and silencing of neurons, ranges from phenomenological models to detailed metabolic models considering a range of ionic concentrations and biological indicators, like cell swelling and energy supply. These models vary in their biological plausibility and computational efficiency, and some are restricted in their capacity to account for different neuronal firing behaviours.

The Hodgkin–Huxley model was the first conductance-based model for excitable cells in computational neuroscience (Hodgkin and Huxley, 1952, 1953). It consists of four differential equations accounting for the membrane potential, the activation of Na^+ and K^+ currents and the inactivation of the Na^+ current. This model exhibits the particular biological properties of spiking neurons and is one of the fundamental models for neuronal excitation. The FitzHugh–Nagumo model, a system of two ordinary differential equations (ODEs), describes the membrane potential and one recovery variable, that acts as a negative feedback on the membrane potential (FitzHugh, 1955; Nagumo, 1962; Rogers and McCulloch, 1994). It describes the typical characteristics of spiking neurons but fails to account for tonic bursting, periods of rapid action potential spiking followed by a quiescent period, and spike frequency adaptation, a reduction of the frequency of the action potential response to a stimulus. The Izhikevich model is a simplified model for neuronal spiking with an auxiliary after-spike resetting of the two variables representing the membrane potential and the recovery variable, respectively (Izhikevich, 2003, 2010). This model exhibits firing patterns of all types of cortical neurons (Izhikevich, 2004). However, the big drawback of this model is the lack of autonomous behaviour. These phenomenological models are able to reproduce the firing behaviour of highly activated cells and also account for the following refractory period, but lack in providing detailed insight into the ionic shifts in Na^+, K^+, Cl^- and into the changes in oxygen concentration during the time course of a spreading depression. The oxygen concentration is often used as an indicator for the energy supply.

The model introduced by Cressman is an extension of the Hodgkin–Huxley formalism incorporating dynamic Na^+ and K^+ ionic concentrations (Barreto and Cressman, 2011; Cressman et al., 2009, 2011). It describes the sodium current, calcium-gated potassium, and the leak current, and it also takes into account the sodium-potassium pump current, glial current and potassium diffusion.

An extension of the previously mentioned model was proposed by Wei, Ullah, and Schiff, 2014. This model supports various neuronal behaviours varying from seizure to spreading depression, tonic firing and a steady state, whose occurrence depends on the values of the potassium and oxygen bath concentrations, respectively.

Depending on the scale of the neuronal network, the required accuracy and the scope of the biological description, each of these models could be most suited. In the following, we introduce these models in a detailed manner and discuss their relevance for the application of interest in this thesis.

2.1 Phenomenological models

2.1.1 Hodgkin–Huxley model

The first to develop a model describing the ionic mechanism underlying the action potential were Hodgkin and Huxley, 1952, 1953. They described the propagation of the electrical current along giant axons of squids and received the Nobel Price in Physiology or Medicine in 1963 for their work.

With the cell membrane acting as a capacitor, each component of the cell is treated as an electrical element, and the voltage-gated and leak channels are treated as electrical conductances. The temporal evolution of the membrane potential V can be expressed as

$$C\frac{dV}{dt} = I - I_{\text{ion}}, \tag{2.1}$$

where I is the applied current or the input stimulus, I_{ion} is the total ionic current and C is the membrane capacitance. The total ionic current is a combination of the sodium and potassium current, plus a leak current combining the dynamics of the other ions, namely

$$I_{\text{ion}} = I_{\text{Na}} + I_{\text{K}} + I_{\text{l}}. \tag{2.2}$$

These ionic currents can be written in terms of the ionic conductances as

$$\begin{aligned} I_{\text{Na}} &= g_{\text{Na}}(V - V_{\text{Na}}), \\ I_{\text{K}} &= g_{\text{K}}(V - V_{\text{K}}), \\ I_{\text{l}} &= g_{\text{l}}(V - V_{\text{l}}), \end{aligned} \tag{2.3}$$

where g_{Na}, g_{K}, g_{l} are the sodium, potassium and leak conductances and V_{Na}, V_{K}, V_{l} are the reversal potentials of the respective channels. The ionic conductances are defined in terms of suitable gating variables (denoted by n, m, h) as

$$\begin{aligned} g_{\text{Na}} &= \bar{g}_{\text{Na}} m^3 h, \\ g_{\text{K}} &= \bar{g}_{\text{K}} n^4. \end{aligned}$$

The gating variables n, m and h vary between 0 and 1 and describe the percentage of open channels (1 indicates open channels, while 0 denotes closed ones). The temporal dynamics of n, m and h is governed by suitable ODEs, naturally depending on the membrane potential V, defined as follows

$$\frac{dw}{dt} = \alpha_w(V)(1 - w) - \beta_w(V)w, \quad \text{for } w = n, m, h.$$

Note that, n, m and h are dimensionless variables, while the functions α_w, β_w (with $w = n, m, h$) depend on the membrane potential. Like in Hodgkin and Huxley, 1952, we define

$$\alpha_n(V) = 0.01\left(\frac{10-V}{\exp\left(\frac{10-V}{10}\right)-1}\right), \qquad \beta_n(V) = 0.125\exp\left(-\frac{V}{80}\right),$$

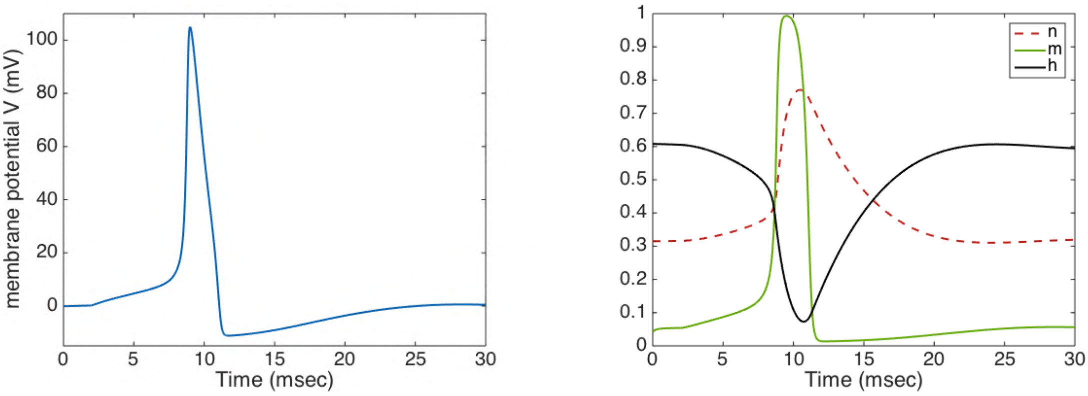
$$\alpha_m(V) = 0.1\left(\frac{25-V}{\exp\left(\frac{25-V}{10}\right)-1}\right), \qquad \beta_m(V) = 4\exp\left(-\frac{V}{18}\right),$$

$$\alpha_h(V) = 0.07\exp\left(-\frac{V}{20}\right), \qquad \beta_h(V) = \frac{1}{\exp\left(\frac{30-V}{10}\right)+1}.$$

The parameters provided by Hodgkin and Huxley correspond to a shifted membrane potential so that the resting potential is 0 mV instead of -65 mV. The choice of the remaining parameters is given in Table 2.1. As initial values we choose $V(0) = 0$, and the gating variables $n(t)$, $m(t)$ and $h(t)$ have their steady-state value when $t = 0$ (Hodgkin and Huxley, 1952). This means $n(0) = n_\infty = 0.315$, $m(0) = m_\infty = 0.042$ and $h(0) = h_\infty = 0.608$. For illustrative purposes, with $\mathbb{1}_{[a,b]}(t)$ denoting the indicator function of an interval $[a, b]$, we add a stimulus $I = 2 \cdot \mathbb{1}_{[2,8]}(t)$ μA/cm^2. The resulting spiking dynamics generated with the Hodgkin–Huxley model are shown in Figure 2.1.

Parameter	**Description**	**Value**	**Unit**
V_K	reversal potential of the potassium current	-12	mV
V_Na	reversal potential of the sodium current	120	mV
V_l	reversal potential of the leak current	10.6	mV
$\bar{g}_\mathrm{K}$	potassium conductance	36	mS/cm^2
$\bar{g}_\mathrm{Na}$	sodium conductance	120	mS/cm^2
g_l	leak conductance	0.3	mS/cm^2
C	membrane capacitance	1	μF/cm^2

TABLE 2.1: Model parameters for the simulation of the Hodgkin–Huxley model.



(a) membrane potential V

(b) gating variables n, m and h

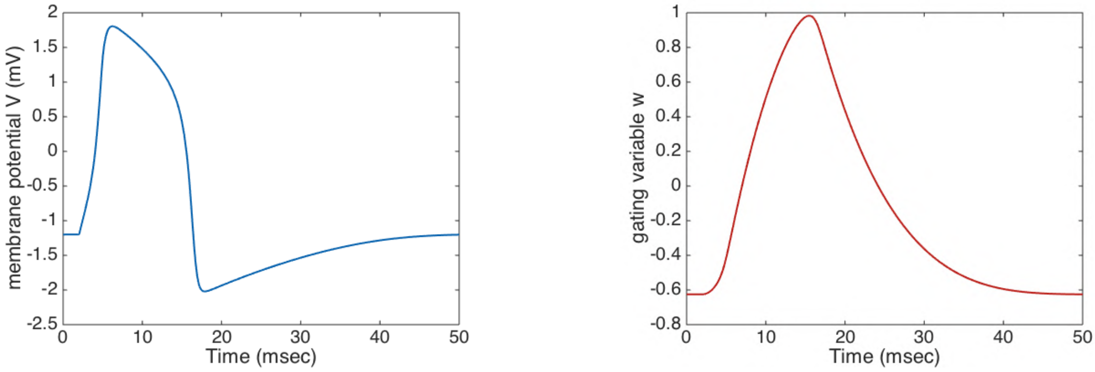
FIGURE 2.1: Neuronal spiking dynamics simulated with the Hodgkin–Huxley model.

2.1.2 FitzHugh–Nagumo model

A simplified version of the Hodgkin–Huxley model is the FitzHugh–Nagumo model (FitzHugh, 1955, 1961; Nagumo, 1962). It describes the spiking of neurons through a cubic ionic term coupling the excitation variable to only one recovery variable. The voltage-like variable exhibits regenerative self-excitation while the recovery variable acts as a slow negative feedback. This model is a typical example of a relaxation oscillator; if the external stimulus surpasses a threshold value, the system exhibits a characteristic deflection in the phase space followed by variables relaxation and return to their resting values. This model features the main qualitative characteristics of the Hodgkin–Huxley model but in a simplified form. The traditional FitzHugh–Nagumo equations (FitzHugh, 1961) for the excitation variable V and the recovery variable w read

$$\begin{aligned} \frac{dV}{dt} &= V - \frac{V^3}{3} - w + I, \\ \frac{1}{\phi}\frac{dw}{dt} &= V + a - bw, \end{aligned}$$

where I is the applied current and a, b, and ϕ are parameters. A simulation of this model response is given in Figure 2.2 with the parameters chosen as $a = 0.7$, $b = 0.8$ and $\phi = 0.08$ and the applied current $I = 0.5 \cdot \mathbb{1}_{[2,5]}(t)$.



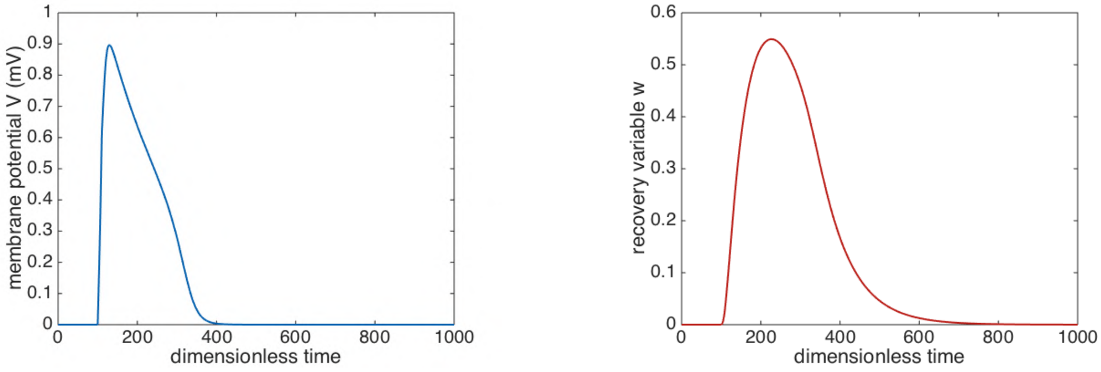
(a) membrane potential V (b) recovery variable w

FIGURE 2.2: Neuron spiking dynamics simulated with the FitzHugh–Nagumo model with parameter values chosen as $a = 0.7$, $b = 0.8$ and $\phi = 0.08$.

A modified variant of the FitzHugh–Nagumo model is the Rogers–McCulloch model that was developed in the framework of cardiac action potential propagation studies (Rogers and McCulloch, 1994). The original model reads

$$\begin{aligned} \frac{dV}{dt} &= c_1 V(V-a)(1-V) - c_2 V w + I, \\ \frac{dw}{dt} &= b\,(V - dw)\,, \end{aligned}$$

where V is the membrane potential, w the recovery variable and I the input stimulus while a, b, c_1, c_2 and d are parameters. An example of simulated spiking behaviour with the parameters $a = 0.13$, $b = 0.013$, $c_1 = 0.26$, $c_2 = 0.1$ and $d = 1$ and the applied current $I = 0.05 \cdot \mathbb{1}_{[100,110]}(t)$ is given in Figure 2.3.



(a) membrane potential V (b) recovery variable w

FIGURE 2.3: Neuronal spiking dynamics simulated with the Rogers–McCulloch variant of the FitzHugh–Nagumo model.

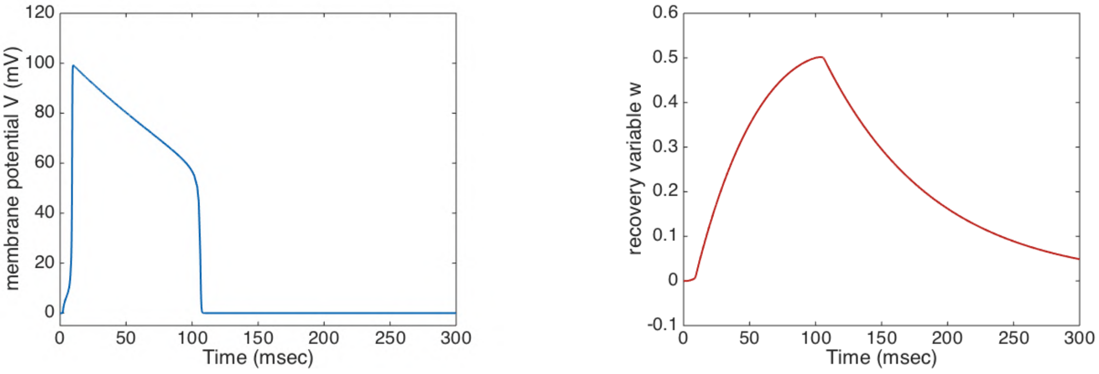
An alternative and more descriptive way to express these equations highlights the dependence of the dynamics on the initial, threshold and peak value. These values represent the baseline value, the value of excitation when an all-or-nothing response of an action potential is triggered and the maximum of the membrane potential, which indicates the beginning of the recovery phase. This system can be written as

$$\begin{aligned} \frac{dV}{dt} &= G(V - v_0)\left(1 - \frac{V}{v_{th}}\right)\left(1 - \frac{V}{v_p}\right) + \eta_1(V - v_0)w + I, \\ \frac{dw}{dt} &= \eta_2\left(V - v_0 - \eta_3 w\right), \end{aligned}$$

where V is the membrane potential, I the applied current, and w the recovery variable; G, η_1, η_2, η_3 are positive parameters and v_0, v_{th} and v_p are the initial value, the threshold parameter, and peak potential, respectively. Through the above formulation of the model, the resulting spiking dynamics can be more easily tuned to the specific characteristics and temporal behaviour of a certain type of neuronal response. A simulation of this model is shown in Figure 2.4 with the parameter choice given in Table 2.2.

Parameter	**Description**	**Value**	**Unit**
G		1.5	msec^{-1}
v_0	initial value	0	mV
v_{th}	threshold value	13	mV
v_p	peak value	100	mV
η_1		4.4	msec^{-1}
η_2		0.012	msec^{-1}
η_3		1	–

TABLE 2.2: Model parameters for the simulations of a variant of the Rogers–McCulloch model.



(a) membrane potential V (b) recovery variable w

FIGURE 2.4: Neuronal spiking dynamics simulated with a variant of the Rogers–McCulloch model.

2.1.3 Izhikevich model

Another simplified spiking model was introduced by Izhikevich (Izhikevich, 2003, 2010). This model is a system of two ODEs featuring the behaviour of a fast voltage variable and a slower recovery variable describing the activation of potassium or the inactivation of sodium (or both) and acting as a negative feedback on the fast variable. It is computationally less expensive but reproduces the spiking and bursting of cortical neurons with the help of a non-autonomous system. The membrane potential V and the recovery variable w are described by the following system of ordinary differential equations

$$C\frac{dV}{dt} = k(V - v_r)(V - v_{th}) - w + I, \tag{2.4}$$

$$\frac{dw}{dt} = a(b(V - v_r) - w), \tag{2.5}$$

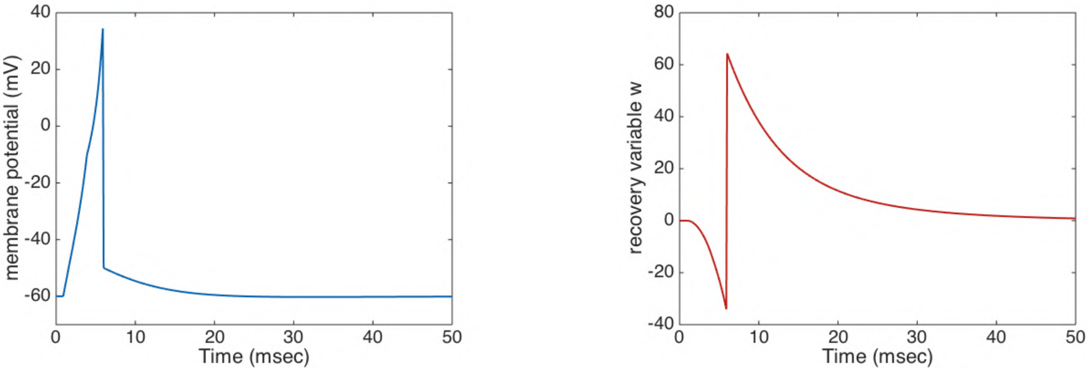
where C is the membrane capacitance, v_r is the resting membrane potential, v_{th} is the threshold, I an input stimulus and k, a and b are parameters. After exhibiting a spike ($V \geq 35$ mV) the system is reset through the following condition:

$$\text{if } V \geq v_p, \quad \text{then} \quad \begin{cases} V \leftarrow c \\ w \leftarrow w + d \end{cases}, \tag{2.6}$$

meaning that once V surpasses the peak value v_p, then V and w are reassigned and given the values c and $w + d$, respectively, where c and d are given model parameters. All the parameters can be adjusted to obtain the characteristic spiking of any particular type of neuron. The parameter a is a recovery time constant; the smaller a, the longer the recovery time. The parameter b describes the sensitivity of the recovery variable w to the behaviour of the membrane potential V below the threshold v_{th}. The after-spike reset value of the membrane potential is c, and the rest value of the recovery value is d. The parameters k and b are closely related to the neurons' spiking characteristics and input resistance. An example of the spiking behaviour simulated with the Izhikevich model (2.4) - (2.6) and parameters listed in Table 2.3 is given in Figure 2.5.

Parameter	Description	Value	Unit
C	membrane capacitance	1	pF
v_r	resting membrane potential	-60	mV
v_{th}	threshold potential	-40	mV
k	peak value	0.7	pF/(mV msec)
a		0.03	msec^{-1}
b		-2	pF/msec
c		-50	mV
d		100	pA

TABLE 2.3: Model parameters for the simulations of the Izhikevich model.



(a) membrane potential V (b) recovery variable w

FIGURE 2.5: Neuronal spiking dynamics simulated with the Izhikevich model.

2.2 Detailed electrophysiological models

2.2.1 Cressman model

The Cressman model describes the spiking behaviour of neurons in a more detailed manner and additionally includes micro-environmental factors, such as sodium and potassium dynamics as they play a vital role in the generation of action potentials and the subsequent restoration of the resting potential (Barreto and Cressman, 2011; Cressman et al., 2009, 2011). It describes the conductance of a single neuron compartment including sodium, potassium, calcium-gated potassium and leak currents in the intracellular and extracellular space. The separation of the different ions across the cell membrane and their concentrations governs the ionic current of the neuron. In addition, the sodium-potassium pump, a glial current, and potassium diffusion have an important influence on the change of ion concentration in the neuron. The total membrane potential V can be expressed as

$$C\frac{dV}{dt} = I_{\mathrm{Na}} + I_{\mathrm{K}} + I_{\mathrm{Cl}},$$

where C is the membrane capacitance, I_{Na} is the sodium current, I_{K} the potassium current, and I_{Cl} the chloride current. These ionic currents are defined in a similar manner as in the Hodgkin–Huxley model (2.1)-(2.3) but contain an additional term that accounts for the leak current of each ion involved:

$$\begin{aligned}
I_{\text{Na}} &= -g_{\text{Na}} m_\infty^3(V) h (V - V_{\text{Na}}) - g_{\text{NaL}}(V - V_{\text{Na}}), \\
I_{\text{K}} &= \left(g_{\text{K}} n^4 + \frac{g_{\text{AHP}}[\text{Ca}]_i}{1 + [\text{Ca}]_i} \right)(V - V_{\text{K}}) - g_{\text{KL}}(V - V_{\text{K}}), \\
I_{\text{Cl}} &= -g_{\text{ClL}}(V - V_{\text{Cl}}).
\end{aligned}$$

Here, the parameters g_{Na}, g_{K}, and g_{AHP} describe the conductance of the sodium, of the potassium and of the after hyperpolarisation current, g_{NaL}, g_{KL} and g_{ClL} denote the conductances for the sodium, for the potassium and for the chloride leak currents. V_{Na}, V_{K} and V_{Cl} are the reversal potential of the sodium, of the potassium and of the chloride current and $[\text{Ca}]_i$ denotes the intracellular calcium concentration. The different gating variables n, m and h correspond to the activating and inactivating sodium gate and the activating potassium gate, respectively. The dynamics of the gating variables n and h is defined as

$$\frac{dw}{dt} = \varphi \left(\alpha_w(V)(1 - w) - \beta_w(V) w \right), \quad \text{for } w = n, h,$$

with φ denoting the time constant of both gating variables. The model assumes that the activating sodium gate m is fast compared to the voltage change. Thus, m is considered to reach its equilibrium instantly which implies $m = m_\infty$, where

$$m_\infty(V) = \frac{\alpha_m(V)}{\alpha_m(V) + \beta_m(V)}. \tag{2.7}$$

The corresponding rate equations are

$$\begin{aligned}
\alpha_n(V) &= 0.001 \frac{V + 34}{1 - \exp\left(0.1(V + 34)\right)}, & \beta_n(V) &= 0.125 \exp\left(-\frac{V + 44}{80} \right), \\
\alpha_m(V) &= \frac{0.1(V + 30)}{1 - \exp\left(-0.1(V + 30)\right)}, & \beta_m(V) &= 4 \exp\left(-\frac{V + 55}{18} \right), \\
\alpha_h(V) &= 0.07 \exp\left(-\frac{V + 44}{20} \right), & \beta_h(V) &= \frac{1}{\left(1 + \exp\left(-0.1(V + 4)\right)\right)}.
\end{aligned} \tag{2.8}$$

The intracellular calcium concentration, which influences the calcium gated potassium current, is defined as

$$\frac{d[\text{Ca}]_i}{dt} = -0.002\, g_{\text{Ca}} \frac{V - V_{\text{Ca}}}{1 + \exp\left(-0.4\,(V + 25)\right)} - \frac{[\text{Ca}]_i}{80},$$

where g_{Ca} describes the calcium conductance and V_{Ca} the calcium reversal potential. The extracellular potassium concentration $[\text{K}]_o$ can be described in terms of the potassium current I_{K} across the membrane, of the ionic current of the sodium-potassium pump I_{pump}, of the uptake of potassium by astrocytes I_{glia} and of the diffusion of potassium I_{diff} in the extracellular space to nearby reservoirs. In particular, the extracellular potassium concentration is given by

$$\tau \frac{d[\text{K}]_o}{dt} = \gamma I_{\text{K}} - 2\beta I_{\text{pump}} - I_{\text{glia}} - I_{\text{diff}}, \tag{2.9}$$

with the ratio of intracellular to extracellular volume of the cell β, the time conversion factor $\tau = 1000$ and the unit conversion factor γ that converts the membrane currents into millimolar per seconds. The intracellular sodium concentration $[\text{Na}]_i$ is governed by the ionic sodium current I_{Na} and the ionic current of the sodium-potassium pump I_{pump}

as follows

$$\tau \frac{d[\mathrm{Na}]_i}{dt} = -\gamma I_{\text{Na}} - 3 I_{\text{pump}}.$$

The dynamics of the sodium-potassium pump are modelled as

$$I_{\text{pump}} = \left(\frac{\rho}{1 + \exp\left(\frac{25 - [\mathrm{Na}]_i}{3} \right)} \right) \cdot \left(\frac{1}{1 + \exp\left(5.5 + [\mathrm{K}]_o\right)} \right),$$

where ρ describes the pump strength. The extracellular clearing of the potassium ions by the astrocytes is given by

$$I_{\text{glia}} = \frac{G_{\text{glia}}}{1 + \exp\left(0.4\left(18 - [\mathrm{K}]_o\right)\right)},$$

where G_{glia} is the strength of the glial uptake of potassium. The diffusion of the potassium in the extracellular medium to distant reservoirs is modelled by

$$I_{\text{diff}} = \varepsilon([\mathrm{K}]_o - k_\infty).$$

Here, k_∞ is the steady state of the extracellular potassium concentration and ε is a diffusion constant. From Fick's law we can infer

$$\varepsilon = \frac{2D}{\Delta x^2}, \quad D = 2.5 \cdot 10^{-6}\ \mathrm{cm}^2/\mathrm{sec}, \quad \Delta x \approx 20\ \mu\mathrm{m},$$

and thus, $\varepsilon = 1.2$ Hz. Assuming that the efflux of potassium and the influx of sodium balance each other, we can approximate the intracellular potassium by

$$[\mathrm{K}]_i = 140 + 18 - [\mathrm{Na}]_i,$$

where the values 140 mM and 18 mM account for the concentration of potassium and sodium at rest, respectively. To describe the concentration of the extracellular sodium, we assume that the total amount of sodium is conserved

$$[\mathrm{Na}]_o = 144 - \beta([\mathrm{Na}]_i - 18).$$

Finally, in order to close the description of this model, the reversal potentials are derived from the ion concentrations via the Nernst equation

$$V_{\text{Na}} = 26.64 \ln\left(\frac{[\mathrm{Na}]_o}{[\mathrm{Na}]_i}\right), \quad V_{\text{K}} = 26.64 \ln\left(\frac{[\mathrm{K}]_o}{[\mathrm{K}]_i}\right), \quad V_{\text{Cl}} = -26.64 \ln\left(\frac{[\mathrm{Cl}]_i}{[\mathrm{Cl}]_o}\right),$$

where the intracellular and extracellular concentration of chloride ions at $[\mathrm{Cl}]_i = 6$ mM and $[\mathrm{Cl}]_o = 130$ mM are fixed. A simulation of the model with the choice of parameters and initial conditions given in Table 2.4 and Table 2.5 is shown in Figure 2.6. The steady-state of the extracellular potassium concentration in a nearby reservoir for a neuron at rest is $k_\infty = 4$ mM, but in order to generate active spiking behaviour we set $k_\infty = 12$ mM. As this model is autonomous, there is no need for an applied stimulus to trigger excitation.

Parameter	Description	Value	Unit
C	membrane capacitance	1	μF/cm^2
g_{Na}	conductance of the sodium current	100	mS/cm^2
g_{K}	conductance of the potassium current	40	mS/cm^2
g_{AHP}	conductance of the after hyperpolarisation current	0.01	mS/cm^2
g_{KL}	conductance of the sodium leak current	0.05	mS/cm^2
g_{NaL}	conductance of the potassium leak current	0.0175	mS/cm^2
g_{ClL}	conductance of the chloride leak current	0.05	mS/cm^2
g_{Ca}	conductance of the calcium current	0.1	mS/cm^2
V_{Ca}	reversal potential of calcium	120	mV
φ	time constant of the gating variables	30	msec^{-1}
β	ratio of intracellular to extracellular volume of the cell	7	–
γ	unit conversion factor	0.33	mM cm^2/μC
τ	time conversion factor	1000	–
ρ	pump strength	1.25	mM/sec
ε	diffusion constant	1.2	Hz
G_{glia}	strength of the glial uptake of potassium	66	mM/s
k_∞	modified steady state of the extracellular potassium concentration	12	mM
$[\text{Cl}]_i$	intracellular chloride concentration	6	mM
$[\text{Cl}]_o$	extracellular chloride concentration	130	mM

TABLE 2.4: Model parameters for the simulation of the Cressman model.

Parameter	Description	Value	Unit
V	membrane potential	-68.2102	mV
n	gating variable	20.1432	–
h	gating variable	0.9931	–
$[\text{Ca}]_i$	intracellular calcium concentration	0.0008	mM
$[\text{K}]_o$	extracellular potassium concentration	3.8155	mM
$[\text{Na}]_i$	intracellular sodium concentration	20.1432	mM

TABLE 2.5: The initial conditions for the simulation of the Cressman model.

2.2.2 Wei model

The model introduced by Wei, Ullah, and Schiff, 2014 is an extension of the neural model proposed in Cressman et al., 2009 and accounts for a wider range of neuronal activities than the original one. Wei's model supports various neuronal behaviours varying from seizure to spreading depression, tonic firing and a steady state, whose occurrence depend on the values of the potassium and oxygen bath concentrations, denoted by k_{bath} and O_{bath}, respectively. A detailed analysis of the bifurcation parameters can be found in Wei, Ullah, and Schiff, 2014.

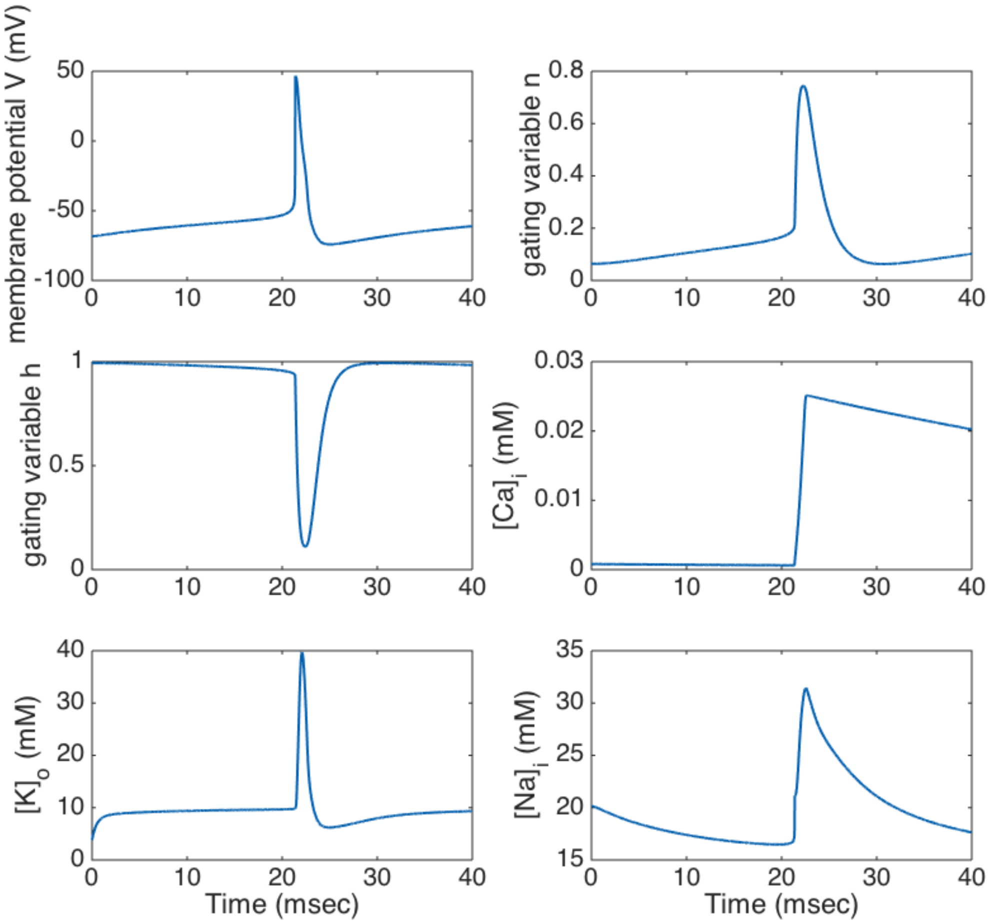


FIGURE 2.6: Neuronal spiking dynamics simulated with the electrophysiology model including sodium and potassium dynamics.

Membrane potential dynamics The membrane potential V is described by Hodgkin–Huxley type equations

$$\begin{aligned}
C\frac{dV}{dt} &= -I_{\mathrm{Na}} - I_{\mathrm{K}} - I_{\mathrm{Cl}} - \frac{I_{\mathrm{pump}}}{\gamma}, \\
I_{\mathrm{Na}} &= g_{\mathrm{Na}} m^3 h (V - V_{\mathrm{Na}}) + g_{\mathrm{NaL}}(V - E_{\mathrm{Na}}), \\
I_{\mathrm{K}} &= g_{\mathrm{K}} n^4 (V - V_{\mathrm{K}}) + g_{\mathrm{KL}}(V - E_{\mathrm{K}}), \\
I_{\mathrm{Cl}} &= g_{\mathrm{ClL}}(V - V_{\mathrm{Cl}}), \\
\frac{dw}{dt} &= \alpha_w(1-w) - \beta_w w, \qquad \text{for } w = n, m, h,
\end{aligned} \tag{2.10}$$

where C is the membrane capacitance, I_{Na} and I_{K} are the sodium and potassium currents, also including their respective leak currents, while I_{Cl} is the Cl^- leak current; g_{Na} and g_{K} are the maximal conductances of Na^+ and K^+, while g_{NaL}, g_{KL} and g_{ClL} are the leak Na^+, K^+ and Cl^- conductances, respectively; I_{pump} is the pump current of the neuronal ATP-dependent sodium-potassium pump, that will be described in detail in the next section. Finally, the conversion factor $\gamma = S/(Fv_i) \cdot 10^{-2}$, where S is the cell surface, v_i the intracellular volume and $F = 96.485$ C/mol the Faraday constant, transforms the current units (μA/cm^2) into concentration units (mM/s). The activation and inactivation variables $w = n, m, h$ represent the percentage of ion-selective channels in the open and closed state and vary between 0 and 1, where 0 indicates closed channels, and 1 open ones. The functions α_w and β_w for $w = n, m, h$ are once again voltage-dependent opening and closing rates of the ion channels and for this model are defined as

$$\begin{aligned} \alpha_n &= 0.032\,\tfrac{V+52}{1-\exp\left(-\frac{V+52}{5}\right)}, & \beta_n &= 0.5\,\exp\left(-\frac{V+57}{40}\right), \\ \alpha_m &= 0.32\,\tfrac{V+54}{1-\exp\left(-\frac{V+54}{4}\right)}, & \beta_m &= 0.28\,\frac{V+27}{\exp\left(\frac{V+27}{5}\right)-1}, \\ \alpha_h &= 0.128\,\exp\left(-\tfrac{V+50}{18}\right), & \beta_h &= \frac{4}{1+\exp\left(-\frac{V+27}{5}\right)}. \end{aligned}$$

The reversal potentials of Na^+, K^+ and Cl^- are linked to the ion concentrations via the Nernst equation

$$V_{\mathrm{Na}} = 26.64 \ln\left(\frac{[\mathrm{Na}]_o}{[\mathrm{Na}]_i}\right) \quad V_{\mathrm{K}} = 26.64 \ln\left(\frac{[\mathrm{K}]_o}{[\mathrm{K}]_i}\right) \quad V_{\mathrm{Cl}} = 26.64 \ln\left(\frac{[\mathrm{Cl}]_i}{[\mathrm{Cl}]_o}\right)$$

where $[\cdot]_i$ and $[\cdot]_o$ denote intracellular and extracellular concentrations, respectively.

Ion concentrations The dynamics of the ion concentrations is governed by the combined action of ionic currents and fluxes. The concentrations involved are computed through the number of ions per volume component, taking into account the possible swelling of the cell during activity. The ion numbers are defined as

$$N_{\mathrm{X}_i} = v_i[\mathrm{X}]_i, \qquad N_{\mathrm{X}_o} = v_o[\mathrm{X}]_o, \qquad \text{for } X \in \{\mathrm{Na}, \mathrm{K}, \mathrm{Cl}\},$$

with v_i and v_o being the intracellular and extracellular volumes, respectively. In the intracellular space, the ion number dynamics read

$$\begin{aligned} \frac{d}{dt} N_{\mathrm{K}_i} &= \frac{1}{\sigma}(-\gamma I_{\mathrm{K}} + 2I_{\mathrm{pump}} - I_{\mathrm{kcc2}} - I_{\mathrm{nkcc1}})v_i, \\ \frac{d}{dt} N_{\mathrm{Na}_i} &= \frac{1}{\sigma}(-\gamma I_{\mathrm{Na}} - 3I_{\mathrm{pump}} - I_{\mathrm{nkcc1}})v_i, \\ \frac{d}{dt} N_{\mathrm{Cl}_i} &= \frac{1}{\sigma}(\gamma I_{\mathrm{ClL}} - I_{\mathrm{kcc2}} - 2I_{\mathrm{nkcc1}})v_i, \end{aligned} \tag{2.11}$$

while in the extracellular space they read

$$\begin{aligned} \frac{d}{dt} N_{\mathrm{K}_o} &= \frac{1}{\sigma}(\gamma\beta I_{\mathrm{K}} - 2\beta I_{\mathrm{pump}} - I_{\mathrm{diff}} - I_{\mathrm{glia}} - 2I_{\mathrm{gliapump}} + \beta I_{\mathrm{kcc2}} + \beta I_{\mathrm{nkcc1}})v_o, \\ \frac{d}{dt} N_{\mathrm{Na}_o} &= \frac{1}{\sigma}(\gamma\beta I_{\mathrm{Na}} + 3\beta I_{\mathrm{pump}} + \beta I_{\mathrm{nkcc1}})v_o, \\ \frac{d}{dt} N_{\mathrm{Cl}_o} &= \frac{1}{\sigma}(-\gamma\beta I_{\mathrm{ClL}} + \beta I_{\mathrm{kcc2}} + 2\beta I_{\mathrm{nkcc1}})v_o. \end{aligned} \tag{2.12}$$

In the above equations, $\sigma = 1000$ is the time conversion factor from seconds to milliseconds, and $\beta = v_i/v_o$ is the ratio between intracellular and extracellular volume. By defining

$$N_i(t) = \begin{bmatrix} N_{\mathrm{K}_i}(t) \\ N_{\mathrm{Na}_i}(t) \\ N_{\mathrm{Cl}_i}(t) \end{bmatrix}, \qquad N_o(t) = \begin{bmatrix} N_{\mathrm{K}_o}(t) \\ N_{\mathrm{Na}_o}(t) \\ N_{\mathrm{Cl}_o}(t) \end{bmatrix},$$

the vector C_i and C_o of intracellular and extracellular ion concentrations can be written as

$$C_i(t) = \frac{N_i(t)}{v_i(t)}, \qquad C_o(t) = \frac{N_o(t)}{v_o(t)}.$$

In equations (2.11) and (2.12), I_{pump} is the neuronal sodium-potassium pump current, I_{diff} is the K^+ diffusion current, I_{glia} is the current associated with the glial buffering, I_{gliapump} is the glial sodium-potassium pump current. Their formulations are adapted from the ones in Cressman et al., 2009 and extended by an oxygen dependency

$$\begin{aligned}
I_{\text{pump}} &= \frac{\rho([\text{O}_2]_o)}{1+\exp((25-[\text{Na}]_i)/3)} \cdot \frac{1}{1+\exp(3.5-[\text{K}]_o)}, \\
I_{\text{gliapump}} &= \frac{1}{3} \cdot \frac{\rho([\text{O}_2]_o)}{1+\exp((25-[\text{Na}^+]_{\text{gi}})/3)} \cdot \frac{1}{1+\exp(3.5-[\text{K}]_o)}, \\
I_{\text{glia}} &= \frac{G_{\text{glia}}(\text{O}_{\text{bath}})}{1+\exp((18-[\text{K}]_o)/2.5)}, \\
I_{\text{diff}} &= \varepsilon_k(\beta, \text{O}_{\text{bath}})\,([\text{K}]_o - k_{\text{bath}}).
\end{aligned} \tag{2.13}$$

In the above expressions, k_{bath} represents the bath potassium concentration and O_{bath} the bath oxygen concentration. The intracellular sodium concentration in the glia compartment $[\text{Na}]_{\text{gi}}$ is assumed to be constant. The cost of pumps and currents in (2.13) can be used to estimate the oxygen, as the neural activity consumes most of the energy in the brain. Additionally, the diffusion of K^+ to the blood is constrained by the instantaneous volume fraction β. The dependence of the sodium-potassium pump rate $\rho([\text{O}_2]_o)$, the K^+ uptake of the glial cells $G_{\text{glia}}(\text{O}_{\text{bath}})$, and potassium diffusion coefficient $\varepsilon_k(\beta, \text{O}_{\text{bath}})$ of the available oxygen in the extracellular space are in turn defined as

$$\begin{aligned}
\rho([\text{O}_2]_o) &= \frac{\rho_{\max}}{1+\exp((20-[\text{O}_2]_o)/3)}, \\
G_{\text{glia}}(\text{O}_{\text{bath}}) &= \frac{G_{\text{glia,max}}}{1+\exp((2.5-\text{O}_{\text{bath}})/0.2)}, \\
\varepsilon_k(\beta, \text{O}_{\text{bath}}) &= \frac{1}{1+\exp((-20+\beta)/2)} \cdot \frac{\varepsilon_{k,max}}{1+\exp(2.5-\text{O}_{\text{bath}})/0.2)}.
\end{aligned} \tag{2.14}$$

The extracellular oxygen concentration $[\text{O}_2]_o$ is assumed to be supplied diffusively by the bath solution O_{bath}, and its dynamics is modelled as

$$\frac{d[\text{O}_2]_o}{dt} = \frac{1}{\sigma}\left(-\alpha(I_{\text{pump}} + I_{\text{gliapump}}) + \varepsilon_0(\text{O}_{\text{bath}} - [\text{O}_2]_o)\right),$$

where ε_0 is the diffusion constant and α is a conversion factor from the pump current units (mM/sec) to the units of the oxygen concentration variation ($\text{mg}\,\text{L}^{-1}\text{sec}^{-1}$).

The chloride homeostasis is mainly regulated by cation-chloride co-transporters, in particular by the $\text{Na}^+/\text{K}^+/2\text{Cl}^-$ (NKCC1) and the K^+/Cl^- (KCC2) co-transporters (see Wei, Ullah, and Schiff, 2014 for detailed physiological description). The co-transporter currents I_{nkcc1} and I_{kcc2} are modelled in a Nernst-like manner as

$$\begin{aligned} I_{\text{kcc2}} &= U_{\text{kcc2}} \ln\left(\frac{[\mathrm{K}]_i[\mathrm{Cl}]_i}{[\mathrm{K}]_o[\mathrm{Cl}]_o}\right), \\ I_{\text{nkcc1}} &= U_{\text{nkcc1}} f\left([\mathrm{K}]_o\right)\left(\ln\left(\frac{[\mathrm{K}]_i[\mathrm{Cl}]_i}{[\mathrm{K}]_o[\mathrm{Cl}]_o}\right) + \ln\left(\frac{[\mathrm{Na}]_i[\mathrm{Cl}]_i}{[\mathrm{Na}]_o[\mathrm{Cl}]_o}\right)\right), \\ f([\mathrm{K}]_o) &= \frac{1}{1+\exp(16-[\mathrm{K}]_o)}, \end{aligned}$$

where U_{kcc2} and U_{nkcc1} are the co-transporter strengths.

Volume dynamics As the cell volume varies according to the level of activity (entailing high ion influx or efflux), the expected intracellular neuronal volume $\hat{v}_i$ depends on the difference between the extracellular and the intracellular osmotic pressures π_o and π_i. Both the extracellular and intracellular fluids consist of water, sodium, potassium and chloride ions. Negatively charged proteins (A^-), also called anions and mainly present in the intracellular space, cannot permeate the membrane and are one of the reasons for the overall negative charge of the membrane potential at rest. The intracellular and extracellular osmotic pressures can thus be written as

$$\begin{aligned} \pi_o &= [\mathrm{Na}]_o + [\mathrm{K}]_o + [\mathrm{Cl}]_o + [\mathrm{A}]_o, \\ \pi_i &= [\mathrm{Na}]_i + [\mathrm{K}]_i + [\mathrm{Cl}]_i + [\mathrm{A}]_i, \end{aligned}$$

where $[\mathrm{A}]_i = 132$ mM and $[\mathrm{A}]_o = 18$ mM are the intra- and extracellular concentration of anions.

The expected intracellular volume is given by

$$\hat{v}_i = v_i^0 \cdot \left(1.1029 - 0.1029 \exp\left(\frac{\pi_o - \pi_i}{20}\right)\right), \tag{2.15}$$

where v_i^0 is the initial intracellular volume of a cell with radius 7 μm, whereas the change of cell volume is defined as

$$\frac{dv_i}{dt} = \frac{\hat{v}_i - v_i}{250}. \tag{2.16}$$

Assuming a constant total volume, the extracellular volume is then computed as

$$v_o = \left(1 + \frac{1}{\beta_0}\right) v_i^0 - v_i, \tag{2.17}$$

where β_0 is the initial ratio of intra- and extracellular volume. The instantaneous volume ratio $\beta = v_i/v_0$ is updated according to the temporal variations of v_i and v_0.

Vectorial representation To simplify the notation, we introduce a compact vector representation of the Wei model, and define

$$N(t) = \begin{bmatrix} N_i(t) \\ N_o(t) \end{bmatrix} \in \mathbb{R}^6, \qquad w(t) = [\, n(t),\ m(t),\ h(t)\,]^T \in \mathbb{R}^3.$$

Furthermore, by setting

$$\Lambda(t) = \begin{bmatrix} v_i(t)\,\mathbb{I}_3 & 0 \\ 0 & v_o(t)\,\mathbb{I}_3 \end{bmatrix} \in \mathbb{R}^{6\times 6},$$

where $\mathbb{I}_3 \in \mathbb{R}^{3\times 3}$ is the identity matrix, we have

$$C(t) = \begin{bmatrix} C_i(t) \\ C_o(t) \end{bmatrix} = \Lambda(t)^{-1} N(t) \in \mathbb{R}^6.$$

By introducing the vectors

$$U(t) = \begin{bmatrix} V(t) \\ w(t) \\ N(t) \\ [\mathrm{O}_2]_\mathrm{o}(t) \\ v_i(t) \end{bmatrix} \in \mathbb{R}^{12}, \qquad W(t) = \begin{bmatrix} C(t) \\ v_o(t) \end{bmatrix} \in \mathbb{R}^7, \tag{2.18}$$

and by defining

$$\Phi_U(U,W) = \begin{bmatrix} \Phi_V \\ \Phi_w \\ \frac{1}{\sigma}\Lambda\,\Phi_N \\ \frac{1}{\sigma}\Phi_{O_2} \\ \Phi_{v_i} \end{bmatrix} \in \mathbb{R}^{12}, \quad \Phi_W(U) = \begin{bmatrix} \Lambda^{-1}N \\ \left(1+\dfrac{1}{\beta_0}\right) v_i^0 - v_i \end{bmatrix} \in \mathbb{R}^7,$$

with

$$\Phi_V = -I_\mathrm{Na} - I_\mathrm{K} - I_\mathrm{Cl} - \frac{I_\mathrm{pump}}{\gamma}, \qquad \Phi_w = \begin{bmatrix} \alpha_n(1-n) - \beta_n n \\ \alpha_m(1-m) - \beta_m m \\ \alpha_h(1-h) - \beta_h h \end{bmatrix},$$

$$\Phi_N = \begin{bmatrix} -\gamma I_\mathrm{K} + 2I_\mathrm{pump} - I_\mathrm{kcc2} - I_\mathrm{nkcc1} \\ -\gamma I_\mathrm{Na} - 3I_\mathrm{pump} - I_\mathrm{nkcc1} \\ \gamma I_\mathrm{ClL} - I_\mathrm{kcc2} - 2I_\mathrm{nkcc1} \\ \gamma\beta I_\mathrm{K} - 2\beta I_\mathrm{pump} - I_\mathrm{diff} - I_\mathrm{glia} - 2I_\mathrm{gliapump} + \beta I_\mathrm{kcc2} + \beta I_\mathrm{nkcc1} \\ \gamma\beta I_\mathrm{Na} + 3\beta I_\mathrm{pump} + \beta I_\mathrm{nkcc1} \\ -\gamma\beta I_\mathrm{ClL} + \beta I_\mathrm{kcc2} + 2\beta I_\mathrm{nkcc1} \end{bmatrix},$$

$$\Phi_{\mathrm{O}_2} = -\alpha(I_\mathrm{pump} + I_\mathrm{gliapump}) + \varepsilon_0([\mathrm{O}_2]_\mathrm{bath} - [\mathrm{O}_2]_o), \qquad \Phi_{v_i} = \frac{\hat{v}_i - v_i}{250},$$

we can represent Wei's differential-algebraic system in compact form as

$$\begin{cases} \dfrac{dU}{dt} & = & \Phi_U(U,W) \\ W & = & \Phi_W(U) \end{cases}. \tag{2.19}$$

A simulation of the model is given in Figure 2.7, with initial values given in Table 2.6 and the parameters set as in Table 2.7.

Keeping all the other model parameters fixed, according to the bifurcation diagram in Wei, Ullah, and Schiff, 2014, a resting state featuring a spiking frequency around 10-12 Hz is obtained for the parameter choices $k_\mathrm{bath} = 5.5$ mM and $\mathrm{O}_\mathrm{bath} = 30$ mM, while a spreading depression behaviour is triggered by a higher k_bath concentration, in the

Parameter	Description	Value	Unit
C	membrane capacity of the lipid bilayer	1	μF/cm^2
g_{Na}	maximal conductance of sodium current	30	mS/cm^2
g_{K}	maximal conductance of potassium current	25	mS/cm^2
g_{NaL}	conductance of leak sodium current	0.0247	mS/cm^2
g_{KL}	conductance of leak potassium current	0.05	mS/cm^2
g_{ClL}	conductance of leak chloride current	0.1	mS/cm^2
β_0	initial ratio between intra- and extracellular volume	7	–
ρ_{max}	maximal Na$^+$/K$^+$ pump rate	0.8	mM/s
$\varepsilon_{k,max}$	maximal potassium diffusion rate	0.25	s^{-1}
$G_{\text{glia,max}}$	maximal glial uptake strength of potassium	5	mM/s
$[\text{K}]_{\text{bath}}$	normal bath potassium concentration	3.5	mM
ε_0	oxygen diffusion rate	0.17	sec^{-1}
$[\text{Na}]_{\text{gi}}$	glial intracellular sodium concentration	18	mM
α	conversion factor	5.3	g/mol
$[\text{O}_2]_{\text{bath}}$	normal bath oxygen concentration	32	mg/L
U_{kcc2}	maximal KCC2 co-transporter strength	0.3	mM/sec
U_{nkcc1}	maximal NKCC1 co-transporter strength	0.1	mM/sec
σ	conversion from seconds to milliseconds	1000	sec

TABLE 2.6: Model parameters for the simulation of the Wei model.

Parameter	Description	Value	Unit
V	membrane potential	-74.30	mV
v_i	intracellular volume	$1.4368 \cdot 10^{-15}$	mm^3
v_o	extracellular volume	v_i/β_0	mm^3
β_0	intra-/extracellular volume ratio	7	–
$[\text{K}]_o$	extracellular potassium concentration	4	mM
$[\text{K}]_i$	intracellular potassium concentration	140	mM
$[\text{Na}]_o$	extracellular sodium concentration	144	mM
$[\text{Na}]_i$	intracellular sodium concentration	18	mM
$[\text{Cl}]_o$	extracellular chloride concentration	130	mM
$[\text{Cl}]_i$	intracellular chloride concentration	6	mM
$[\text{O}_2]_o$	extracellular oxygen concentration	29.3	mM
k_{bath}	high bath concentration of potassium	64	mM
m		0.0031	–
h		0.9994	–
n		0.0107	–

TABLE 2.7: Initial conditions of the Wei model.

specific $k_{\text{bath}} = 64$ mM, provided that g_{NaL} is doubled with respect to the original value. In Figure 2.8 we give an overview of all the different neuronal behaviours that can be obtained with this model, including tonic firing, periodic seizure, steady-state, wave of death and spreading depression.

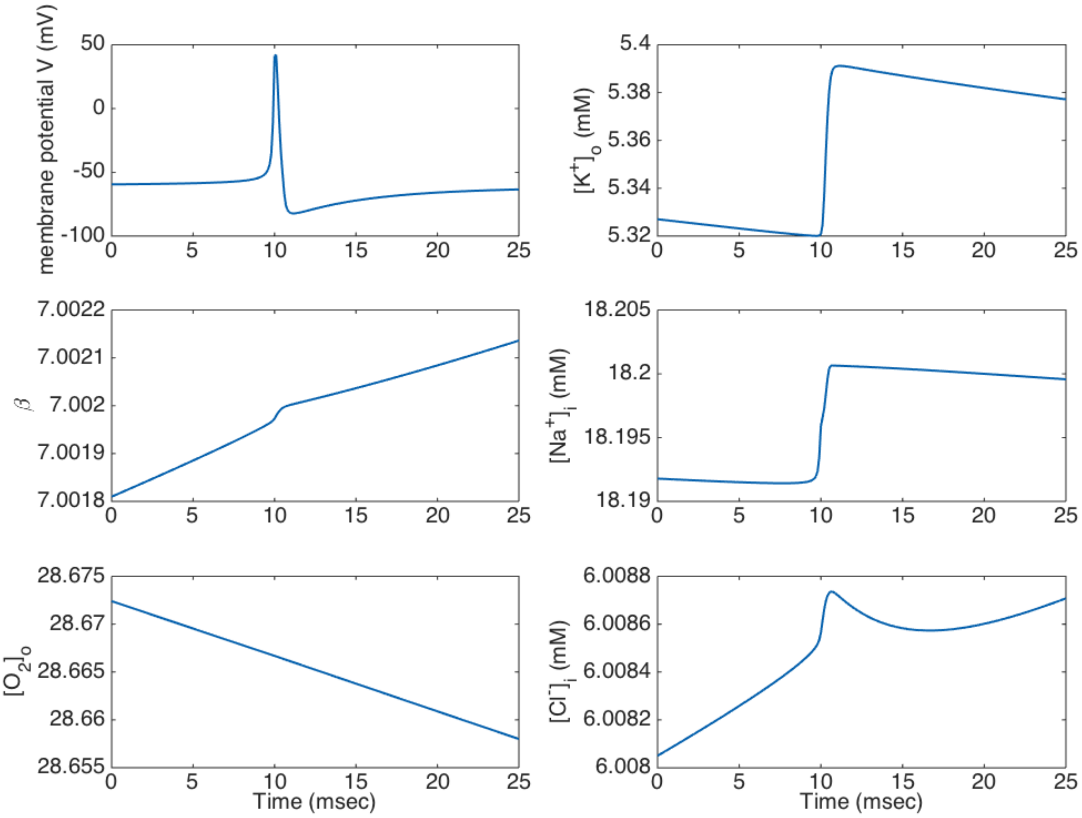


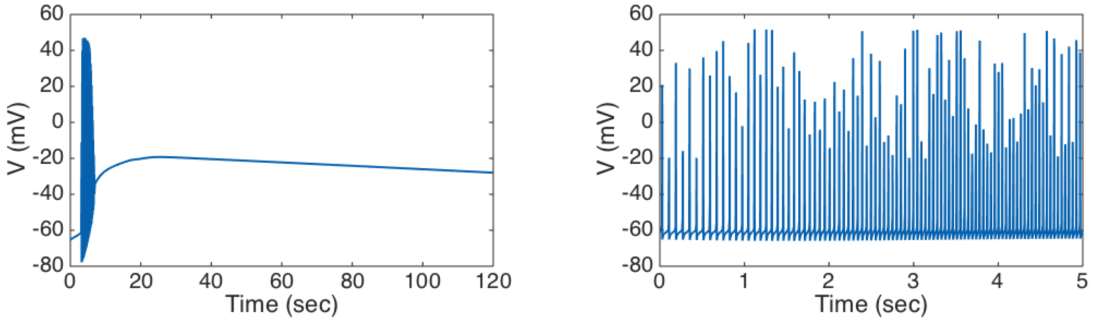
FIGURE 2.7: Neuronal spiking dynamics simulated with the Wei model.

2.3 Summary

A variety of models have been proposed in computational neuroscience to describe the biological complexity of neuronal activity. In this chapter, we presented the Hodgkin–Huxley model, the first conductance-based model for excitable media and the FitzHugh–Nagumo model that describes the membrane potential with the help of only one recovery variable. These phenomenological models were the first to mathematically describe the neuronal excitation and serve as basis and reference for more sophisticated and detailed models. Right after the FitzHugh–Nagumo model, we presented a variant of the Roger–McCulloch model that allows to adjust resting, threshold and peak value of the action potential characteristics in an effortless manner. For the sake of completeness, we shortly summarised the Izhikevich model, a simplified model for neuronal spiking with an auxiliary after-spike resetting of the membrane potential and the recovery variable. These models give a basic introduction to the phenomenological modelling of neuronal excitation.

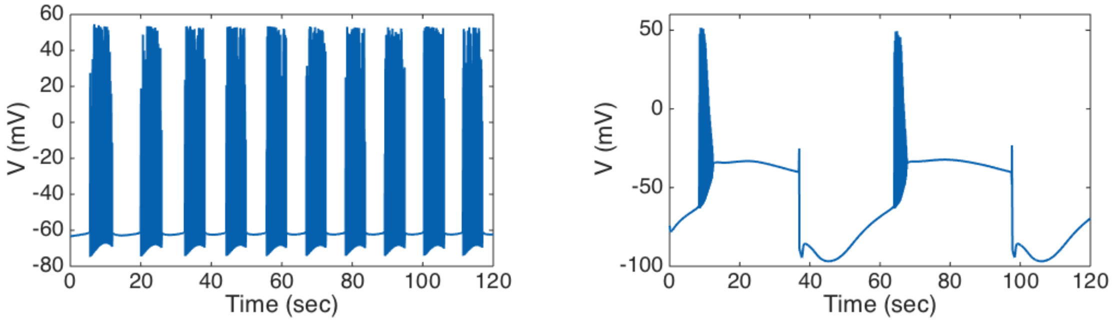
A more detailed electrophysiological model was introduced by Cressman et al., 2009 and extended by Barreto and Cressman, 2011 and subsequently by Wei, Ullah, and Schiff, 2014. Including pumps and glia currents, potassium diffusion and later oxygen dynamics and energy (oxygen) dependence of the ion exchange pumps, these models account for a wide variety of neuronal activity patterns, such as spikes, seizure and tonic firing. In particular, the model proposed by Wei, Ullah, and Schiff, 2014 allows the description of high potassium induced spreading depression.

On the basis of these models we are going to propose a detailed phenomenological approach to model CSD in the following chapter.



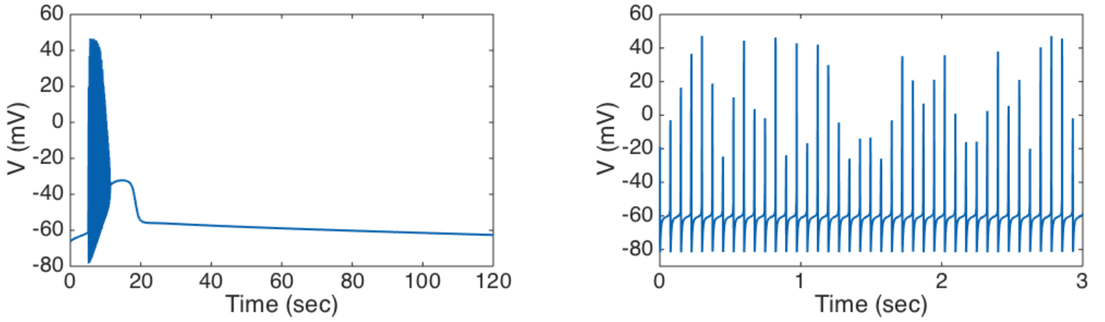
(a) wave of death, $k_{\text{bath}} = 5.5$ mM, $\text{O}_{\text{bath}} = 2$ mM

(b) tonic firing, $k_{\text{bath}} = 15$ mM, $\text{O}_{\text{bath}} = 30$ mM



(c) periodic seizure, $k_{\text{bath}} = 10$ mM, $\text{O}_{\text{bath}} = 30$ mM

(d) seizure, $k_{\text{bath}} = 25$ mM, $\text{O}_{\text{bath}} = 30$ mM



(e) mixed dynamics of seizure and spreading depression, $k_{\text{bath}} = 5.5$ mM, $\text{O}_{\text{bath}} = 5$ mM

(f) periodic singlet firing, $k_{\text{bath}} = 5.5$ mM, $\text{O}_{\text{bath}} = 30$ mM

FIGURE 2.8: The different neuronal behaviours simulated with the Wei model.

Chapter 3

Distributed CSD model

"Science, my lad, is made up of mistakes,
but they are mistakes which it is useful to make, because they lead little by little to the truth."

Jules Verne, in *A Journey to the Center of the Earth* (1873)

Biophysical models aim at providing insight into interactions and dynamics at the microscopic (cellular or subcellular) level, whereas phenomenological models aim at providing information about the general propagation of the CSD on the macroscopic scale. In this chapter, we create a first bridge between the biophysical and phenomenological modelling approach, to provide a better understanding of the complex dynamics underlying the propagation of the depolarisation wave.

Following the potassium hypothesis for CSD propagation, we introduce a multi-scale coupled PDE-ODE system accounting for the macroscopic propagation of the depolarisation wave and the microscopic neuronal dynamics. The problem is inherently multiscale in time: the propagation of the CSD wave is on the minute scale (taking about 20 minutes to propagate from the visual cortex to the frontal cortex), whereas the electrophysiological dynamics features a much faster temporal scale (of the order of milliseconds). We model the extracellular potassium bath concentration at the macroscopic level with a modified version of the FitzHugh–Nagumo model. Referring to the potassium hypothesis, the propagation of the extracellular potassium is the driving force of CSD propagation on the cerebral cortex (Zandt et al., 2015). The resulting dynamics of the potassium bath concentration is then used as a driver for the activation of the detailed electrophysiological model for the neurons. All models introduced in Chapter 2 are theoretically suited to describe the microscopic neuronal response, especially the detailed electrophysiological models introduced in Section 2.2. However, we chose to build our model upon the one proposed by Wei, Ullah, and Schiff, 2014 (see also Section 2.2.2), as it supports various neuronal activity patterns depending on the extracellular potassium and oxygen bath concentration. A constant potassium concentration around 5 mM triggers a spiking frequency of about 12 Hz which coincides with the neuronal activity at rest. Increasing the potassium concentration up to 64 mM results in CSD-like behaviour. In particular, constantly high extracellular potassium leads to periodic seizures in the microscopic neuronal model dynamics with initial high intensity firing followed by a period of neuronal silencing. We are going to take advantage of the model's specific reaction to varying potassium concentrations in order to reproduce the characteristic neuronal behaviour during a CSD.

As the available literature describes the behaviour of a single cell, the behaviour of the microscopic component of the coupled model can be considered a non-standard mean field model (following the classical definition in Deco et al., 2008). The model variables in a generic point $x \in \Omega$ represent the average value of the corresponding quantities for the collection of neurons present at that particular spatial location. In addition to that, it is advantageous to consider an ensemble of neurons in space, as it allows to use a computational grid with a characteristic size of the order of millimetres, based on the geometry obtained from MRI imaging, rather than the characteristic size of individual neurons (typically of the order of μm).

In this chapter, we first describe the microscopic electrophysiological formulation of neuronal behaviour and then the macroscopic propagative model of the potassium bath concentration for the depolarisation wave. Subsequently, the coupling and the numerical approximation are discussed. The last part of the chapter is devoted to numerical evaluations in one and two space dimensions to demonstrate the model features and its correspondence to measurements and experiments on the CSD. In Chapter 4 we numerically simulate CSD propagation on a brain geometry reconstructed from medical imaging using the model described in the following.

3.1 Electrophysiological part of the model

For the description of the microscopic level we use the model proposed by Wei, Ullah, and Schiff, 2014, introduced in Section 2.2.2. This model features various neuronal behaviours from seizure to spreading depression, tonic firing and a steady state. The model behaviour depends on the extracellular potassium (k_{bath}) and oxygen bath concentration (O_{bath}). We refer to the summary of the model in Section 2.2.2 and to Wei, Ullah, and Schiff, 2014 for a detailed bifurcation analysis. Here, we just repeat the outline and general formulation of the model and its variables in order to clarify the coupling with its propagating driver k_{bath}.

In contrast to the original model proposed by Wei, where k_{bath} is a constant, we follow the potassium hypothesis and assume that the extracellular potassium varies in time and thus acts as a driver for the CSD propagation. This implies that the potassium diffusion current I_{diff} in equation (2.13) explicitly depends on k_{bath}, namely $I_{\text{diff}}(k_{\text{bath}})$. This dependency is passed on to the functions depending on I_{diff} and the compact notation in the vectorial representations of the model (see Section 2.2.2).

For the sake of presentation, we recall here the Wei model and, as in the previous chapter, we collect the number of intra- and extracellular ions in a vector variable N and group the activation and inactivation variables n, m, h in the vector w:

$$N(t) = \begin{bmatrix} N_i(t) \\ N_o(t) \end{bmatrix} \in \mathbb{R}^6, \qquad w(t) = [\, n(t),\ m(t),\ h(t)\,]^T \in \mathbb{R}^3.$$

By setting

$$\Lambda(t) = \begin{bmatrix} v_i(t)\,\mathbb{I}_3 & 0 \\ 0 & v_o(t)\,\mathbb{I}_3 \end{bmatrix} \in \mathbb{R}^{6\times 6},$$

where $\mathbb{I}_3 \in \mathbb{R}^{3\times 3}$ is the identity matrix, the intra- and extracellular ion concentrations can be written as

$$C(t) = \begin{bmatrix} C_i(t) \\ C_o(t) \end{bmatrix} = \Lambda(t)^{-1} N(t) \in \mathbb{R}^6.$$

We then introduce the variable vector U and W as

$$U(t) = \begin{bmatrix} V(t) \\ w(t) \\ N(t) \\ [\mathrm{O_2}]_\mathrm{o}(t) \\ v_i(t) \end{bmatrix} \in \mathbb{R}^{12}, \qquad W(t) = \begin{bmatrix} C(t) \\ v_o(t) \end{bmatrix} \in \mathbb{R}^7, \tag{3.1}$$

where V denotes the membrane potential, $[\mathrm{O_2}]_\mathrm{o}$ is the extracellular oxygen concentration, v_i is the cell's volume and v_o is the extracellular volume. The Wei model depending on k_bath is a differential-algebraic system that can be expressed in compact form as

$$\begin{cases} \dfrac{dU}{dt} &= \Phi_U(U, W, k_\mathrm{bath}) \\ W &= \Phi_W(U) \end{cases}. \tag{3.2}$$

The functions Φ_U and Φ_W describe the system's dynamics including the ion and channel pump dynamics, as well as volume dynamics. Specifically, by letting

$$\Phi_U(U, W, k_\mathrm{bath}) = \begin{bmatrix} \Phi_V \\ \Phi_w \\ \frac{1}{\sigma}\Lambda\,\Phi_N(k_\mathrm{bath}) \\ \frac{1}{\sigma}\Phi_{\mathrm{O_2}} \\ \Phi_{v_i} \end{bmatrix} \in \mathbb{R}^{12}, \quad \Phi_W(U) = \begin{bmatrix} \Lambda^{-1}N \\ \left(1 + \dfrac{1}{\beta_0}\right) v_i^0 - v_i \end{bmatrix} \in \mathbb{R}^7,$$

with

$$\Phi_V = -I_\mathrm{Na} - I_\mathrm{K} - I_\mathrm{Cl} - \frac{I_\mathrm{pump}}{\gamma}, \qquad \Phi_w = \begin{bmatrix} \alpha_n(1-n) - \beta_n n \\ \alpha_m(1-m) - \beta_m m \\ \alpha_h(1-h) - \beta_h h \end{bmatrix},$$

$$\Phi_N(k_\mathrm{bath}) = \begin{bmatrix} -\gamma I_\mathrm{K} + 2I_\mathrm{pump} - I_\mathrm{kcc2} - I_\mathrm{nkcc1} \\ -\gamma I_\mathrm{Na} - 3I_\mathrm{pump} - I_\mathrm{nkcc1} \\ \gamma I_\mathrm{ClL} - I_\mathrm{kcc2} - 2I_\mathrm{nkcc1} \\ \gamma\beta I_\mathrm{K} - 2\beta I_\mathrm{pump} - I_\mathrm{diff}(k_\mathrm{bath}) - I_\mathrm{glia} - 2I_\mathrm{gliapump} + \beta I_\mathrm{kcc2} + \beta I_\mathrm{nkcc1} \\ \gamma\beta I_\mathrm{Na} + 3\beta I_\mathrm{pump} + \beta I_\mathrm{nkcc1} \\ -\gamma\beta I_\mathrm{ClL} + \beta I_\mathrm{kcc2} + 2\beta I_\mathrm{nkcc1} \end{bmatrix},$$

$$\Phi_{\mathrm{O_2}} = -\alpha(I_\mathrm{pump} + I_\mathrm{gliapump}) + \varepsilon_0([\mathrm{O_2}]_\mathrm{bath} - [\mathrm{O_2}]_o), \qquad \Phi_{v_i} = \frac{\hat{v}_i - v_i}{250}.$$

We introduce in (3.2) the dependence of Φ_U on k_bath, as the latter will be the slow dynamic variable governing the propagation of the CSD.

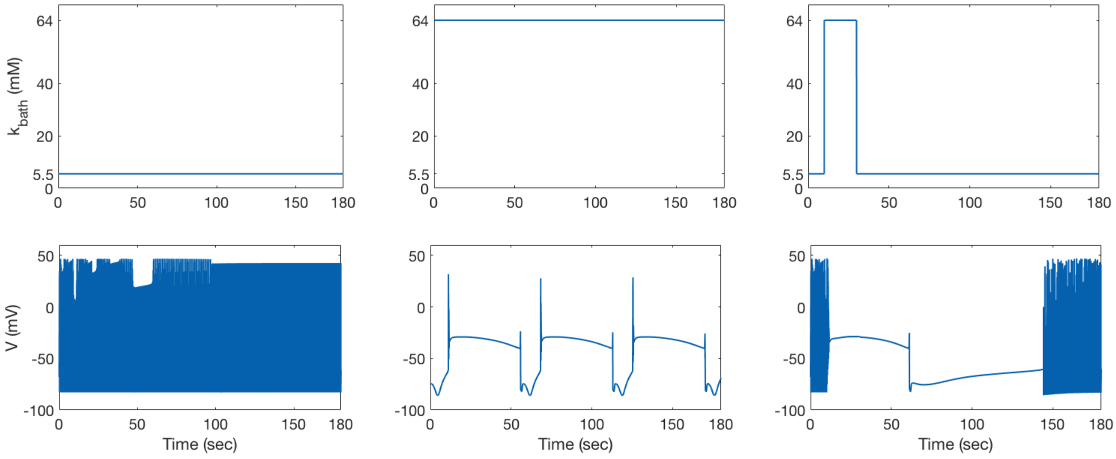
This model reproduces a variety of different neuronal dynamics, depending on the oxygen and extracellular potassium concentration (for more details see Figure 2.8 and the bifurcation analysis in Wei, Ullah, and Schiff, 2014). However, these concentrations were previously assumed to be constant and for a given choice of $(\mathrm{O}_\mathrm{bath}, k_\mathrm{bath})$ the resulting patterns are constantly repeated. In reality, during CSD, the respective neuronal firing at rest is interrupted by a short period of intense firing followed by a neuronal silencing.

These different dynamics can only be achieved by different parameter combinations of the pair $(\mathrm{O}_{\mathrm{bath}}, k_{\mathrm{bath}})$.

Consequently, to reproduce realistic CSD dynamics, the pair $(\mathrm{O}_{\mathrm{bath}}, k_{\mathrm{bath}})$ has to be a function of time. To highlight this aspect, we show in Figure 3.1 the temporal dynamics of the membrane potential obtained from Wei's model under three different conditions: at rest $k_{\mathrm{bath}} = 5.5$ mM, high concentration $k_{\mathrm{bath}} = 64$ mM, and varying k_{bath} in time, switching between the high and low concentration state. In particular, modelling k_{bath} as a step function with a short period of high potassium

$$k_{\mathrm{bath}}(t) = \begin{cases} 64\ \mathrm{mM}, & \text{if } t \in [10, 30] \\ 5.5\ \mathrm{mM}, & \text{else} \end{cases} \tag{3.3}$$

results in a CSD-like behaviour of the membrane potential. However, the choice of a step function for an ion distribution is not a natural choice and does not represent natural dynamics. We just use the step function to illustrate that a k_{bath} varying in time triggers CSD-like reactions at the microscopic scale. In the following subsection, we introduce a more appropriate choice for the k_{bath} concentration and its propagation.



(a) $k_{\mathrm{bath}} = 5.5$ mM (b) $k_{\mathrm{bath}} = 64$ mM (c) k_{bath} varying in time

FIGURE 3.1: The different temporal dynamics of k_{bath} and the membrane potential. The top row shows the extracellular potassium concentration: $k_{\mathrm{bath}} = 5.5$ mM (a), $k_{\mathrm{bath}} = 64$ mM (b) and $k_{\mathrm{bath}}(t)$ depending on time (c), as defined in (3.3). The bottom row demonstrates the corresponding dynamics of the membrane potential V.

3.2 Propagative part of the model

In accordance with the potassium hypothesis, we model the propagation of the extracellular potassium as the driving force of CSD propagation and its electrophysiological dynamics at the microscopic scale. We describe a dynamical system for the temporal evolution and propagation of k_{bath} with a coupled reaction-diffusion PDE-ODE model as follows

$$
\begin{aligned}
\frac{\partial k_{\text{bath}}}{\partial t} &= \operatorname{div}\left(D\nabla k_{\text{bath}}\right) - F(k_{\text{bath}}, w),\\
\frac{\partial w}{\partial t} &= G(k_{\text{bath}}, w),\\
F(k_{\text{bath}}, w) &= \eta_1\left(k_{\text{bath}} - k_0\right)\left(1 - \frac{k_{\text{bath}}}{k_{th}}\right)\left(1 - \frac{k_{\text{bath}}}{k_p}\right) + \eta_2\left(k_{\text{bath}} - k_0\right)w,\\
G(k_{\text{bath}}, w) &= \eta_3\left(k_{\text{bath}} - k_0 - \eta_4 w\right),
\end{aligned}
\tag{3.4}
$$

where $D \in \mathbb{R}^{d\times d}$ for $d = 1, 2, 3$ is the diffusion tensor, w is a recovery variable, η_1, η_2, η_3, and η_4 are parameters, while k_0, k_{th}, k_p are the resting, threshold, and peak values of the k_{bath} concentration, respectively. The values of these parameters are reported in Table 3.1. The coupled PDE-ODE problem (3.4) is defined on a computational domain $\Omega \times (0, T)$, with $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$. A mathematically well-posed problem is obtained by imposing initial conditions $k^0_{\text{bath}}(x) = k_{\text{bath}}(x, 0)$ and $w^0(x) = w(x, 0)$ in Ω and suitable boundary conditions on $\partial\Omega$: homogeneous Neumann boundary conditions are imposed to model an isolated environment. When considering a brain geometry, the computational domain is a 2D surface $\Sigma \subset \mathbb{R}^3$; thus, the classical divergence and gradient operators are replaced by their tangential counterparts div_Σ and ∇_Σ and boundary conditions are not necessary as the brain geometry is a closed 2D surface.

Parameter	Description	Value	Unit
k_0	resting value of the k_{bath} concentration	4	mM
k_{th}	threshold value	11.8	mM
k_p	peak value of the k_{bath} concentration	64	mM
η_1		0.2667	sec^{-1}
η_2		0.4806	sec^{-1}
η_3		3.333e-5	sec^{-1}
η_4		60	mM

TABLE 3.1: Model parameters for the propagation of the k_{bath}.

This system is a modified version of the Rogers–McCulloch variant (Rogers and McCulloch, 1994) of the FitzHugh–Nagumo model (FitzHugh, 1961; Nagumo, 1962). The model parameters have been tuned to obtain the typical CSD characteristics from the Wei model introduced in Section 2.2.2: resting frequency at 12 Hz and a brief period of intense firing in the wake of the potassium wavefront followed by a phase of neuronal silencing and a slow recovery back to the resting state. These characteristics correspond to an extracellular potassium concentration of 2.7-5 mM at rest and a short period of high potassium concentration (30-60 mM) in the onset of the CSD wave (Zandt et al., 2015).

3.3 The coupled system

While the cortical spreading depression propagates very slowly across the cortex, the electrophysiological dynamics feature a faster temporal scale (in the order of milliseconds). As a consequence, the coupled problem is inherently multi-scale in time, which is posing a non-negligible computational challenge. Given a region $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$ the complete model for the CSD propagation in $\Omega \times [0, T]$, including the electrophysiological dynamics, is a differential-algebraic system, where the differential part consists of a coupled PDE-ODE system. With suitable initial conditions (k^0_{bath}, w^0, U^0, W^0) this system reads

$$\begin{cases} \dfrac{\partial k_{\text{bath}}}{\partial t} & = & \operatorname{div}\left(D\nabla k_{\text{bath}}\right) - F(k_{\text{bath}}, w) & \text{in } \Omega \times (0,T) \\[2ex] \dfrac{\partial w}{\partial t} & = & G(k_{\text{bath}}, w) & \text{in } \Omega \times (0,T) \\[2ex] \dfrac{1}{\sigma}\dfrac{\partial U}{\partial t} & = & \Phi_U(U, W, k_{\text{bath}}) & \text{in } \Omega \times (0,T) \\[2ex] W & = & \Phi_W(U) & \text{in } \Omega \times (0,T) \end{cases} . \tag{3.5}$$

In the above system, all variables depend on space, $F(k_{\text{bath}}, w)$ and $G(k_{\text{bath}}, w)$ are the ones defined in equation (3.4), U and W are defined in equation (3.1), while $\sigma = 1000$ is the conversion factor to the temporal scale of the electrophysiology, converting seconds to milliseconds. We obtain a well-posed problem by imposing initial conditions $k_{bath}(x, 0) = k^0_{\text{bath}}$, $w(x, 0) = w^0$, $U(x, 0) = U^0$ and $W(x, 0) = W^0$ in Ω and suitable boundary conditions on $\partial\Omega$ for the first equation in (3.5), as previously mentioned in Section 3.2.

With this coupled system we can now obtain the typical dynamics of CSD, at the macroscopic scale as well as on the microscopic scale. To emphasise this and show the microscopic dynamics of the electrophysiology resulting from the temporal dynamics of the extracellular potassium, we visualise k_{bath} and the membrane potential V in Figure 3.2 as functions of time. In this example, we consider the dynamics without the diffusion part of the potassium, that is, in one single point $x \in \Omega$. A closer look at the V dynamics highlights the key characteristics of CSD: a short period of intense firing at the onset of the CSD, followed by a silencing of the neurons and a slow recovery back to the normal spiking frequency of neurons at rest.

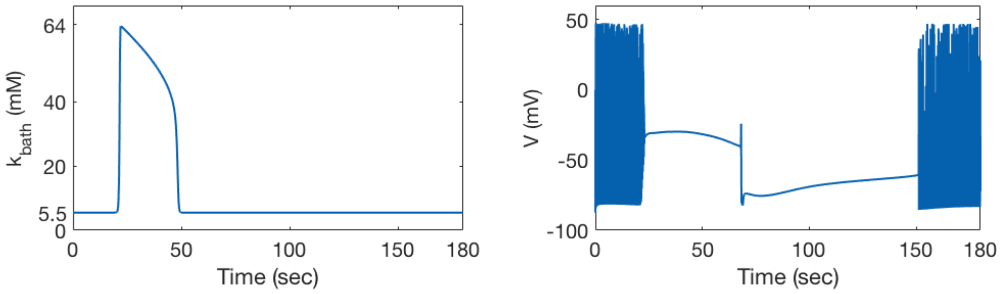
3.4 Numerical approximation

The final aim is to simulate and describe the propagation of a CSD wave on a real brain geometry without any simplifying assumptions. Brain geometries can be reconstructed from medical imaging and require the use of unstructured meshes. Consequently, we will formulate the numerical discretisation of our problem in the broadest sense possible to allow the application to real geometries.

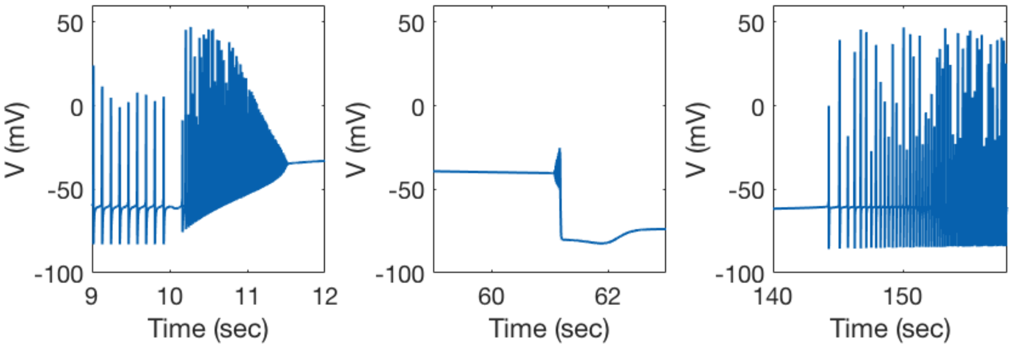
We will first formulate the variational formulation of the coupled system, then we will describe the details of the semi-discrete formulation, and finally discuss the multi scale time steps required to deal with the different time scales of the model.

3.4.1 Variational formulation

For the variational formulation of problem (3.5) let us denote by $L^2(\Omega)$ and $H^1(\Omega)$, the classical space of square integrable functions over Ω, and the Sobolev space of $L^2(\Omega)$ functions with weak first derivatives in $L^2(\Omega)$, respectively. In addition, we will use the



(a) temporal dynamics of k_{bath} and the membrane potential V



(b) detailed dynamics of the membrane potential V

FIGURE 3.2: The temporal dynamics of k_{bath} and the membrane potential V, solution to system (3.5) in (a) and three more detailed snapshots of the membrane potential dynamics to emphasise the key characteristics of the CSD (b).

Sobolev space $W^{1,1}(\Omega)$ which is a subset of functions in $L^1(\Omega)$ with weak first derivatives in $L^1(\Omega)$. Then the variational formulation of system (3.5) can be stated as follows:

given $k^0_{\mathrm{bath}}, w^0, U^0, W^0 \in L^2(\Omega)$, find $k_{\mathrm{bath}} \in W^{1,1}(0,T;H^1(\Omega))$, $w \in L^2(0,T;L^2(\Omega))$, $U \in L^2(0,T;[L^2(\Omega)]^{12})$, and $W \in L^2(0,T;[L^2(\Omega)]^7)$ such that for all $t \in (0,T)$

$$\begin{aligned}
&\frac{\partial}{\partial t}\langle k_{\mathrm{bath}}(t),\phi\rangle + a(k_{\mathrm{bath}}(t),\phi) + \langle F(k_{\mathrm{bath}}(t),w(t)),\phi\rangle = 0 && \forall\phi \in H^1(\Omega),\\
&\frac{\partial}{\partial t}\langle w(t),\psi\rangle = \langle G(k_{\mathrm{bath}}(t),w(t)),\psi\rangle && \forall\psi \in L^2(\Omega),\\
&\frac{\partial}{\partial t}\langle U(t),\Psi_U\rangle = \langle \Phi_U(U(t),W(t),k_{\mathrm{bath}}(t)),\Psi_U\rangle && \forall\Psi_U \in [L^2(\Omega)]^{12},\\
&\langle W(t),\Psi_W\rangle = \langle \Phi_W(U(t)),\Psi_W\rangle && \forall\Psi_W \in [L^2(\Omega)]^7,
\end{aligned} \tag{3.6}$$

with suitable initial conditions, as described in Section 3.2. In the above equation we denote by $\langle\cdot,\cdot\rangle$ the inner product in $L^2(\Omega)$, namely

$$\langle \eta,\zeta\rangle = \int_\Omega \eta\zeta\,dx, \qquad \forall\eta,\zeta \in L^2(\Omega)$$

and by $a(\cdot,\cdot)$ the bilinear form on $H^1(\Omega) \times H^1(\Omega)$

$$a(\varphi,\chi) = \int_\Omega (\nabla\varphi)^T D(x) \nabla\chi \, dx, \qquad \forall \varphi,\chi \in H^1(\Omega).$$

There are several theoretical results on reaction-diffusion equations that can be applied to equation (3.6). As the bilinear form $a(\cdot,\cdot)$ is weakly coercive on $H^1(\Omega) \times H^1(\Omega)$, a detailed well-posedness analysis of this problem would follow the one presented in Colli Franzone and Savare, 2002 for an analogous system.

3.4.2 Semi-discrete formulation

For the numerical approximation of the coupled system, we first start with the macroscopic model for the k_{bath} propagation. Then, in the following subsection, we add the discretisation of the microscopic part, and we conclude the section with the multi-scale numerical approximation of the coupled model.

Assuming that Ω is a triangulated surface (in $\mathbb{R}^3$), let $\mathcal{T}_h$ be a regular triangulation of Ω, namely $\Omega = \bigcup_{K \in \mathcal{T}_h} K$, where each element $K \in \mathcal{T}_h$ is the image of the reference element E, a triangle with vertices $(0,0,0)$, $(1,0,0)$, and $(0,1,0)$, through an invertible affine map T_K ($K = T_K(E)$). We define h as the maximum diameter of the elements of the triangulation. The corresponding finite element spaces X_h^k are defined as

$$X_h^k = \{\varphi_h \in C^0(\overline{\Omega}) \,|\, \varphi_{h|K} \circ T_K \in \mathbb{P}_k(E)\},$$

where $\mathbb{P}_k(E)$ is the space of polynomials of degree less than or equal to k on E. In the following, we use piecewise linear elements ($k = 1$). For a more detailed introduction to finite element methods see Quarteroni and Valli, 1994.

For the semi-discrete approximation of our problem, we apply the Galerkin method selecting the finite dimensional space $V_h = X_h^1$, the grid points $\{x_j\}_{j=1}^{N_h}$ and the finite element basis $\{\varphi_j(x) \,|\, j = 1,\dots,N_h\}$ associated to them; N_h is the dimension of V_h. The semi-discrete unknowns on the grid nodes are given by

$$\begin{aligned}
\boldsymbol{k}(t) &= \begin{bmatrix} \boldsymbol{k}_1(t) \\ \vdots \\ \boldsymbol{k}_{N_h}(t) \end{bmatrix}, & \boldsymbol{U}(t) &= \begin{bmatrix} \boldsymbol{U}_{1,1}(t) & \dots & \boldsymbol{U}_{12,1}(t) \\ \vdots & & \vdots \\ \boldsymbol{U}_{1,N_h}(t) & \dots & \boldsymbol{U}_{12,N_h}(t) \end{bmatrix}, \\
\boldsymbol{w}(t) &= \begin{bmatrix} \boldsymbol{w}_1(t) \\ \vdots \\ \boldsymbol{w}_{N_h}(t) \end{bmatrix}, & \boldsymbol{W}(t) &= \begin{bmatrix} \boldsymbol{W}_{1,1}(t) & \dots & \boldsymbol{W}_{12,1}(t) \\ \vdots & & \vdots \\ \boldsymbol{W}_{1,N_h}(t) & \dots & \boldsymbol{W}_{12,N_h}(t) \end{bmatrix},
\end{aligned}$$

and by defining $\boldsymbol{\varphi}(x) = [\varphi_1(x),\dots,\varphi_{N_h}(x)]^T$, the semi-discrete solutions are given by

$$\boldsymbol{k}_h(x,t) = \boldsymbol{k}(t)^T \boldsymbol{\varphi}(x), \qquad \boldsymbol{U}_h(x,t) = \boldsymbol{U}(t)^T \boldsymbol{\varphi}(x),$$

$$\boldsymbol{w}_h(x,t) = \boldsymbol{w}(t)^T \boldsymbol{\varphi}(x), \qquad \boldsymbol{W}_h(x,t) = \boldsymbol{W}(t)^T \boldsymbol{\varphi}(x).$$

For the sake of simplicity in notations, we simply denote by $\boldsymbol{k}$ the variable k_{bath}, from this point forward.

Following the Green formula of integration, we deduce from the variational formulation (3.6) the approximation

$$\begin{aligned}\frac{d}{dt}\sum_{i=1}^{N_h}\boldsymbol{k}_i\sum_{K\in\mathcal{T}_h}\int_K\varphi_i(x)\varphi_j(x)dx+\sum_{i=1}^{N_h}\boldsymbol{k}_i\sum_{K\in\mathcal{T}_h}\int_K(\nabla\varphi_i(x))^T D\nabla\varphi_j(x)dx\\+\sum_{i=1}^{N_h}F(\boldsymbol{k}_i,\boldsymbol{w}_i)\sum_{K\in\mathcal{T}_h}\int_K\varphi_i(x)\varphi_j(x)dx=0.\end{aligned}\tag{3.7}$$

Using the product approximation method for the nonlinear function F, namely

$$F(\boldsymbol{k}_h,\boldsymbol{w}_h)\approx\sum_{i=1}^{N_h}F(\boldsymbol{k}_i,\boldsymbol{w}_i)\varphi_i$$

simplifies the integral calculations (Langtangen, 2003).

We define the mass and stiffness matrix as $M=(m_{ij})_{i,j=1,\dots,N_h}$ and $S=(s_{ij})_{i,j=1,\dots,N_h}$ with

$$m_{ij}=\sum_{K\in\mathcal{T}_h}\int_K\varphi_i(x)\varphi_j(x)dx\quad\text{and}\quad s_{ij}=\sum_{K\in\mathcal{T}_h}\int_K(\nabla\varphi_i(x))^T D\nabla\varphi_j(x)dx.\tag{3.8}$$

For the approximation of the integrals in these matrices, we use the symmetric Gaussian quadrature rule for triangles (Dunavant, 1985; Quarteroni, Sacco, and Saleri, 2007), with seven nodes which we denote by $\mathcal{I}_7(\cdot)$ and a degree of exactness equal to three. For a given triangle K, the formula uses the vertices a_j^K, $j=1,2,3$, the midpoints of the edges $a_{(j)}^K$, $j=1,2,3$ and the centroid a^K. For a given function $f:\mathbb{R}^3\to\mathbb{R}$ this quadrature rule reads

$$\int_K f(x)\,dx\approx\mathcal{I}_7(f)=\frac{|K|}{60}\left(3\sum_{j=1}^{3}f\left(a_j^K\right)+8\sum_{j=1}^{3}f\left(a_{(j)}^K\right)+27f\left(a^K\right)\right),\quad\text{for } x\in\Omega.$$

Owing to (3.7) and the definition of the mass and stiffness matrix, the finite elements approximation of the macroscopic model (3.4) describing k_{bath} is the solution $\boldsymbol{k}$ such that

$$M\frac{d\boldsymbol{k}}{dt}+S\boldsymbol{k}-MF(\boldsymbol{k},\boldsymbol{w})=0,\tag{3.9}$$

coupled with the semi-discrete formulation of the dynamics of the recovery variable and the microscopic part of the model

$$\begin{aligned}M\frac{d\boldsymbol{w}}{dt}&=MG(\boldsymbol{k},\boldsymbol{w}),\\\frac{1}{\sigma}M_{12}\frac{d\boldsymbol{U}}{dt}&=M_{12}\boldsymbol{\Phi}_{\boldsymbol{U}}(\boldsymbol{U},\boldsymbol{W},\boldsymbol{k}),\\M_7\boldsymbol{W}&=M_7\boldsymbol{\Phi}_{\boldsymbol{W}}(\boldsymbol{U}),\end{aligned}\tag{3.10}$$

that are derived analogously. In the above equations, $F(\boldsymbol{k},\boldsymbol{w})$, $G(\boldsymbol{k},\boldsymbol{w})$, $\boldsymbol{\Phi}_{\boldsymbol{U}}(\boldsymbol{U},\boldsymbol{W},\boldsymbol{k})$ and $\boldsymbol{\Phi}_{\boldsymbol{W}}(\boldsymbol{U})$ describe the vector of the corresponding functions evaluated in the nodal values of the unknown and

$$M_n = \begin{bmatrix} M & & \\ & \ddots & \\ & & M \end{bmatrix} \in \mathbb{R}^{(nN_h)\times(nN_h)}, \quad n \in \mathbb{N}$$

denotes the block diagonal matrix containing the mass matrix $M \in \mathbb{R}^{N_h \times N_h}$.

3.4.3 Multi-scale time steps

To deal with the multi-scale nature of the coupled system, we consider a global time step for both levels but additionally solve the microscopic model on a finer time scale within the global time frame. The results are synchronised at the end of every slow (global) time step of the macroscopic model. By Δt we denote the time step at which we integrate the dynamics of the propagative wave, and by $\Delta\tau$ the time step at which we solve the electrophysiology.

The coupling of the system is unidirectional: the electrophysiology is ruled by the macroscopic part, but there is no feedback from the electrophysiological component to the propagative part of the model. Consequently, to advance one global time step from t^n to $t^{n+1} = t^n + \Delta t$, we propagate the pair (k_{bath}, w) along one time step Δt, we interpolate k_{bath} on (t^n, t^{n+1}) along the fine temporal scale $\Delta\tau$, and we solve for the electrophysiological variables (U, W) on the fine temporal scale. For the sake of simplicity, we assume a constant ratio N_τ between the temporal discretisation steps namely, we let $\Delta t = N_\tau \Delta\tau$. A time adaptive scheme in both components with a varying ratio could also be considered, but it would need some additional care in the proper choice of the synchronisation instants.

Let $t^n = n\Delta t$, for $n = 0, \dots, N_t$, be a discretisation of the time interval $(0, T)$. We denote by $\boldsymbol{k}^n$ and $\boldsymbol{w}^n$ the approximation of k_{bath} and w at time $t = t^n$, and with $\boldsymbol{U}^{n,m}$ and $\boldsymbol{W}^{n,m}$ ($m = 0, .., N_\tau$) the approximation of U and W, respectively, at time $t = t^n + m\Delta\tau$. Here, for a generic variable X, the convention is $X^n = X^{n,0} = X^{n-1,N_\tau}$. A visualisation of this scheme is given in Figure 3.3.

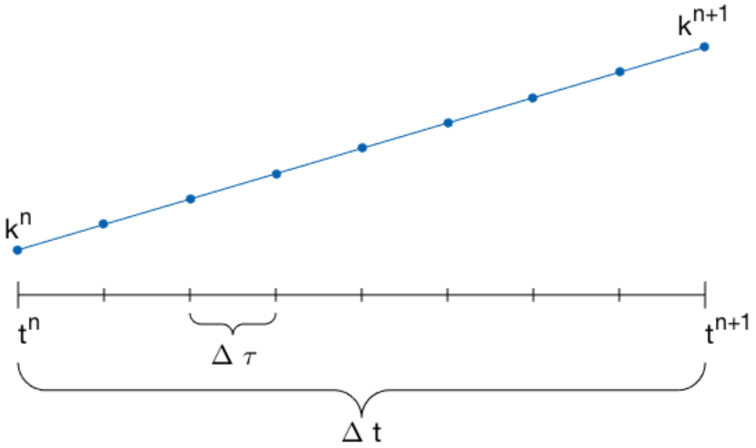


FIGURE 3.3: The multi-scale discretisation in time and the interpolation of k_{bath} on (t^n, t^{n+1}) along the fine temporal scale $\Delta\tau$.

A fully discrete approximation of the problem is then obtained by integrating system (3.9) - (3.10) in time. Let us first focus on the propagative part of the coupled model. We can integrate this part between t^n and t^{n+1} by means of a mixed implicit/explicit (IMEX) scheme: the linear diffusion term is discretised implicitly, while the nonlinear reaction term $F(\boldsymbol{k}, \boldsymbol{w})$ is treated explicitly. Because of the particular form of the problem, the ODE for the recovery variable $\boldsymbol{w}$ can be linearised around $\boldsymbol{k}$ at the previous time step, and integrated exactly on the interval (t^n, t^{n+1}), in any spatial node independently:

$$\begin{aligned}\boldsymbol{w}(t) &= \left(w^0 + \int\limits_{t_0}^{t} \eta_3(\boldsymbol{k}(t) - k^0) \exp\left(\int\limits_{t_0}^{t} \eta_3\eta_4 \mathrm{d}\xi \right) \mathrm{d}\theta \right) \exp\left(- \int\limits_{t_0}^{t} \eta_3\eta_4 \mathrm{d}\theta \right) \\ &= \frac{\boldsymbol{k} - k^0}{\eta_4} + \left(w^0 - \frac{\boldsymbol{k} - k^0}{\eta_4} \right) \exp\left(-\eta_3\eta_4(t - t_0)\right).\end{aligned}$$

This allows us to decouple the PDE-ODE system by solving it with respect to the recovery variable $\boldsymbol{w}^{n+1}$. We approximate the first-order derivative by the backward finite difference scheme

$$\frac{d\boldsymbol{k}^{n+1}}{dt} \approx \frac{\boldsymbol{k}^{n+1} - \boldsymbol{k}^n}{\Delta t}.$$

Substituting this into the respective equation in (3.9) and plugging in the updated recovery variable w^{n+1} into the nonlinear reaction term $F(\boldsymbol{k}^n, \boldsymbol{w}^{n+1})$, we are left with the problem

$$(M + \Delta t S)\boldsymbol{k}^{n+1} = M\boldsymbol{k}^n - \Delta t F(\boldsymbol{k}_h^n, \boldsymbol{w}^{n+1}). \tag{3.11}$$

The full discretisation of the macroscopic model (3.4) for every time step t^n to t^{n+1} is given in Algorithm 3.1.

Algorithm 3.1 Numerical approximation scheme for the macroscopic model

```
initialise: $\boldsymbol{k}^0 \in \mathbb{R}^{N_h}, \boldsymbol{w}^0 \in \mathbb{R}^{N_h}$
for $n = 0, \dots, N_t - 1$ do
    update: $\boldsymbol{w}^{n+1} = \frac{\boldsymbol{k}^n - k_0}{\eta_4} + \left(\boldsymbol{w}^n - \frac{\boldsymbol{k}^n - k_0}{\eta_4}\right) \exp\left(-\eta_3\eta_4\Delta t\right)$
    update: $F^{n+1} = F(\boldsymbol{k}^n, \boldsymbol{w}^{n+1})$
    solve: $(M + \Delta t S)\boldsymbol{k}^{n+1} = M\boldsymbol{k}^n - \Delta t M F^{n+1}$
end for
```

After solving system (3.9) for the time step Δt, we linearly interpolate the solution between $\boldsymbol{k}^n$ and $\boldsymbol{k}^{n+1}$ to obtain the values $\boldsymbol{k}^{n,m}$ for $m = 1, \dots, N_\tau$ as input for the electrophysiological model. Let $\boldsymbol{k}^n$ and $\boldsymbol{k}^{n+1}$ be the solution for the k_{bath} concentration at time step t^n and t^{n+1} and let $t^{n,m} = t^n + m\Delta\tau$ be the time steps at which we seek the interpolation $\boldsymbol{k}^{n,m}$. In general, for every time $t \in [t^n, t^{n+1}]$ a linear interpolant y satisfies

$$\frac{y - \boldsymbol{k}^n}{t - t^n} = \frac{\boldsymbol{k}^{n+1} - \boldsymbol{k}^n}{t^{n+1} - t^n},$$

which is equivalent to

$$y = \boldsymbol{k}^n + \frac{\boldsymbol{k}^{n+1} - \boldsymbol{k}^n}{\Delta t}(t - t^n).$$

For the interpolation at $t = t^{n,m}$ this expression can be rewritten as

$$\boldsymbol{k}^{n,m} = \left(1 - \frac{m\Delta\tau}{\Delta t}\right)\boldsymbol{k}^{n} + \frac{m\Delta\tau}{\Delta t}\boldsymbol{k}^{n+1}.$$

Approximating the time derivative of the variable $\boldsymbol{U}$ by a backward finite difference scheme, we obtain

$$\frac{d\boldsymbol{U}^{n,m+1}}{d\tau} \approx \frac{\boldsymbol{U}^{n,m+1} - \boldsymbol{U}^{n,m}}{\Delta\tau},$$

and substituting this into the respective equation in (3.10), we are left with:

$$\frac{1}{\sigma}\frac{\boldsymbol{U}^{n,m+1} - \boldsymbol{U}^{n,m}}{\Delta\tau} = \boldsymbol{\Phi}_{\boldsymbol{U}}(\boldsymbol{U}^{n,m}, \boldsymbol{W}^{n,m}, \boldsymbol{k}^{n,m+1}).$$

The full discretisation of the coupled model is given in Algorithm 3.2. The differential-algebraic system for the electrophysiology variables features stiff components and thus has to be integrated with a suitable ODE solver. For the sake of simplicity, the time integrator for the ODE system of electrophysiology in Algorithm 3.2 is a Forward Euler method for $\boldsymbol{U}^{n,m+1}$ coupled with a linearisation around the previous state of the algebraic unknowns $\boldsymbol{W}^{n,m}$, but other options (such as Backward Euler, Crank-Nicholson, Runge-Kutta, or Multistep) could be applied to this problem to improve the computational efficiency and accuracy.

Algorithm 3.2 Numerical approximation scheme for the multi-scale coupled model

```
initialise: $\boldsymbol{k}^0 \in \mathbb{R}^{N_h}$,  $\boldsymbol{w}^0 \in \mathbb{R}^{N_h}$,  $\boldsymbol{U}^{0,0} \in \mathbb{R}^{N_h \times 12}$,  $\boldsymbol{W}^{0,0} \in \mathbb{R}^{N_h \times 7}$
for $n = 0, \dots, N_t - 1$ do
    update: $\boldsymbol{w}^{n+1} = \frac{\boldsymbol{k}^n - k^0}{\eta_4} + \left(\boldsymbol{w}^n - \frac{\boldsymbol{k}^n - k^0}{\eta_4}\right)\exp\left(-\eta_3\eta_4\Delta t\right)$
    update: $F^{n+1} = F(\boldsymbol{k}^n, \boldsymbol{w}^{n+1})$
    solve: $(M + \Delta t\, S)\ \boldsymbol{k}^{n+1} = M\boldsymbol{k}^n - \Delta t M F^{n+1}$
    interpolate between $\boldsymbol{k}^n$ and $\boldsymbol{k}^{n+1}$ on $(t^n, t^{n+1})$ at $N_\tau + 1$ equally spaced nodes:
    $\boldsymbol{k}^{n,m+1} = \left(1 - \frac{m\Delta\tau}{\Delta t}\right)\boldsymbol{k}^n + \frac{m\Delta\tau}{\Delta t}\boldsymbol{k}^{n+1}$    for $m = 0, ..N_\tau$
    for $m = 0, \dots, N_\tau - 1$ do
        update: $\boldsymbol{U}^{n,m+1} = \boldsymbol{U}^{n,m} + \Delta\tau\sigma\Phi_U(\boldsymbol{U}^{n,m}, \boldsymbol{W}^{n,m}, \boldsymbol{k}^{n,m+1})$
        update: $\boldsymbol{W}^{n,m+1} = \Phi_W(\boldsymbol{U}^{n,m+1})$
    end for
end for
```

3.5 Simulation

In this section, we present the numerical simulation of a CSD wave propagation in one and two dimensional domains. With the final aim to simulate the CSD propagation, the potassium propagation and the detailed electrophysiological characteristics, on the brain cortex, we illustrate the model features in different domains increasing the dimension step by step.

The numerical simulations of the multi-scale coupled model are performed with a self-developed code in Matlab (MathWorks Inc., Natick, MA), where we choose a constant time step $\Delta t = 50$ msec for the propagation and a constant time step $\Delta\tau = 0.05$ msec for the electrophysiology, resulting in $N_\tau = 1000$ electrophysiology steps within a propagative time step. The IMEX scheme described in Section 3.4.3 requires to solve a linear system associated with a linear parabolic system, for which a wide range of literature on preconditioning is available (Saad, 2003).

Keeping in mind the final aim of applying this model to a brain geometry, we observe that a detailed discretisation of the cerebral cortex leads to a high number of grid points and thus a high dimensionality of the associated finite element matrix in the system (3.11). The matrix of this system is badly conditioned but symmetric. As of now, we solve system (3.11) with a conjugate gradient (CG) method, preconditioned by incomplete Cholesky factorisation, with pivoting and with drop tolerance set at 10^{-6}. The stopping criterion for the CG is set at a relative error of 10^{-6}, and a maximum number of iteration is set at 60. Combining the low drop tolerance in the incomplete Cholesky factorisation and the propagative nature of the solution of system (3.4) entails that, by choosing as initial guess at time t^{n+1} the solution at the previous time step $\boldsymbol{k}^n$, the preconditioned CG converges in very few iterations. The electrophysiology model is solved by an explicit Euler scheme. A detailed description of the numerical optimisation of solving system (3.11) on a brain geometry is discussed in the previous chapter.

3.5.1 Measurable variables

The dynamics of the microscopic and macroscopic part of the model takes place at different timescales: the neuronal activity takes place at a millisecond timescale, whereas the whole CSD propagation event occurs at the larger timescale of minutes. In order to be able to grasp the overall dynamics on both timescales without resorting to the fine timescale of the electrophysiology, we introduce two auxiliary variables that describe the typical characteristics of the electrophysiology but at the timescale of seconds.

The first quantity is the *maximum membrane potential* in a given point $x \in \Omega$, defined, at each second t, as

$$V_{max}(x,t) = \max_{s\in[t-1,t]} V(x,s), \tag{3.12}$$

which is also commonly measured by experimentalists (see e.g., Pietrobon, 2005; Sawat-Pokam et al., 2016).

Another common way to characterise the neuronal activity is through its *spiking frequency* (in Hz), which is naturally defined at the second timescale. We denote by $\phi(x,t)$ the spiking frequency in a given point $x \in \Omega$, during the time interval $[t-1,t]$. The procedure to obtain the spiking frequency, that is, the number of spikes in a given interval, is not straightforward and needs to be handled with care. Let $V^{n,m}$ be the membrane potential $V(x,t)$ in a generic point $x \in \Omega$ at the discrete time steps $t^{n,m} \in [t^n, t^{n+1}]$, with $t^{n,m} = t^n + m\Delta\tau$ for $m = 1, \ldots, N_\tau$. Then $[t^n, t^{n+1}]$ is a time interval of the length Δt like in the previous section. We set ϑ as a threshold value above which we consider a rise in the membrane potential as a spike. The set of indices indicating a membrane potential above the threshold is then defined as

$$j := \{m \in \{1, \ldots, N_\tau\} \,|\, V^{n,m} > \vartheta\} = \{m_1, \ldots, m_k\}.$$

We then shift this index set by inserting a zero and deleting the last entry of j

$$j_{\text{shift}} = \{0, m_1, \ldots, m_{k-1}\}$$

and take the difference between those two sets per entry, namely

$$j_{\text{diff}} = j - j_{\text{shift}} = \{m_1 - 0, m_2 - m_1, \ldots, m_k - m_{k-1}\}.$$

We then define the set

$$a = \{l \in \{1, \ldots, k\} \,|\, j_{\text{diff}}(l) > 1\},$$

where $j_{\text{diff}}(l)$ refers to the l-th entry of the set j_{diff}. With these positions, the frequency is finally defined as

$$\phi(t) = \frac{|a|}{\Delta t},$$

where $|a|$ denotes the cardinality of the set a. Note that if the first point ($m = 1$) is already above the threshold the procedure does not consider this as a spike. However, this does not have a significant impact on the results, as we are interested in distinguishing high from low frequency spiking. In this case, one spike more or less does not significantly change the results. We illustrate this procedure with the help of a short example, visualised in Figure 3.4. For this example, the set j of the indices above the threshold ϑ is

$$j := \{m \in \{1, \ldots, 16\} \,|\, V^{n,m} > \vartheta\} = \{5, 6, 9, 12, 13\},$$

then j_{shift}, j_{diff} and a follow to be

$$j_{\text{shift}} = \{0, 5, 6, 9, 12\}, \qquad j_{\text{diff}} = \{5, 1, 3, 3, 1\}, \qquad \text{and} \qquad a = \{1, 3, 4\}.$$

As the cardinality of a is three, this number gives us the amount of spikes in the considered time interval and thus a spiking frequency $\phi(t) = 3/\Delta t$ at time t.

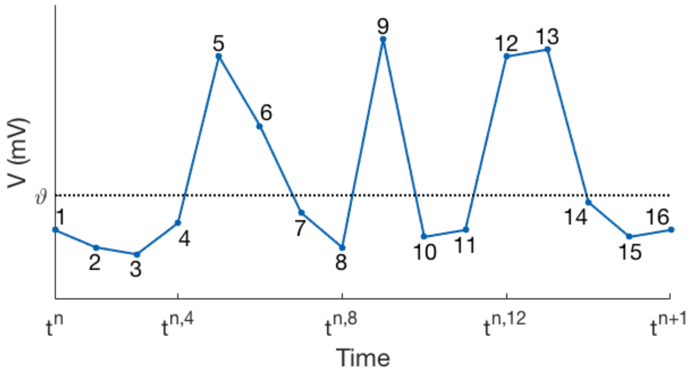
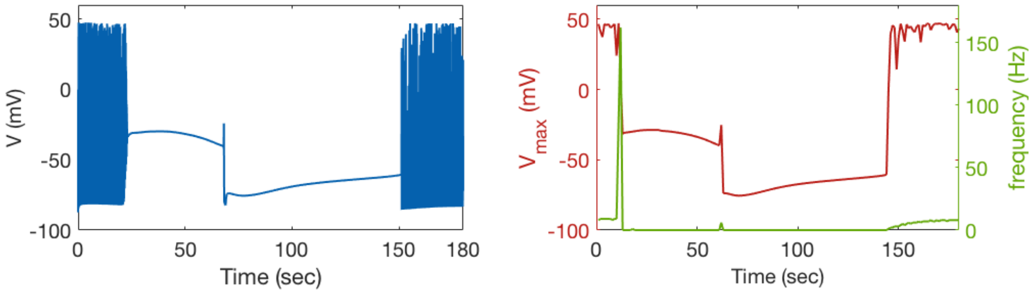


FIGURE 3.4: An illustration of the computation of the spiking frequency from the membrane potential.

In Figure 3.5 we plot the membrane potential V, computed at the millisecond scale, the comprehensive visualisation of the maximum membrane potential V_{max} and the frequency ϕ, both computed at the seconds timescale. The maximum membrane potential and the frequency simplify the membrane potential and characterise it in a comprehensive manner on the seconds scale without loss of information.



(a) membrane potential

(b) maximum membrane potential and frequency

FIGURE 3.5: The membrane potential computed at the milliseconds scale during the CSD (a), and the maximum membrane potential and the frequency (b), computed at the seconds scale.

3.5.2 CSD simulation in 1D

As an academic example, we consider a one-dimensional domain $\Omega = (0, 1)$ to demonstrate the propagative behaviour and the electrophysiological dynamics it triggers in detail. For that, we consider uniform diffusion coefficients on the whole domain, namely $D = 10^{-4}$ mm^{2}sec^{-1}, and homogenous Neumann boundary conditions modelling an isolated environment. We consider a uniform spatial mesh with a mesh size $h = 0.01$ mm and a uniform temporal mesh on $[0, 350]$ with a time step $\Delta t = 0.05$ sec for the propagative part and $\Delta\tau = 0.05$ msec for the electrophysiology. We initialise the problem by imposing

$$\boldsymbol{k}^0(x) = \begin{cases} 5.5 \text{ mM}, & \text{for } x > 0.15 \\ 64 \text{ mM}, & \text{else} \end{cases},$$

and $\boldsymbol{w}^0(x) = 0$ for all $x \in \Omega$, thus starting the potassium wave in the left extreme of the interval and simulating the propagation of the CSD towards the right. The other initial conditions are listed in Table 2.7.

In Figure 3.6 we show the propagation of the extracellular potassium in space by plotting its profile at different time steps: $t = 5$ sec, $t = 25$ sec and $t = 50$ sec. We also plot the temporal evolution of the membrane potential, and the frequency in two points $x = 0.1$ and $x = 0.9$ (identified by the red and green dot in the plot of the k_{bath} propagation). These plots show the time shift in the onset of the CSD due to the k_{bath} propagation. In Figure 3.7, we show the temporal evolution of k_{bath} and the dynamics it triggers in the electrophysiology variables at the microscopic scale. Specifically, we plot the membrane potential V, the extracellular potassium [K$^+$]$_o$, the intracellular sodium [Na$^+$]$_i$, the intracellular chloride [Cl$^-$]$_i$ and the extracellular oxygen [O$_2$]$_o$ at point $x = 0.9$.

3.5.3 CSD simulation on the unit square

In the second example, we consider a two-dimensional domain given by the unit square $\Omega = (0, 1) \times (0, 1)$, as a representation of a generalised slab of grey matter tissue of 1 mm × 1 mm. Therefore, we consider uniform diffusion coefficients for k_{bath} on the whole domain, namely $D = 5 \cdot 10^{-4}$ mm^{2}sec^{-1}, and homogenous Neumann boundary conditions to model an isolated environment. We discretise the space domain Ω with

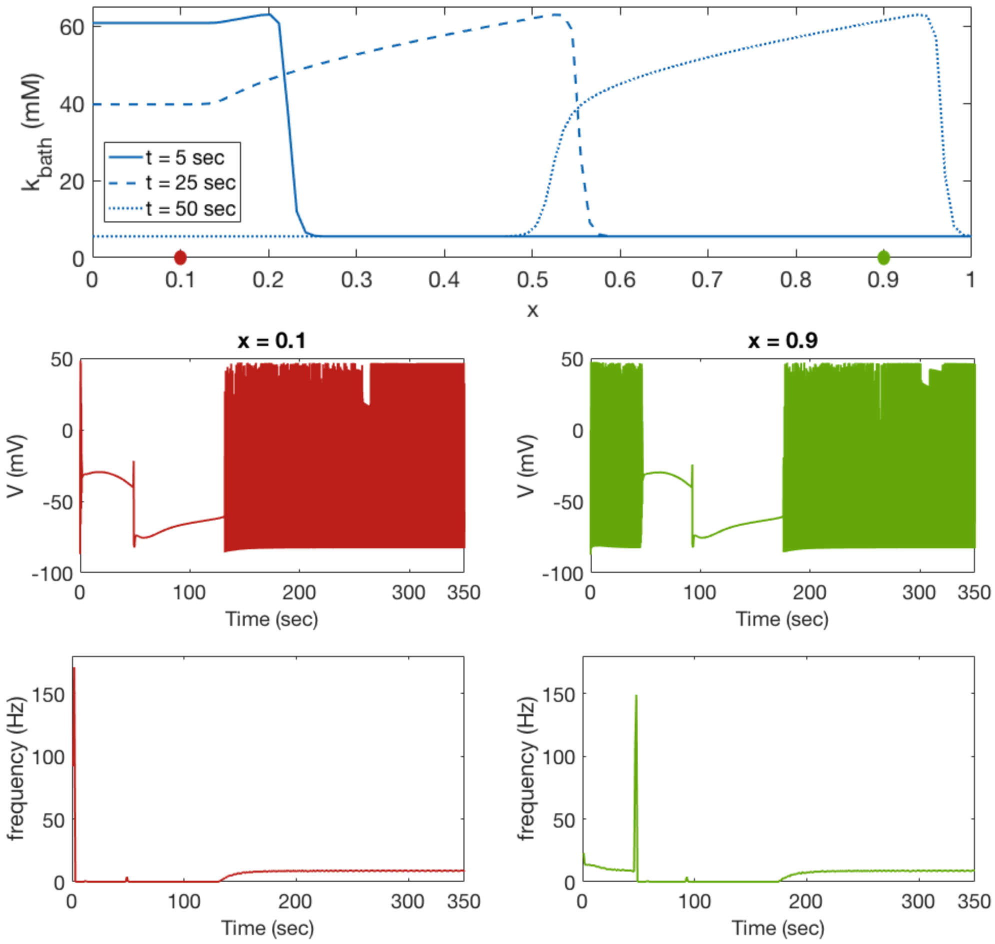


FIGURE 3.6: The extracellular potassium bath k_{bath} dynamics in space at times $t = 5$ sec, $t = 25$ sec and $t = 50$ sec and the membrane potential V, and the frequency ϕ at point $x = 0.1$ and $x = 0.9$.

an unstructured mesh with characteristic size $h = 0.01$ mm, and we choose uniform time steps for both the macroscopic part, $\Delta t = 0.05$ sec, and for the microscopic one, $\Delta\tau = 0.05$ msec. We simulate a CSD wave starting the propagation from the centre of the slab, $x \in B_r(0)$ where $B_r(0)$ is the circle around the point $(0, 0)$ with radius r. In $B_r(0)$ we initially impose $k_{\mathrm{bath}} = 64$ mM (in the rest of the domain $k_{\mathrm{bath}} = 5.5$ mM) and run the simulation for 280 seconds. The initial conditions for the other variables are given in Table 2.7.

In Figure 3.8 we visualise the maximum membrane potential V_{max} and the frequency ϕ at four different time steps, $t = 5$, $t = 50$, $t = 135$ and $t = 280$ sec. In Figure 3.9 we plot the k_{bath} distribution at $t = 5$ sec over the whole domain, and give the whole temporal dynamics on the microscopic scale in the points $(0.5, 0.5)$ and $(0.05, 0.05)$. These include the k_{bath}, maximum membrane potential V_{max}, the frequency ϕ, the membrane potential V, the extracellular potassium $[K^+]_o$, the intracellular sodium $[Na^+]_i$, the intracellular chloride $[Cl^-]_i$, the extracellular oxygen $[O_2]_o$ and the ratio between the intra- and extracellular volume β.

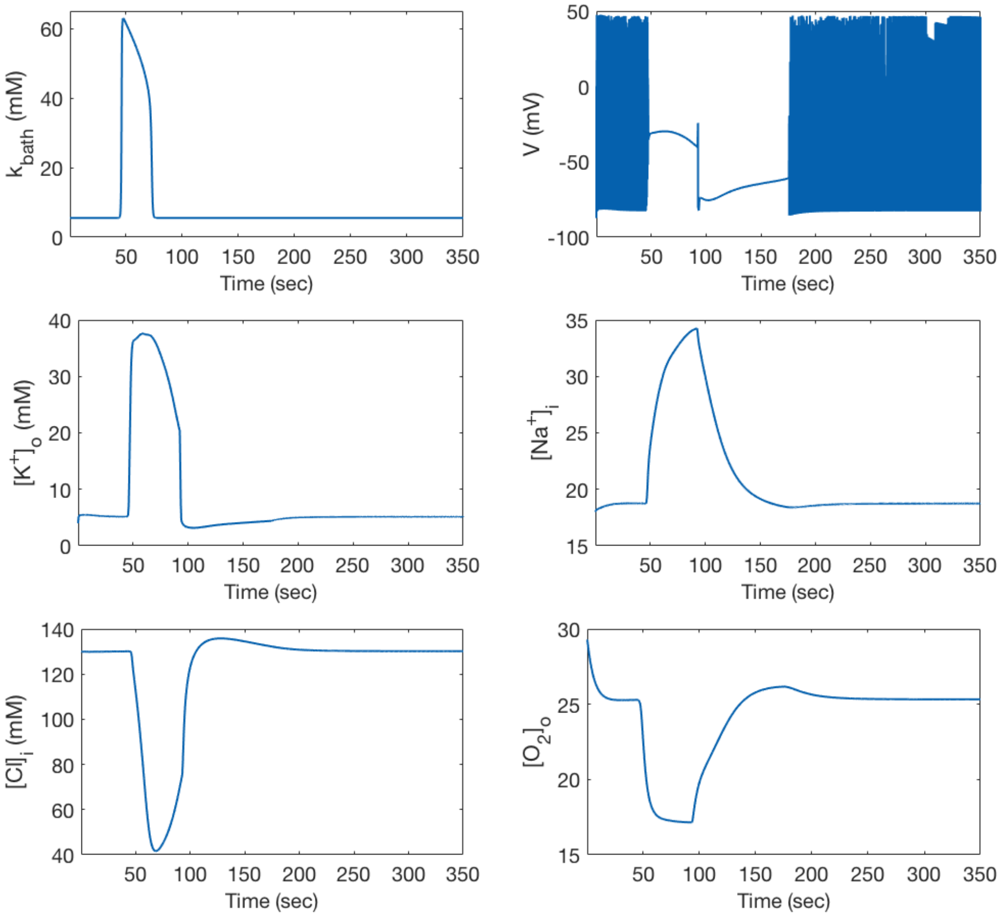


FIGURE 3.7: The extracellular potassium concentration k_{bath}, the membrane potential V, the extracellular potassium $[K^+]_o$, the intracellular sodium $[Na^+]_i$, the intracellular chloride $[Cl^-]_i$ and the extracellular oxygen $[O_2]_o$ at point $x = 0.9$.

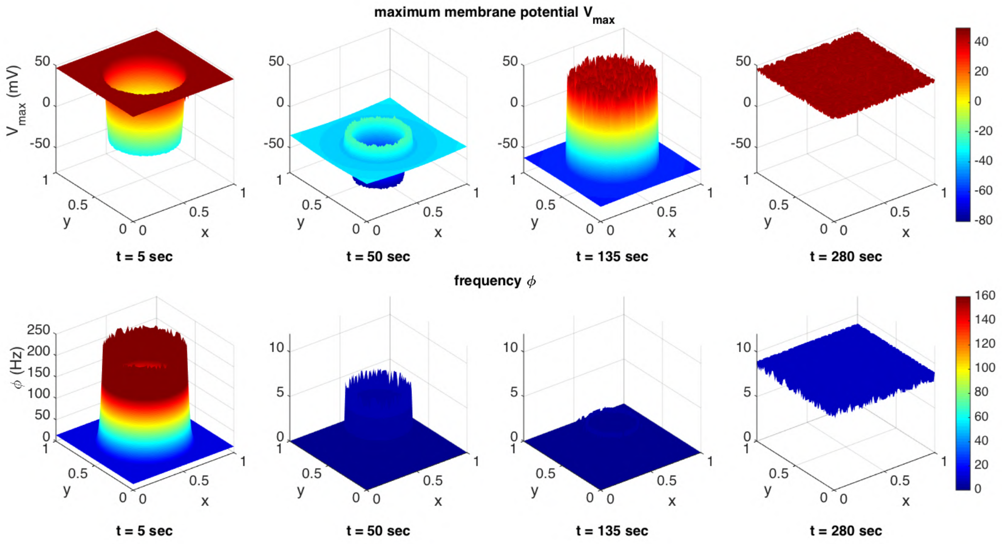


FIGURE 3.8: The maximum membrane potential V_{max} and the frequency ϕ at four different time steps of the CSD propagating on the unit square.

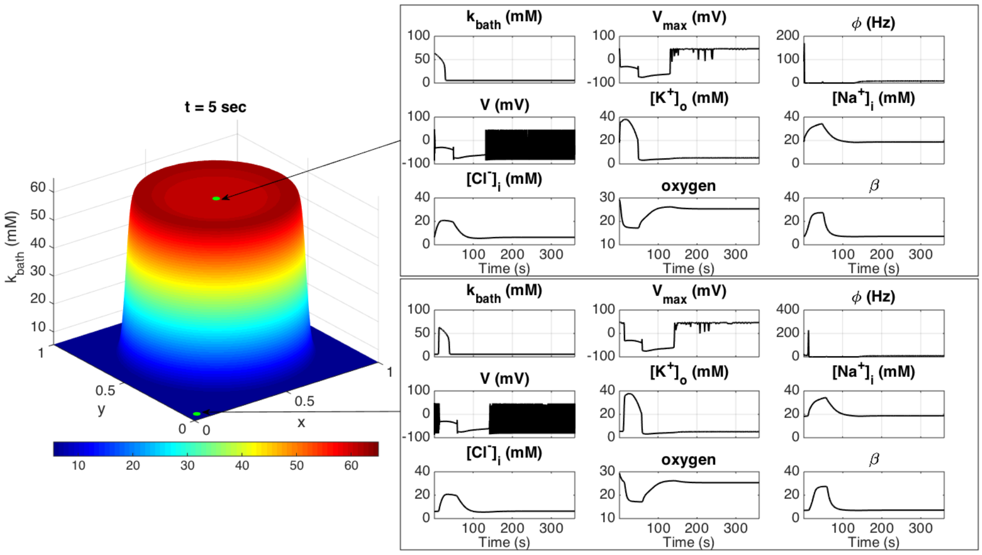


FIGURE 3.9: The k_{bath} distribution at $t = 5$ seconds over the whole domain, and the whole temporal dynamics for the k_{bath}, maximum membrane potential V_{max}, the frequency ϕ, the membrane potential V, the extracellular potassium $[\text{K}^+]_o$, the intracellular sodium $[\text{Na}^+]_i$, the intracellular chloride $[\text{Cl}^-]_i$, the extracellular oxygen $[\text{O}_2]_o$ and the ratio between the intra- and extracellular volume β in the points $(0.5, 0.5)$ (top panel) and $(0.05, 0.05)$ (bottom panel).

3.6 Summary

In order to simulate the propagation of CSD we proposed in this chapter a multi-scale PDE-ODE model which takes into account not only the massive changes in ionic homeostasis and its corresponding electrophysiological details but also the propagative character of CSD waves.

We modified the electrophysiological model proposed by Wei, Ullah, and Schiff, 2014 (introduced in Section 2.2.2) in such a way that it accounts for realistic CSD-like behaviour and coupled it with a propagative model for the extracellular potassium concentration. This coupled model unifies the propagation of the CSD on the minutes scale and the electrophysiology dynamics that takes place in the order of milliseconds. We illustrate the interaction of these two levels of action on a simple example in 1D space and a 2D slab representing neuronal tissue.

The detailed description of the electrophysiological activity in addition to the potassium propagation improves the modelling accuracy and underlines the connection between these phenomena.

The results of this chapter are collected in the paper by Gerardo-Giorda and Kroos, 2017.

Chapter 4

Model personalisation

"It is more important to know what sort of person has a disease,
than to know what sort of disease a person has."

Author unknown

Precision medicine (or personalised medicine) is the concept of customising healthcare with medical decisions, treatments, practices or products tailored to the individual patient. While patient-specific therapies go back to the time of Hippocrates, it is still a pending topic in modern science and medicine as big data and fast genomic sequencing allow a deeper insight in the molecular basis of diseases (Engnew, 2009). Mathematical models applied to medicine can help to understand diseases, develop drugs and hospital technology. Especially including patient-specific information in a model can improve its reliability and meaningfulness.

In the case of migraine with aura and the simulation of the underlying CSD, there are two levels at which individual patient data can be integrated into the model introduced in Chapter 3: the first step is to use a brain geometry obtained from an MRI scan; the second one is to incorporate patient-specific diffusion data collected either from diffusion tensor imaging (DTI) or diffusion-weighted imaging (DWI).

As the sulci and gyri of the human cortex provide a unique structure that is comparable to the friction ridges of the human fingerprint, their patterns are unique to every human being. Thus, including the individual geometry of the cerebral cortex is a first step to personalise the computational domain. Though measurements of diffusion anisotropy were traditionally used as a white matter imaging technique, there is growing evidence that DTI data can be useful to describe the grey matter microstructure (McNab et al., 2013). Consequently, the combination of the patient-specific geometry and the diffusion data representing the microstructure of the cerebral cortex provide highly individual information on the patient's brain for the model.

In the following sections we introduce the basics of MRI, DTI and DWI imaging, give a description of the data at hand and a detailed explanation of how we incorporate them in the CSD model. Finally, we conclude the chapter by comparing the isotropic to the anisotropic case, stressing the improvement we make by adding patient-specific diffusion data.

4.1 MRI-T1 data

The brain cortex is a thin layer of grey matter, characterised by a very large aspect ratio between its surface and its volume. At the spatial scale we are considering, such layer can be approximated by a two-dimensional surface obtained from MRI measurements.

The cortical geometry we use in this study is a closed surface $\Omega \subset \mathbb{R}^3$ that has been reconstructed from an MRI scan with FreeSurfer[1] image analysis suite, see Fischl, 2012 and references therein for further details. This data set was acquired and published in Diez et al., 2015 and used in Kroos et al., 2016, and its acquisition was approved by the Ethics Committee at the Cruces University Hospital. The data set corresponds to one healthy subject, male, age 28, and was acquired with a Philips Achieva 1.5 T Nova scanner. The cortical geometry was obtained from a high-resolution anatomical MRI, acquired using a T1-weighted 3D sequence with the following parameters: TR = 7.482 msec, TE = 3.425 msec; parallel imaging (SENSE) acceleration factor = 1.5; acquisition matrix size = 256 x 256; FOV = 26 cm; slice thickness = 1.1 mm; and 170 contiguous sections.

The resulting computational mesh is a triangulation of the reconstructed cortex, consisting of 140,208 nodes and 280,412 triangles for the left hemisphere, and 139,953 nodes and 279,902 triangles for the right hemisphere. We visualise the mesh triangulation of the left hemisphere in Figure 4.1. This geometry already appeared previously in Section 1.2 in the visualisation of the Desikan-Killiany atlas and the lobes (Figure 1.3 and Figure 1.4).

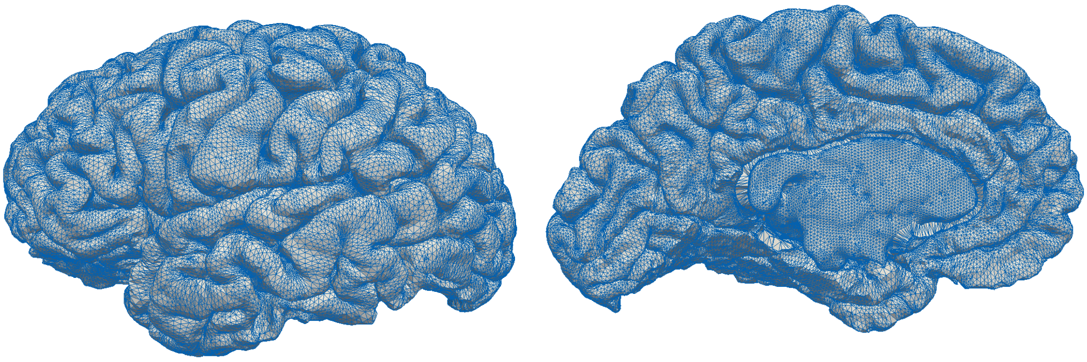


FIGURE 4.1: The triangular mesh of the reconstructed cortex in the lateral and medial view of the left hemisphere.

A priori the corpus callosum, an area made of white matter linking the two hemispheres, could be excluded from the computational domain, but in this case, boundary conditions should be imposed at the interface between corpus callosum and grey matter. Thus, at a first stage, we keep the corpus callosum as part of the brain geometry. In Section 5.1 we improve the mesh triangulation, remove noise by applying a smoothing technique and in this step also remove the corpus callosum from the computational domain.

---

[1] `http://surfer.nmr.mgh.harvard.edu/`

4.1.1 CSD propagation on a cortical geometry

We apply the model introduced in the previous chapter to the real brain geometry obtained from MRI imaging and can observe the propagation of the depolarisation wave triggering the microscopic dynamics. We consider an isotropic diffusion tensor, $D = \delta\mathbb{I}$ with $\delta = 0.7174$ mm^{2}sec^{-1} and no boundary conditions, as the cortical geometry at hand is a closed surface $\Omega \subset \mathbb{R}^3$. We initialise the problem by imposing $k_{\text{bath}} = 64$ mM in the lateral occipital region in the occipital lobe, to which the visual cortex belongs.

To show the behaviour of the model on the real brain geometry, we simulate the CSD on the left hemisphere. In Figure 4.2 we plot the k_{bath} profile on the cerebral cortex at time $t = 5$ min in the lateral and medial view, as well as the detailed temporal dynamics for two different points on the cortex, one on the lateral and one on the medial side, highlighted by green circles. This example shows the propagation of the extracellular potassium across the cortex triggering the characteristic CSD dynamics of the electrophysiology.

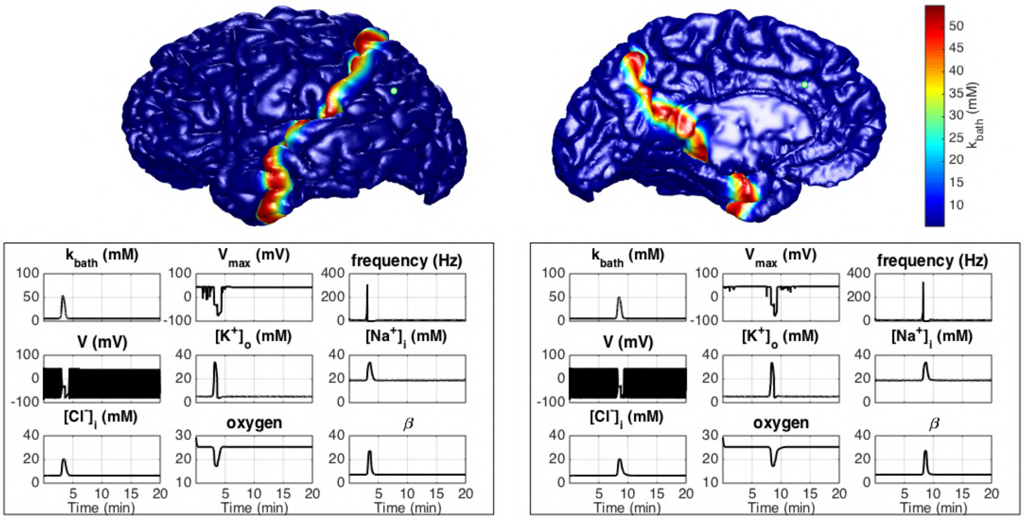


FIGURE 4.2: The k_{bath} profile on the cerebral cortex at time $t = 5$ min in the lateral and medial view of the left hemisphere, and the detailed temporal dynamics for two different points on the cortex.

The computation to simulate CSD propagation is complex and highly expensive. In particular, the memory necessary to store the information at the small time step $\Delta\tau$, to evaluate the functions involved at the bigger time steps Δt and to compute the measurable variables V_{max} and ϕ at the seconds scale, quickly increases with the number of data points in the spatial dimension. Saving the values for all 12 variables, describing the electrophysiology for 1000 small time steps to synchronise with the big time step (or even 20000 small time steps to synchronise the measurable variables at the seconds scale) and evaluate them at the 140,208 discrete nodes of the brain geometry results in an array of the size $12 \times 1000 \times 140,208$. Taking into account that each number with double precision requires 8 bytes of memory, this array would easily need several GB of memory just to be synchronised with one big time step. As this information is not the only one that has to be stored in the course of the simulation, running out of memory becomes an issue as soon as we start considering real brain geometries with a high number of discrete data points.

To avoid this problem, we will only consider the propagation of the extracellular potassium in the following, keeping in mind that the potassium wave triggers the electrophysiological characteristics of the CSD.

4.2 Diffusion tensor and diffusion-weighted imaging data

In neurology, brain pathologies can be detected by using DTI that provides a measurement for diffusion and anisotropy, the quality of exhibiting physical properties along different directional axis. DTI is mainly used to study the architecture of white matter in human brains in vivo. In white matter, diffusion tends to be more anisotropic than in grey matter, while in the cerebrospinal fluid the diffusion is equal in all directions (isotropic).

DTI imaging uses MRI techniques to measure the diffusion of water molecules in different directions in every voxel of an image. Its acquisition requires DWI measurements in at least seven non-collinear directions which are necessary to describe the rate and preferred direction of diffusion in each voxel (Huisman, 2010; Kingsely, 2006). Diffusion-weighted images are raw images where the signal has been sensitised to diffusion by using a strong magnetic field (Le Bihan, 2013).

The underlying concept of DTI is that the random movement of water molecules is influenced by its surrounding tissue. Water molecules interact and collide with tissue components like cell membranes, fibres or macromolecules. In free water at 37 °C, the diffusion coefficient of water molecules is $3 \cdot 10^{-9}\ \mathrm{m}^2\mathrm{sec}^{-1}$ which can be interpreted as a movement of 17μm in 50 msec (Le Bihan, 2003). In a specific type of tissue, the movement of water gives an insight into the structure and geometric organisation of the tissue itself. To measure the displacement of these molecules, a pair of magnetic field gradient pulses is used to make the MR signal sensitive to diffusion. Along this gradient, the change of location of the water nuclei is measured (Le Bihan, 2003). This change of position is proportional to the displacement of water molecules in a given space. The conductivity tensor for the tissue can be approximated from the water diffusion tensor obtained through DTI (Tuch et al., 1999, 2001).

In addition, DTI measurements can be used to describe the directional flow of information between non-contiguous regions of the brain. The process of extracting this directional information to obtain neural tracts throughout the brain is called *tractography*. Including the tractography to connect the two hemispheres and fast connections between distant regions could be further added to the modelling, but we leave this as a possible future extension of the proposed work.

DTI data is given in the form of three eigenvalues and three normalised orthogonal eigenvectors at each grid point and has to be processed in order to contribute to the model. As the cortex is a surface in 3D, we need to derive a 2D diffusion tensor from the 3D DTI diffusion data. In the following, we introduce a method to obtain these 2D diffusion tensors.

4.2.1 Description of the DTI data

The DTI data was acquired and published in Diez et al., 2015 and used in Kroos et al., 2017. Its acquisition was approved by the Ethics Committee at the Cruces University Hospital and all the methods employed were in accordance with approved guidelines.

The DTI data was processed with the FSL software[2]. First, an eddy current correction was applied to overcome the artefacts produced by head movement and to correct the changes produced by the variations in gradient field directions during the acquisition. Then, the gradient directions were corrected using the head movement matrices obtained from the previous step. Next, a local fitting of the diffusion tensor was then applied to compute the tensor model at each voxel. For each voxel, the tensor model provided the three eigenvectors and the associated eigenvalues of the diffusion of water molecules within brain tissues. Then, to project the tensor data onto the points of the mesh triangulation, we first computed a nonlinear transformation between the T1 image and the fractional anisotropy map. This transformation took as input a previous linear transformation using 12 degrees of freedom and chose the mutual information as the cost function. Then, FreeSurfer was used to project the diffusion data onto the brain mesh. A more detailed description of the whole process can be found in Appendix C.

The data from the DTI is given in the form of three eigenvalues $\lambda_1, \lambda_2, \lambda_3$ and three normalised and orthogonal eigenvectors $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3$ at each grid point. The eigenvectors are sorted according to the size of the eigenvalue. Thus, $\boldsymbol{w}_1$ describes the principal direction of the movement. These eigenvalues and eigenvectors describe the principal direction and strength of the displacement of water molecules in the underlying tissue. The diffusion tensor in 3D space can be described as

$$D = \lambda_1 \boldsymbol{w}_1 \boldsymbol{w}_1^T + \lambda_2 \boldsymbol{w}_2 \boldsymbol{w}_2^T + \lambda_3 \boldsymbol{w}_3 \boldsymbol{w}_3^T \tag{4.1}$$

or equivalently as a 3×3 matrix

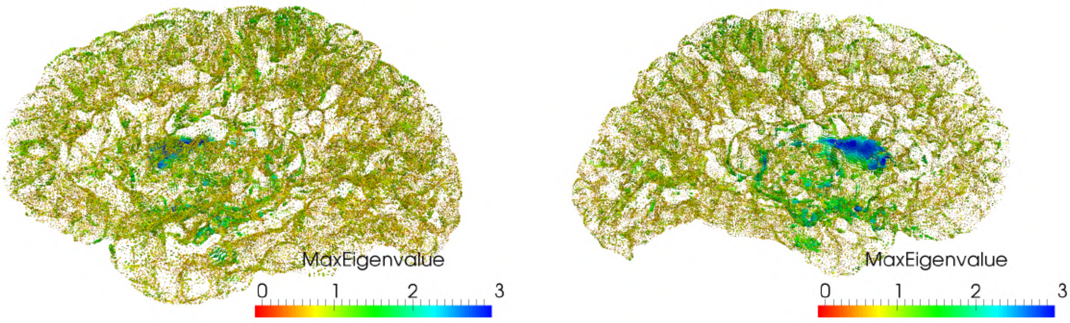
$$D = \begin{pmatrix} D_{xx} D_{xy} D_{xz} \\ D_{yx} D_{yy} D_{yz} \\ D_{zx} D_{zy} D_{zz} \end{pmatrix} . \tag{4.2}$$

The diffusion tensor D can be visualised as an ellipsoid that describes the average displacement and direction of the molecules in a particular spot. The eigenvectors determine the principal directions of the ellipsoid's axis and the eigenvalues their length. An example of diffusion tensors representing the conductivity of the brain cortex and visualised as ellipsoids is given in Figure 4.3. Here, we plot the DTI data on the nodes of the left hemisphere mesh. We provide a global view of the anisotropy by a lateral and medial visualisation of the cortex, where the colour of the ellipsoids refers to the size of the largest eigenvalue. In addition, we plot different levels of zoom to make the anisotropy of the diffusion provided by the DTI data visible on the local scale. Both the orientation and size of the ellipsoids exhibit huge variations across the cortex. In homogenous media, as the diffusion is isotropic, the corresponding tensor is a multiple of the unit matrix, and the corresponding ellipsoid is a sphere.

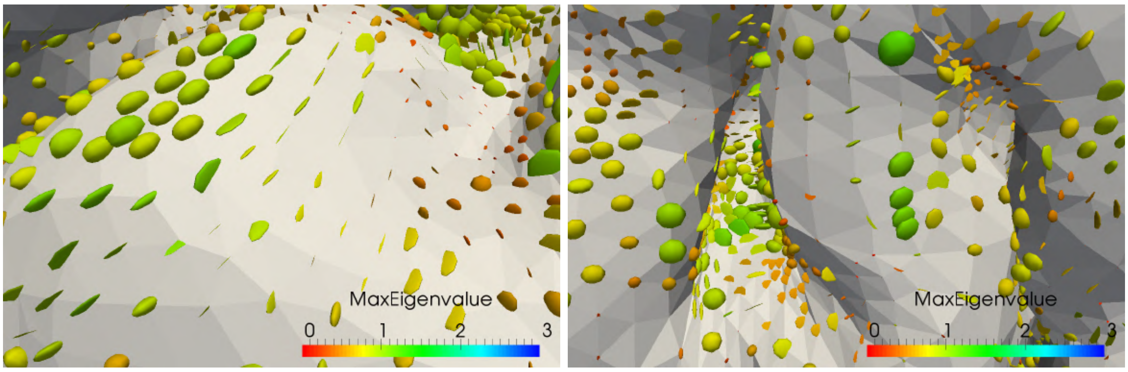
The data acquisition techniques for DTI and the low signal-to-noise ratio of this measurement can produce negative or vanishing eigenvalues (Soares et al., 2013). Another source for artefacts and missing data in the DTI datasets can originate from the gradient system hardware, the pulse sequence, and motion. The main problem is that DTI measurements are usually recorded with the aim to analyse diffusion in white matter. Thus, data acquisition is terminated as soon as it reaches border regions of the white matter. As a consequence, some values in these border regions are missing and set to zero. The mesh points of the geometry under consideration, featuring vanishing or negative eigenvalues

---

[2] `http://fsl.fmrib.ox.ac.uk/`



(a) lateral view (b) medial view



(c) different levels of zoom of the lateral surface

FIGURE 4.3: A visualisation of DTI data with ellipsoids coloured according to the size of the maximum eigenvalue on the left hemisphere in the lateral (a) and medial (b) view of the whole left cortex and two levels of zoom of the lateral surface (c).

or eigenvectors are visualised in Figure 4.4. Here, the termination of data acquisition is one of the main reasons for missing data, as the majority is located at the border regions at the very top and front of the brain geometry. Proper treatment of these artefacts is addressed in the next sections.

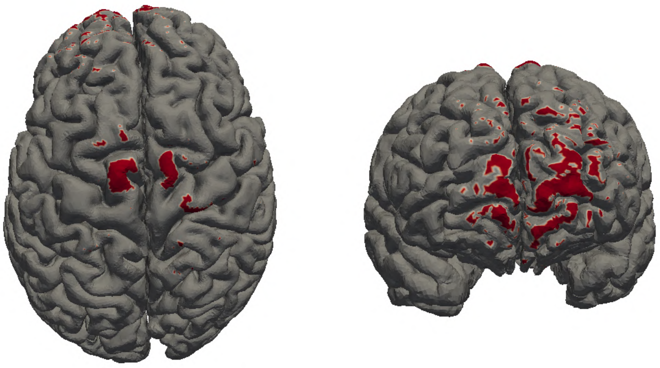


FIGURE 4.4: A visualisation of the missing or noisy DTI data in red on the cerebral cortex in the superior and anterior view.

4.2.2 Quantifying the diffusion data

The degree of anisotropy in the diffusion ellipsoid gives information about the diffusion data and provides information on the underlying tissue structure in each measuring unit (Le Bihan et al., 2001). There are three different ways to extract information from DWI data: the mean diffusivity describes the mean-squared displacement of molecules in a specific region or voxel, the main direction of diffusivity focuses on the orientation of the underlying structures, and the degree of anisotropy is a measure of how much the diffusion in the point deviates from the isotropic case.

The mean diffusivity (MD), giving a general evaluation of the diffusion, is defined as the trace of the diffusion tensor D in (4.2) divided by three or equivalently the mean of its diagonal entries:

$$\mathrm{MD} = \frac{Tr(D)}{3} = \frac{D_{xx} + D_{yy} + D_{zz}}{3}.$$

This quantity, as well as its mean values across the cortex, can give important information about the overall diffusion behaviour and speed. In a generic point in space, the MD can also be represented as the mean of the eigenvalues of the diffusion tensor D, namely

$$\mathrm{MD} = \frac{1}{3}\sum_{i=1}^{3} \lambda_i = \bar{\lambda}. \tag{4.3}$$

In Figure 4.5 we plot the MD normalised with respect to its mean on the cortex of the brain geometry for both hemispheres, and give the normalised histogram that describes the probability density of the MD values. The normalisation highlights areas in which the MD is higher or lower than the average. In this figure, we visualise the values on each triangle to make it comparable to the 2D case. For this, we average the MD values over all the vertices of the respective triangle.

For the degree of anisotropy a variety of measures have been proposed (Kingsley, 2006; Le Bihan et al., 2001). We will focus on the most common ones: the volume ratio (VR), the relative anisotropy (RA) and the fractional anisotropy (FA).

The volume ratio describes the ratio between the ellipsoid volume and a sphere volume with radius $\bar{\lambda}$ (or equivalently, with radius MD), see (4.3). It is defined as

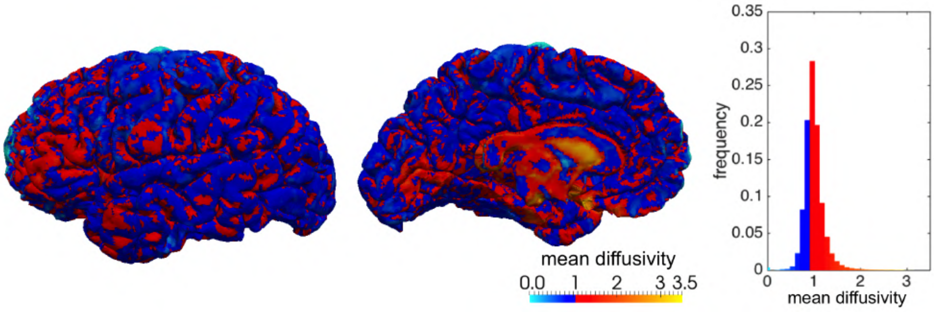
$$\mathrm{VR} = \frac{\lambda_1 \lambda_2 \lambda_3}{\bar{\lambda}^3},$$

and ranges from 0 to 1, where 1 is the fully isotropic situation.

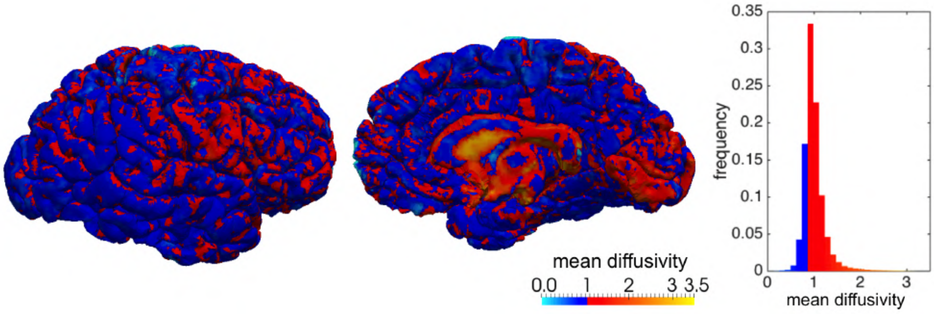
The relative anisotropy represents (like FA) the ratio of the anisotropic part of the diffusion tensor D to its isotropic part. It is a normalised standard deviation defined as

$$\mathrm{RA} = \sqrt{\frac{\left(\lambda_1 - \bar{\lambda}\right)^2 + \left(\lambda_2 - \bar{\lambda}\right)^2 + \left(\lambda_3 - \bar{\lambda}\right)^2}{3\bar{\lambda}^3}}$$

and varies between 0 and $\sqrt{2}$, where 0 represents a fully isotropic condition.



(a) left hemisphere



(b) right hemisphere

FIGURE 4.5: The normalised mean diffusivity of the three-dimensional DTI data of the cortex of the left (a) and right (b) hemisphere and a normalised histogram displaying the probability density of the MD values.

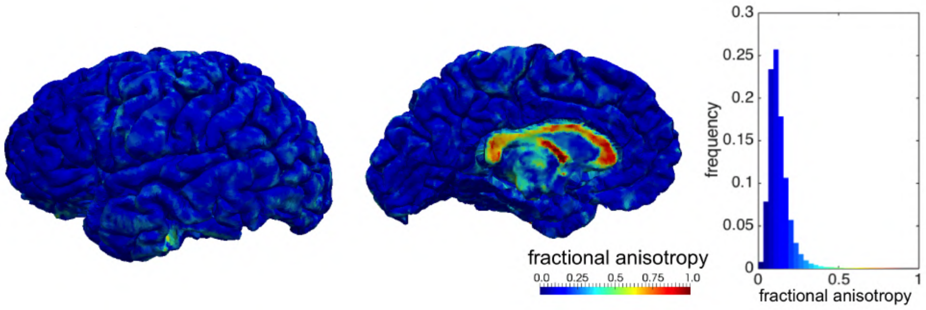
Finally, the fractional anisotropy expresses the normalised variance of the eigenvalues, defined as

$$\mathrm{FA} \;=\; \sqrt{\frac{3[(\lambda_1-\bar{\lambda})^2+(\lambda_2-\bar{\lambda})^2+(\lambda_3-\bar{\lambda})^2]}{2(\lambda_1^2+\lambda_2^2+\lambda_3^2)}}. \tag{4.4}$$

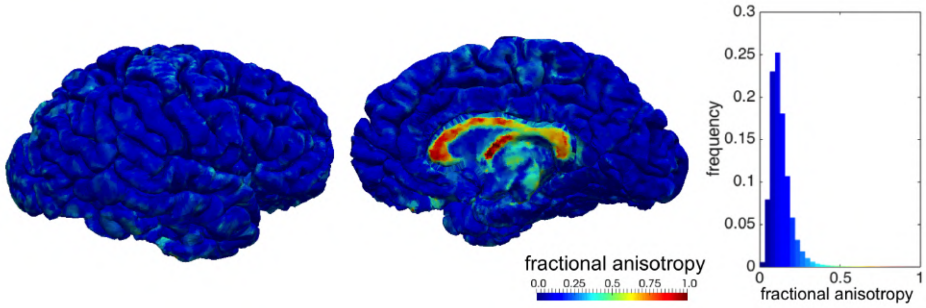
The fractional anisotropy is positive and smaller than 1, where 0 represents a fully isotropic condition and is the most commonly used measurement for the degree of anisotropy due to its convenient domain of definition. In Figure 4.6 we plot the FA of the left and right hemisphere as well as a normalised histogram displaying the relative frequency of the FA values. Again, we plot on each triangle the average FA value over its three vertices to make it comparable to the 2D case later on. In the original 3D data set, the diffusion is marginally anisotropic, with some areas of high anisotropy, like in the corpus callosum.

In Figure 4.7 we plot the joint distribution of 3D FA and the MD normalised by the mean over all triangles, for both left and right hemisphere. Again we consider the values in the centroids of the triangles by averaging the values over the vertices of each triangle. The colours represent density, and the difference to an isotropic situation, where mean diffusivity and fractional anisotropy are 1 and 0, respectively, is evident. Thus, this figure strongly supports the contribution of DTI-based diffusion coefficients.

In some cases, there are not enough DWI images to generate the diffusion tensor. When this occurs, the apparent diffusion coefficient (ADC) can be computed by averaging over images recorded with different diffusion weighting, giving the average diffusion of water

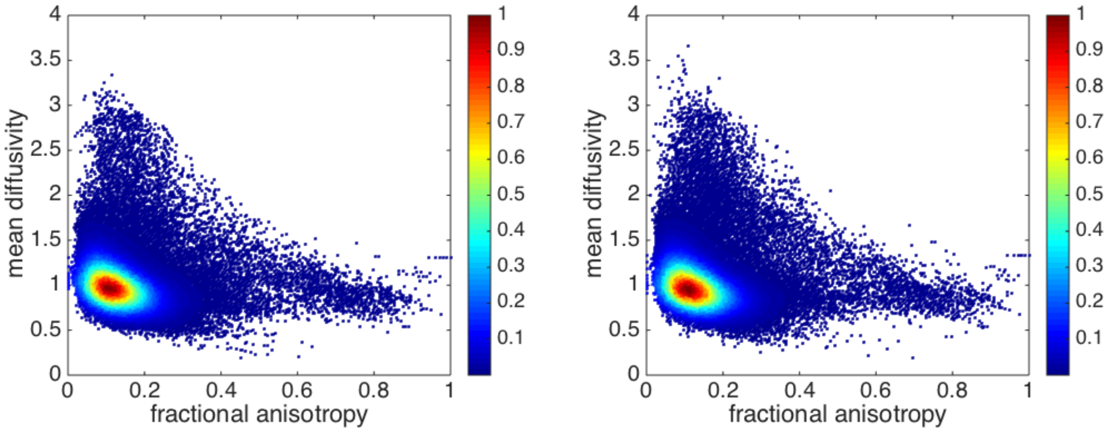


(a) left hemisphere



(b) right hemisphere

FIGURE 4.6: The fractional anisotropy of the three dimensional DTI data of the cortex of the left (a) and right (b) hemisphere and a normalised histogram displaying the probability density of the FA values.



(a) left hemisphere (b) left hemisphere

FIGURE 4.7: The joint distribution of the fractional anisotropy and the normalised mean diffusivity of the 3D tensor for each triangle of the mesh, both for the left (a) and right (b) hemisphere. Colours represent the density.

molecules in each voxel. However, the ADC only provides information about the diffusion in a voxel, whereas the anisotropy maps go one step further and give information about the orientation of the underlying structure.

Depending on the acquisition modalities either the ADC or the eigenvalues and eigenvectors for the diffusion tensor are available.

4.2.3 Implementation in the model

We use finite elements to approximate the problem and since the stiffness matrices are assembled by local integration on the triangles, we need to derive a 2D diffusion tensor from the 3D DTI data. The main idea is to intersect the ellipsoids describing the diffusion with the plane identified by the corresponding triangle of the surface mesh. The procedure consists of two main steps. First, the considered triangle and the ellipsoids (describing the diffusion in the three vertices) are moved with an isometric transformation to the xy-plane $z = 0$. The isometry guarantees that all angles and distances are preserved. In a second step, we intersect the transformed ellipsoids with the plane $z = 0$. The resulting tensors describe the diffusion on the 2D surface. In the following, we describe the details of this procedure.

Local construction of the diffusion tensor on the surface We consider a generic element K of the triangulation with vertices P_1, P_2 and P_3 and the ellipsoid from the DTI data in vertex P_1. We describe the procedure only for the diffusion tensor in P_1 as an example. The final definition of the diffusion tensor in 2D requires this information in all the vertices of the triangle which can be obtained analogously in P_2 and P_3 with the transformation computed for P_1. In Figure 4.8, we show the element K and colour in grey the plane identified by the triangle. The labelling of the vertices P_1, P_2 and P_3 of the triangle is given by the adjacency matrix of the mesh. Following this local labelling, we consider a basis for $\mathbb{R}^3$ composed of the vectors $\boldsymbol{v}_1 = \overline{P_1P_2}$ and $\boldsymbol{v}_2 = \overline{P_1P_3}$ and the unit vector $\boldsymbol{e}_3 = (0,0,1)^T$ that we assume not to belong to the plane. In case the vector $\boldsymbol{e}_3$ belongs to the plane of the triangle, it is enough to replace it by another vector from the canonical basis of $\mathbb{R}^3$ not belonging to the considered plane. By $\langle \cdot, \cdot \rangle$ we denote the scalar product in $\mathbb{R}^3 \times \mathbb{R}^3$ and by $|| \cdot ||$ the Euclidean norm on $\mathbb{R}^3$.

For the first step of the procedure, we aim for a 3D linear transformation matrix $T = (T_{ij})_{i,j=1,2,3} \in \mathbb{R}^{3\times 3}$ that transforms the original 3D space to the space in which the triangle K is situated in the xy-plane, $z = 0$. Setting as a constraint that the transformation puts P_1 in the origin of coordinates $(0,0,0)^T$ and that the edge $\boldsymbol{v}_1$ coincides with the x-axis, we obtain the system

$$T\boldsymbol{v}_1 = \begin{pmatrix} ||\boldsymbol{v}_1|| \\ 0 \\ 0 \end{pmatrix}, \qquad T\boldsymbol{v}_2 = \begin{pmatrix} ||\boldsymbol{v}_2||\cos(\theta) \\ ||\boldsymbol{v}_2||\sin(\theta) \\ 0 \end{pmatrix}, \qquad T\boldsymbol{e}_3 = \begin{pmatrix} T_{13} \\ T_{23} \\ T_{33} \end{pmatrix}, \tag{4.5}$$

where θ is the angle between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, with

$$\cos\theta = \frac{\langle \boldsymbol{v}_1, \boldsymbol{v}_2 \rangle}{||\boldsymbol{v}_1||\,||\boldsymbol{v}_2||},$$

and $T\boldsymbol{e}_3$ has to be determined in a way that it maintains the angles between $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{e}_3$, respectively. By denoting with α and β the angles between $\boldsymbol{v}_1$ and $\boldsymbol{e}_3$, and $\boldsymbol{v}_2$ and $\boldsymbol{e}_3$, we impose

$$\cos\alpha = \frac{\langle \boldsymbol{v}_1, \boldsymbol{e}_3 \rangle}{||\boldsymbol{v}_1||} = \frac{\langle T\boldsymbol{v}_1, T\boldsymbol{e}_3 \rangle}{||T\boldsymbol{v}_1||\,||T\boldsymbol{e}_3||}, \qquad \cos\beta = \frac{\langle \boldsymbol{v}_2, \boldsymbol{e}_3 \rangle}{||\boldsymbol{v}_2||} = \frac{\langle T\boldsymbol{v}_2, T\boldsymbol{e}_3 \rangle}{||T\boldsymbol{v}_2||\,||T\boldsymbol{e}_3||}.$$

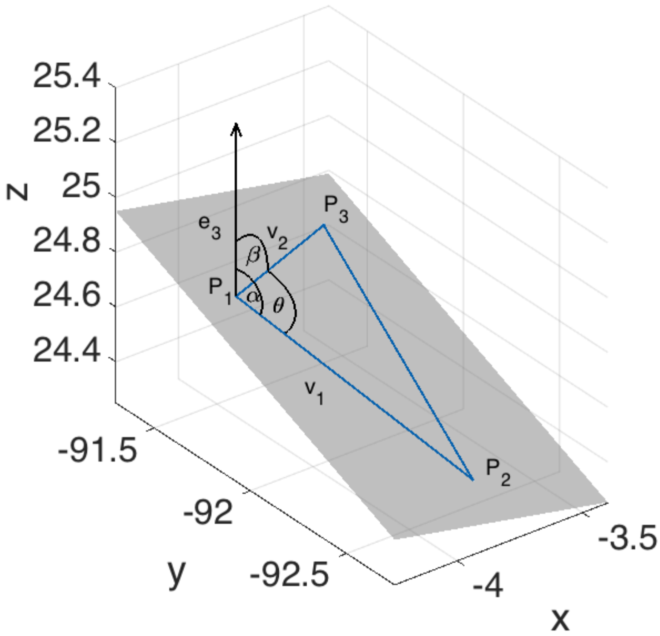


FIGURE 4.8: A generic element (blue) of the triangulation with the notations of the vertices, angles and vectors. The grey plane is an extension of the plane on which the triangle lies.

Substituting the definition of $T\boldsymbol{v}_1$ and $T\boldsymbol{v}_2$ from equation (4.5) into the previous equation and keeping in mind that $||T\boldsymbol{e}_3|| = 1$, we obtain the entries of the third column of T as

$$T_{13} = \frac{\boldsymbol{v}_1(3)}{||\boldsymbol{v}_1||}, \qquad T_{23} = \left(\frac{\boldsymbol{v}_2(3)}{||\boldsymbol{v}_2||} - T_{13}\cos\theta\right)\frac{1}{\sin\theta}, \qquad T_{33} = \sqrt{1 - T_{13}^2 - T_{23}^2},$$

where $\boldsymbol{v}_j(i)$ denotes the i-th component of the vector $\boldsymbol{v}_j$. By backward substitution in equation (4.5), simple algebra shows that the remaining entries of T can be obtained by solving the linear systems

$$\begin{aligned} A\begin{pmatrix} T_{11} \\ T_{12} \end{pmatrix} &= \begin{pmatrix} ||\boldsymbol{v}_1|| - T_{13}\boldsymbol{v}_1(3) \\ ||\boldsymbol{v}_2||\cos\theta - T_{13}\boldsymbol{v}_2(3) \end{pmatrix}, \\ A\begin{pmatrix} T_{21} \\ T_{22} \end{pmatrix} &= \begin{pmatrix} -T_{23}\boldsymbol{v}_1(3) \\ ||\boldsymbol{v}_2||\sin\theta - T_{23}\boldsymbol{v}_2(3) \end{pmatrix}, \\ A\begin{pmatrix} T_{31} \\ T_{32} \end{pmatrix} &= \begin{pmatrix} -T_{33}\boldsymbol{v}_1(3) \\ -T_{33}\boldsymbol{v}_2(3) \end{pmatrix}, \end{aligned} \tag{4.6}$$

where A is the matrix

$$A = \begin{pmatrix} v_1(1) & v_1(2) \\ v_2(1) & v_2(2) \end{pmatrix}.$$

In Figure 4.9 we visualise the original triangle with the diffusion ellipsoid in P_1, and the triangle K and the ellipsoid after the transformation.

An important point in building the transformation T is the unique solvability of the linear systems in equation (4.6), which is guaranteed as long as the matrix A is not singular. The only way for A to be singular is that the first two components of $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are aligned, but this option is ruled out as a consequence of choosing $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{e}_3$ as a basis. However, from the computational standpoint, even though the matrix in equation (4.6) is not singular, it can be badly conditioned and trigger round-off errors in the solution of the linear systems. To improve computational accuracy, we check the determinant of matrix A and, if it is less than 10^{-4}, centre the transformation in one of the other two vertices of the triangle.

After the transformation, the second step to obtain the diffusion data in 2D on the surface is to intersect the ellipsoid with the plane identified by the triangle obtaining an ellipse. The ellipsoid representing the 3D DTI data in point P_j for $j = 1, 2, 3$ is given by

$$(X - P_j)^T \; W_j \, \Lambda_j^{-2} \, W_j^T \; (X - P_j) = 1, \tag{4.7}$$

where $W_j = [\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3]$ is the matrix whose columns are the eigenvectors in point P_j, while $\Lambda_j = \operatorname{diag}(\lambda_1, \lambda_2, \lambda_3)$ is the diagonal matrix of the eigenvalues in point P_j and $X = (x, y, z) \in \mathbb{R}^3$. Letting $T \in \mathbb{R}^{3\times 3}$ denote the previously introduced isometric transformation, we indicate by $\widehat{P}_j = T P_j$ the transformed vertices of the triangle. By definition we have $\widehat{P}_1 = (0, 0, 0)$, while $\widehat{P}_j = (x_j, y_j, 0)$ for suitable x_j, y_j when $j = 2, 3$. The equation of the image of the ellipsoid in $\widehat{P}_j$ under the transformation T is then given by

$$\left(X - \widehat{P}_j\right)^T \; (T\,W_j) \; \Lambda_j^{-2} \; (T\,W_j)^T \; \left(X - \widehat{P}_j\right) = 1. \tag{4.8}$$

Let $B_j = (T\,W_j) \; \Lambda_j^{-2} \; (T\,W_j)^T$. By intersecting the ellipsoid (4.8) with the plane $z = 0$, we obtain the ellipse

$$\begin{pmatrix} x - x_j \\ y - y_j \end{pmatrix}^T \; B_{j,33} \; \begin{pmatrix} x - x_j \\ y - y_j \end{pmatrix} = 1, \tag{4.9}$$

where $x, y \in \mathbb{R}$ and we denote by $B_{j,33} \in \mathbb{R}^{2\times 2}$ the principal submatrix of B_j of order 2, obtained by discarding the third row and the third column.

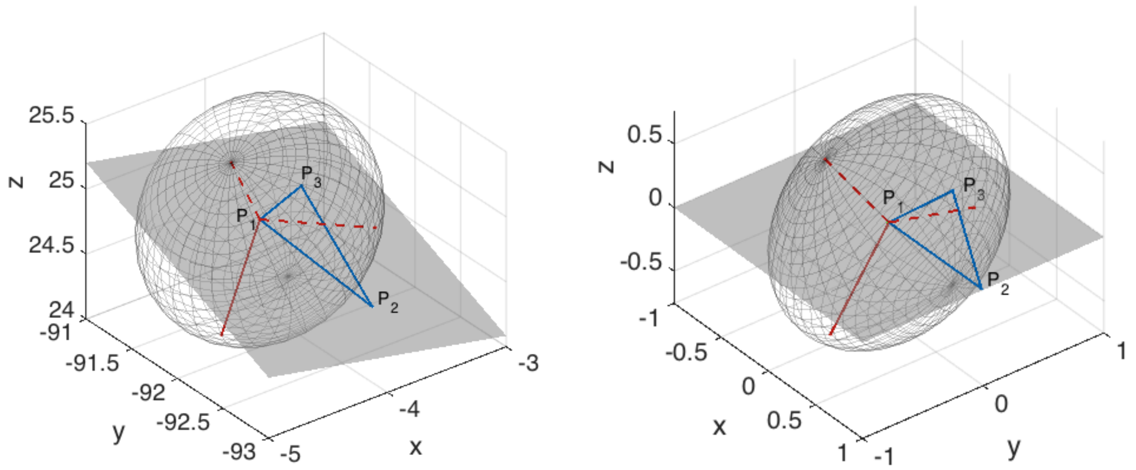
For each vertex of the triangle $\widehat{P}_j$, let $(\xi_{1,j}, \boldsymbol{w}_{1,j})$ and $(\xi_{2,j}, \boldsymbol{w}_{2,j})$ be the pair of eigenvalue and eigenvector of $B_{j,33}$ (with normalised eigenvectors): the ellipse semi-axes are oriented along $\boldsymbol{w}_{1,j}$ and $\boldsymbol{w}_{2,j}$, and their lengths are given by $\mu_{1,j} = 1/\sqrt{\xi_{1,j}}$, and $\mu_{2,j} = 1/\sqrt{\xi_{2,j}}$.

We sort them according to their size, namely

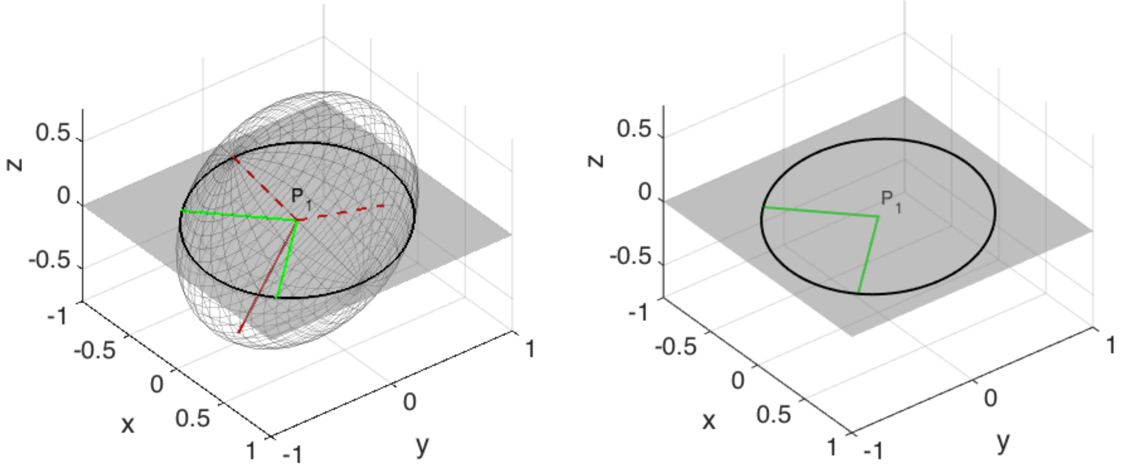
$$\mu_j^+ = \max(\mu_{1,j}, \mu_{2,j}), \qquad \mu_j^- = \min(\mu_{1,j}, \mu_{2,j}),$$

and denote by $\boldsymbol{p}_j$ and $\boldsymbol{q}_j$ the eigenvectors associated with μ_j^+, and μ_j^-, respectively. By construction, $\boldsymbol{p}_j$ corresponds to the principal direction of diffusion in 2D, whereas $\boldsymbol{q}_j$ is the secondary direction of diffusion. Notice that for an isotropic point P_j, as the intersection of a sphere with any plane through its centre is a circle, we are left with $\mu_j^+ = \mu_j^-$. In Figure 4.9 we plot the transformed ellipsoid in point P_1 of the triangle and the resulting ellipse with its principal axis from the intersection with the xy-plane.

Since the diffusion tensors are built at the triangular level, additional information from DTI data can be obtained by considering the MD and FA values associated with each triangle. A natural choice is to consider such values in its centroid. In the numerical simulation of CSD propagation on a cerebral cortex, we chose linear finite elements, and we can assume that both the orientation and the size of the diffusion coefficients change linearly across the triangle K. Thus, the diffusion coefficients in the centroid, that we



(a) notations triangle (b) after transformation



(c) intersecting ellipsoid and plane (d) ellipse of diffusion in 2D

FIGURE 4.9: A generic element of the triangulation (blue), and the example ellipsoid from DTI data in vertex P_1. The plane identified by the triangle is shown in grey. The original triangle with its notations (a), the triangle and the ellipsoid associated with P_1 after the transformation (b), intersection between the transformed ellipsoid and the xy-plane (c), the main diffusion directions (green) on the xy-plane and the ellipse (black) associated with the 2-D conductivity tensor (d).

denote by μ_K^+ and μ_K^-, can be interpolated as

$$\mu_K^+ = \frac{1}{3}\sum_{j=1}^{3} \mu_j^+ \qquad \text{and} \qquad \mu_K^- = \frac{1}{3}\sum_{j=1}^{3} \mu_j^- . \tag{4.10}$$

Owing to (4.10), the 2D FA and MD values of the triangle K are given by

$$\mathrm{FA}_K = \frac{\mu_K^+ - \mu_K^-}{\sqrt{(\mu_K^+)^2 + (\mu_K^-)^2}} \qquad \text{and} \qquad \mathrm{MD}_K = \frac{\mu_K^+ + \mu_K^-}{2}, \tag{4.11}$$

respectively. We denote by MD_{mean} the mean of MD_K over all the elements of the triangulation.

We can define the 2D diffusion tensor in the vertex P_j ($j = 1, 2, 3$) of triangle K as

$$D_j = \frac{\delta}{\mathrm{MD}_{mean}} \left(\mu_j^+ \, \boldsymbol{p}_j \, \boldsymbol{p}_j^T + \mu_j^- \, \boldsymbol{q}_j \, \boldsymbol{q}_j^T \right), \tag{4.12}$$

where δ is a scaling factor normalised by MD_{mean}.

The coefficient δ in (4.12) was tuned to adapt the velocity of the extracellular potassium wave across the cortex to a propagation time that is comparable to the one of CSD. In the case that real measurements were available, the parameter δ could be estimated through a data assimilation procedure (see, e.g., D'Elia et al., 2011; Delingette et al., 2012).

Treatment of artefacts and missing data In order to assign reasonable values for the diffusion at the points where the data is zero or incorrect, we assume isotropic conductivity ($D = d\,\mathbb{I}_2$) and select a diffusivity d depending on the mean of the diffusion tensor intensity in the centroids of all vertices that belong to a specific region. Therefore, we consider the anatomical subdivision of the brain cortex introduced in Section 1.2, which divides each hemisphere into 34 regions of interest (ROI) (Desikan et al., 2006). To assign reasonable values in these critical points, we average the mean diffusivity of all points that do not feature missing or negative values and lie in the same ROI as the point itself.

In the specific, by denoting with $\mathcal{R}$ the ROI to which the critical point belongs, with $\mathcal{P}_\mathcal{R}$ the set of points in $\mathcal{R}$ featuring data, and with $|\mathcal{P}_\mathcal{R}|$ the cardinality of $\mathcal{P}_\mathcal{R}$, we assign as eigenvalues at the critical points the values

$$\lambda_i = \sum_{P \in \mathcal{P}_\mathcal{R}} \frac{\mathrm{MD}_\mathcal{R}}{|\mathcal{P}_\mathcal{R}|}, \quad \text{for } i = 1, 2, 3 \tag{4.13}$$

where $\mathrm{MD}_\mathcal{R}$ is the mean diffusivity of all points in the considered ROI computed on the 3D data and we choose the eigenvectors $\boldsymbol{w}_i, i = 1, 2, 3$ as the standard basis for the Euclidean space.

Computational aspects In the previous section, we explained how we derive the 2D diffusion tensor in the vertices of a triangle. However, for the quadrature rule in the computation of the stiffness matrix in equation (3.8), we need to evaluate the diffusion tensor also in other points of the triangle. In the following, we show how to compute the diffusion tensor in the midpoints of the sides and the centroid of the triangle, as these are quadrature points required by the third-order Gaussian rule we use in our implementation. The basic idea is to compute the conductivity tensor at the sides' midpoints and at the centroid by interpolating both orientation and length of the semi-axis of the conductivity ellipses in the vertices. The following procedure can easily be extended to further additional points whenever a higher quadrature rule is used.

Let P_i and P_j with $i, j = 1, 2, 3, i \neq j$ be points with anisotropic diffusion and ϑ the angle between the major diffusion directions in the two points P_i and P_j, that is,

$$\vartheta = \arccos \Big| \langle \boldsymbol{p}_i, \boldsymbol{p}_j \rangle \Big|,$$

where $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ are normalised vectors. The principal diffusion direction in the side's midpoint between P_i and P_j is obtained by rotating $\boldsymbol{p}_i$ by half the angle ϑ between the two vectors $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. This rotation can be performed either clockwise or counterclockwise, but for the sake of consistency, we implement it in the direction of the shortest possible rotation between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. We define the rotation matrix as

$$R_{\vartheta/2} = \begin{pmatrix} \cos(\vartheta/2) & -\sin(\vartheta/2) \\ \sin(\vartheta/2) & \cos(\vartheta/2) \end{pmatrix} \quad \text{or} \quad R^*_{\vartheta/2} = \begin{pmatrix} \cos(\vartheta/2) & \sin(\vartheta/2) \\ -\sin(\vartheta/2) & \cos(\vartheta/2) \end{pmatrix}.$$

To select the rotation direction, we compute the vector corresponding to a full rotation of ϑ, namely $R_\vartheta \boldsymbol{p}_i$, and consider the angle between $R_\vartheta \boldsymbol{p}_i$ and $\boldsymbol{p}_j$. If this angle is zero, the rotation $R_{\vartheta/2}$ is along the shortest rotation path between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. In other words, we set

$$R = \begin{cases} R_{\vartheta/2}, & \text{if } \langle R_\vartheta \boldsymbol{p}_i, \boldsymbol{p}_j \rangle = 1 \\ R^*_{\vartheta/2}, & \text{else} \end{cases}. \tag{4.14}$$

Due to computational reasons, we also consider angles that are very close to zero as the shortest rotation path, that is, even when $|\langle R_\vartheta \boldsymbol{p}_i, \boldsymbol{p}_j \rangle - 1| < 10^{-6}$. Thus, the diffusion direction vectors $\boldsymbol{p}_{ij}$ and $\boldsymbol{q}_{ij}$ of the midpoint are defined as

$$\boldsymbol{p}_{ij} = R\,\boldsymbol{p}_i \quad \text{and} \quad \boldsymbol{q}_{ij} = R\,\boldsymbol{q}_j$$

The eigenvalues μ^+_{ij} and μ^-_{ij} are computed by linearly interpolating between μ^+_i and μ^+_j and μ^-_i and μ^-_j, respectively:

$$\mu^+_{ij} = \frac{\mu^+_i + \mu^+_j}{2} \quad \text{and} \quad \mu^-_{ij} = \frac{\mu^-_i + \mu^-_j}{2}.$$

The case when P_i or P_j is isotropic requires some additional care. For example, let us assume that P_j is isotropic, then any direction is the principal one, and for simplicity we choose the principal directions coinciding with the ones of P_i. Thus, $\vartheta = 0$, and we only interpolate the vector length.

For the diffusion tensor in the centroid, we already computed in equation (4.10) the corresponding diffusion coefficients. To obtain the associated eigenvectors, we consider the principal diffusion directions in the midpoint P_{ij} of a side of the triangle ($\boldsymbol{p}_{ij}$ and $\boldsymbol{q}_{ij}$ with $i, j \in \{1, 2, 3\}, i \neq j$) and the ones on the opposite vertex P_l ($\boldsymbol{p}_l$ and $\boldsymbol{q}_l$ with $l \notin \{i, j\}$). The choice of $\boldsymbol{p}_{ij}$ and $\boldsymbol{p}_l$ does not affect the result by construction. Letting ψ be the angle between the major vectors $\boldsymbol{p}_l$ and $\boldsymbol{p}_{ij}$ in the two points, given by

$$\psi = \arccos\Big|\langle \boldsymbol{p}_l, \boldsymbol{p}_{ij} \rangle\Big|,$$

we rotate the principal diffusion directions $\boldsymbol{p}_l$ and $\boldsymbol{q}_l$ in the vertex by $2/3$ of the angle ψ. Again, the rotation is performed in the shortest possible path for consistency. The diffusion directions in the centroid $\boldsymbol{p}_K$ and $\boldsymbol{p}_K$ are then given by

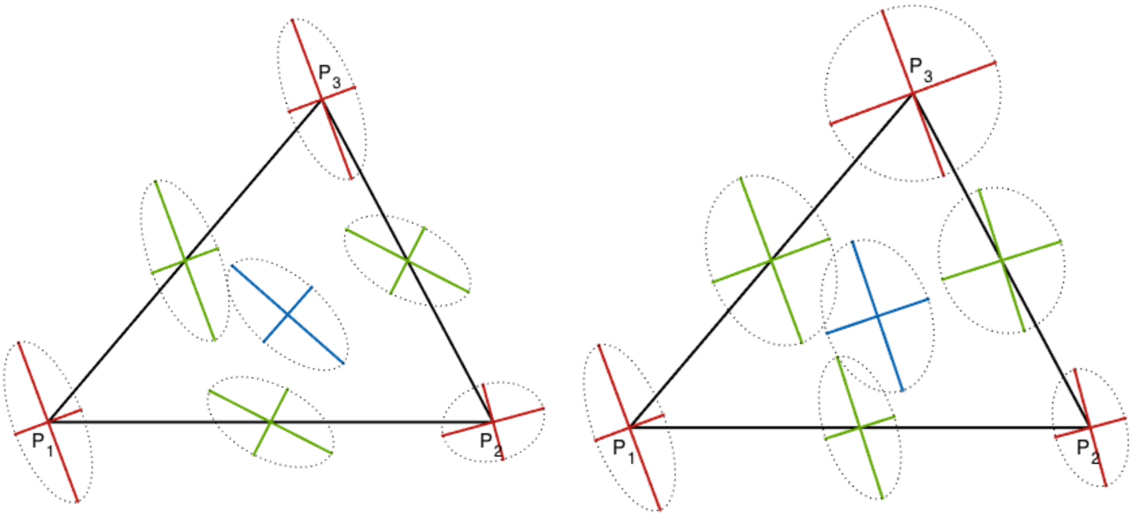
$$\boldsymbol{p}_K = Q\boldsymbol{p}_l \quad \text{and} \quad \boldsymbol{q}_K = Q\boldsymbol{q}_l,$$

where Q is the rotation matrix defined either as

$$Q = \begin{pmatrix} \cos(2\psi/3) & -\sin(2\psi/3) \\ \sin(2\psi/3) & \cos(2\psi/3) \end{pmatrix} \quad \text{or} \quad Q = \begin{pmatrix} \cos(2\psi/3) & \sin(2\psi/3) \\ -\sin(2\psi/3) & \cos(2\psi/3) \end{pmatrix},$$

following the same idea previously used to determine R in (4.14). In the case when either P_{ij} or P_k is an isotropic point, the same procedure highlighted before is applied.

In Figure 4.10 we show the interpolation of the diffusion direction on a triangle, both for anisotropic points and the case in which one point features isotropic diffusion. We plot the ellipses representing the 2D diffusion tensor and their major axis at the vertices, at the midpoints and at the centroid.



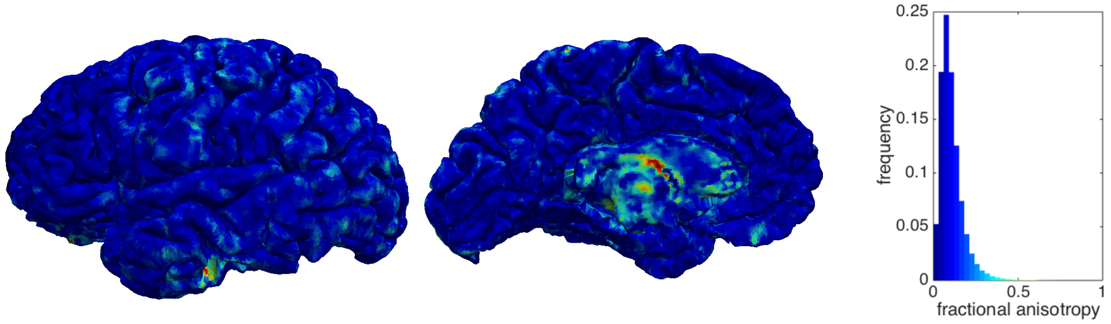
(a) anisotropic diffusion in all vertices (b) isotropic diffusion in P_3

FIGURE 4.10: Interpolation of diffusion direction and strength between the vertices of a triangle, with the axis of the diffusion ellipsoid at the vertices (red), at the midpoints (green) and at the centroid (blue). For anisotropic diffusion in all vertices (a) and isotropic diffusion in vertex P_3 (b).

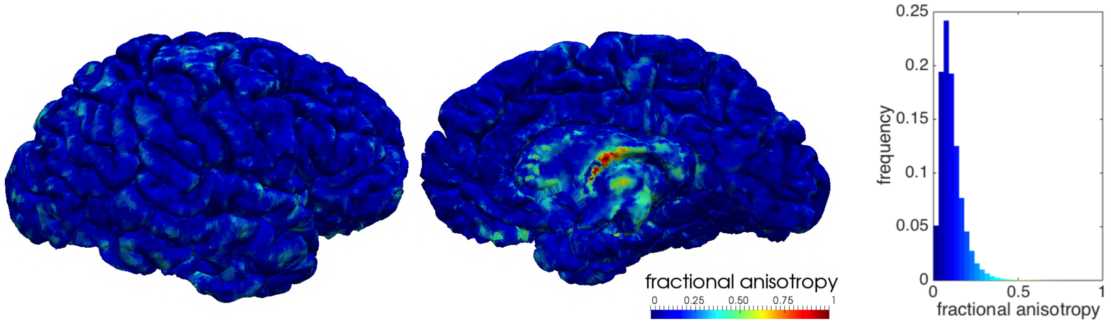
4.2.4 Two-dimensional diffusion on the cortex

The DTI data preprocessing from 3D to 2D diffusion tensors, introduced in the previous section, modifies the anisotropy characteristics of diffusion. Although some areas, like the corpus callosum, feature a higher anisotropy level in the 3D data set, FA values are centred around the value 0.2, showing that most areas feature mild anisotropy for the underlying tissue (see Figure 4.6). In Figure 4.11 we plot the 2D FA values at the centroid of each triangle for both the left and right hemisphere and a normalised histogram that describes the probability density of the FA values. This data maintains the overall structure of a distributed mild anisotropy, but it exhibits a slightly larger amount of isotropic points compared to the original 3D data. This feature depends strongly on the orientation of the 3D eigenvectors of the diffusion with respect to the surface. Nevertheless, areas supporting higher FA in the 3D data set maintain these characteristics also after the transformation of the diffusion tensor to 2D.

The reduction to 2D diffusion data on the surface does not only affect the FA but also the MD values. In Figure 4.12 we plot the MD values of the 2D transformed data at the centroids of each triangle, for both hemispheres. To highlight their variability, the plotted data is normalised with respect to the mean across the whole cortex. With this approach, we get insight into the behaviour of the parameter $\mathrm{MD}_K/\mathrm{MD}_{mean}$, which we use in the definition of the 2D diffusion tensor (see equation (4.11)). From Figure 4.12 we can infer that larger MD values appear in the sulci and the corpus callosum. In the latter, the mesh geometry is much flatter and high anisotropy values of white matter tracts yield a much



(a) left hemisphere



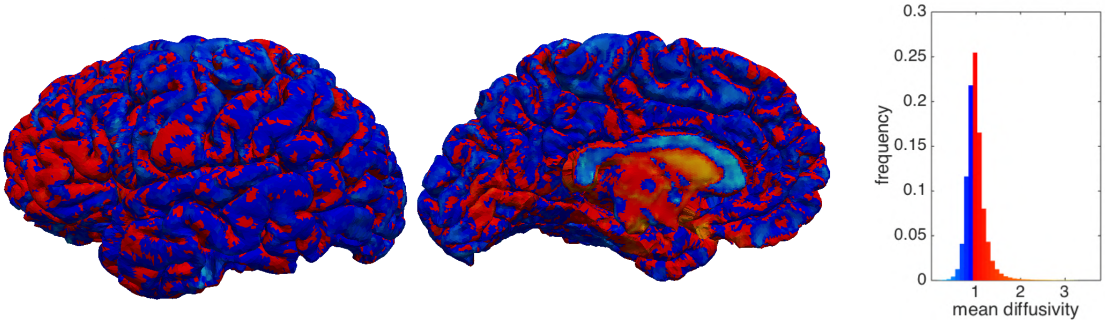
(b) right hemisphere

FIGURE 4.11: The fractional anisotropy of the 2D diffusion data of the cortex on the left (a) and right (b) hemisphere and a normalised histogram displaying the probability density of the FA values (right).

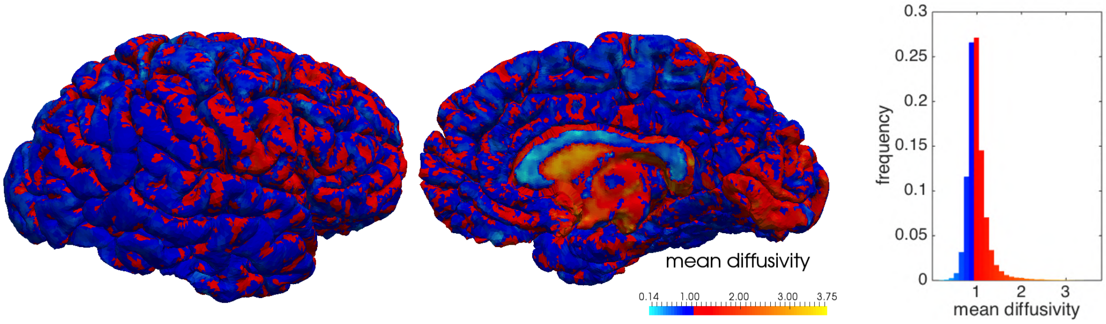
larger diffusion with respect to other cortical areas: as a consequence, the simulated CSD propagation is expected to travel faster in this region.

In Figure 4.13 we plot the joint distribution of 2D FA and the normalised MD values in the centroids of all triangles, for both left and right hemisphere. The colours represent the density, and one can observe the significant difference with respect to an isotropic situation, whose MD and FA values are 1 and 0, respectively. Evidence from Figure 4.13 is a major advocate for the use of DTI-based diffusion coefficients. In general, the reduction to 2D diffusion data led to a minor reduction of the anisotropies as more FA values are closer to 0 compared to the 3D case (see Figure 4.7). Comparing the MD values one can observe that the values are slightly more spread out. However, all these changes are minor and result from the reduction by one dimension.

Finally, we plot in Figure 4.14 the principal diffusion direction, its strength on the medial side of the left hemisphere and a zoom of the lateral side. The direction of the lines indicates the diffusion direction, and the diffusion strength is given by the vector's length and the colour. Here again, principal diffusion directions are considered in the centroids of the triangles.

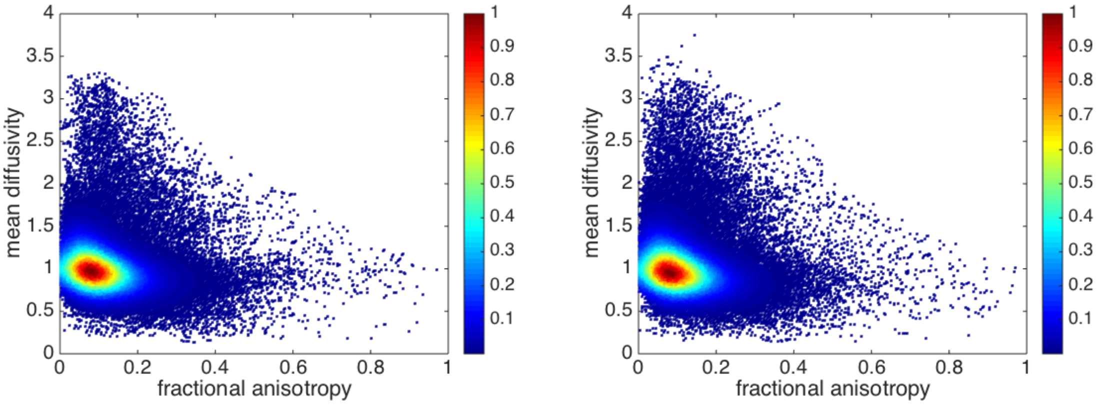


(a) left hemisphere



(b) right hemisphere

FIGURE 4.12: The normalised mean diffusivity of the 2D diffusion data of the cortex of the left (a) and right (b) hemisphere. Plotted data are normalised with respect to the average mean diffusivity across the whole cortical hemispheres.



(a) left hemisphere (b) right hemisphere

FIGURE 4.13: The joint distribution of the fractional anisotropy and the normalised mean diffusivity of the 2D diffusion tensors for each triangle of the mesh, both for the left (a) and right (b) hemisphere. Colours represent the density.

4.3 Simulation of the CSD progression with DTI data

After describing the DTI data processing, we are now left with a 2D diffusion tensor describing the diffusion direction and strength on the surface, the cerebral cortex. With

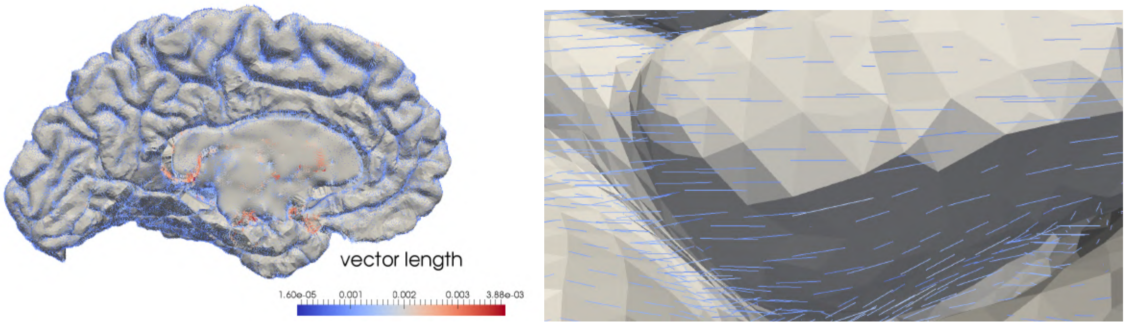


FIGURE 4.14: The principal diffusion direction in the centroids of each triangle of the left hemisphere, where the length and the colour describe the strength of the diffusion at this centroid.

this, we can simulate the CSD propagation on the cortex with spatially varying diffusion based on DTI data. For the sake of simplicity, we simulate the progressing wave of extracellular potassium, that is, system (3.4). The dynamics on the microscopic scale can be inferred from the potassium dynamics afterwards, if necessary.

In the following, we improve the scheme for the numerical solution and show the impact of diffusion tensors obtained from DTI data on the CSD propagation across the whole cerebral cortex.

4.3.1 Numerical optimisation

At every time step, we have to solve a linear system $Ax = b$ that can be costly. As matrix A does not change, we optimise the performance of the solver. To increase the efficiency and reduce computation times we use the Parallel Computing Toolbox of Matlab (MathWorks Inc., Natick, MA) that allows running computations on multicore processors. We parallelise all inner for-loops, when possible, and observe that some built-in functions are already optimised and run automatically in parallel.

The problem at hand is ill-conditioned, but with a reasonable choice of preconditioner and iterative solver, the numerical approximation can be sped up significantly.

Condition number of the problem The high number of grid points of the cortex triangulation results in a large sparse matrix $A = S + M/\Delta t$, with the mass and stiffness matrices M and S in the linear system (3.11) for the CSD propagation. The condition of matrix A

$$\kappa(A) = ||A||\,||A^{-1}||$$

is a measure for the sensitivity of the relative error of the solution to changes or errors in the input (Quarteroni, Sacco, and Saleri, 2007). Using the Matlab built-in function **condest**, the estimates of the condition number are

$$\kappa(A_{\text{left}}) = 5015.84, \quad \kappa(A_{\text{right}}) = 34305.18 \tag{4.15}$$

for the left and right hemisphere, respectively. Both linear systems are poorly conditioned, so that preconditioning is useful to speed up the numerical solution of the linear system.

Preconditioning The IMEX scheme described in Section 3.4 requires to solve a linear system associated to a system of linear PDEs, for which a vast literature on preconditioning is available (Saad, 2003). As potential preconditioners, we consider an incomplete LU factorisation without threshold and pivoting (ILU(0)), an incomplete LU factorisation with both threshold and pivoting (ILUTP), and an incomplete Cholesky factorisation with a drop tolerance (ICT). For a more detailed description of the preconditioners see Appendix B.1.

To quantify the different possible preconditioners P for a problem $Ax = b$, the most natural way is to compare the condition number of the matrix A to the one of $P^{-1}A$. However, such high dimensional matrices are difficult to invert. In the case of a symmetric positive definite matrix we have

$$\kappa_2(A) = \frac{\lambda_{\max}}{\lambda_{\min}},$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalues of A (Quarteroni, Sacco, and Saleri, 2007). Eigenvalues close to one (or in the complex case close to the unit circle) indicate that the problem is well-conditioned. To avoid the inversion of the matrix P in the eigenvalue problem

$$P^{-1}Ax = \lambda x$$

we consider the generalised eigenvalue problem

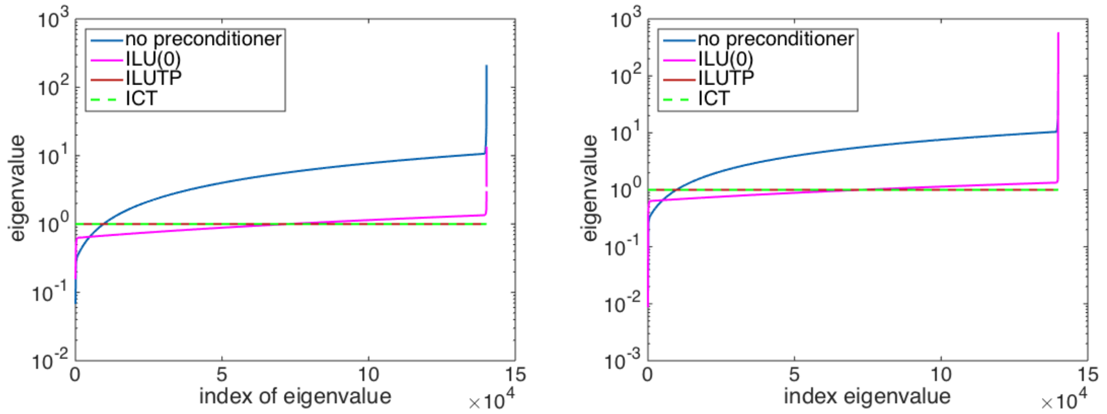
$$Ax = \lambda Px.$$

For the computation of the eigenvalues, we use the Matlab built-in function **eigs** which approximates the eigenvalues of large sparse matrices using the Lanczos algorithm (Lanczos, 1950).

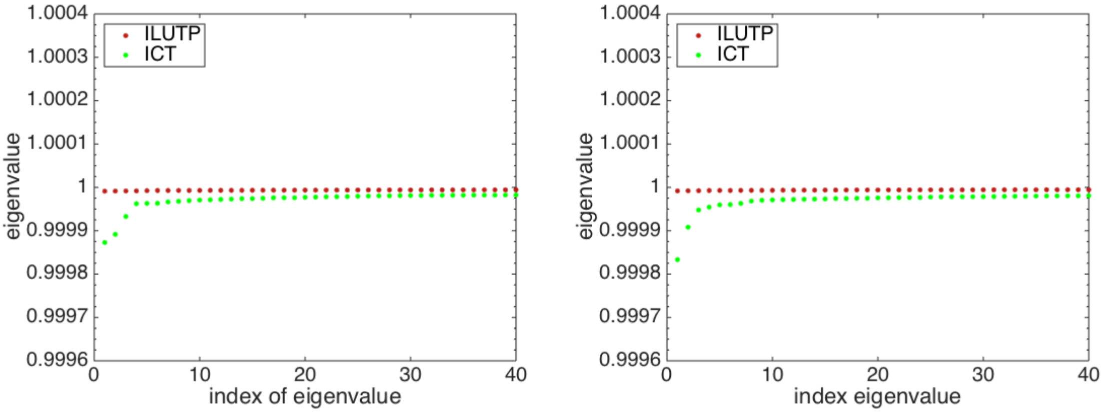
In Figure 4.15 we plot an approximation of the eigenvalues of the generalised eigenvalue problem with the different preconditioners for the left and right hemisphere. We approximate the smallest and biggest 500 eigenvalues and interpolate linearly between them. The drop tolerance for the ILUTP and ICT is set to 10^{-6}. In the top row, we plot the eigenvalues (sorted by size) of the different preconditioned systems for the left and right hemisphere. The incomplete Cholesky factorisation with no fill-up of positive definite matrices does not necessarily always exist, and in our case, it does not exist neither for the left nor right hemisphere. ILU(0) preconditioning improves the condition of the problem, but preconditioning with the ILUTP and the ICT perform the best.

Iterative solver The preconditioned gradient method (PCG) is an effective method that is already parallelised as a built-in function in Matlab to solve symmetric linear systems. An alternative might have been the biconjugate gradient stabilised method (BiCGSTAB), but this method is not monotonously converging. To find the most efficient combination of solver and preconditioners we compare PCG applied to the original system to the preconditioned one. We use the solution from the previous time step as initial value for the iterative solver and consider the relative residual norm at each iteration step, with and without preconditioning. For a detailed description of the PCG, we refer to Appendix B.1.

In Figure 4.16 we plot the relative residual at each iteration step of the PCG for the different preconditioners in the left and right hemisphere. Here, we solve system (3.11) at the very first time step for a propagation wave starting in the lateraloccipital region.



(a) left hemisphere (b) right hemisphere



(c) left hemisphere (d) right hemisphere

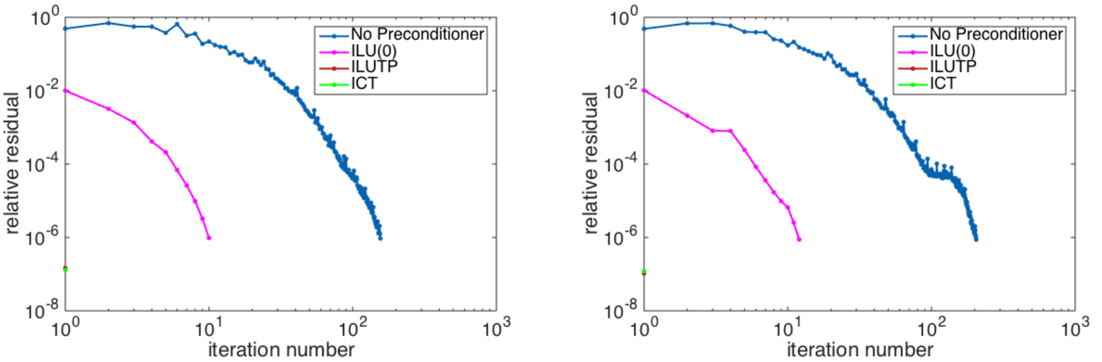
FIGURE 4.15: The smallest and biggest 500 eigenvalues of the original matrices, preconditioned with incomplete LU factorisation without fill-up (ILU(0)), with threshold and pivoting (ILUTP) and with incomplete Cholesky factorisation with a drop tolerance of 10^{-6} (ICT), for the left (a) and right (b) hemisphere. A close up showing the difference between ILUTP and ICT for the left (c) and right (d) hemisphere.

For a more detailed comparison we give the exact number of iterative steps and the relative residual in Table 4.1. Again, different preconditioners for the system in the left and right hemisphere are considered.

	left hemisphere		right hemisphere	
	iterations	rel. residual	iterations	rel. residual
no preconditioner	156	9.403e-07	205	8.6915e-07
ILU(0)	10	9.6725e-07	12	8.809e-07
ILUTP	1	1.4769e-07	1	1.0432e-07
ICT	1	1.2996e-07	1	1.2148e-07

TABLE 4.1: The iteration number and relative residual for the PCG with different preconditioners in the left and right hemisphere.

Since the coefficients do not change in time during the simulation, neither does the finite element matrix in (3.11). Thus, it is convenient to implement a low drop tolerance



(a) left hemishpere

(b) right hemishpere

FIGURE 4.16: The relative residual norm at each iteration step of the PCG with different preconditioning: ILU(0), ILUTP, and ICT for the left (a) and right (b) hemisphere.

in the ICT factorisation, as the same preconditioner will be used for a large number of time steps needed to simulate a whole CSD propagation. To investigate the impact of the drop tolerance of the ICT, we consider as initial condition a fully depolarised lateraloccipital region, and we run the simulation until the whole cortex has been activated. In Figure 4.17 we give an overview of the iteration number and the relative residual for the different drop tolerances of the ICT and consider the average number of iterations and the average relative residual over time for the left and right hemisphere.

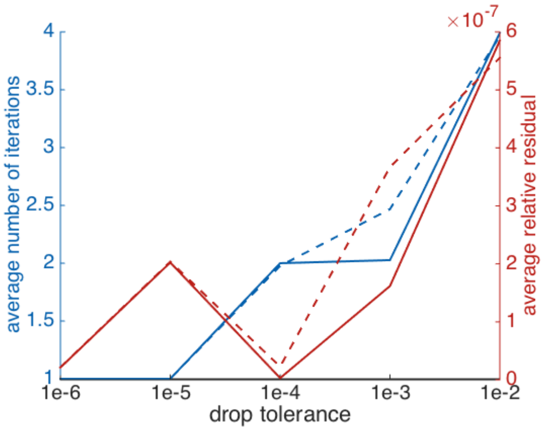


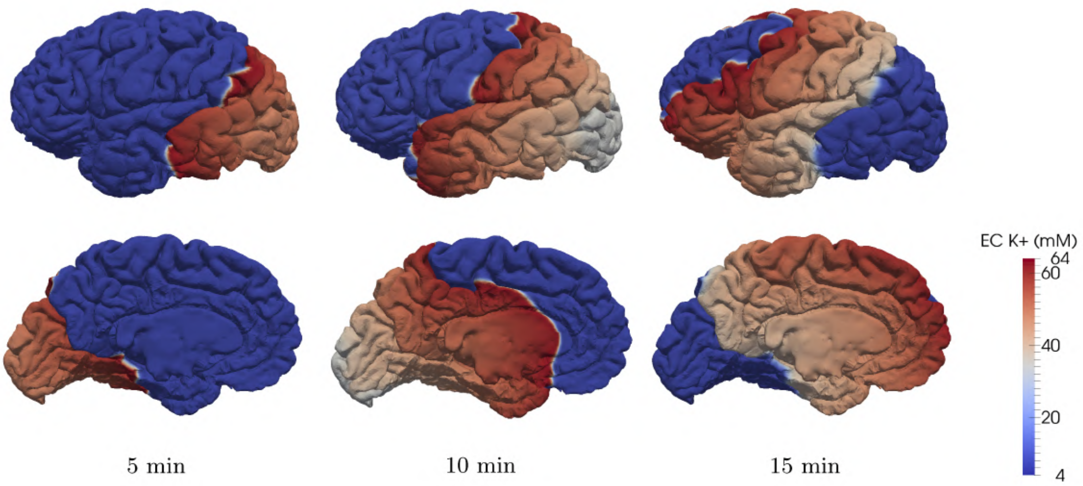
FIGURE 4.17: The average number of iterations (blue) and the relative residual (red) for the ICT with different drop tolerances, solved with PCG with the initially activated lateraloccipital in the left (solid line) and right (dashed line) hemisphere.

We also collect in Table 4.2 the average iteration number and average relative residual for both hemispheres for different values of the drop tolerance in ICT. The combined effect of the low drop tolerance in the ICT factorisation, and the propagative nature of the solution of system (3.4) entails that choosing as initial guess (at time t^{n+1}) the solution at the previous time step $\boldsymbol{k}^n$, PCG converges in very few iterations. Moreover, the single iteration needed in most cases for the 10^{-6} drop tolerance advocates for the use of more accurate preconditioning for the CSD simulation.

	left hemisphere		right hemisphere	
drop tol.	iterations	rel. residual	iterations	rel. residual
10^{-6}	1	2.1160e-8	1	2.1680e-8
10^{-5}	1.0005	2.0239e-7	1.0005	2.0378e-7
10^{-4}	2	3.0384e-9	1.9768	2.2775e-8
10^{-3}	2.0266	1.6174e-7	2.4679	3.6689e-7
10^{-2}	3.9924	5.8514e-7	3.9628	5.5549e-7

TABLE 4.2: Average number of iterations and average relative residual for different drop tolerances in the preconditioner computation for the left and right hemisphere.

To conclude, we solve system (3.11) with PCG, preconditioned by ICT, with pivoting and drop tolerance set at 10^{-6}. We consider as initial condition a fully depolarised lateraloccipital region, which corresponds to the area of the visual cortex where CSD typically is generated, and we run the simulation with the diffusion coefficients described in Section 4.2.3 until every point of the cortical mesh has been reached by the potassium wave. The scaling coefficient $\delta = 0.18\,\text{mm}^2\,\text{sec}^{-1}$ for the diffusion (4.12) has been tuned to obtain a propagation time of around 20 minutes across the whole cortex, in agreement with the physics of CSD. In Figure 4.18 we plot the potassium wavefronts on the left hemisphere at times $t = 5$ min, $t = 10$ min and $t = 15$ min.



5 min

10 min

15 min

FIGURE 4.18: Lateral and medial views of extracellular potassium propagation dynamics on the left hemisphere with DTI-based diffusion at time $t = 5\,\text{min}$, $t = 10\,\text{min}$ and $t = 15\,\text{min}$.

4.3.2 Impact of tensor anisotropies on the CSD

In this series of tests, to assess the effect of the anisotropy in the coefficients on the progression of CSD across the cortex, we compare the DTI-based diffusion tensors with the isotropic diffusion and a mixed isotropic diffusion tensor. The latter represents the case in which the diffusion imaging data only provides local diffusion coefficients, like the ADC, and not a full diffusion tensor. We define the diffusion tensors in 2D locally on each element K of the triangulation as:

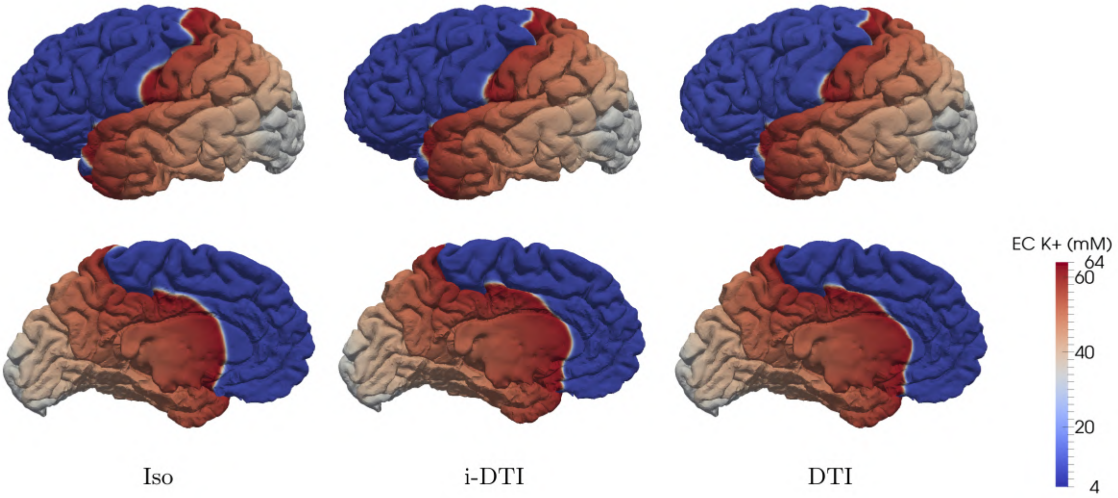
1. **Isotropic diffusion (Iso):** For purely isotropic diffusion, $\mathrm{FA}_K = 0$. The diffusion tensor in a generic point P_j is given by
$$D_j = \delta \mathbb{I}_2.$$
2. **DTI-based isotropic diffusion (i-DTI):** We consider $\mu_j^+ = \mu_j^-$ in the diffusion tensor definition (4.12). This means that $\mathrm{MD}_j = \mu_j^+$ and $\mathrm{FA}_K = 0$. Thus, the diffusion coefficient is locally isotropic, but changes in space. In a generic point P_j the diffusion tensor reads
$$D_j = \delta \frac{\mu_j^+}{\mathrm{MD}_{mean}} \mathbb{I}_2.$$
3. **DTI-based anisotropic diffusion (DTI):** We consider the diffusion tensor introduced in equation (4.12). Thus, in the generic point P_j, it reads
$$D_j = \frac{\delta}{\mathrm{MD}_{mean}} \left(\mu_j^+ \boldsymbol{p}_j \boldsymbol{p}_j^T + \mu_j^- \boldsymbol{q}_j \boldsymbol{q}_j^T \right).$$

Using the same scaling coefficient $\delta = 0.18\,\mathrm{mm}^2\,\mathrm{sec}^{-1}$ for all three simulations, we consider as initial condition a fully depolarised lateraloccipital region and run the simulations until every node of the computational grid, i.e., the whole cortex, has been reached by the CSD wave. From Section 4.2.4 we expect differences in both propagation times and wavefront patterns. In Table 4.3 we collect the different activation times, the moment in time in which the last point is reached by the CSD wave, for the three different types of diffusion tensors and both hemispheres. Compared to the isotropic diffusion types, the DTI-based anisotropic coefficients feature a slower wave propagation on the cortex, whereas the activation time with DTI-based isotropic diffusion is comparable to the isotropic diffusion case.

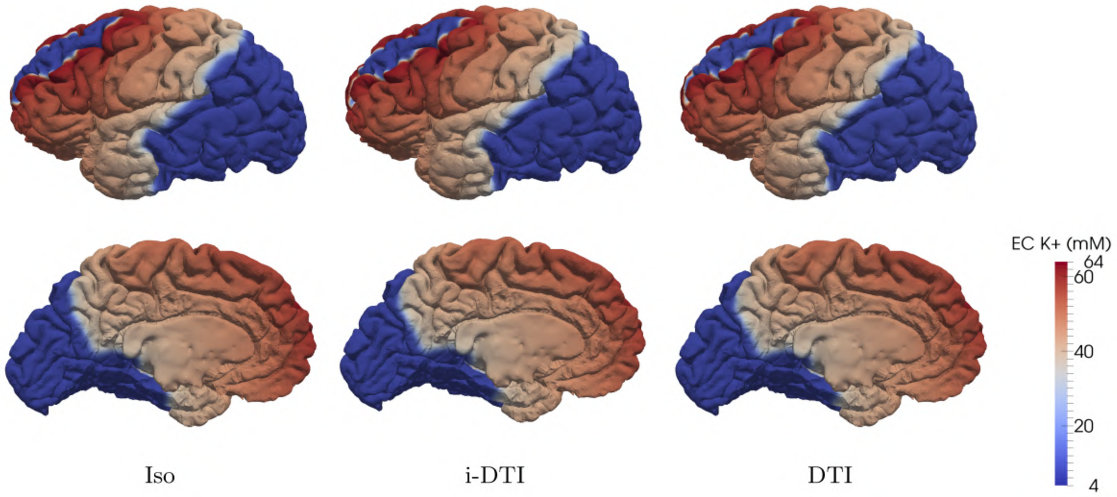
	Iso	**i-DTI**	**DTI**
left hemisphere	18.29	18.34	18.39
right hemisphere	18.52	18.44	18.55

TABLE 4.3: The total activation times (in minutes) for a CSD wave originating from the lateraloccipital region for the 3 diffusion tensors, and both hemispheres.

Moreover, not only the final activation times are affected by the different diffusion types, but also the local propagation speed and direction are different when considering DTI data, resulting in different depolarisation wavefronts. In Figure 4.19 we plot the depolarisation wavefront on the left hemisphere originating from the lateraloccipital at times $t = 10$ min and $t = 16$ min for the different diffusion types. Even though the overall propagation speed differs only slightly, up to 0.1 minutes in activation time delay, the wavefront propagation is different for the three cases of diffusion considered.



(a) $t = 10$ min



(b) $t = 16$ min

FIGURE 4.19: Lateral and medial view of the CSD wave propagation via the extracellular potassium concentration (EC K^+) of the left hemisphere at time $t = 10$ min (a) and $t = 16$ min (b), for the three different choices of diffusion tensor.

In Figure 4.20 and Figure 4.21 we show the difference in the activation times between the isotropic and the DTI-based conductivity (Iso - DTI) and between the isotropic DTI and the DTI-based conductivity (i-DTI - DTI). Discrepancies in the activation patterns are not just visible between the different diffusion tensors, but also between the left and right hemisphere. Some areas feature similar activation times, especially in the proximity of the initial region of the depolarisation wave.

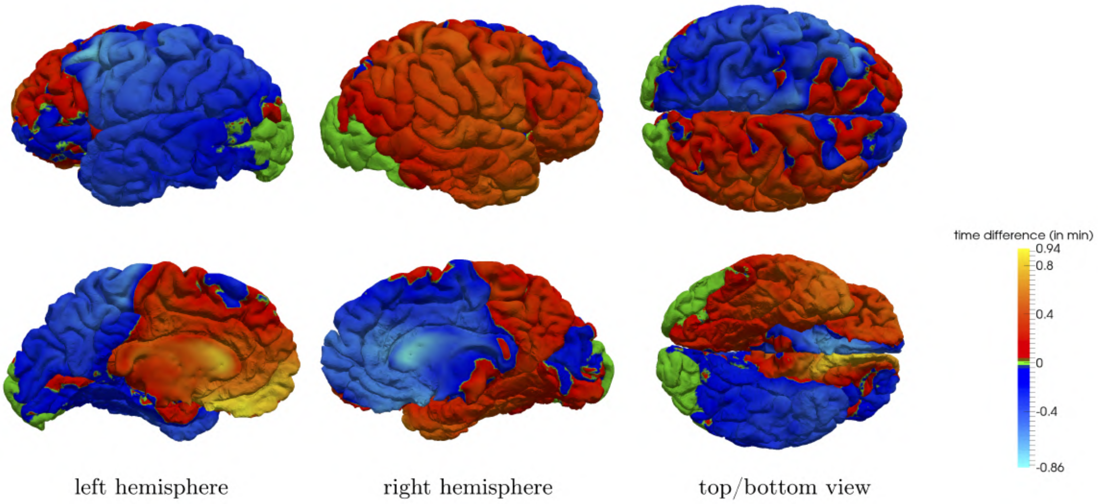


FIGURE 4.20: The difference between the activation times (in minutes) for the isotropic and the DTI-based diffusion tensor originating in the lateraloccipital region for both hemispheres.

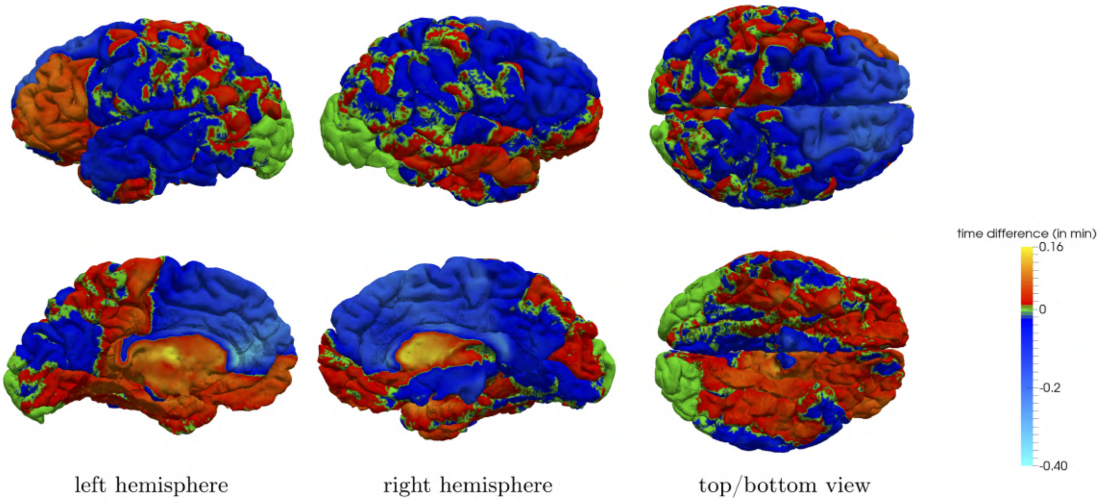


FIGURE 4.21: The difference between the activation times (in minutes) for the isotropic DTI and the DTI-based diffusion tensor originating in the lateraloccipital region for both hemispheres.

4.4 Summary

In this chapter we simulated a CSD wave on a real brain geometry obtained from MRI data and were able to reproduce the propagation of the depolarisation wave triggering the microscopic dynamics of CSD that have been observed in experiments. After studying the coupled model and observing the CSD behaviour on the microscopic scale, we only focused on the propagation of the extracellular potassium, that is, the macroscopic part of the model, which already provides the essential information. From DTI data, we derived patient-specific conductivity tensors which provide detailed information about the anisotropy and the electrical and ionic conductivity properties of the cortical tissue. In addition to the cortical geometry reconstructed from MRI imaging, diffusion coefficients based on DTI data bring the personalisation of the model to a higher level.

The algorithm to reduce the 3D DTI data to 2D diffusion data on the cortical surface was described in detail: the diffusion tensors were computed triangle by triangle in a finite elements numerical approximation setting and the ellipsoids representing the diffusion in 3D were intersected with the plane identified by the considered triangle. Numerical results showed differences in propagation time with respect to an isotropic diffusion that was assumed so far. The main contribution of the proposed model resides in the use of a more detailed description of the tissue diffusivity: numerical simulations are expected to be a closer match to CSD propagation, especially given Figure 4.13, Figure 4.20 and Figure 4.21, at a minimal additional computational effort.

The computational load of the proposed model is comparable with the one of the model introduced in Chapter 3. The only additional cost is associated with the pre-processing of the DTI data to compute the diffusion tensors in 2D, a procedure to be performed just once for any given dataset, not affecting the actual computational time of the propagation: the number of linear systems to be solved is the same as in the propagative part of the model in Chapter 3, with no additional computational issue. However, to reduce the computational load and speed up the numerical approximation we briefly elaborated on possible preconditioners and the iterative solver.

Here, we just focused on the potassium dynamics, disregarding the electrophysiological dynamics at the microscopic scale. However, as these two parts of the model are coupled unidirectionally, the electrophysiology can be quickly reproduced from the results of the propagating potassium wave on the cerebral cortex.

The results of this chapter have been published in two papers, in Kroos et al., 2016 and Kroos et al., 2017.

Chapter 5

Cortical assessment

"Mathematics is biology's next microscope, only better;
biology is mathematics' next physics, only better."

J.R. Cohen, in *PLoS Biology (2004)*

After describing a way to personalise the CSD model, it is essential to assess the cortical geometry and the CSD propagation behaviour on the cortex from Chapter 3 for later group analysis. Our aim is to describe the unique structure of sulci and gyri of the human cortex and CSD propagation in terms of quantities that can be compared between subjects.

Cortical curvature appears to be a specific and quantitative marker for white matter volume irregularities. Fissures and sulci that consistently appear in all human brains show variability in size and configuration (Thompson et al., 1996). Some studies support the hypothesis that cortical curvature abnormalities may indicate white matter atrophy, a characteristic symptom in early multiple sclerosis (Deppe et al., 2014), or subtle malformations of the cortex linked to seizure in temporal lobe epilepsy (Ronan et al., 2010).

To quantify the surface convolution of the cortex, we estimate the local curvature and compute a surface regularity index. The curvature of a geometry is a differential invariant property that provides information about the direction of the bending of the surface and how far the bending deviates from the plane, while the surface regularity index is the ratio between the given volume and the volume of a sphere with the same surface size (Pérez-Beteta et al., 2018).

Some recent studies started to consider the effect of cortical geometries on CSD propagation, at different levels of detail. Gyri and sulci affect the propagation of depolarisation waves and can stop the CSD wave in different positions depending on the patient. Pocci et al., 2010 studied the effect of the cortical bending by simulating a wave propagation in a 2D duct containing a bend. They show how sharp bends naturally block the wave propagation. Above a critical radius, blocking can be achieved by changing the system parameters, assuming that adapted therapeutic agents can stop migraine aura. In a personalised approach to migraine aura treatment, Dahlem et al., 2015 propose to use the Gaussian curvature of the cortex to identify potential targets for neuromodulation. Applying a generic reaction-diffusion model they highlight the local effects of the curvature on a simple 2D geometry including a bump.

In this chapter the objective is to study the wave propagation on a whole 3D individual geometry to identify symmetries and asymmetries in its behaviour and analyse regions to assess their potential to play a crucial role in CSD episodes. Introducing quantities of interest (QoI) that can be computed by post-processing the results from the simulation of the CSD propagation across the whole cortex, makes the CSD propagation comparable for different patients.

In the following sections we introduce a smoothing and a curvature approximation technique for triangulated meshes, define the surface regularity index and derive computational QoI that identify symmetries and asymmetries in the propagation of the depolarisation wave. We conclude the chapter with an overview of the resulting QoI for the patient-specific brain geometry and CSD propagation based on the DTI-data discussed in Chapter 4.

5.1 Smoothing and curvature

Data acquisition techniques for medical imaging such as MRI can feature artefacts and noisy data. Figure 5.1 shows fragments of a brain cortex reconstructed from medical imaging that suffers from these artefacts. These artefacts can be treated during the surface extraction process or at the mesh generation phase. The main objective of mesh smoothing algorithms is to remove undesirable noise and uneven edges while retaining characteristic geometric features.

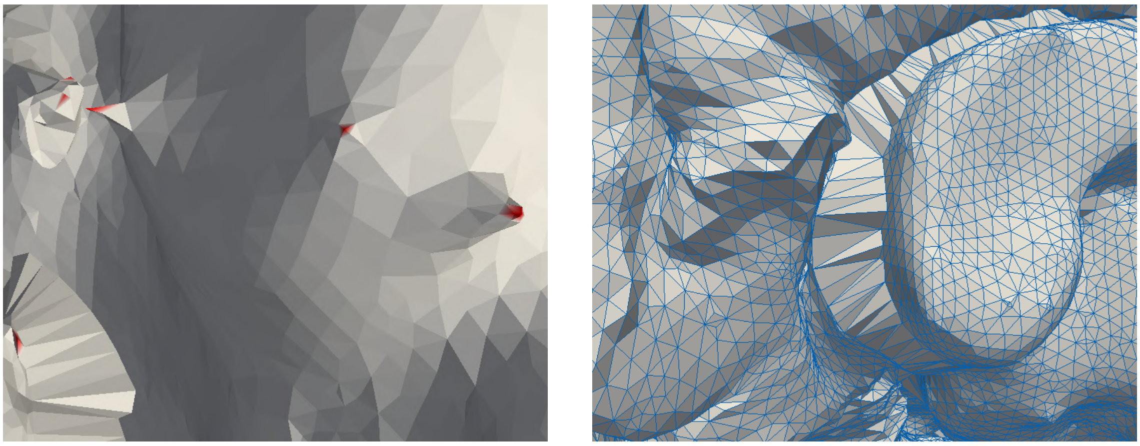


FIGURE 5.1: Areas with artefacts and critical mesh triangulation of the brain geometry.

A multitude of smoothing algorithms and techniques for mesh simplification and improvement have been proposed (Bossen and Heckbert, 1996; Dassi et al., 2015; Semenova, Savchenko, and Hagiwara, 2004). The work of Bade, Haase, and Preim, 2006 is dedicated to finding optimal smoothing techniques and parameters for medical images, with a focus on geometries that can be classified as compact, flat or elongated. However, the complex shape of the cerebral cortex does not permit the classification in one of these three groups. When approximating the curvature of a discrete mesh, artefacts and extreme mesh configurations can feature extreme curvature values. To overcome this issue, we smooth the cortical mesh obtained from FreeSurfer to eliminate artefacts and improve the mesh quality before approximating the curvature of the discrete mesh.

The code for the smoothing and curvature approximation that we introduce in this section was developed in collaboration with Franco Dassi from the University of Milan-Biccoca in Italy.

5.1.1 Data smoothing techniques

One of the most popular techniques to smoothen curves and surfaces is *Gaussian smoothing*. In the continuous case, a convolution of the vector function parameterising the curve is performed with a Gauss kernel. For meshed surfaces, Gaussian smoothing has to be formulated in a discrete manner. In the following, we define this smoothing for discrete surfaces as well as the algorithm proposed by Gabriel Taubin to prevent shrinkage caused by the original Gaussian smoothing (Taubin, 1995).

Gaussian smoothing is an iterative algorithm in which at every step the smoothed position of a mesh point is defined by a weighted average of its current position and its direct neighbours without changing the mesh topology. The direct neighbours of a mesh vertex are all vertices that share an edge of the mesh with the considered vertex. Let $\boldsymbol{p} \in P$ be a generic point in a set of points describing a mesh Γ_h, where h refers to the maximum diameter of the elements of the triangulation. Then a vertex $\boldsymbol{p}_i \in P$ is considered an immediate or direct neighbour of $\boldsymbol{p}$ if the edge $\overline{\boldsymbol{p}\boldsymbol{p}_i}$ is part of the given mesh Γ_h. In the Gaussian smoothing, the position of $\boldsymbol{p}$ is replaced in the smoothed mesh by

$$\boldsymbol{p}_{new} = \boldsymbol{p} + \frac{\lambda}{c} \sum_{\boldsymbol{p}_i \in P^*} w(\boldsymbol{p}, \boldsymbol{p}_i)(\boldsymbol{p} - \boldsymbol{p}_i),$$

where $\lambda \in (0, 1]$ is a scaling factor, P^* is the set of neighbouring vertices, w is a positive weight function with $\sum_{\boldsymbol{p}_i \in P^*} w(\boldsymbol{p}, \boldsymbol{p}_i) = 1$ and c is a normalisation factor. The different ways to select the weight function w are discussed below.

Weight functions The weight functions can be chosen by taking into account the neighbourhood structure, the specific variations in the edge lengths and the face angles. The equal weights proposed by Taubin provide good smoothing results for meshes with small variations in the edge length and face angles across the mesh. For more irregular meshes, like the ones obtained from medical imaging, other weights can be better suited to compensate for the irregular triangulation. Several choices of weight functions w have been proposed in Taubin, 2000:

- *equal weights:* all edges connected to $\boldsymbol{p}$ are weighted with the same constant value, i.e.,

  $$w(\boldsymbol{p}, \boldsymbol{p}_i) := \frac{1}{|P^*|}, \tag{5.1}$$

  where $|\cdot|$ denotes the cardinality of the set.

- *Fujiwara weights:* the edges are weighted according to the inverse of the edge length, i.e.,

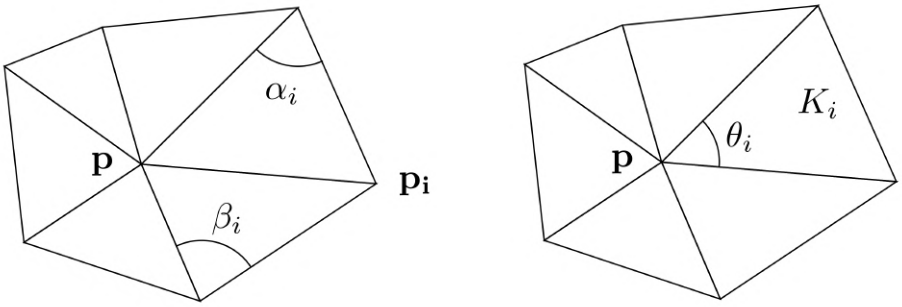
  $$w(\boldsymbol{p}, \boldsymbol{p}_i) := \frac{1}{||\overline{\boldsymbol{p}\boldsymbol{p}_i}||}, \tag{5.2}$$

  where $||\cdot||$ is the Euclidean norm giving the length of the edge $\overline{\boldsymbol{p}\boldsymbol{p}_i}$.

- *Desbrun weights:* the edges are weighed by the adjacent angles

$$w(\boldsymbol{p},\, \boldsymbol{p}_i) := \cot(\alpha_i) + \cot(\beta_i)\,, \tag{5.3}$$

where the angles α_i and β_i are the angle opposite to the edge $\overline{\boldsymbol{p}\boldsymbol{p}_i}$ in the two adjacent triangles sharing the edge $\overline{\boldsymbol{p}\boldsymbol{p}_i}$, as shown in Figure 5.2.



(a) angles opposite the edge $\overline{\boldsymbol{p}\boldsymbol{p}_i}$ (b) angle at $\boldsymbol{p}$

FIGURE 5.2: The notations for the angles opposite the edge $\overline{\boldsymbol{p}\boldsymbol{p}_i}$ (a) and for the angle at point $\boldsymbol{p}$ (b).

The resulting smoothed mesh depends on the choice of the weights. Applying the smoothing technique reduces the triangles' stretch and transforms them into more equilateral triangles. To measure the performance of the smoothing algorithm, we introduce a quality measure based on the triangles' stretch.

Mesh quality measure To derive a global indicator for the mesh quality, we start by considering the aspect ratio defined for a generic triangle K as

$$q(K) := 2\frac{r_K}{R_K},$$

where r_K and R_K are the radii of the inscribed and circumscribed circle, respectively (Shewchuk, 2002). For an equilateral triangle, this measure yields 1, while $q(K)$ close to zero means that triangle K is highly stretched. We then compute the mean of $q(K)$ over all the triangles to obtain a single quantity characterising the mesh quality, namely

$$q_{\text{mean}} := \frac{1}{|\Gamma_h|} \sum_{K \in \Gamma_h} q(K)\,, \tag{5.4}$$

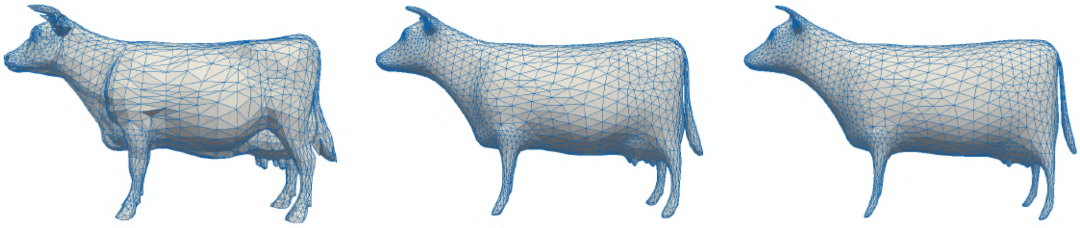
where $|\Gamma_h|$ is the number of triangles in the surface Γ_h. In the following, we use q_{mean} as a proxy for mesh quality.

To illustrate the effect of smoothing in a particular example, we apply Gaussian smoothing with equal weights and $\lambda = 0.33$ to a cow geometry[1] with 2903 nodes and 5804 triangles. The original geometry, the results of $N = 10$ and $N = 20$ iterative steps of the smoothing algorithm and the corresponding mesh qualities are shown in Figure 5.3. This comparison highlights the significant loss of volume in the fine features of the geometry (like the legs) with an increasing number of iterations. This is known as the *shrinkage*

---

[1] from the INRIA Gamma mesh repository

effect associated to Gaussian smoothing. The mesh quality is not necessarily increasing with the iteration number. After 10 smoothing iterations the mesh quality of the cow geometry increases but after 10 more iterations the mesh quality is decreases. However, there is still an improvement in the mesh quality after 20 smoothing steps compared to the initial mesh.



(a) original mesh, $q_{mean} = 0.7339$ (b) $N = 10$, $q_{mean} = 0.8046$ (c) $N = 20$, $q_{mean} = 0.7734$

FIGURE 5.3: The Gaussian smoothing with equal weights applied to the original geometry (a) with 2903 nodes and 5804 triangles with the scaling factor $\lambda = 0.33$ and $N = 10$ (b) and $N = 20$ (c) iterations.

Taubin smoothing To prevent shrinkage, the Taubin algorithm consists of two consecutive Gaussian smoothing steps with positive and negative scaling factors (Taubin, 1995). After a first Gaussian smoothing step with positive scaling λ is used, a second Gaussian smoothing step is performed with negative scaling μ. These two values are chosen such that $0 < \lambda < -\mu$, and to produce a significant smoothing effect on the final mesh these two steps have to be repeated iteratively. The precise scheme is given in Algorithm 5.1.

Algorithm 5.1 Taubin smoothing scheme

```
$\Gamma_h$ = SMOOTHING ($\Gamma_h^{init}$, $\lambda$, $\mu$, $w$, $N$)
initialise: $\Gamma_h = \Gamma_h^{init}$
for $i = 1, \dots, N$ do
    $\Gamma_h$ = Gaussian smoothing with $\lambda$ and $w$ on $\Gamma_h$
    $\Gamma_h$ = Gaussian smoothing with $\mu$ and $w$ on $\Gamma_h$
end for
```

To quantify the shrinkage caused by the smoothing algorithm, we consider a fast way to compute the volume enclosed by a triangular mesh.

Volume computation for triangulated surfaces We start by redefining a generic volume $V \subset \mathbb{R}^3$ as

$$\int_V 1 \, dV.$$

Let V be a bounded open set (here in $\mathbb{R}^3$) with smooth or piecewise smooth enclosing surface S (or, in general, boundary ∂V) and $\mathbf{F}$ be a continuously differentiable vector field up to the boundary. Then the divergence theorem relates the flux of the vector field through a surface enclosing V to the behaviour of the vector field inside the volume, namely

$$\int_V \nabla \cdot \mathbf{F}\, dV = \int_S \mathbf{F} \cdot \mathbf{n}\, dA. \tag{5.5}$$

where $\nabla \cdot$ is the divergence operator and $\mathbf{n}$ the outward pointing normal of the surface S (Kellogg, 1929). For volumes enclosed by a triangulated surface, the right hand side of (5.5) can be split into a summation over triangular faces on which the normal $\mathbf{n}$ to the surface remains constant. Choosing a vector field $\mathbf{F} = (x, 0, 0)^T$ whose divergence is equal to 1, we can hence write

$$\int_V 1\, dV = \int_V \nabla \cdot \mathbf{F}\, dV = \int_S \mathbf{F} \cdot \mathbf{n}\, dA = \sum_{K \in \Gamma_h} \int_K \mathbf{F} \cdot \mathbf{n}\, dA \tag{5.6}$$

for a set of triangles Γ_h describing the surface mesh and a normal vector $\mathbf{n} = (n_1, n_2, n_3)^T$. Considering the triangle K with vertices $Q = (q_1, q_2, q_3)^T$, $R = (r_1, r_2, r_3)^T$ and $S = (s_1, s_2, s_3)^T$, every point P in the triangle K can be expressed as

$$P = (1 - u - v)Q + uR + vS, \qquad \text{with } u \in [0, 1], v \in [0, 1 - u]$$

using barycentric coordinates. If $\mathbf{F}$ is linear, then

$$F(P) = (1 - u - v)F(Q) + uF(R) + vF(S)$$

and in our particular case with the definition of the vector field $\mathbf{F}$, we have $F(P) = ((1 - u - v)q_1 + ur_1 + vs_1, 0, 0)^T$. Using the triangle parametrisation, we can rewrite the surface integral for the vector field $\mathbf{F} = (x, 0, 0)^T$ as

$$\int_K \mathbf{F} \cdot \mathbf{n}\, dA = 2n_1|K| \int_0^1 \int_0^{1-u} (1 - u - v)q_1 + ur_1 + vs_1\, dv\, du, \tag{5.7}$$

where $|K|$ is the area of the triangle with

$$|K| = \frac{||\hat{\mathbf{n}}||}{2} \quad \text{and} \quad \hat{\mathbf{n}} = (R - Q) \times (S - Q).$$

Using the fact that q_1, r_1 and s_1 are constants and performing simple integration one can show that

$$\int_0^1 \int_0^{1-u} (1 - u - v)q_1 + ur_1 + vs_1\, dv\, du = \frac{q_1 + r_1 + s_1}{6}.$$

Hence, equation (5.7) simplifies to

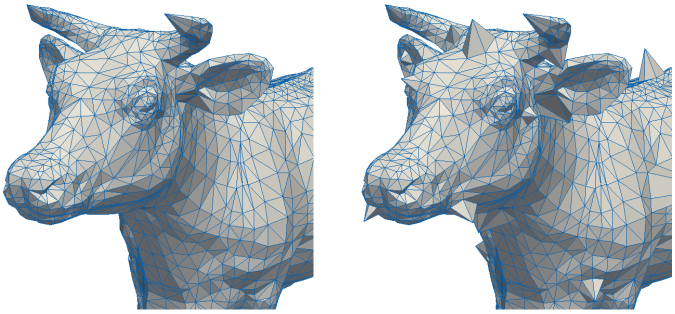
$$\int_K \mathbf{F} \cdot \mathbf{n}\, dA = \frac{n_1 ||\hat{\mathbf{n}}||}{2} \frac{q_1 + r_1 + s_1}{3},$$

where the last fraction is the x-component of the barycentre $\frac{Q+R+S}{3}$ of triangle K. Consequently, the volume enclosed by a triangulated surface can be summarised as

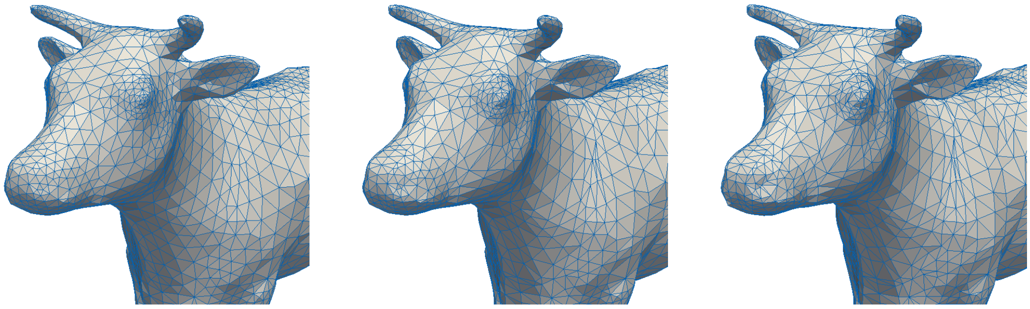
$$\frac{1}{2} \sum_{K \in \Gamma_h} n_1 ||\hat{\mathbf{n}}|| \frac{q_1 + r_1 + s_1}{3}.$$

Meshes with noise To evaluate the smoothing algorithm, the different weight functions and the impact of the scaling factor, we consider again the example of the cow geometry.

As medical images usually include noise and artefacts, we artificially add noise to the cow geometry by moving some points along the normal to the surface, see Figure 5.4. We apply the smoothing procedure described in Algorithm 5.1 for the different weight functions and show the results for $\lambda = 0.33$ and $\mu = -0.34$ in Figure 5.4.



(a) original (b) artificially added noise



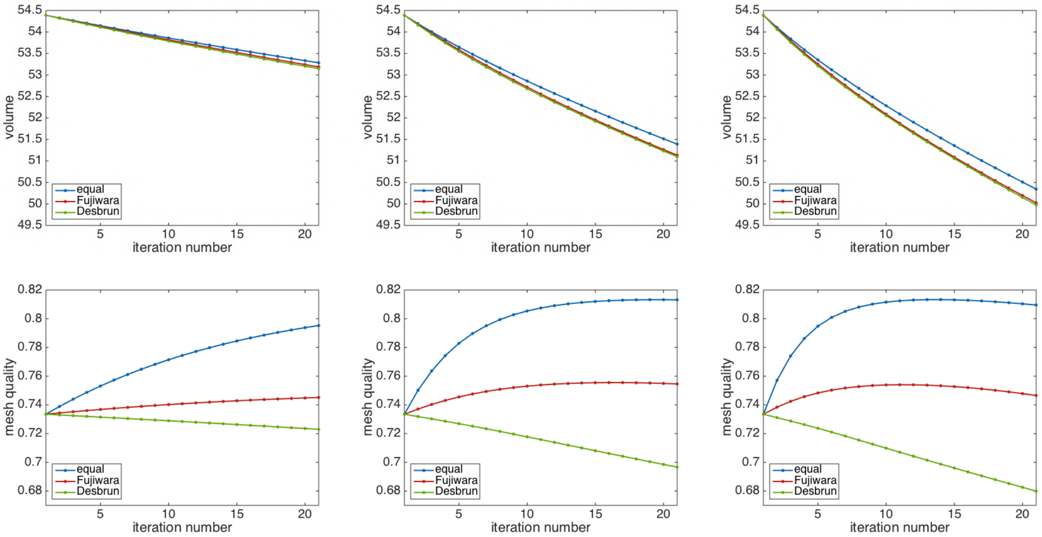
(c) equal weights (d) Fujiwara weights (e) Desbrun weights

FIGURE 5.4: The original mesh of the cow geometry (a), with artificially added noise (b) and the results of Taubin smoothing with equal (c), Fujiwara (d) and Desbrun (e) weights after $N = 20$ iterations and with $\lambda = 0.33$ and $\mu = -0.34$.

To evaluate the impact of the smoothing weights and scaling factors on the mesh, we plot the volume and the (mean) mesh quality (5.4) of the cow geometry at each iteration step for the different weight functions in Figure 5.5. Here, we choose different values for the scaling factors λ and μ but keep their ratio as proposed by Taubin, Zhang, and Golub, 1996.

All the weight functions cause a similar loss of volume and the noise is removed (see Figure 5.4). However, the equal weights perform better when considering the mesh quality. In the beginning, both the equal and Fujiwara weights improve the mesh quality significantly but this improvement levels off with the number of iterations. This is a numerical evidence that the smoothing at some point cannot improve the mesh quality anymore. Consequently, other mesh operations such as edge flipping, edge contraction and point insertion have to be included to achieve further improvement (Frey, 2004; Frey and Borouchaki, 1998, 2003; Geuzaine and Remacle, 2009) but this goes beyond the scope of this thesis.

Since the Taubin smoothing is based on the Gaussian one, there is always a shrinkage in the triangular surface mesh. A standard stopping criterion for the iterative smoothing scheme is the maximum number of iterations. However, an appropriate choice of the



(a) $\lambda = 0.2, \mu = -0.21$ (b) $\lambda = 0.33, \mu = -0.34$ (c) $\lambda = 0.4, \mu = -0.42$

FIGURE 5.5: Volume and mesh quality of the cow geometry after applying the Taubin smoothing with equal, Fujiwara and Desbrun weights at each iteration step for scaling parameters $\lambda = 0.2$ and $\mu = -0.21$ (a), $\lambda = 0.33$ and $\mu = -0.34$ (b) and $\lambda = 0.4$ and $\mu = -0.42$ (c).

iteration number and the scaling factor is important for the quality of the final smoothing result. To avoid choosing the number of iterations, we adopt an alternative stopping criterion in line 2 of Algorithm 5.1, and restrict it by the volume loss. Therefore, we assume that the discrete mesh Γ_h encloses a volume V_i after i iterations. We apply the Taubin smoothing until the percentage of volume loss exceeds the threshold V_{tol}, that is, we keep applying the smoothing as long as

$$\frac{|V_i - V_0|}{V_0} < V_{tol},$$

where V_0 is the volume enclosed by the initial surface mesh. This way the stopping criterion is a trade-off between the volume reduction and the smoothness. The Taubin smoothing scheme with the new stopping criterion is given in Algorithm 5.2.

Algorithm 5.2 Taubin smoothing with volume restriction

```
$\Gamma_h$ = SMOOTHING ( $\Gamma_h^{init}$, $\lambda$, $\mu$, $w$, $V_{tol}$ )
$V_0$ = volume of $\Gamma_h^{init}$
initialise: $\Gamma_h = \Gamma_h^{init}$, $V = V_0$
while $\left(\frac{|V-V_0|}{V_0} < V_{tol}\right)$ do
    $\Gamma_h$ = Gaussian smoothing with $\lambda$ and $w$ on $\Gamma_h$
    $\Gamma_h$ = Gaussian smoothing with $\mu$ and $w$ on $\Gamma_h$
    $V$ = volume of $\Gamma_h$
end while
```

Applying this smoothing algorithm to the example of the cow with a maximum volume loss of 5% we present the number of iterations and the mesh quality of the final mesh in Table 5.1. In this example the Taubin smoothing with equal weights results in a mesh with the best quality.

	initial mesh	Taubin smoothing		
weight function		equal	Fujiwara	Desbrun
iterations		28	26	26
mesh quality	0.7335	0.8107	0.7517	0.6873

TABLE 5.1: The iteration number and mesh quality of the noisy cow geometry after applying the Taubin smoothing with equal, Fujiwara and Desbrun weights with $\lambda = 0.33$, $\mu = -0.34$ and $V_{tol} = 0.05$.

5.1.2 Discrete curvature approximation

In 2D curvature can be defined in several equivalent ways: as the deviation of the curve from the tangent line, as the rate of change of the tangent direction, or as the reciprocal of the radius of the osculating circle. For surfaces in 3D the definition of curvature is more complicated. In analogy to the 2D definition, the curvature should reproduce the deviation of the surface from the tangent plane but in contrast to planar curves, there are several ways to measure this deviation. *Gaussian curvature* and *mean curvature* are two approaches to estimate this and to give precise information about the shape of the considered surface. Gaussian curvature is defined as the product of the principal curvatures, while the mean curvature is defined as the average of the principal curvatures.

Differential geometry properties like Gaussian and mean curvature are well-known for parametrically defined curves and surfaces in the classical literature, and they provide essential and invariant information about a geometry (Do Carmo, 1976; Goldman, 2005; Kühnel, 2006). Considering a sufficiently regular surface Γ and the normal $\boldsymbol{n}$ in a generic point $\boldsymbol{p} \in \Gamma$, we can construct a set of normal planes at the point $\boldsymbol{p}$ which contain the normal vector $\boldsymbol{n}$ and the point $\boldsymbol{p}$. All these planes cut the surface Γ in different plane curves that are characterised by a distinct curvature behaviour. We denote by k_1 and k_2 the principal curvatures, i.e., the maximum and minimum values of the curvature of all these curves in $\boldsymbol{p}$. Then, the Gaussian curvature K and the mean curvature H are defined as

$$K(\boldsymbol{p}) := k_1\, k_2, \qquad \text{and} \qquad H(\boldsymbol{p}) := (k_1 + k_2)/2.$$

For a summary of the main definitions and results in differential geometry to clarify the concept of Gaussian and mean curvature of surfaces in the framework of parametric curves, we refer to Appendix B.2.

For an analytic evaluation of the curvature, a parametrisation of the surface Γ is necessary. In the case of discrete surfaces, like the brain geometry reconstructed from MRI imaging, we do not have such parametrisation, but a triangulated mesh instead. Moreover, since we are dealing with a complex triangulated surface of a brain reconstruction, finding such parametrisation would require high computational effort and sophisticated techniques, see for instance Marchandise, Remacle, and Geuzaine, 2014.

Numerous approaches to estimate the curvature of 3D triangular meshes have been made and a comprehensive overview of Gaussian and mean curvature estimation methods with a subsequent comparison of the errors can be found in Meyer et al., 2003 and Magid, Soldea, and Rivlin, 2007.

The discrete Gaussian and mean curvature approximations at a generic point $\boldsymbol{p}$ of the mesh do not require any parametrisation of the surface. In fact, they are based on the computation of angles, areas and the distance to directly neighbouring points and read

$$\begin{aligned} K_h(\boldsymbol{p}) &= \frac{1}{A_{mixed}} \left(2\pi - \sum_{\boldsymbol{p}_i \in P^*} \theta_i \right), \\ H_h(\boldsymbol{p}) &= \frac{1}{2} \left\| \frac{1}{A_{mixed}} \sum_{\boldsymbol{p}_i \in P^*} (\cot \alpha_i + \cot \beta_i)(\boldsymbol{p} - \boldsymbol{p}_i) \right\|, \end{aligned} \tag{5.8}$$

where A_{mixed} is the discrete cell Voronoi area around $\boldsymbol{p}$, P^* is the set of neighbouring points $\boldsymbol{p}_i$, θ_i is the angle in $\boldsymbol{p}$ of the triangle K_i, α_i and β_i are the angles opposite to the edge $\overline{\boldsymbol{p}\boldsymbol{p}_i}$ (see Figure 5.2 for notations), and $||\cdot||$ is the standard Euclidean vector norm. For a deeper analysis of the above equations, we refer to Meyer et al., 2003.

Numerical validation of discrete curvature As an example with well-known curvature values we consider a sphere $\mathcal{S}$ with radius r. Its curvature is constant, so the Gaussian and the mean curvatures are constant and their values in a generic point $\boldsymbol{p} \in \mathcal{S}$ are

$$K(\boldsymbol{p}) = \left(\frac{1}{r}\right)^2 \quad \text{and} \quad H(\boldsymbol{p}) = \frac{1}{r}\,.$$

For a detailed derivation of these values we refer to the example in Appendix B.2. We choose as examples a structured and an unstructured mesh discretisation of the sphere with radius $r = 0.5$ consisting of 2402 nodes and 4800 triangles, and 2144 nodes and 4284 triangles, respectively. A parametrisable sphere of this radius would have

$$K(\boldsymbol{p}) = 4 \quad \text{and} \quad H(\boldsymbol{p}) = 2$$

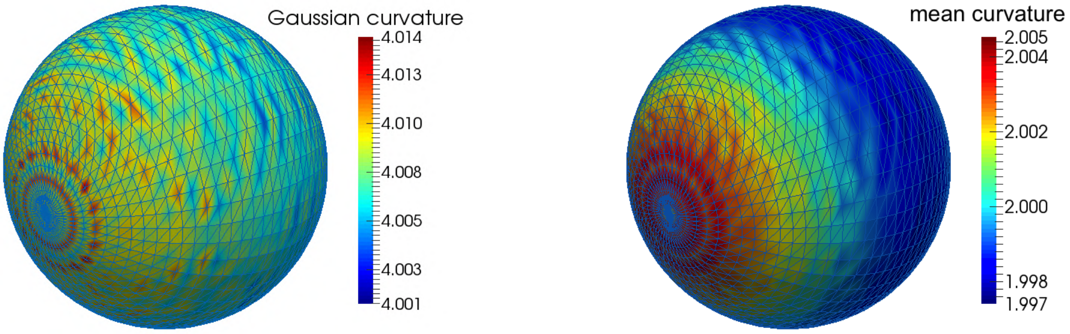
as Gaussian and mean curvature values at any point $\boldsymbol{p}$ on the surface of the sphere. In Figure 5.6 and Figure 5.7 we provide the discrete Gaussian and mean curvature approximations with (5.8) for the different triangulations of the sphere.

Although the estimated curvature values exhibit larger variations at the poles of the structured mesh and at points with only four neighbours in the unstructured mesh, both approximations provide overall reasonably good estimates for the curvature. To get a more complete view on the results we provide in Table 5.2 the mean value of both curvature measures and consider the relative errors defined as

$$e_K = \frac{\max_{\boldsymbol{p} \in \mathcal{S}} |K_h(\boldsymbol{p}) - 4|}{4} \quad \text{and} \quad e_H = \frac{\max_{\boldsymbol{p} \in \mathcal{S}} |H_h(\boldsymbol{p}) - 2|}{2}$$

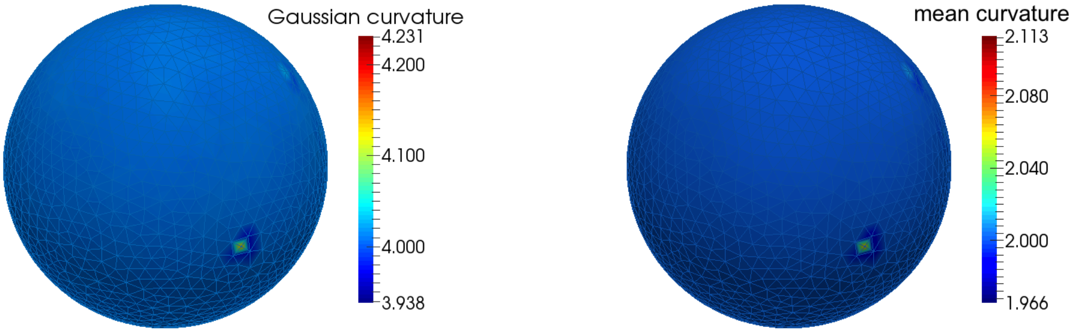
on the two mesh triangulations of the sphere.

From Table 5.2 it can be deduced that the discrete curvature approximation gives reasonable values in the case of the sphere with a low relative error. Even though larger errors occur in the unstructured mesh, these approximations are still good enough for our purpose.



(a) Gaussian curvature (b) mean curvature

FIGURE 5.6: The approximation of Gaussian curvature (a) and mean curvature (b) for a sphere of radius $r = 0.5$ with a structured mesh of 2402 nodes and 4800 triangles.



(a) Gaussian curvature (b) mean curvature

FIGURE 5.7: The approximation of Gaussian curvature (a) and mean curvature (b) for a sphere of radius $r = 0.5$ with an unstructured mesh of 2411 nodes and 4284 triangles.

	Gaussian curvature		mean curvature	
mesh	mean	e_K	mean	e_H
structured	4.008	3.61e-3	2.001	2.265e-3
unstructured	4.006	5.78e-2	2	5.654e-2

TABLE 5.2: The mean and the relative error of the discrete Gaussian and mean curvature of a sphere with radius $r = 0.5$ with a structured and with an unstructured mesh.

5.1.3 Numerical results on the brain

Sharp configurations in the cortex are the result of imaging and processing errors because the soft tissue of the brain is typically does not exhibit any steep formations. We apply the proposed smoothing technique to the brain geometry to remove noise and obtain a mesh that corresponds to the natural smoothness of tissue. Setting $\lambda = 0.33$, $\mu = -0.34$ and $V_{tol} = 0.05$, we consider the different weight functions introduced in Section 5.1.1. In Figure 5.8 we compare the initial mesh with the smoothed ones and focus on different parts of the left hemisphere. These areas highlight the noise removal, maintaining the complexity of the geometry. In the case of equal weights, the triangulation is evidently improved, converting stretched triangles of the initial mesh into more regular ones. This fact is numerically confirmed by the mesh quality, as it can be observed in Table 5.3.

	initial mesh	Taubin smoothing		
weight function		equal	Fujiwara	Desbrun
left hemisphere	0.7187	0.7452 (43)	0.6819 (51)	0.6696 (78)
right hemisphere	0.7241	0.7525 (44)	0.6867 (52)	0.6726 (81)

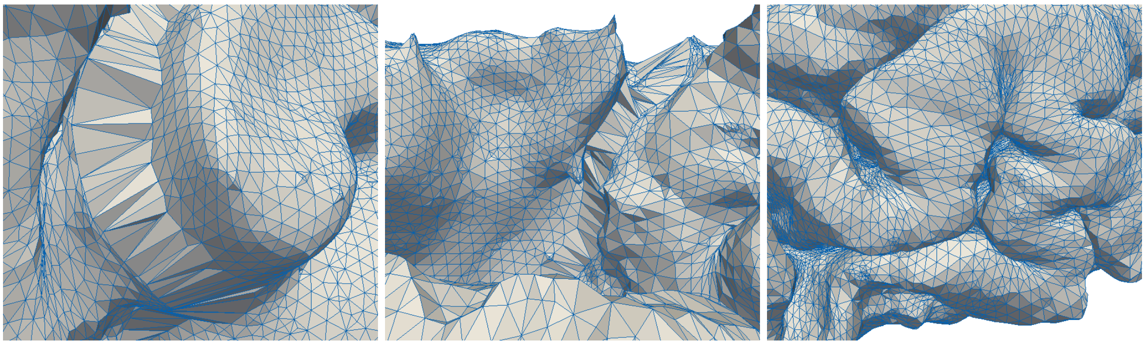
TABLE 5.3: Mesh quality and iteration number (in brackets) for Taubin smoothing with $\lambda = 0.33$, $\mu = -0.34$ and $V_{tol} = 0.05$ and equal, Fujiwara and Desbrun weights for the left and right hemisphere.

From now on, we only consider equal weights as they perform best on the cortex geometry regarding the mesh quality. To elaborate a bit more on the choice of scaling parameters, we set the maximum volume loss to 5% and plot the volume loss and mesh quality for different parameters λ and μ in Figure 5.9. Choosing small parameters increases the iteration number to reach the maximum volume loss and obtain a specific mesh quality, significantly. For all considered choices of scaling parameters the smoothing algorithm constantly improves the mesh quality and at the same time decreases the volume, for both hemispheres. Thus, we choose the same scaling parameters as proposed in the literature on Taubin smoothing. In what follows, we will smooth the brain mesh with equal weights and scaling parameters $\lambda = 0.33$ and $\mu = -0.34$ up to a volume loss of 5%.

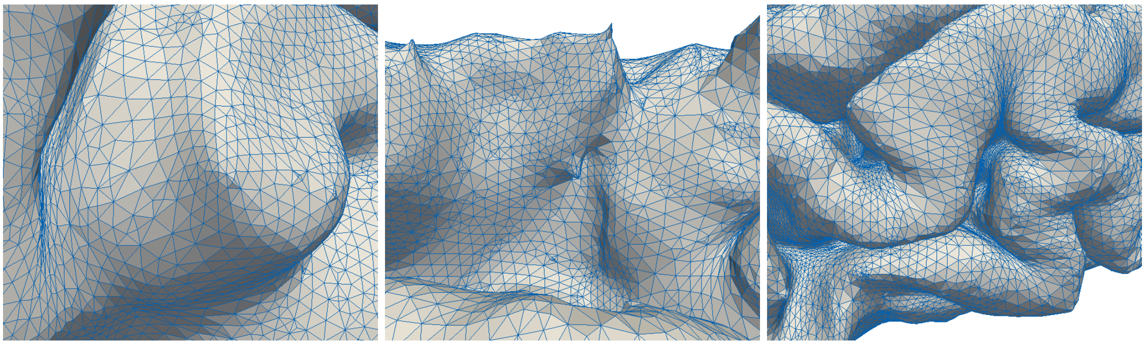
The results of applying the curvature approximation from Section 5.1.2 to the smoothed brain geometry is shown in Figure 5.10. We present a close-up of the Gaussian curvature and the mean curvature on the smoothed left hemisphere and in Table 5.4 we list the mean and standard deviation of the Gaussian and mean curvature of the left and right hemisphere of the smoothed cortex geometry. Gaussian curvature values have a high standard deviation which can be explained by a few very large curvature values that result from extreme mesh formations even after applying Taubin smoothing, as illustrated in Figure 5.8.

	Gaussian curvature		mean curvature	
hemisphere	mean	std	mean	std
left	0.1893	12.6791	0.4278	1.2869
right	0.3677	84.1972	0.4177	1.8088

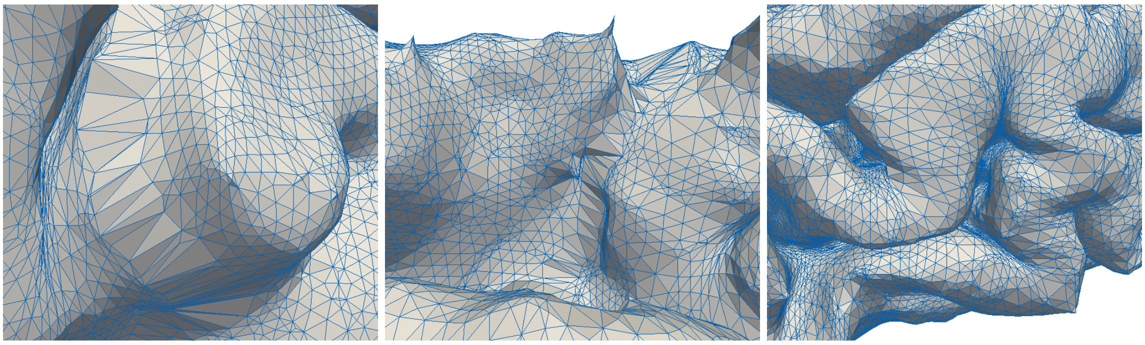
TABLE 5.4: The mean and standard deviation of the Gaussian and mean curvature of the (smoothed) left and right hemisphere.



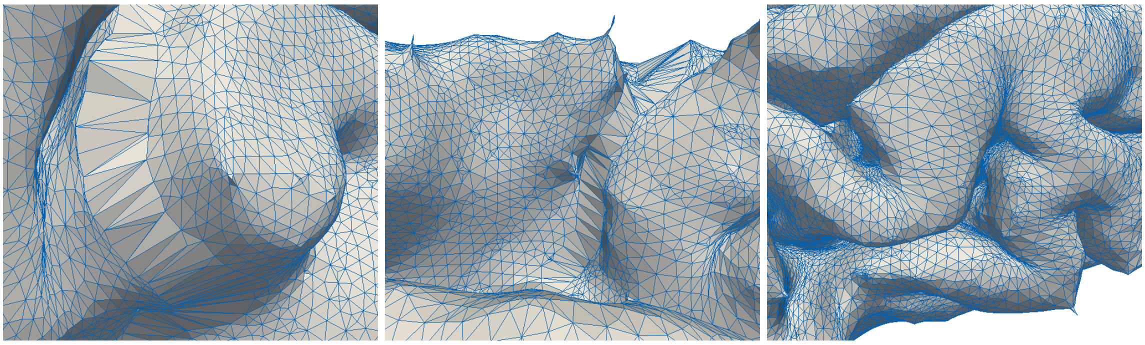
(a) original



(b) equal weights

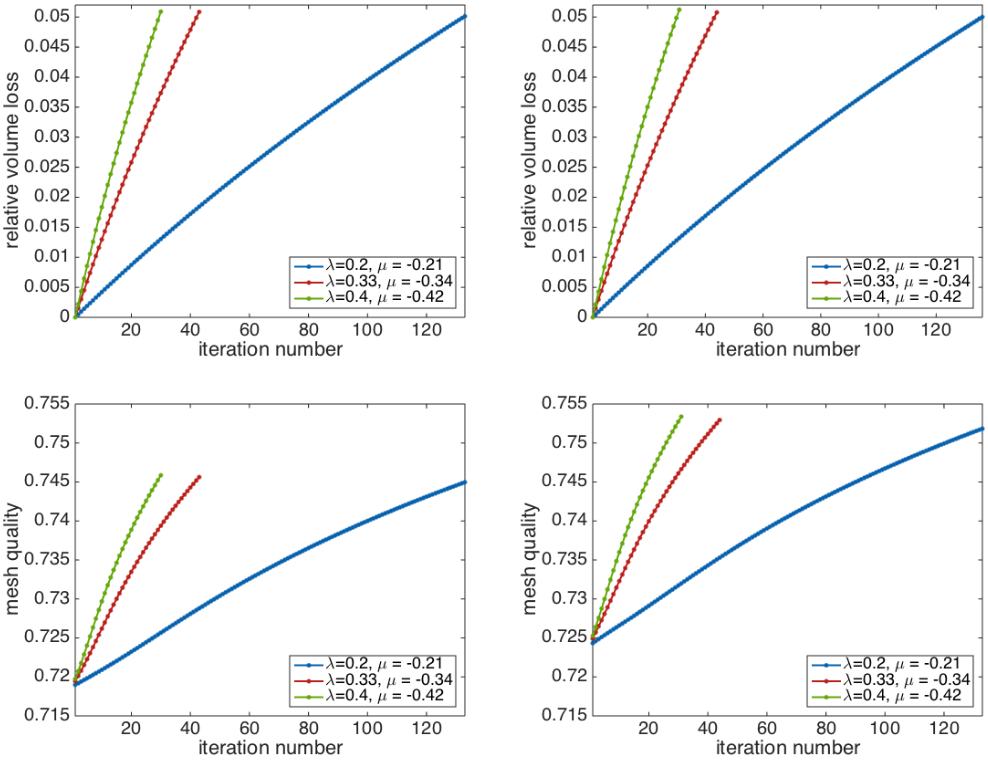


(c) Fujiwara weights



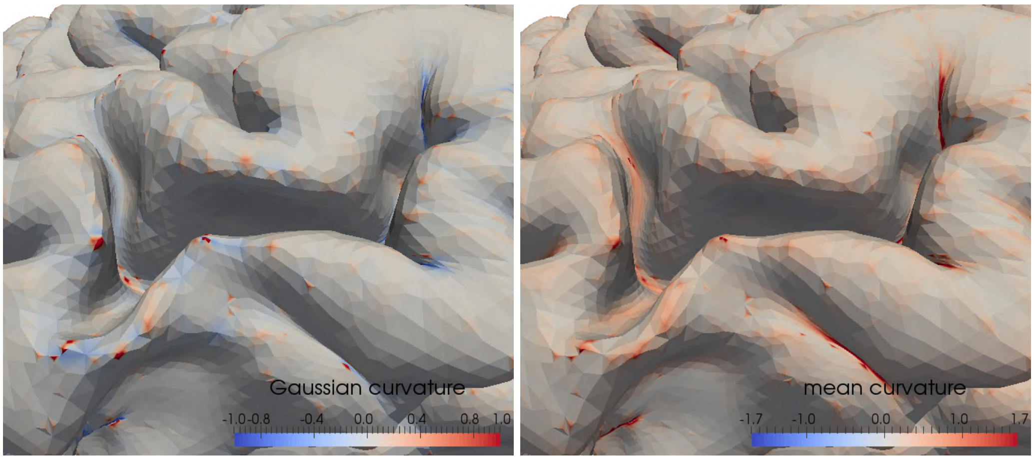
(d) Desbrun weights

FIGURE 5.8: Results of the Taubin smoothing with scaling factors $\lambda = 0.33$ and $\mu = -0.34$, a maximum volume loss of $V_{tol} = 0.05$ and equal (b), Fujiwara (c) and Desbrun (d) weights for particular regions of the left hemisphere.



(a) left hemisphere (b) right hemisphere

FIGURE 5.9: Volume loss and mesh quality for different parameter choices of λ and μ, for the left (a) and right (b) hemisphere.



(a) Gaussian curvature (b) mean curvature

FIGURE 5.10: Gaussian and mean curvature of the smoothed brain geometry of the left hemisphere.

5.2 Surface regularity index

For the computation of the surface regularity index (SRI) we refer to Pérez-Beteta et al., 2018. They introduce the surface regularity as a dimensionless ratio between the given volume of a geometry and the volume of a sphere with the same surface size, namely

$$\mathrm{SRI} = 6V\sqrt{\frac{\pi}{S^3}},$$

where V is the total volume and S is the total surface. The surface regularity varies between 0 (very irregular surface) and 1 (spherical geometry). To compute the volume and the surface of a closed surface in 3D, we use the Python shell in Paraview (Ahrens, Geveci, and Law, 2005).

For the geometry at hand, the surface regularity index is 0.1799 for the left and 0.1807 for the right hemisphere.

5.3 Quantities of interest in CSD propagation

CSD is known to originate from the visual cortex, but to gain deeper insight in the way the geometry shapes the CSD propagation, we simulate the spread of depolarisation waves between all the regions of the anatomical classification, in both hemispheres. In each simulation, we consider an area around the centroid of each region (out of the 34 ROI) as initial condition. Each simulation is run until all remaining regions have been reached by the depolarisation wave and the arrival times of this wave in the remaining 33 regions are recorded. In previous works, the whole region was considered as initial conditions but here we focus on the centroids to make the comparison between the propagation path and speed more accurate (Kroos et al., 2016).

The corpus callosum is the only compartment that is not considered as initially activated as it constitutes the intersection between the two hemispheres and obeys different rules for diffusion. For this reason, we eliminate the corpus callosum from the mesh and impose zero-flux boundary conditions on the boundary of the corpus callosum.

Using the data obtained from the 34 CSD simulations run per hemisphere, each one with CSD initiated in a different ROI, we can introduce different ways to assess symmetry and asymmetry in the propagation, with the aim of identifying regions featuring a peculiar propagation behaviour.

In this section we start with some comments on the computation of the centroids and the removal of the corpus callosum from the geometry. Then, we introduce the QoI describing CSD propagation on the cortex and conclude with the results on the smoothed brain geometry at hand with CSD propagation with DTI-based diffusion coefficients.

5.3.1 Centroids of the regions of interest

For the centroid computation we consider all points $\mathcal{P}_{\mathcal{R}}$ in region $\mathcal{R}$ and denote by $\mathcal{T}_{\mathcal{R}}$ all triangles that have at least two vertices in that region.

The surface area of a generic triangle K with vertices P_1, P_2 and P_3 is given by

$$S_K = \frac{||(P_2 - P_1) \times (P_3 - P_1)||}{2},$$

where $||\cdot||$ denotes the Euclidean norm, while the centroid of this triangle is $C_K = (P_1 + P_2 + P_3)/3$. Then, by defining the total mass of region $\mathcal{R}$ as

$$S_{\mathcal{R}} = \sum_{K \in \mathcal{T}_{\mathcal{R}}} S_K,$$

we compute the weighted sum of all centroids of the respective triangles (weighted by each triangle's mass) divided by the total mass, that is,

$$\widetilde{C} = \frac{1}{S_{\mathcal{R}}} \sum_{K \in \mathcal{T}_{\mathcal{R}}} C_K S_K.$$

Finally, we consider as centroid of region $\mathcal{R}$ the point $C_{\mathcal{R}}$ in region $\mathcal{R}$ that is closest to $\widetilde{C}$ in the Euclidean distance, namely

$$C_{\mathcal{R}} = \arg\min_{p \in \mathcal{P}_{\mathcal{R}}} ||\widetilde{C} - p||.$$

To obtain the initial condition for the CSD wave propagation, we consider a set of points in the proximity of the region's centroid, namely

$$P^* = \{p \in \mathcal{P}_{\mathcal{R}} \,|\, ||C_{\mathcal{R}} - p|| \leq s\},$$

where s is the maximum distance between the centroid and the points we consider as initially excited (typically we set $s = 10$ mm). These initially excited points belong to the same area and should not be located on opposite sides of a sharp bend, like in the example in Figure 5.11. Here, all points in P^* are visualised with dots but some of them are not connected (in the sense that will be described in detail below) to the centroid itself. To avoid the problem of starting CSD propagation in several areas, we only consider points in the initial region that are connected to the centroid, that we identify by means of some definitions and results from the area of graph theory, see Appendix B.3.

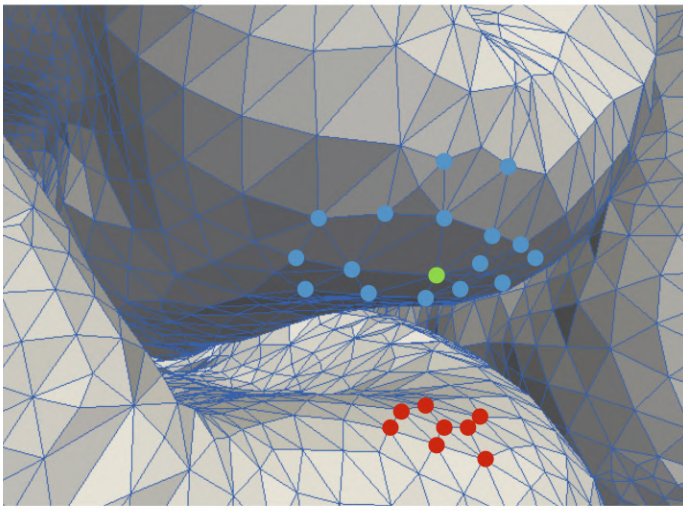


FIGURE 5.11: A visualisation of all points considered within the radius of $s = 10$ mm of the centroid (green). The non-connected points are coloured in red.

Connectivity centroids To verify that all points in the initially activated area around the centroid $C_{\mathcal{R}}$ are connected, we compute the adjacency matrix $\mathcal{A} \in \mathbb{R}^{l \times l}$ where l is the cardinality of P^*. A well-known result in graph theory states that l vertices are *connected* if and only if the matrix power $(\mathcal{A}+\mathbb{I}_l)^{l-1}$ has no zero entries. However, the size of $\mathcal{A}$ and the exponent l make this evaluation very expensive. Even using the diameter, that is, the maximum distance in a graph, as a lower boundary for the path length (see Definition B.3.3 and Lemma B.3.3) blows up the matrix. When calculations overflow the largest representable floating point number in Matlab (which is 10^{309}), big values are treated as infinity and this leads to not defined values when multiplying infinity with zero.

Another approach is a modified version of the Dijkstra algorithm that finds the shortest path between two nodes (Dijkstra, 1959). Most variations of this algorithm fix a single node as the source and find the shortest path from the source to any other node in the graph. We are interested in the distance of all the points to the centroid. As the problem at hand is an undirected graph with constant edge weights, we use a modified version of this algorithm that is outlined in Algorithm 5.3 (Dijkstra, 1959; Peng et al., 2012).

Algorithm 5.3 Modified Dijkstra algorithm

```
$D$ = DIJKSTRA($\mathcal{A}, P^*, C_{\mathcal{R}}$)
initialise: $Q = P^*$                                  ▷ all nodes initially in Q (unvisited nodes)
$D(p) = \infty$ for all $p \in Q$ with $p \neq C_{\mathcal{R}}$, $D(C_{\mathcal{R}}) = 0$    ▷ initiating the distance matrix
while $Q$ is not empty do
    $p = \arg\min_{p \in Q} D(p)$                      ▷ vertex in $Q$ with minimum distance to $C_{\mathcal{R}}$
    remove $p$ from $Q$
    for each neighbour $u$ of $p$ do                   ▷ that is all $u$ with $\mathcal{A}(p,u) = 1$
        $a = D(p) + 1$             ▷ the distance between $u$ and $p$ is 1, as they are neighbours
        if $a < D(u)$ then                             ▷ a shorter path to $u$ has been found
            $D(u) = a$
        end if
    end for
end while
```

Running the modified Dijkstra algorithm for the adjacency matrix $\mathcal{A}$ of all the points that are considered excited and the centroid as the source point, the vector D gives the distance of every vertex to the centroid. In the case of $D(p) = \infty$, this means that vertex p does not have any connection to the centroid and is thus dismissed from the initial condition.

5.3.2 Removal of corpus callosum

Removing the corpus callosum from the cortical geometry is essential to prevent the CSD wave from propagating through an area of white matter that would not take place naturally. However, eliminating a set of points from a given mesh needs to be done carefully because of the triangulation, its node numbering and adjacencies.

We start by creating a vector that gives the index number of all points belonging to region $\mathcal{R}$. These points and all triangles with a vertex in $\mathcal{R}$ are eliminated. An index exchange vector gives the new and old index number of all the points and has to be applied to the list of triangles to account for the new vertex numbering after deleting region $\mathcal{R}$. The precise scheme is given in Algorithm 5.4.

Algorithm 5.4 Removing region $\mathcal{R}$ from the mesh

```
input: set of points $P$, mesh $\Gamma_h$, point label $label$, region $\mathcal{R}$ to eliminate
output: $P^{\text{new}}, \Gamma_h^{\text{new}}, label^{\text{new}}$
Set: $N = |P|$                                        ▷ number of points
$ind :=$ find$(label = \mathcal{R})$                  ▷ vector with indices of points in $\mathcal{R}$
$M := \{1, \dots, N\} \backslash ind$                 ▷ row vector of remaining indices
$m := |M|$ and $swap := [(1:m)^T; M^T]$               ▷ matrix defining old and new labels
delete points $ind$ from $P$ and $label$
delete all triangles in $\Gamma_h$ with vertex in $ind$ and rename them according to $swap$
```

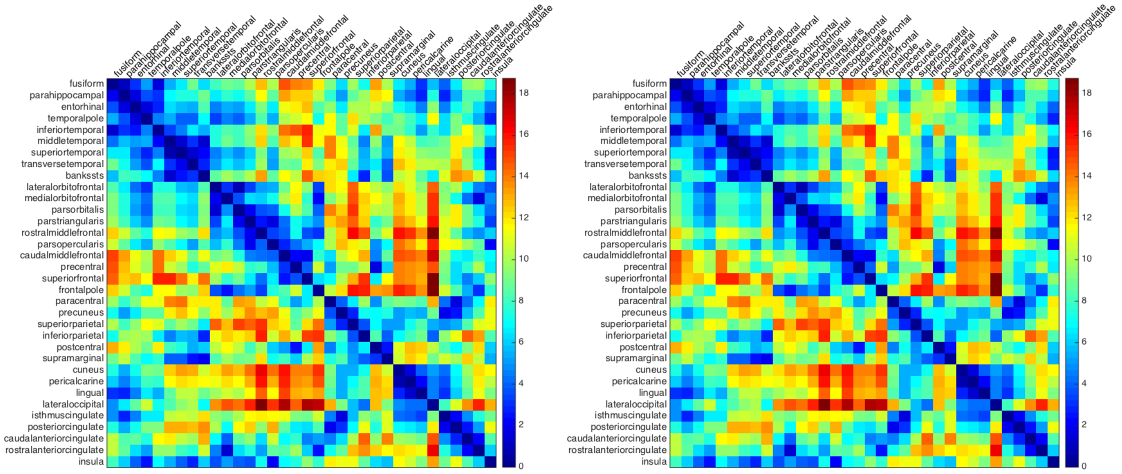
5.3.3 Quantities of interest

In each simulation, the centroid of one region is initially activated, and we record three values for each of the 33 remaining regions: the activation time of the centroid and the minimum and maximum activation times. The *activation in the centroid* is the moment in which the wave reaches the centroid of a region. The *minimum activation time* is the moment when the first point of the region considered is depolarised and reached by the CSD wave, while the *maximum activation time* is the moment at which the last point of the region is reached by the CSD wave. On the one hand, the minimum activation time for a given arrival region does not depend on the region shape or size, but only on the initially activated area and the area of the cortex previously traveled by the wave. On the other hand, the maximum activation time is related to the shape and area of the arrival region. These three recorded quantities allow assessing different aspects of the propagation. Thus, the minimum activation time for a region provides information about the propagation behaviour of its neighbouring regions, while the maximum activation time is also accounting for the effect of the region's geometry on the propagation. The activation in the centroid allows comparable measurements regarding the symmetry and relating traveling distance to propagation speed.

These quantities are accumulated in six 34×34 matrices, that we denote by L_c, L_{min}, L_{max}, R_c, R_{min} and R_{max} where L and R refer to the left and right hemisphere, respectively. In all these matrices, rows represent the starting region of the wave propagation, while columns correspond to the arrival region. As an example, the (i, j)-th element of L_c represents the arrival time in the centroid of region j of a wave originated from the neighbourhood of the centroid in region i.

The ordering of the regions in building such matrices plays a crucial role in the clustering of the results. Different sorting choices, as the clustering of all regions with respect to their centroids distances or the ordering concerning the activation times, lead to different types of ordering. To emphasise the spatial connection between the regions, we chose to rearrange their ordering according to their affiliation to lobes as introduced in Section 1.2.

Regions belonging to one lobe are then clustered according to the mutual distance of their centroids in the Euclidean norm. The activation times in the centroids for both the left and right hemispheres are given in Figure 5.12.



(a) left hemisphere

(b) right hemisphere

FIGURE 5.12: The activation times in the centroids, L_c and R_c, for the left (a) and right (b) hemisphere.

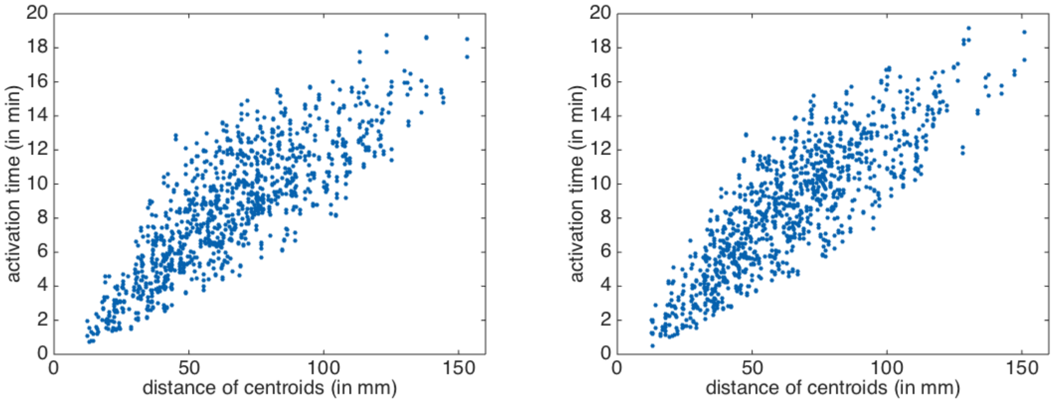
Independently from the ordering of the regions, a lack of symmetry in the matrices L_c and R_c is a first clear indicator that the geometry plays a role in shaping the propagation. A spherical cortex with isotropic conductivity coefficient would have resulted in the matrices being symmetric.

Correlation between activation times and distances Despite the complex geometry of the cortex, a direct relation between the propagation time of wavefronts and the distance traveled is expected. To further assess this association, we explicitly relate activation times to distances from the initially activated region. We consider the Euclidean distance between the centroids of two regions as a proxy of the distance traveled by the wavefront. From the off-diagonal entries of matrices L_c and R_c, we create two datasets. Diagonal elements of these matrices as well as the corresponding Euclidean distances vanish, since the starting and arrival points coincide. Thus, we can neglect diagonal entries in the definition of each dataset, obtaining a total of 1122 pairs in both L_c and R_c. In Figure 5.13 we plot these two datasets for the left and right hemisphere.

Clearly, a correlation between the distance of the centroids and the activation times can be observed. This correlation can be assessed by the Pearson product-moment correlation coefficient, a measure of the linear correlation between two variables X and Y (Pearson, 1895). The Pearson test returns values between +1 and -1 inclusive, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation. For a given sample of paired data $(x_i, y_i)_{i=1,\dots,n}$ the Pearson correlation coefficient is defined as

$$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{5.9}$$

with $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$.



(a) left hemisphere

(b) right hemisphere

FIGURE 5.13: The activation times in the centroids, L_c and R_c plotted against the distance of the centroids, for the left (a) and right (b) hemisphere.

The Pearson correlation coefficients are high, 0.826 for the left and 0.8411 for the right hemisphere, revealing a positive correlation between the distance of the regions and the activation times. Comparing the two hemispheres, we can observe a slightly stronger correlation between activation times and distances on the right hemisphere. On a flat surface with isotropic diffusion this correlation would be 1. So the more the Pearson correlation coefficient differs from 1, the higher the impact of the geometry and the diffusion data on the CSD propagation.

Global asymmetry index To assess propagation differences between brain hemispheres, we analyse how CSD waves propagate in the two hemispheres. Comparing propagation speed in the different directions provides substantial information about the global geometry and how easily depolarisation waves propagate through certain areas. The activation time matrices in the centroids L_c and R_c are the least dependent on the size and shape of the regions, and provide valuable information about the propagation between two regions. As a consequence, these matrices are natural candidates to assess global symmetry properties on both hemispheres.

First, we focus on propagation asymmetry within each hemisphere, and we consider the time difference for a depolarisation wave to travel back and forth between any two regions. Such difference is given by the matrices

$$L = L_c - L_c^T \quad \text{and} \quad R = R_c - R_c^T, \tag{5.10}$$

where L_c^T and R_c^T denote the transposes of matrices L_c and R_c. As an example, the entry L_{ij} expresses the difference in arrival time between a wave traveling from region i to region j, and a wave traveling from region j to region i in the left hemisphere. If the propagation of the CSD wave is symmetric in any hemisphere, the corresponding matrix L or R vanishes. The values of the entries of L and R provide a global measure for the asymmetry in the geometry between two different regions.

Since both matrices L and R are skew-symmetric with zero diagonal, their values can be represented, without loss of information, by their lower or upper triangular, off-diagonal, portion, combined in a single matrix G, whose entries are given by

$$G_{ij} = \begin{cases} L_{ij} & \text{for} \;\; i > j \\ 0 & \text{for} \;\; i = j \\ R_{ij} & \text{for} \;\; i < j \end{cases} . \tag{5.11}$$

Matrix G collects at once all the information on the lack of symmetry of both hemispheres, that can be compared in a single plot, see Figure 5.14. For the cortical geometry at hand, we can observe significant asymmetries in both hemispheres, in particular in the neighbourhood of the postcentral region in the left hemisphere and the superiortemporal region in the right hemisphere.

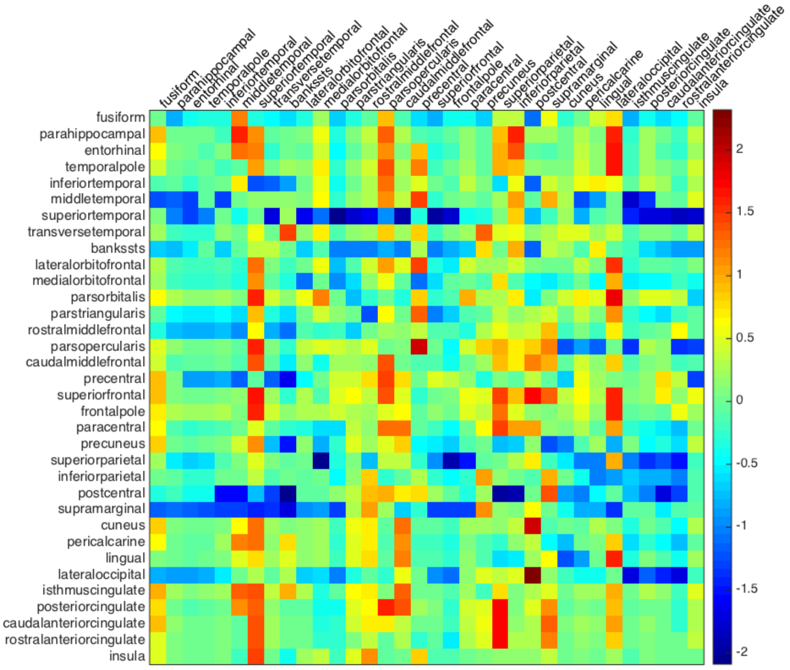


FIGURE 5.14: The time difference of the depolarisation wave to propagate between two regions back and forth. The lower triangle gives the values of the left hemisphere and the upper triangle the values for the right hemisphere.

To quantify the lack of symmetry in the propagation, we consider the normalised difference matrices $\widehat{L}$ and $\widehat{R}$, whose (i, j)-th entries are defined as

$$\widehat{L}_{ij} = \begin{cases} 0, & \text{for} \;\; i = j \\ \dfrac{L_{ij}}{(L_c)_{ij}}, & \text{for} \;\; i \neq j \end{cases}, \qquad \widehat{R}_{ij} = \begin{cases} 0, & \text{for} \;\; i = j \\ \dfrac{R_{ij}}{(R_c)_{ij}}, & \text{for} \;\; i \neq j \end{cases}. \tag{5.12}$$

We denote by l_j and r_j the mean of the j-th column of matrix $\widehat{L}$ and $\widehat{R}$, respectively. For each region j, positive values of l_j (or r_j) indicate that depolarisation waves moving away from the region are faster than depolarisation waves approaching that region. A positive mean $\bar{l}_j$ (respectively, $\bar{r}_j$) implies that the corresponding region is, in general, behaving as a source (or a facilitator) for the propagation of a depolarisation wave, while a negative mean implies that the region is behaving as a sink for the propagation. In Figure 5.15 we summarise in the same plot the means $\bar{l}_j$ and $\bar{r}_j$ and identify regions that exhibit different behaviour in the two hemispheres. We can observe that not only the magnitude of the effect changes significantly between the two hemispheres but in some cases the source/sink role was even reversed (as in the highlighted regions in Figure 5.15). In all the plots, m.t. denotes the medial temporal lobe, l.t. the lateral temporal lobe, o.l. the occipital lobe, and c.c. the cingulate cortex.

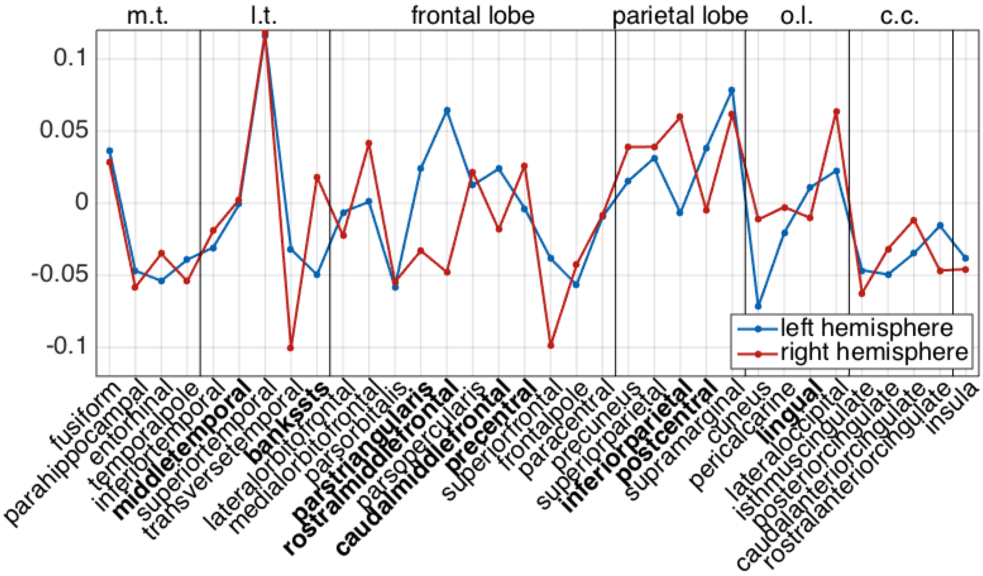


FIGURE 5.15: Mean of the normalised differences $\widehat{L}$ and $\widehat{R}$ for the left (blue) and right hemisphere (red). The regions that exhibit different behaviour in the two hemispheres are highlighted with bold font in the horizontal axis labels.

A global asymmetry index, for each region j, is then obtained by taking the sign of the quantities l_j and r_j, namely

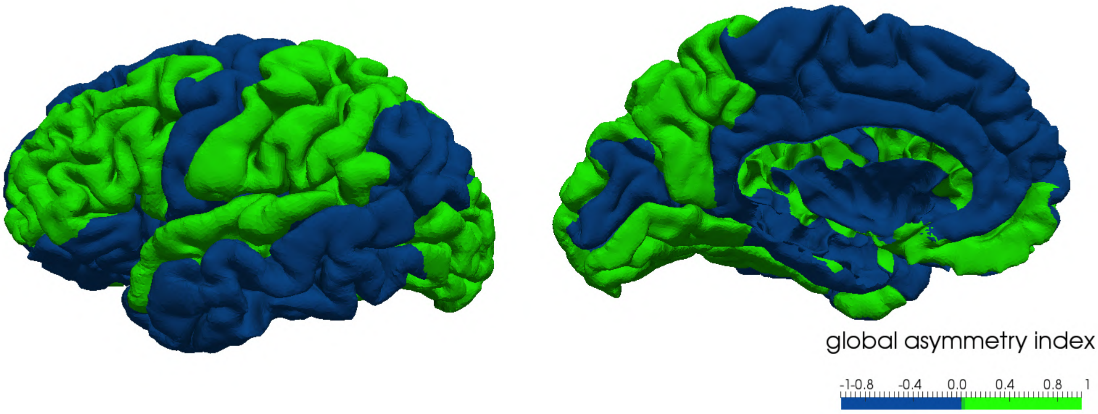
$$\mathfrak{A}_j^L = \operatorname{sign}(l_j) \qquad \text{and} \qquad \mathfrak{A}_j^R = \operatorname{sign}(r_j)\,. \tag{5.13}$$

In Figure 5.16 we plot the global asymmetry index on the brain geometry to visually highlight the behavioural differences between homologous regions in the two hemispheres.

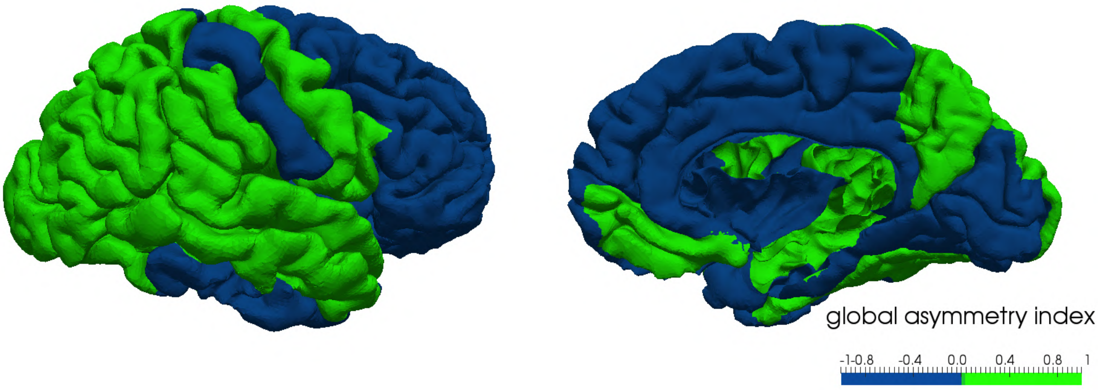
Retention index For a given region, the time a depolarisation wave needs to reach the whole region provides important information about its geometrical regularity, shape and size. To assess possible differences in the local behaviour between the left and right hemisphere, we introduce *residence matrices* defined as the matrices of the difference between maximum and minimum activation times, namely

$$\mathcal{D}^L = L_{max} - L_{min} \qquad\qquad \mathcal{D}^R = R_{max} - R_{min}. \tag{5.14}$$

Again, rows indicate the initial region of the depolarisation wave, while columns represent the arrival regions. The (i, j)-th element of $\mathcal{D}^L$ and $\mathcal{D}^R$ represents the residence time in region j of a wavefront originated from region i. Hence, the j-th column of $\mathcal{D}^L$ gives the residence times for the j-th region in the left hemisphere for all initially excited regions, while the j-th column of $\mathcal{D}^R$ provides the same information for the j-th region in the right hemisphere.



(a) left hemisphere

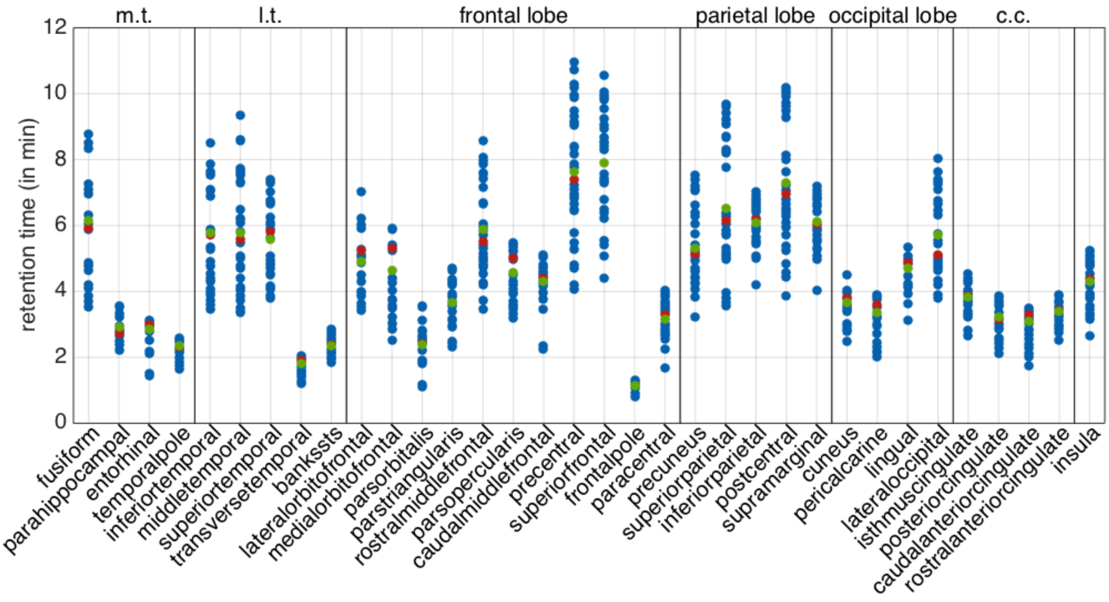


(b) right hemisphere

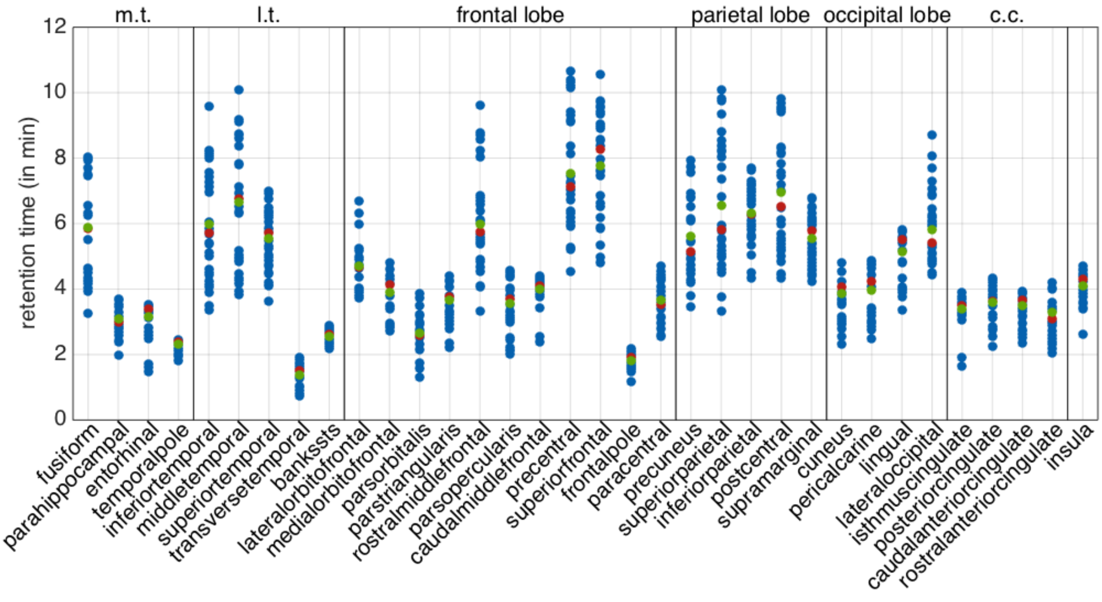
FIGURE 5.16: The global asymmetry index $\mathfrak{A}_j^L$ and $\mathfrak{A}_j^R$ on the left (a) and right hemisphere (b).

The residence times for each of the 34 regions are summarised in Figure 5.17. We visualise all entries of a column, compute the median and the arithmetic mean of the column entries for both hemispheres and compare the mean in a separate graph in Figure 5.18. The diagonal elements of the matrices $\mathcal{D}^L$ and $\mathcal{D}^R$ are zero and are neglected in the following. The behavioural symmetry between corresponding regions of the two hemispheres can thus be assessed by comparing the average residence time (both mean and median) from the columns of $\mathcal{D}^L$ and $\mathcal{D}^R$. Wide variations in the residence times with different initial regions hint at elongated or very irregularly shaped regions, in terms of their curvature or basal area.

We point out that the times presented in Figure 5.17 and Figure 5.18 do not actually represent the duration of the activation of the whole region. The activation period for the cortical cells after the passage of the CSD is of the order of 10 minutes. As a consequence, elongated regions can return to their resting state before the whole region has been activated.



(a) left hemisphere



(b) right hemisphere

FIGURE 5.17: Duration of stay of the depolarisation wave for the regions of the left (a) and right (b) hemisphere. The red markers denote the median, the green ones the arithmetic mean.

Comparing the average retention times (Figure 5.18), we can observe more similarities between the two hemispheres than in the previously observed back and forth propagation. A few regions exhibit appreciably different behaviour, likely the consequence of the lack of anatomical symmetry between the two hemispheres. Looking at this difference visually such regions are the middletemporal, the medialorbitofrontal, the parsopercularis, the frontalpole and the pericalcarine. From Figure 5.18 we can observe that, for the cortical geometry at hand, the major differences are localised in the frontal lobe and in the cingulate cortex.

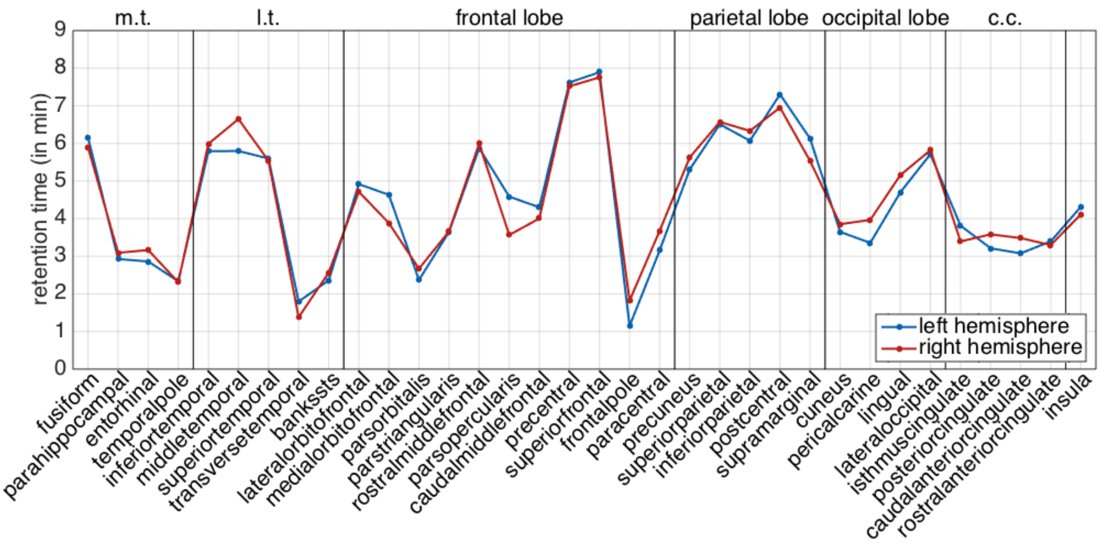


FIGURE 5.18: Comparative plot of the mean of the retention time for regions in the left (blue) and right (red) hemisphere.

To obtain a global measure for the retention of the depolarisation wave for each region independently of the initial region of the depolarisation wave, we sum up the entries of each column of $\mathcal{D}^L$ and $\mathcal{D}^R$. Thus, for both hemispheres we define the retention index for each region j as

$$\mathfrak{R}_j^L = \sum_{i=1}^{34} \mathcal{D}_{ij}^L, \quad \text{and} \quad \mathfrak{R}_j^R = \sum_{i=1}^{34} \mathcal{D}_{ij}^R, \tag{5.15}$$

that provides a measure of the overall time in which the CSD wave stays in a specific region.

A direct relation between the residence time of the depolarisation wave in a given region and the area of the region itself is expected. We assess it by considering the datasets

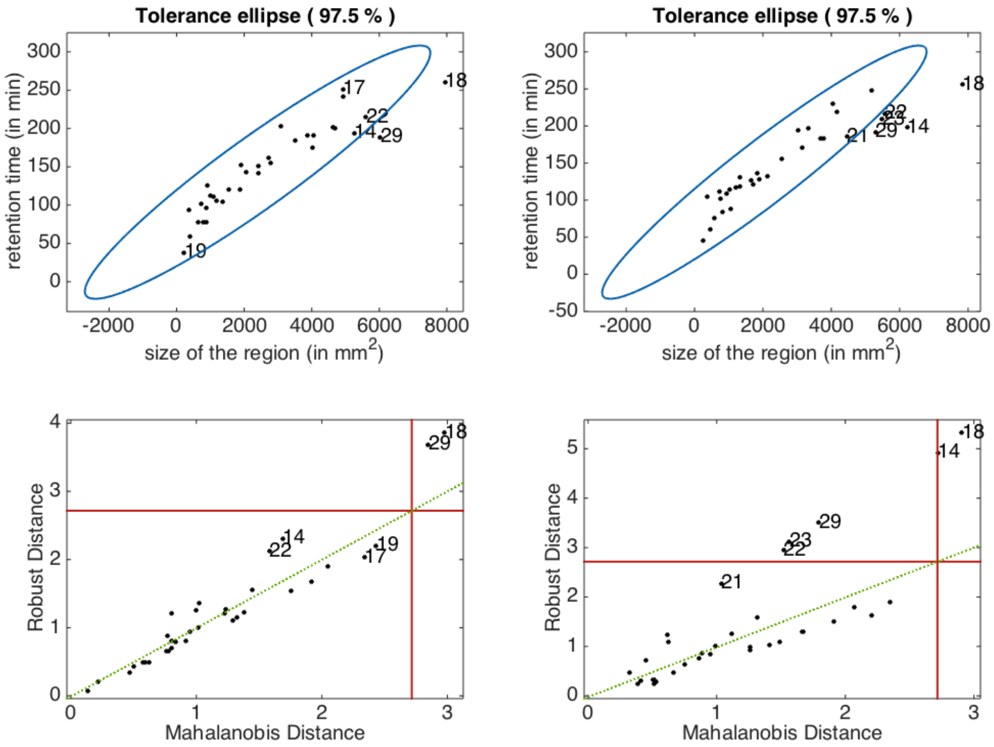
$$(S_j, \mathfrak{R}_j^L)_{j=1,\dots,34}, \quad \text{and} \quad (S_j, \mathfrak{R}_j^R)_{j=1,\dots,34}, \tag{5.16}$$

where S_j describes the surface of region j and $\mathfrak{R}_j$ the corresponding retention index. Since we focus on the identification of symmetries or the lack of them, we are particularly interested in regions whose propagative behaviour deviates significantly from the others'. A natural choice to detect regions of unusual behaviour, is to look for outliers in the database. The standard method for multivariate outlier detection is the Mahalanobis distance (De Maesschalck, Jouan-Rimbaud, and Massart, 2000; Mahalanobis, 1936). For an $n \times p$ dimensional set of data $(x_1, \dots, x_n)^T$ with the i-th observation defined as $x_i = (x_{i1}, \dots, x_{ip})$ the Mahalanobis distance M is defined as

$$M_i = \sqrt{(x_i - \mu)^T \, \Sigma^{-1} \, (x_i - \mu)} \quad \text{for } i = 1, \dots, n, \tag{5.17}$$

where $\mu = (\mu_1, \dots, \mu_n)^T$ is the estimated mean and $\Sigma \in \mathbb{R}^{p \times p}$ the estimated covariance matrix. For multivariate normally distributed data the values are approximately χ^2-distributed with p degrees of freedom. Multivariate outliers can be defined as observations having a large Mahalanobis distance. Thus, a quantile of the χ_p^2 distribution, e.g., the 97.5% quantile, can be considered. A drawback of the Mahalanobis distance is that it uses the classical estimators for mean and covariance, which can be highly affected by outlying values. In order to obtain more reliable results for the data analysis, more robust estimators are required. The minimum covariance determinant estimator is most frequently used and can be computed with a fast algorithm (Rousseeuw and Van Driessen, 1999). Using robust estimators of location and scatter in equation (5.17) we obtain the so-called robust distance (Rousseeuw and Van Zomeren, 1990; Rousseeuw and Van Zomeren, 1991).

In Figure 5.19 we visualise the relationship between the surface of a region and its retention index. We plot the size against the retention index and the robust tolerance ellipse describing the 97.5% quantile for the datasets of the left and right hemisphere. The Mahalanobis distance and the robust distance can be compared in a distance - distance plot. In both graphs the 97.5% quantile of the χ_2^2-distribution is drawn as a threshold value and the numbers mark the six maximum distances and refer to the index of the dataset and thus the index of the regions. For a list of the indices and the corresponding regions, we refer to Table D.3.



(a) left hemisphere (b) right hemisphere

FIGURE 5.19: The tolerance ellipse (top row) of the 97.5% quantile for the robust distance and a distance - distance plot (bottom row) comparing Mahalanobis distance and robust distance for the surface - retention index data of the left (a) and right (b) hemisphere. The numbers refer to the index of the datasets and the red lines mark the 97.5% quantile.

In Table 5.5 we list the detected outliers as well as the names corresponding to the indices of the regions. The rostalmiddlefrontal, the superiorfrontal, the superiorparietal, the interiorparietal and the lateraloccipital region in both hemispheres are identified as outliers either by the Mahalanobis distance or the robust distance. Only the lateraloccipital region in the left hemisphere, the rostalmiddlefrontal region in the right hemisphere, and the superiorfrontal region in both hemispheres were considered as outliers by the two methods.

		left		right	
index	region name	Mahalanobis	robust	Mahalanobis	robust
14	rostalmiddlefrontal			✓	✓
18	superiorfrontal	✓	✓	✓	✓
22	superiorparietal				✓
23	interiorparietal				✓
29	lateraloccipital	✓	✓		✓

TABLE 5.5: The index and name of the outliers for the left and right hemisphere detected with the Mahalanobis distance and the robust distance. The checkmarks indicate that this dataset was identified as an outlier by the respective method.

The Pearson correlation coefficients, introduced in (5.9), highlight in both hemispheres a clear positive correlation between the retention index and the size of the corresponding region, $r = 0.9258$ for the left and $r = 0.9225$ for the right hemisphere.

5.4 Summary

In this chapter we introduced a smoothing technique to correct numerical artefacts and make the cortical geometry more realistic and more suited for curvature estimations and simulations of CSD propagation.

Resorting to the Desikan-Killiany atlas, each hemisphere of the brain was divided into 34 different anatomical regions and we identified suitable QoI that can be computed by post-processing the results from the simulation of the CSD propagation across the whole cortex.

We studied propagation symmetries and asymmetries both at the intra and inter-hemispherical level. By introducing the global asymmetry index and the retention index, we have found a clear asymmetry pattern emerging in both assessments. In particular, we observed that the propagation speed between two regions is not the same when comparing CSD traveling in opposite directions. That is, the speed of a wave moving from i to j is different from the one of a wave moving from j to i, and this feature is observed in both left and right hemispheres.

Neuroanatomical differences between the left and right sides of the brain are known to exist at various scales (Toga and Thompson, 2003); our work shows that asymmetry is also present in the propagation of waves of electrophysiological activity, most likely due to geometric effects (but they might also be due to differences in brain circuits within left and right hemisphere). Recent work highlights the local influence of the curvature of two-dimensional surfaces on properties, such as nucleation and propagation of waves (Kneer, Schöll, and Dahlem, 2014); our results investigate this relation at the scale of a whole cortical geometry reconstructed from brain imaging.

We have also identified brain regions whose behaviour significantly differs from the other regions. In particular, some regions appear to trap the propagating depolarisation wave for a longer time. The outliers in the relation between retention index and region sizes are strong candidates to identify areas that may play a vital role in CSD propagation. Such information would be relevant to design therapies using cortical neuromodulation, where target structures for modulation have to be carefully selected.

Chapter 6

Clinical applications

"It is important to understand what you can do before you learn to measure how well you seem to have done it."

John W. Tukey, in *Exploratory Data Analysis (1977)*

The simulation of CSD propagation on a personalised cortical geometry in Chapter 4 and the introduction of QoI in Chapter 5 served as a proof of concept. As a next step, we apply these techniques to a set of patient data with the aim to find significant differences between migraine patients and control groups. In a first case study, we simulate CSD propagation on the cortices of chronic migraine patients with aura to assess the correlation of the symptoms they experience during a migraine attack with the CSD wavefront propagation. Then, in a group analysis, we compare different patient groups in terms of CSD propagation and the QoI introduced in the previous chapter. With these applications, we want to investigate the impact of the geometry on the CSD propagation and its correlation to migraine (with aura).

The data set for the initial case study and the ones for the group analysis were provided from different healthcare institutions and were acquired with different machines, at different magnetic field strengths. Although we have access to various data sets, we have to evaluate them independently of one another. In fact, we cannot set magnetic resonance imaging (MRI) and diffusion tensor imaging (DTI) data acquired from different machines or with different field strengths side by side because cortical reconstruction results obtained with FreeSurfer corresponding to different machine brands and magnetic field strengths are not comparable.

In this chapter we will first comment on the consistency of brain reconstruction with FreeSurfer and the comparability of the results with different acquisition and processing modes. Subsequently, we discuss the case study of the chronic migraine patients with aura and perform a group analysis for migraine patients without aura, abusive painkiller users and a control group. The aim is to explore if the QoI provide a good indicator to distinguish between patient groups.

All the data used in this chapter was provided in MRI and DTI or diffusion-weighted imaging (DWI) raw format and had to be processed in order to extract the cortical geometry and the diffusion values at all mesh points of the triangulated cortical surface. For the first case study the conductivity tensors were provided in the form of DWI data, while in the groups study DTI data was available. For the analysis carried out in this chapter, we use the FreeSurfer image analysis suit. In Appendix C, we give an introduction to

MRI, DWI and DTI data processing and provide detailed information on the programs and functions involved as well as the corresponding code.

6.1 Comparability of brain cortex reconstructions

Several studies have reported that FreeSurfer brain reconstructions from T1-weighted images and diffusion data from DTI images depend on different factors that vary in their impact on the reconstruction results. For example, the acquisition platform and magnetic field strength, the FreeSurfer version, operating system and its version were found to affect in different measure the cortical reconstruction and the corresponding diffusion data. However, the latter two do not have such a significant impact on the reconstruction results as the acquisition platform and the field strength. Despite these limitations, we use FreeSurfer for the brain reconstruction as it is widely used in practice and an open-source software.

Jovicich et al., 2009 assessed the volume differences in specific regions as a measure to compare brain reconstruction results on different platforms (Siemens, GE) and with different field strengths and scanner upgrades. They concluded that automated segmentation of T1 images is reliable within the same scanner platform, even after updates. However, combining data across platforms and field strength introduces bias. Souza et al., 2018 consider the results of different skull stripping programs, i.e., advanced normalization tools[1] (ANTs) and the brain extraction tool[2] (BET) for MRI images recorded with machines of different vendors and different field strength. Their findings support the fact that scanner vendors and magnetic field strength have a significant impact on the brain extraction results. In another study, Gronenschild et al., 2012 found that workstation and MacOS version only have a small impact on the anatomical volume and cortical thickness in the FreeSurfer results. However, the FreeSurfer version has a more significant effect on the results.

The comparability of diffusion data obtained from DTI imaging with different acquisition modes seems controversial. Even though Brander and collaborators found only small differences in the mean fractional anisotropy (FA) and apparent diffusion coefficient (ADC) values when comparing data acquired with different field strengths, other studies proved that ADC values were statistically significantly lower at 3 Tesla compared with 1.5 Tesla and FA values were statistically significantly higher at 3 Tesla compared with 1.5 Tesla (Brander et al., 2010; Huisman et al., 2006; Sasaki et al., 2008).

Consequently, we are going to treat separately the two data sets obtained with different platforms (Siemens Magnetom Aera, Phillips Achieva) and different field strengths (1.5 T and 3 T, respectively) and only compare them with the corresponding control data set obtained with the same acquisition modalities.

6.2 Case study on single patients

For this study, performed in collaboration with Dr. Marina de Tommaso at the Policlinico di Bari in Italy, the available data set consisted of five migraine patients with aura. For

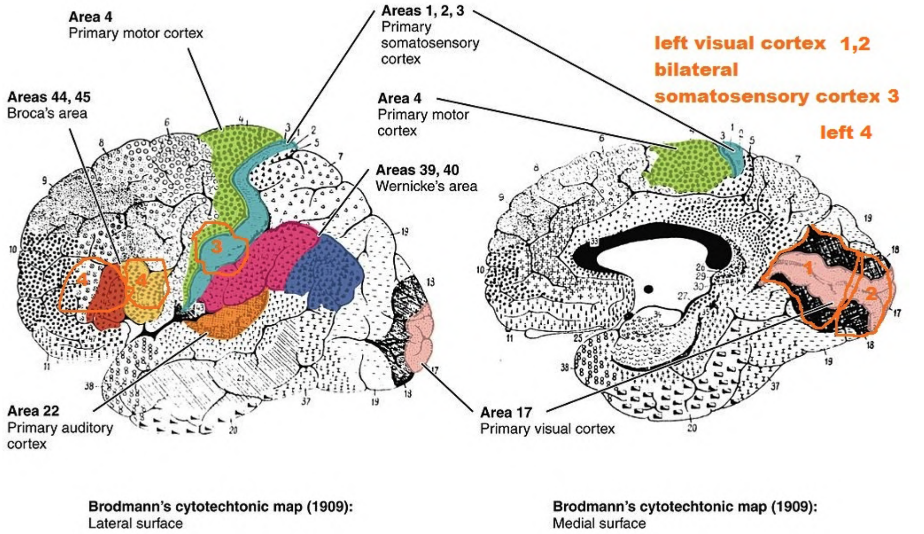
---

[1] `http://stnava.github.io/ANTs/`

[2] `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET`

the sake of anonymity, we refer to them as Subject 1 to Subject 5. Together with MRI and DWI imaging data of all the patients, a sequence of *eloquent areas* during a migraine attack marked on the brain cortex was provided as well as the corresponding symptoms. To avoid displaying the pictures for all five patients, we comment on the results of all of them but only provide a detailed description and illustrations for Subject 1. Information about the clinical symptoms and visualisations of results for all other patients are reported in Appendix D.

Subject 1 has been suffering from migraine with aura for nine years and experiences on average 5 migraine attacks with aura per month. In the course of a migraine attack, this patient reports luminous scotoma in the central visual field followed by a scotoma in the right half of the visual field. These visual symptoms are proceeded by temporary right and left paresthesia (abnormal dermal sensation like tingling, burning or numb sensations in the extremities) and transitory *aphasia* and *apraxia*. *Aphasia* refers to the inability to comprehend and formulate language, while *apraxia* is a motor disorder in which the patient has difficulties with the motor planning to perform tasks or movements. These aura symptoms can be related to specific cortical areas: the left visual cortex, bilateral somatosensory cortex and left Broca's area. These regions correspond to the Brodmann areas (BA) 17/18, 1-4 and 44/45. In Figure 6.1 the eloquent areas are marked in orange and the numbers indicate the sequence during a migraine attack for Subject 1. The areas that are involved and correspond to the specific symptoms differ between the hemispheres and patients. The list of migraine symptoms, information about the clinical history and the eloquent regions for all patients are summarised in Table D.1 and Table D.2.



Brodmann's cytotechtonic map (1909): Lateral surface

Brodmann's cytotechtonic map (1909): Medial surface

FIGURE 6.1: The eloquent areas (orange) in a migraine attack of Subject 1. The numbers refer to the sequence of symptoms.

6.2.1 MRI and DWI data

The data set corresponds to four female and one male subject, with an age of 27.4 ± 9 (range 21 - 43) years, suffering from chronic migraine with aura. The acquisition was performed at the Santissima Annunziata Hospital, Taranto, Italy and was approved by the local ethics committee. Data was acquired with a Siemens Magnetom Aera 1.5 T scanner using a T1-weighted 3D sequence with the following parameters: TR = 2650 ms, TE = 106 ms; flip angle = 10°; parallel imaging (mSENSE) acceleration factor = 1.5;

acquisition matrix size = 256 × 256; FoV = 260 × 260 mm, slice thickness = 1.1 mm, and 176 contiguous sections. The DWI data was acquired with the same scanner using a diffusion-weighted 2D sequence with the following parameters: TR = 2650 msec, TE = 106 msec; flip angle = 90°; acquisition matrix size = 192 × 192 ; FoV = 229 × 229 mm, and slice thickness = 5 mm, and 23 contiguous sections.

The control data set consists of seven patients, corresponding to three female and four male subjects with an age of 36.7 ± 9.9 (range 23 - 54) years also acquired at the Santissima Annunziata Hospital, Taranto in Italy. The local ethics committee approved the acquisition and the data was obtained with the same machine and parameters listed above for the migraine patients.

6.2.2 Data processing

The cortical geometries were obtained from the high-resolution anatomical MRI with the FreeSurfer image analysis suit. The brain reconstruction process was run with the FreeSurfer version 5.3.0 on BCAM's in-house cluster Hipatia featuring at that time 18 nodes (1 with Nvidia Tesla K40 GPU) for 624 cores with 4TB RAM and Infiniband network connectivity using CentOS 7, 64 bits. For the sake of comparability, all reconstructions were run on the same node of the cluster with 32 cores (2xProcessor Intel(R) Xeon(R) CPU E5-2683 v4 @ 2.10 GHz with 16 cores) and 256 GB RAM. For further information on the brain reconstruction procedure, we refer to Appendix C.3.

For the identification of the ROI, we use the **lh.BA.annot** and **rh.BA.annot** files generated by the FreeSurfer reconstruction, which provide the labels for the essential Brodmann areas (Brodmann, 2006). The BA listed in this annotation file include the primary somatosensory cortex (BA 1-3), the primary motor cortex (BA 4), Broca's area (BA 44, BA 45) and the visual cortex (V1, V2 or BA 17, BA 18, respectively).

The DWI data was processed with the FSL software[3]. First, an eddy current correction was applied to correct the changes produced by the variations in gradient field directions during the acquisition and to overcome the artefacts produced by head motion. The T1 and DWI brain data were extracted with BET using a fractional intensity threshold of 0.3. Then, a local fitting of the DWI data was applied to compute the tensor model at each voxel. For each voxel, the DWI data provided the mean diffusion strength of water molecules within brain tissues. To project this data onto the mesh obtained with FreeSurfer from the T1 image, a linear transformation between the T1 image and the ADC map was computed using 6 degrees of freedom and taking the mutual information as the cost function. Then, the transformation from T1 native space to the FreeSurfer structure space was computed with the FreeSurfer **tkregister2** function. Combining these transformation matrices and applying them to the ADC data, the values in the FreeSurfer space were obtained. Finally, the FreeSurfer function **mri_vol2surf** was used to project the diffusion data onto the brain mesh with a projection fraction of 0.5 to sample in the middle of the cortical surface and prevent bias from the white matter or the extracerebral fluid. For more details on the processing of DWI data with FSL, see Appendix C.4.

---

[3] `http://fsl.fmrib.ox.ac.uk/`

6.2.3 Preprocessing and simulation set-up

After obtaining the brain geometry in the form of a triangulated mesh with ADC values associated to each grid point, we run the Taubin smoothing with equal weights up to a volume loss of 5 % choosing the scaling parameters as $\lambda = 0.33$ and $\mu = -0.34$. Then, we eliminate the corpus callosum from the geometry for the CSD simulation (see Section 5.3.2). Figure 6.2 and Figure 6.3 show the smooth surfaces obtained for Subjects 1 to 5 and Controls 1 to 7 in the lateral and medial view of the left hemispheres, respectively. An example of the resulting smooth mesh and the ADC values on this smooth mesh is shown in Figure 6.4 for left hemisphere of Subject 1.

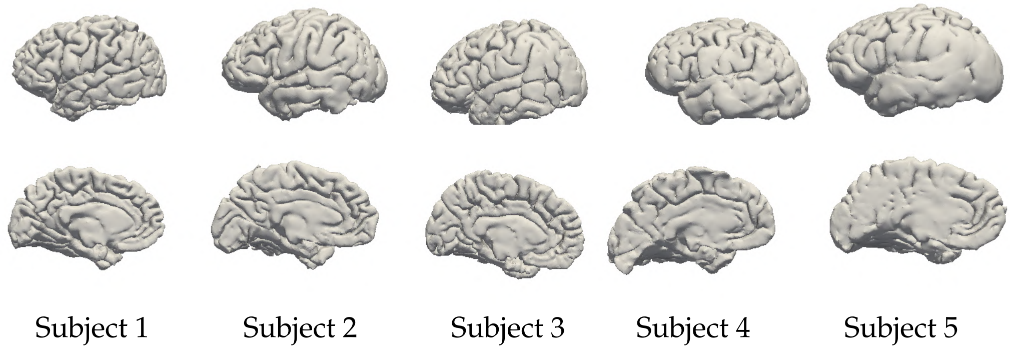


FIGURE 6.2: The left hemisphere of Subjects 1 to 5 in the lateral (top) and medial view (bottom).

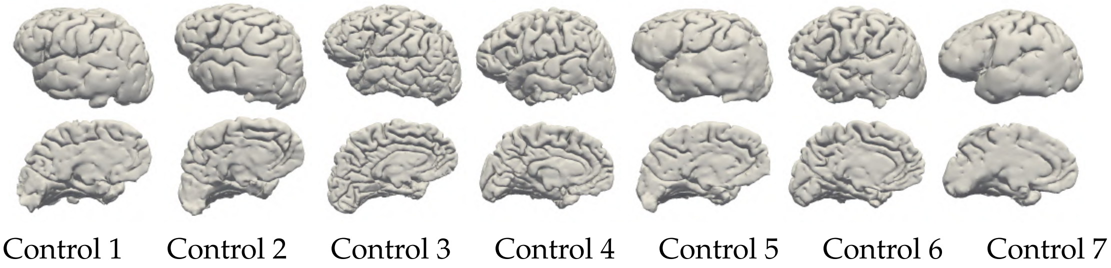
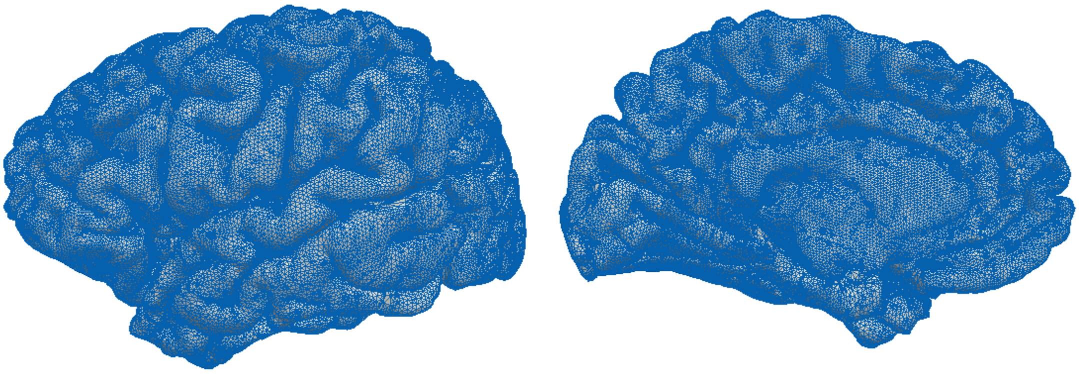


FIGURE 6.3: The left hemisphere of Controls 1 to 7 in the lateral (top) and medial view (bottom).

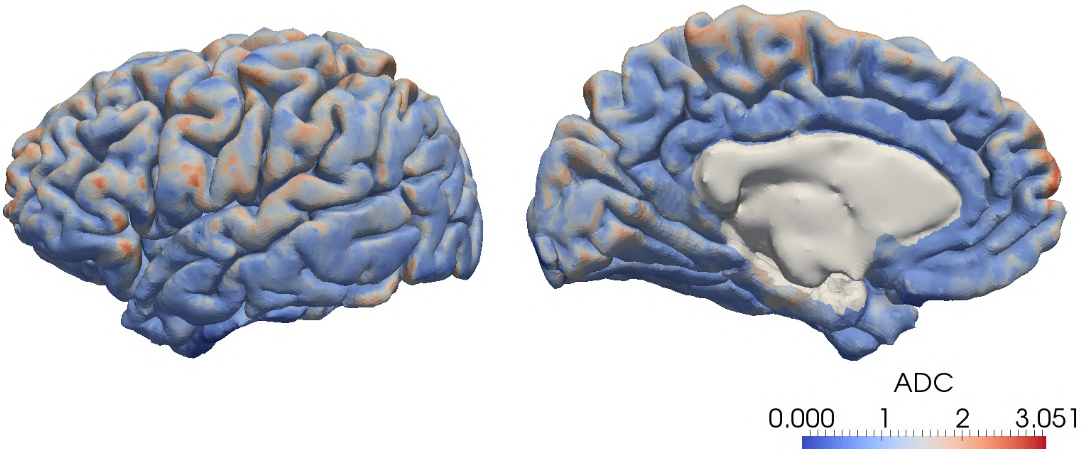
For the simulation of CSD, we normalise the ADC values by their mean value and use the same parameter values as in Chapter 4. For each subject, the wave propagation is initiated in the BA area where the first symptoms were recorded, usually the (primary) visual cortex V1.

6.2.4 Results

From the smoothed cortical mesh, we compute the surface regularity index (SRI) of the migraine patients and compare it to the ones of the control group. Note that for these computations we use the smooth geometry that still includes the corpus callosum because the SRI computation depends on the volume enclosed by the cortical surface. Then, we simulate the CSD propagation and check the conformance of the wavefront propagation with the symptoms of the patient.



(a) mesh reconstructed from T1-weighted image

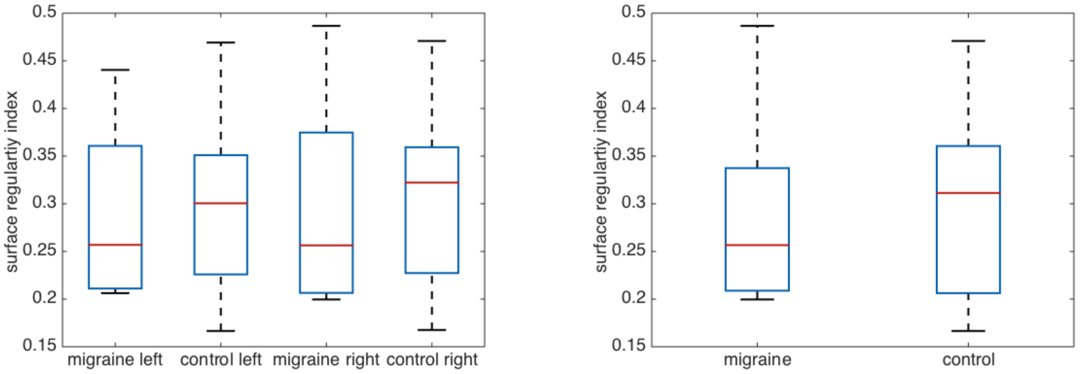


(b) ADC values on the surface mesh

FIGURE 6.4: The smoothed mesh reconstructed from T1-weighted image (a) and the ADC values on the surface mesh after taking out the corpus callosum (b) for the left hemisphere of Subject 1.

Surface regularity index We compute the SRI, introduced in Section 5.2, for the migraine and control data and visualise it in the form of a box plot in Figure 6.5. The central red mark indicates the median, the bottom and top edges of the blue box indicate the 25th and 75th percentiles, respectively, and the whiskers extend to the most extreme data points that are not considered outliers.

From Figure 6.5 we cannot conclude directly that the SRI of migraine patients is significantly different from the one of the control patients. The box plots already hint at different means but the ranges of SRI values are similar. For further investigation, we test for similar distribution of the two data sets. Because the data sets have different sizes, 5 and 7 data points, respectively, and are samples from a bigger sample set whose mean and variance are unknown, we test the hypothesis that the two populations have equal means. For this purpose, we use an adaptation of Student's t-test, called *Welch's t-test* that is more reliable when the two samples have unequal variances or unequal sample sizes (Welch, 1947). This is done in Matlab via the function **ttest2** with the option Vartype = 'unequal'. First, we consider left and right hemisphere for the two groups separately. The



(a) left and right hemisphere (b) whole brain

FIGURE 6.5: Box plots of the surface regularity index of the migraine and control group differentiating between left and right hemisphere (a) and considering the whole brain (b).

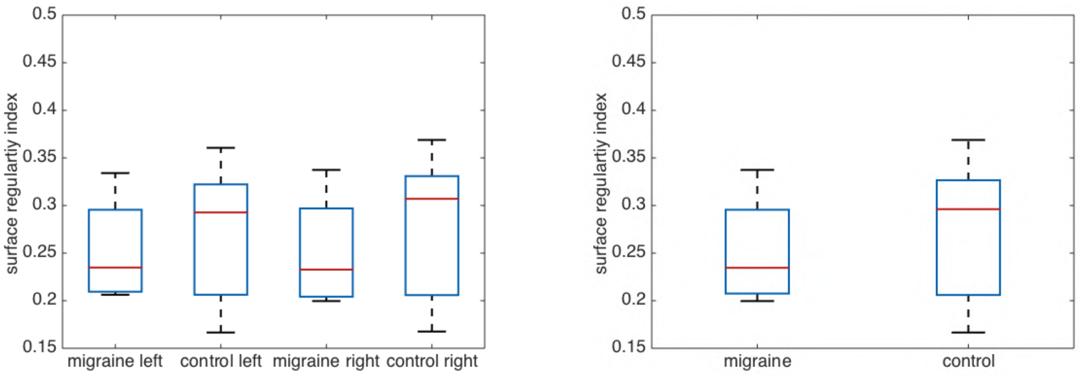
p-value of a test for equal means of the SRI of migraine patients and the control group in the left hemispheres is 0.8483 and in the right hemisphere 0.8764. Then we distinguish simply between patient groups by merging the SRI of left and right hemisphere in one data set for migraine patients and control, respectively. Thus, the data sets consist of 10 and 14 observations, respectively, and by applying the Welch's t-test we obtain a p-value of 0.7947. These high p-values indicate weak evidence against the hypothesis of equal means. Consequently, we cannot say that the data sets are significantly different.

Having a closer look at the reconstructed surfaces one could argue that the cortex of the last patient of each group is not as well reconstructed as the rest, featuring less gyri and in general a smoother cortical surface, see Figure 6.2 and Figure 6.3. We can consider those as outliers due to reconstruction error and visualise the data without the outliers in a box plot in Figure 6.6. Once more, we visualise the SRI of the left and right hemisphere separately for the migraine patients (Subjects 1-4) and the control group (Controls 1-6) in the first box plot and join the data of the two hemispheres in the second box plot.

Figure 6.6 shows slightly more different SRI values between the two groups and less extreme data points compared to Figure 6.5. With this reduced data set we test the hypothesis that the two populations have equal means with the Welch's t-test. The lowest p-value is obtained when considering left and right hemisphere together for the migraine and the control group, namely 0.3832. This p-value is still high and we cannot reject the hypothesis on a significant level. Hence, these tests confirm what could already be expected from the box plots that we cannot make a reliable statement about the distribution of the data and cannot prove a significant difference between the SRI of migraine patients and the control patient group.

CSD simulation Simulating the CSD propagation, we record the arrival time of the wavefront at every node as described in Section 5.3.

We visualise the arrival times of the wave on both the left and the right hemisphere for Subject 1 in Figure 6.7. Even though the CSD wave spreads across the whole cortex, only the BA involved in the migraine attack are coloured by the arrival time of the CSD wave. It can be observed that the temporal progression of the wave starting in the visual cortex



(a) left and right hemisphere (b) whole brain

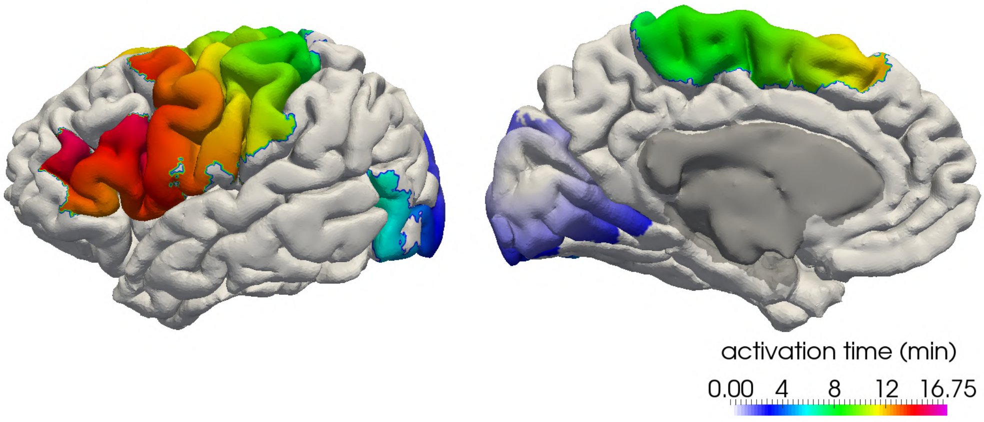
FIGURE 6.6: Box plots of the surface regularity index of the migraine and control group, just considering Subjects 1-4 and Controls 1-6. We differentiate between left and right hemisphere (a) and consider the whole brain (b).

matches the sequence of the regions involved in the migraine attack, see Figure 6.1. From the visual cortex the wave propagates to the somatosensory cortex (BA 1-3) and Broca's area (BA 44/45).

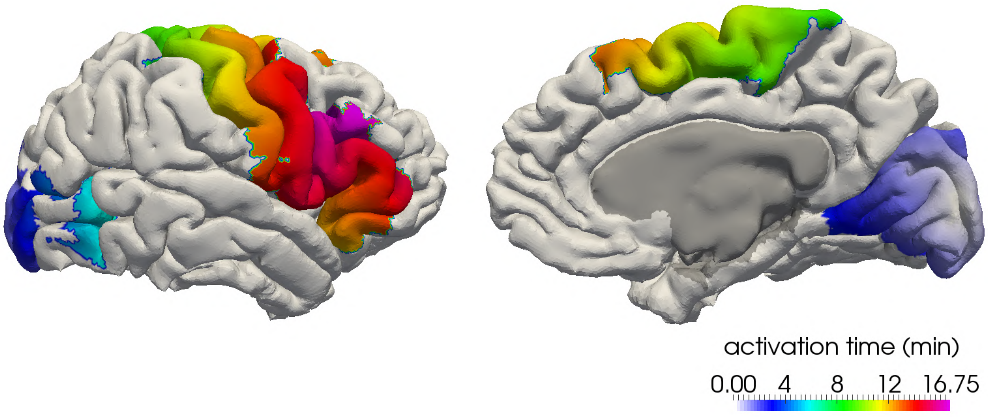
Considering the CSD propagation in Subjects 2 to 5 (see Appendix D) we can observe that for all patients the simulated CSD wave arrives in the eloquent areas in the sequence described by the patients during a migraine attack with only some minor exceptions. For example, in the right hemisphere of Subject 4 and of Subject 5 we do not observe the full sequence because only BA 17 and 18 are involved in the migraine attack. Moreover, in the left hemisphere of Subject 4, BA 1-4 and 44 are reached by the CSD wave nearly simultaneously. Although, we lack data about the exact timing of the appearance of different symptoms, we can conclude that the simulated CSD wavefront propagation matches the sequence of the affected areas during a migraine attack in all patients.

To sum up the CSD propagation behaviour and structure of the cortical geometry, we list the maximum time the wave takes to spread across the whole cortex (propagation time), the surface area, the SRI and the propagation speed for all migraine patients in Table 6.1. The propagation speed describes the amount of space invaded in a certain amount of time. To demonstrate how these values relate to each other, we visualise the data for Subjects 1 to 5 in the correlation plot of Figure 6.8. In this graph, we plot the 5 observations for each pair of variables for the left hemisphere in the lower triangular part of the plot and for the right hemisphere in the upper triangular part. The Pearson correlation coefficients (see equation (5.9)) are given in each subplot and coefficients with a p-value below 0.05 are marked in red. The red lines show the linear regression for the given data points.

In Figure 6.8 we can observe a significant correlation between the surface area and the propagation time, and between the propagation speed and the SRI in both hemispheres. Some correlations are more natural, like surface area, propagation speed, and SRI as the definitions of propagation speed and SRI both involve the surface area. The propagation speed also relates by definition to the propagation time. Only in the right hemisphere we can observe a significant (negative) correlation between the propagation time and the SRI. The closer the SRI is to 1 the closer the geometry is a sphere. If the SRI is smaller,



(a) left hemsiphere



(b) right hemsiphere

FIGURE 6.7: The arrival time (activation time) of the CSD wave in the left (a) and right (b) hemisphere of Subject 1.

the geometry at hand is rougher, and thus the propagation of a CSD wave takes longer. Comparing the two hemispheres, we can observe a slightly stronger correlation between each variable pair on the right hemisphere. Other correlations are counterintuitive, like the SRI and the propagation speed that are strongly negatively correlated, or the surface area and the propagation speed that correlate positively. An explanation could be that the DWI data influences the propagation speed, i.e., speeding up the CSD propagation in deep sulcus. This behaviour would not be expected with homogenous diffusion coefficients. However, these statements are only based on evaluating data from 5 patients and might not give enough evidence supporting the claim that there are significant differences between hemispheres in this particular context.

patient	**hemisphere**	**propagation time (min)**	**surface area (cm^2)**	**propagation speed (cm^2/min)**	**SRI**
Subject 1	left	16.23	80895.75	49.8433	0.2062
	right	16.13	82739.12	51.2952	0.1996
Subject 2	left	15.72	79969.21	50.8710	0.2568
	right	15.44	80394.32	52.0689	0.2564
Subject 3	left	16.25	86070.91	52.9667	0.2128
	right	16.23	87556.00	53.9470	0.2088
Subject 4	left	14.10	68030.99	48.2489	0.3341
	right	14.30	66850.20	46.7484	0.3374
Subject 5	left	14.51	60303.24	41.5598	0.4405
	right	14.13	53962.26	38.1899	0.4866

TABLE 6.1: The propagation time for CSD starting in the visual cortex, the surface area, the wave propagation speed and the SRI for the left and right hemisphere of Subjects 1 to 5.

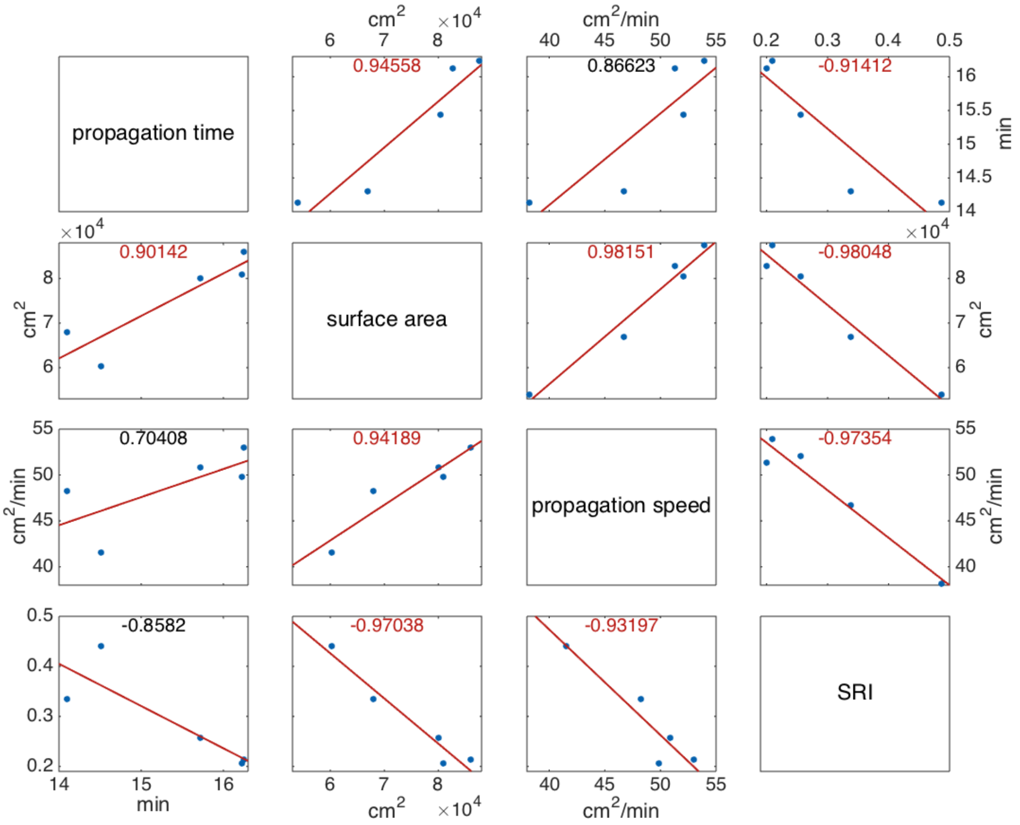


FIGURE 6.8: A correlation plot of the propagation time, the surface area, the wave propagation speed and the SRI for the left and right hemisphere of Subjects 1 to 5. The data of the left hemisphere is plotted in the lower triangular part while the data of the right hemisphere is given in the upper triangular part. The Pearson correlation coefficient is given in every subfigure, coefficients with a p-value below 0.05 are marked in red and the red lines show the linear regression for the given data points.

6.3 Patient group analysis

For the second study, performed in collaboration with Marian Gomez Beldarrain from the Department of Neurology at the Hospital Galdakao-Usansolo in Spain, the available data set consisted of 52 patients.

6.3.1 MRI and DTI data

The data set is anonymised and corresponds to 18 migraine patients without aura, 15 control patients, and 19 patients with chronic headache that have been abusively using painkillers. These patients have been taking analgesics at least 15 times per month in a period of more than three months. The data was acquired at the Hospital Galdakao-Usansolo in Spain, and its acquisition was approved by the local ethics committee. The data was obtained from a Philips 3 T Achieva system with a 32-channel head coil (Philips Medical Systems, Best, the Netherlands), and underwent an anatomical acquisition, a high-resolution structural T1-weighted 3D volume, using a spoiled gradient recalled sequence (SPGR-3D) with the following parameters: TR = 7.4 msec, TE = 3.4 msec; flip angle 8°; matrix size = 228 × 227; FoV = 250 × 250 mm; number of slices = 301; in-plane resolution = 1 × 1 × 1 mm; NSA 1; total acquisition time = 4 min 58 sec.

6.3.2 Data processing

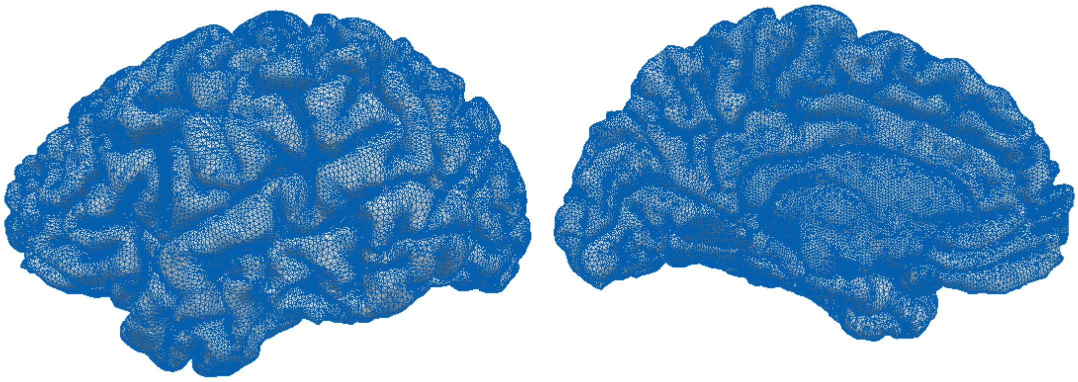
The cortical geometry was obtained from anatomical MRI data with the FreeSurfer image analysis suit in the same set-up as defined previously in Section 6.2.2.

Instead of the Brodmann areas, we use the region label for the Desikan-Killiany atlas defined in the **lh.aparc.annot** and **rh.aparc.annot** files generated by the FreeSurfer reconstruction providing (Desikan et al., 2006).

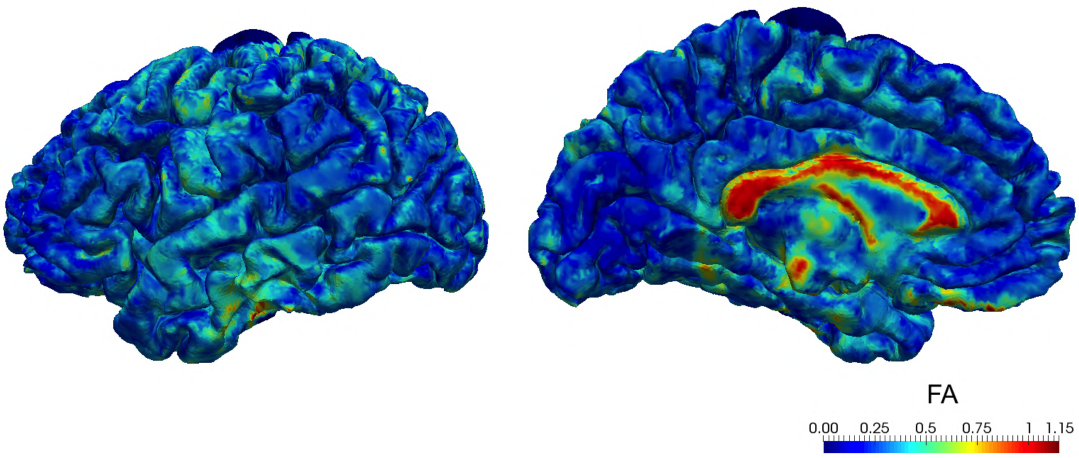
The DTI data processing with the FSL software is very similar to the one for DWI data. After extracting the T1 and DTI brain data with BET, a local fitting of the diffusion tensor was applied to compute the tensor model at each voxel with the function **DTIFIT** from the FSL toolbox. For each voxel, the DTI data provided the three eigenvectors and the associated eigenvalues of the diffusion. The final steps to obtain the diffusion data on the brain mesh is identical for DWI and DTI data. In the case of DTI data this procedure has to be applied to all eigenvalues and eigenvectors, instead of just the ADC values like for the DWI data.

6.3.3 Preprocessing and simulation set-up

After obtaining the brain geometry in the form of a triangulated mesh with diffusion tensors, FA and MD values associated to each grid point, we run the Taubin smoothing with equal weights up to a volume loss of 5 % choosing the scaling parameters as $\lambda = 0.33$ and $\mu = 0.34$. Then, we eliminate the corpus callosum from the geometry for the CSD simulation (see Section 5.3.2). An example of a smooth surface and FA values of the 3D



(a) mesh reconstructed from T1-weighted image



(b) FA values on the surface mesh

FIGURE 6.9: The smoothed mesh reconstructed from T1-weighted image (a) and FA values on the surface mesh (b) for the left hemisphere of a patient from the migraine group.

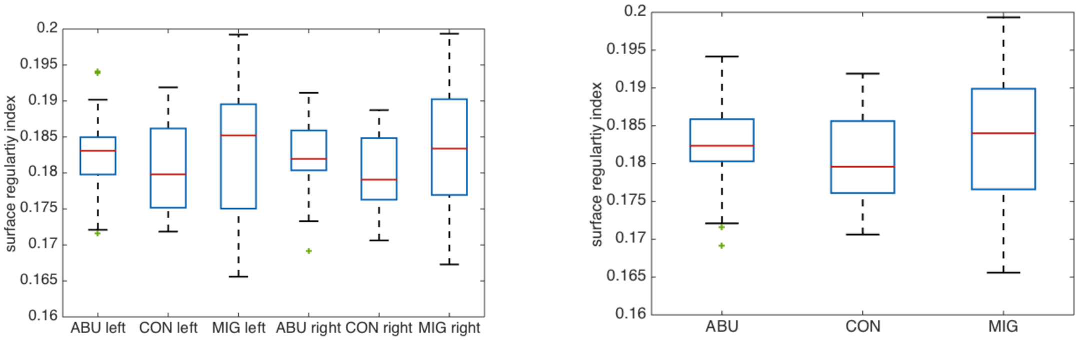
diffusion data for a patient from the migraine group is shown in Figure 6.9 in the lateral and medial view of the left hemisphere.

For the simulation of CSD, we use the technique to obtain 2D diffusion data introduced in Section 4.2.3 and use the same parameter values as proposed in Chapter 3 and Chapter 4. We initiate the wave propagation in all the ROI and record the arrival in each regions' centroid together with the minimum and maximum arrival times as described in Chapter 5.

6.3.4 Results

From the smoothed cortical mesh, we compute the SRI of the migraine patients and compare it to the ones of the other two groups. Note that for these computations we use the smooth geometry that still includes the corpus callosum since the SRI computation depends on the volume enclosed by the cortical surface. Then, we simulate the CSD propagation, compute the QoI introduced in Chapter 5 and compare them between the different patient groups. In the following, we will refer to the patient groups as ABU (abusive painkiller users), CON (control) and MIG (migraine patients without aura).

Surface regularity index We compute the SRI introduced in Section 5.2 for ABU, CON and MIG and visualise it in the form of box plots in Figure 6.10.



(a) left and right hemisphere

(b) whole brain

FIGURE 6.10: Box plots of the surface regularity index of ABU, CON and MIG differentiating between left and right hemisphere (a) and considering the whole brain (b).

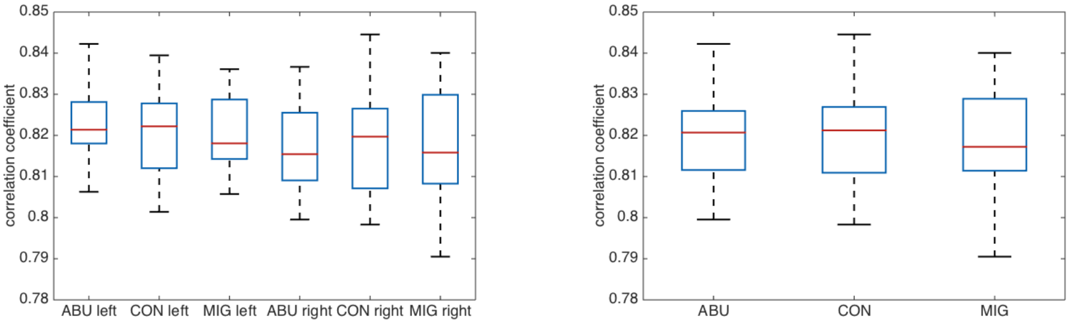
Similarly to what was done in Section 6.2.4 we look for similarities in the distribution of the ABU, CON and MIG data sets, and we do so by using again Welch's t-test. First, we consider left and right hemisphere for the three groups separately. The p-values of Welch's t-test of the SRI for the different pairs of patient groups are displayed in Table 6.2. We then consider the case in which left and right hemisphere are grouped together. All p-values are bigger than 0.05 and indicate weak evidence against the hypothesis of equal means. Consequently, we cannot say that the SRI of these patient groups is significantly different from one another. The only thing we can infer from Figure 6.10 and Table 6.2 is that the SRI of migraine patients varies more and tends to have higher values than in the other patient groups.

patient groups	left hemisphere	right hemisphere	both hemispheres
ABU - CON	0.3696	0.2376	0.1373
MIG - CON	0.4345	0.3283	0.2072
MIG - ABU	0.9325	0.9405	0.9084

TABLE 6.2: The p-values of the Welch's t-test for equal means of the SRI for the different pairs of patient groups, for the left and right hemisphere separately and considering the whole brain.

CSD simulation and QoI For the simulations of the CSD propagation on the personalised cortices and the computation of the QoI, we proceed like in Section 5.3.3. We consider one region as initially activated and record three values for each of the 33 remaining regions: the minimum and the maximum arrival times of each region and the arrival time of the wave in the centroid. Like in the previous chapter, we compute the QoI by post-processing the results from the simulated CSD propagation across the whole cortex.

Correlation between activation times and distances We consider the Euclidean distance between the centroids of two regions as a proxy for the distance traveled by the wavefront. We compute the Pearson correlation coefficient (see equation (5.9)) between propagation time and the distance between the centroids introduced in Section 5.3.3 for ABU, CON and MIG and visualise it in the form of box plots in Figure 6.11. Neither by the box plots nor by using the Welch's t-test we are able to highlight any significant difference between the three groups.



(a) left and right hemisphere (b) whole brain

FIGURE 6.11: Box plots of the Pearson correlation coefficient between the activation times of each region and the distance between the centroids for the three patient groups differentiating between left and right hemisphere (a) and considering the whole brain (b).

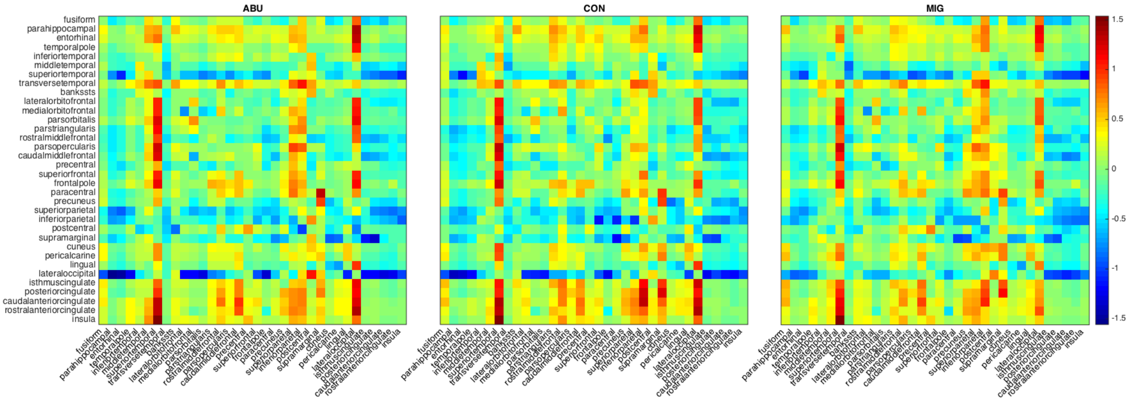
Global asymmetry index To asses the asymmetry of the CSD wave propagation between different regions, we consider the matrix G, see equation (5.11). The mean and standard deviation of this matrix for the different groups are visualised in Figure 6.12.

On average, all patient groups show distinct behaviours in the superiortemporal and lateraloccipital region. Smaller differences can be observed in the transversetemporal, superiorparietal, inferiorparietal and supramarginal region. The standard deviation in MIG and CON exhibits bigger values in the right hemisphere than in the left one.

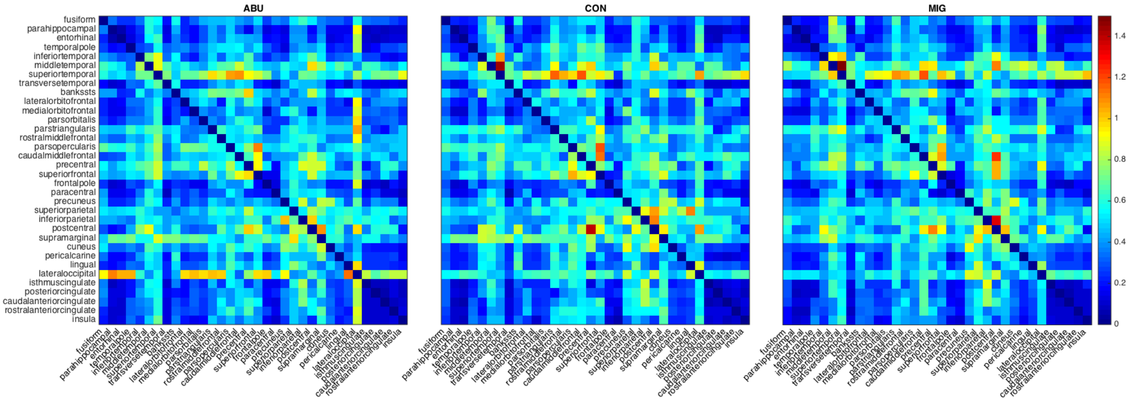
For a better insight into the variability between the different patient groups, we consider the matrices obtained by subtracting the mean of G corresponding to different patient cohorts. The results are shown in Figure 6.13. On the one hand, from this figure we can infer that the behaviour of the middletemporal region and the lateraloccipital region vary substantially both between MIG and CON, and between MIG and ABU. On the other hand, between ABU and CON the differences can mainly be observed in the superiortemporal, inferiortemporal, postcentral and supramarginal region.

Next, we consider the global asymmetry index (GAI) introduced in (5.13). A positive value implies that the corresponding region is, in general, behaving as a source for the propagation of a depolarisation wave, while a negative mean implies that the region is behaving as a sink for the propagation. The mean of this index, namely

$$\mathrm{GAI}^L = \frac{1}{34}\sum_{j=1}^{34} \mathfrak{A}_j^L \qquad \text{and} \qquad \mathrm{GAI}^R = \frac{1}{34}\sum_{j=1}^{34} \mathfrak{A}_j^R \tag{6.1}$$



(a) mean



(b) standard deviation

FIGURE 6.12: The mean (a) and standard deviation (b) of matrix G (5.11) for the different patient group: ABU, MIG and CON, respectively.

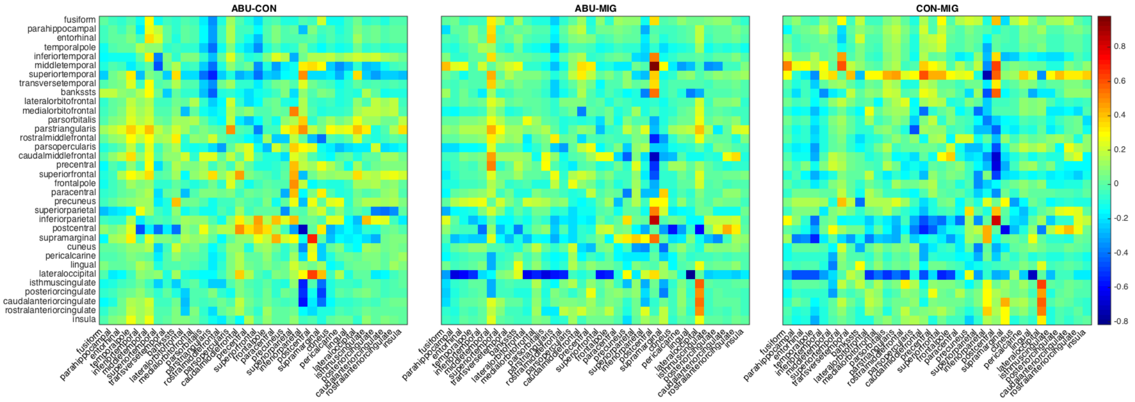
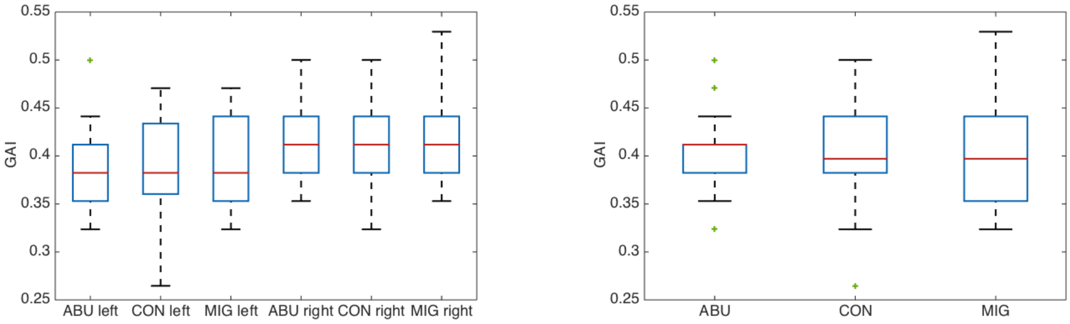


FIGURE 6.13: The difference in the mean of matrix G (5.11) between the different patient groups: ABU, MIG and CON.

gives a way to summarise with only two values the source or sink nature of the considered geometry in the CSD wave propagation for every patient. We visualise the GAI for the three patient groups in box plots in Figure 6.14.

Again, from Figure 6.14 and using Welch's t-test we cannot identify the GAI as a classifier to distinguish the patient groups.



(a) left and right hemisphere

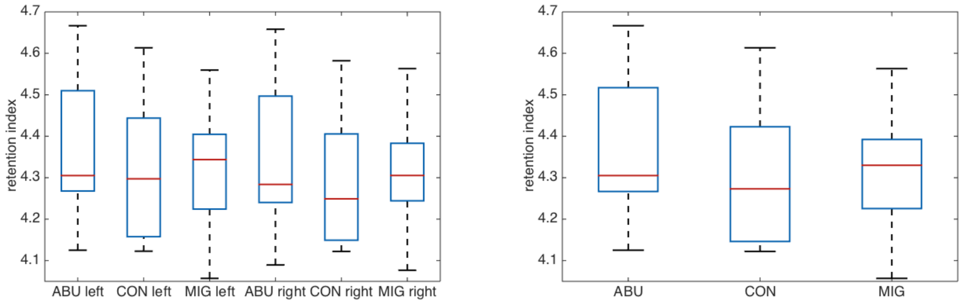
(b) whole brain

FIGURE 6.14: Box plots of the GAI for the three patient groups differentiating between left and right hemisphere (a) and considering the whole brain (b).

Retention index To assess possible differences in the local behaviour, we use the retention index (RI), introduced in (5.15). Because this index gives information about the specific region, we take the mean over all the regions to obtain a value that describes the propagation behaviour in each hemisphere of a patient, namely

$$\mathrm{RI}^L = \frac{1}{34}\sum_{j=1}^{34} \mathfrak{R}_j^L, \qquad \text{and} \qquad \mathrm{RI}^R = \frac{1}{34}\sum_{j=1}^{34} \mathfrak{R}_j^R. \tag{6.2}$$

We visualise the retention index for the different patient groups in the form of box plots in Figure 6.15.



(a) left and right hemisphere

(b) whole brain

FIGURE 6.15: Box plots of the RI for the three patient groups differentiating between left and right hemisphere (a) and considering the whole brain (b).

Again, the box plots in Figure 6.15 and Welch's t-test show that the retention times are not significantly different between the patient cohorts.

Group classification using data mining So far we were unable to distinguish between the patient groups but other techniques, i.e., in data mining might be able to do so. Instead of considering every QoI independently, we consider the data of all QoI as a supervised classification problem in machine learning. We can summarise all the QoI for the different patient groups that we previously discussed in one data set with 52 observations (patients), the three (target) classes ABU, CON and MIG, and 414 numerical features: the average value and standard deviation (std) of the two curvature measures (introduced in Section 5.1), the global asymmetry index $\mathfrak{A}$, the GAI, the retention time $\mathfrak{R}$, the RI and the SRI. The GAI, RI and SRI are scalars describing each hemisphere of a patient, while the other features are values for each region of the left and right hemisphere, respectively. The number and the corresponding name of the ROI can be found in Table D.3 in Appendix D. To analyse the data, we use the Orange data mining software[4] (Demšar et al., 2013). For more details on the supervised classification techniques, we refer to Hastie, Tibshirani, and Friedman, 2009.

Based on all the features, we build a *classification tree*, that is, a series of binary splits of the data set, in order to arrive to a set of criteria that separates the classes. As an example, we visualise a decision tree with depth 5 in Figure 6.16. Nodes are coloured corresponding to the most likely class, and the edge width corresponds to the proportion of patients in the corresponding node with respect to all the instances in the original data.

The decision tree in Figure 6.16 reveals the importance of the features related to the curvature and the retention time, mostly in regions of the left hemisphere. The first split separates the majority of ABU and CON, based on the retention time of the left ROI 31. In the subsequent split the average of the mean curvature in the left ROI 13 and the average Gaussian curvature in the right ROI 34 are used to separate the migraine patients. Although the method fails to completely classify all patients in the correct group, the overall rate to determine the correct group is high. Another way to display this decision tree is to create a *pythagorean tree* which is shown in Figure 6.17. In the pythagorean tree, the sides of each triangle relate to the number of observations split by each decision and the colour of the node denotes the most probable class.

Ranking the features in a classification problem can help to remove noise and focus on relevant information that helps to identify the classes. Ranking methods score the features according to their correlation with the class. In the Orange software, several ranking algorithms are available, like the information gain that gives the expected amount of information and the ANOVA approach that gives the difference between average values of the feature in different classes. The five features ranked highest by the two ranking methods are:

Information gain	**ANOVA**
• std mean curvature left ROI 20	• average mean curvature left ROI 20
• std Gaussian curvature left ROI 33	• std mean curvature left ROI 20
• std Gaussian curvature right ROI 13	• std mean curvature left ROI 33
• retention time left ROI 20	• average Gaussian curvature left ROI 15
• average Gaussian curvature left ROI 7	• average mean curvature left ROI 18

The majority of the important features relate to the left hemisphere and the two curvature measures and the retention time are identified as good classifiers for the three groups. Both ranking methods stress the role of the left ROI 20, with respect to both curvature measures and the retention time. Previously, ROI 7 was already identified as a region displaying significant differences between ABU and CON with respect to the mean of the matrix G, see Figure 6.13.

---

[4] `https://orange.biolab.si`

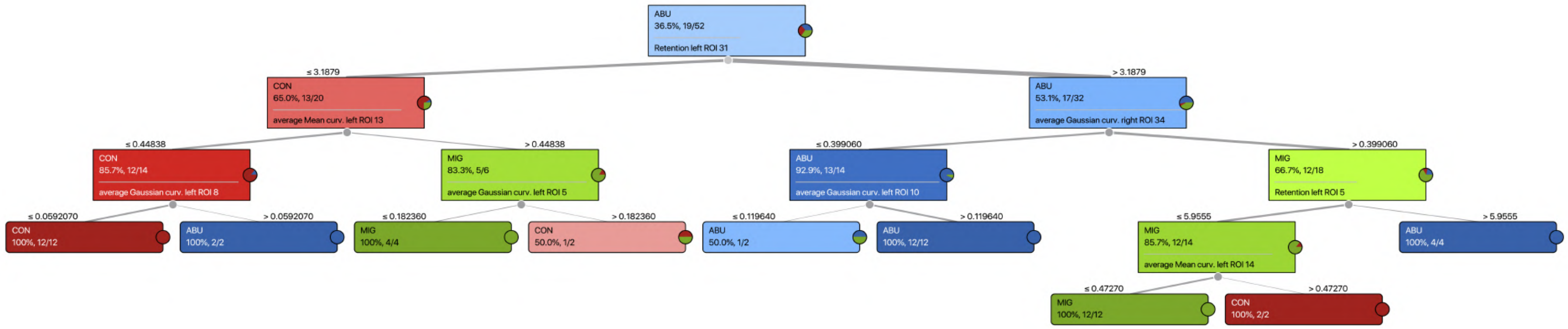


FIGURE 6.16: A decision tree of all features with depth 5 generated with the Orange data mining software for ABU (blue), CON (red) and MIG (green). The node's colour corresponds to the most probable class and the edge width corresponds to the proportion of instances in the corresponding node with respect to all the instances in the original data.

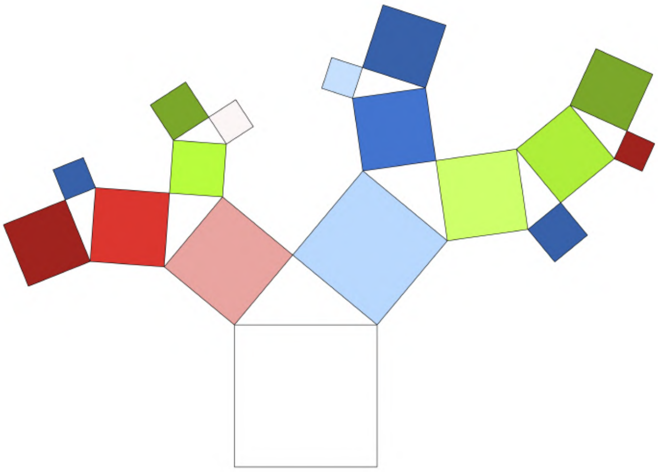


FIGURE 6.17: The pythagorean tree of the decision tree in Figure 6.16 for the three patient groups: ABU (blue), CON (red) and MIG (green). The sides of each triangle relate to the number of observations split by each decision and the node colour denotes the most probable class.

We consider only the 15 best-ranked features (by the two ranking algorithms) and sample from the data with a specific option in the data mining software (cross-validation of 10 folds). The feature reduction improves the *accuracy* of the decision tree, that is, the proportion of correct classification, from 0.327 for all the features to 0.5. For a good classification the accuracy is close to 1. This improvement stresses that some features are more important than others and have a higher potential to distinguish the three patient groups.

To visualise the performance of the classification trees and reveal if certain patients are misclassified, we consider the *confusion matrices*, also called error matrices. Each column of this matrix represents the patients in a predicted group, while rows represent the actual groups the patient belongs to. Correctly identified patients are located on the diagonal of the table, while prediction errors are represented by values outside the diagonal. Evaluating the decision tree with a specific option in the data mining software (cross-validation of 10 folds), the resulting confusion matrices for all features and the reduced features are shown in Table 6.5.

		predicted			
		ABU	**CON**	**MIG**	**Total**
actual	**ABU**	**7**	4	8	19
	CON	3	**6**	6	15
	MIG	7	7	**4**	18
	Total	17	17	18	52

(a) all features

		predicted			
		ABU	**CON**	**MIG**	**Total**
actual	**ABU**	**11**	3	5	19
	CON	4	**9**	2	15
	MIG	8	4	**6**	18
	Total	33	8	11	52

(b) reduced features

TABLE 6.3: The confusion matrices of the decision tree based on all the features (a) and based on the reduced features (b).

These two tables demonstrate that we still make classification errors with the reduced features, but overall the performance of the classification is improved. With reduced features more patients are correctly assigned to their groups. Thus, the important features identified by the ranking methods contribute to the distinction of the patient cohorts.

Even though the accuracy is rather low for both classification approaches, we get a first idea on how the QoI can help to distinguish between the different patient groups.

Obviously, there is a variety of other techniques and measures to evaluate the classification problem in supervised learning. However, this goes beyond the scope of this thesis and is left for future investigations.

6.4 Summary

In this chapter we simulated CSD propagation on different patient data sets. In a first case study of chronic migraine patients with aura, we confirmed that the simulated CSD wavefront propagates according to the sequence of symptoms experienced by the patients during a migraine attack. This is adding computational evidence to the hypothesis of neural correlation between migraine with aura and CSD.

In the subsequent group analysis, we simulated CSD propagation on three different patient groups: migraine patients (without aura), chronic headache patients that abusively use painkillers and a control group. By simply looking at the mean and standard deviation in the QoI introduced in Chapter 5 we were unable to find any feature that significantly distinguishes the patient groups from each other. However, by using classification techniques from data mining, we showed that the two curvature measures and the retention time, especially in the left hemisphere, could be good indicators to differentiate between the considered patient groups. Different ranking methods indicate that these three QoI in the paracentral region (ROI 20) are important features to classify the three patient groups.

Chapter 7

Uncertainty quantification

"We must think differently about our ideas – and how we test them.
We must become more comfortable with probability and uncertainty.
We must think more carefully about the assumptions and beliefs that we bring to a problem."

Nate Silver, in *The Signal and the Noise: The Art of Science and Prediction (2012)*

In the previous chapters we simulated CSD propagation on the cerebral cortex using the model for the extracellular potassium concentration introduced in Chapter 3. The model parameters were tuned to match the characteristics of CSD, including the interval length of high potassium concentration and the overall propagation time across the cortex. By choosing such parameter values, we enforce strong assumptions on the model and its results, but unfortunately, we are unable to validate more thoroughly the obtained results against experiments due to a lack of detailed experimental data. To address this limitation, we apply uncertainty quantification (UQ) to account for the uncertainties in the choice of parameters and perform a sensitivity analysis to explore the parameter space for both a proper tuning of the model and for the improvement of its predictive role. We distinguish between the spatially independent and space-dependent parameters. The former can be considered random variables while the latter is considered a random process.

To estimate the impact of parameter uncertainties on the solution, we apply different techniques of UQ. By assigning certain parameter distributions rather than fixing parameter values, UQ offers sophisticated and efficient methods to explore the parameter space. Note that the selection of a specific parameter distribution still implies having made a particular modelling assumption. However, this guess is clearly less restrictive than selecting a constant parameter value.

The basic idea in UQ is to draw samples from the parameter space according to the assumed distribution, evaluate the problem at hand with these parameter values and compute some output statistics, e.g., by considering the mean and the variance of the obtained model outputs. Especially for a high number of parameters, exploring the parameter space can be costly, but UQ techniques offer efficient strategies to pick samples from this parameter space and reduce the computational effort.

In this chapter we first assume the parameters to be uniformly distributed and study the effect on the solution in this context. Then we briefly comment on the solution statistics when an underlying beta-distribution is assumed. The choice of the two distributions is motivated by the domain they are defined on, continuous finite intervals as opposed to distributions on discrete or continuous but infinite supports. This choice is a natural one for parameters describing biological phenomena where experiments can narrow down the parameter values to reasonable ranges.

The sensitivity analysis concerns the investigation of the sensitivity of simulation results to changes in the different parameters. We answer the question of how much the different parameters contribute to the solution and to other characteristics of the potassium wave, such as the period of high potassium concentration and the maximum potassium concentration.

The diffusion direction and strength of the potassium wave are ruled by DTI measurements. Due to errors in the measurements and the pre-processing, there is noise in the data. Thus, UQ offers a way to account for these uncertainties in the diffusion parameter. However, the diffusion data is spatially dependent and applying UQ is more complex than for the other parameters that do not depend on space. We give a brief idea and introduction to what could be done to address this problem.

In this chapter we first recapitulate the model for the extracellular potassium concentration in the deterministic sense before exploring the parameter space with UQ (for both spatially independent and dependent parameters) and before analysing the solution's sensitivity to the different parameters. The results of this chapter were obtained during my visiting fellowship undertaken under the supervision of Prof. Dr. George Em Karniadakis at the Brown University in Providence, US.

For a basic introduction to UQ and sensitivity analysis, we refer to Appendix B.4. The notations in this chapter are mainly taken from Xiu, 2010.

7.1 Deterministic problem

The deterministic model for the temporal evolution and propagation of the extracellular potassium k_{bath} introduced in Chapter 3 (see system (3.4)) reads

$$\begin{aligned}
\frac{\partial k_{\text{bath}}}{\partial t} &= \operatorname{div}\left(D\nabla k_{\text{bath}}\right) - F(k_{\text{bath}}, w) + I_p,\\
\frac{\partial w}{\partial t} &= G(k_{\text{bath}}, w),\\
F(k_{\text{bath}}, w) &= \eta_1\left(k_{\text{bath}} - k_0\right)\left(1 - \frac{k_{\text{bath}}}{k_{th}}\right)\left(1 - \frac{k_{\text{bath}}}{k_p}\right) + \eta_2\left(k_{\text{bath}} - k_0\right)w,\\
G(k_{\text{bath}}, w) &= \eta_3\left(k_{\text{bath}} - k_0 - \eta_4 w\right),
\end{aligned} \tag{7.1}$$

where k_{bath} and w are defined on $W \times (0, T)$ with $W \subset \mathbb{R}^r$ for $r = 1, 2, 3$, $D \in \mathbb{R}^{r\times r}$ is the diffusion tensor, w is a recovery variable, η_1, η_2, η_3, and η_4 are parameters, while k_0, k_{th}, k_p are the resting, threshold, and peak values of the k_{bath} concentration, respectively. The parameters k_0, k_{th} and k_p are chosen in a way that the resulting potassium wave triggers the CSD-like behaviour in the coupling with the Wei model. The diffusion tensor D is space-dependent and based on DTI measurements while the remaining parameters $\eta_1, \ldots, \eta_4$ have been tuned to match the characteristics of CSD. We start by considering the spatially independent parameters only, neglecting the spatial dimension and focusing merely on the temporal evolution of the problem.

We solve problem (7.1) with the parameter values given in Table 3.1 and set as initial conditions $k^0_{\text{bath}} = k_{\text{bath}}(0) = 4$ mM and $w^0 = w(0) = 0$. Letting $\mathbb{1}_{[a,b]}(t)$ denote the indicator function on an interval $[a, b]$, we add a perturbation $I_p(t) = \mathbb{1}_{[1.8,2]}(t)$ mM sec^{-1} to trigger the onset of the extracellular potassium wave. The result of this simulation is given in Figure 7.1.

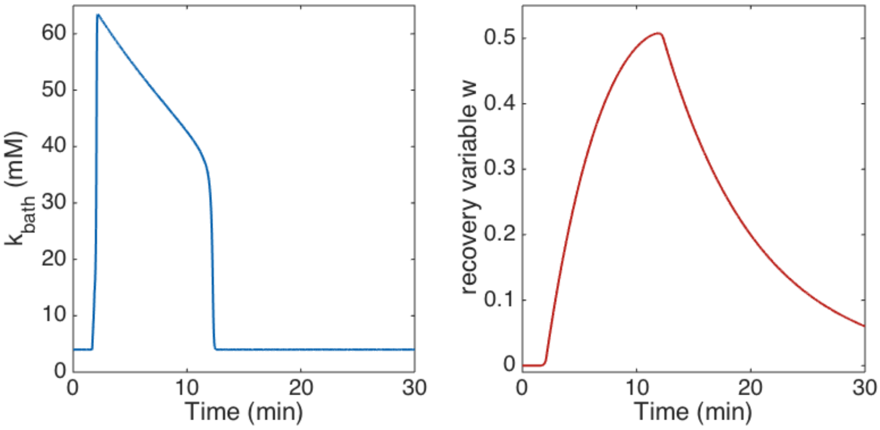


FIGURE 7.1: The temporal evolution of the deterministic solution of the extracellular potassium k_{bath} and the recovery variable w.

7.2 Uncertainties in spatially independent parameters

We start by considering the model for the extracellular potassium propagation (7.1) without spatial component and consider the four parameters $\eta_1(\omega)$, $\eta_2(\omega)$, $\eta_3(\omega)$ and $\eta_4(\omega)$ as mutually independent random variables. Even though k_0, k_{th} and k_p could also be assumed to be random, we restrict this study to uncertainties in η_1, η_2, η_3, and η_4. With these assumptions system (7.1) only depends on time and on the random input parameters, and can hence be rewritten as:

$$\begin{aligned} \frac{dk_{\text{bath}}}{dt}(t;\omega) &= I_p(t) - \eta_1(\omega)(k_{\text{bath}}(t;\omega) - k_0)\left(1 - \frac{k_{\text{bath}}(t;\omega)}{k_{th}}\right)\left(1 - \frac{k_{\text{bath}}(t;\omega)}{k_p}\right) \\ &\quad -\eta_2(\omega)(k_{\text{bath}}(t;\omega) - k_0)w(t;\omega) \\ \frac{dw}{dt}(t;\omega) &= \eta_3(\omega)\left(k_{\text{bath}}(t;\omega) - k_0 - \eta_4(\omega)w(t;\omega)\right), \end{aligned} \tag{7.2}$$

with $\omega \in \Omega$ where Ω defines the sample space. Thus, the input random variables can be summarised as $Z(\omega) = (\eta_1(\omega), \eta_2(\omega), \eta_3(\omega), \eta_4(\omega)) \in \mathbb{R}^4$. Denoting by $I_Z \subset \Omega$ the support of Z that depends on the assumption on the parameter distribution and will be defined later on. With this, the solutions k_{bath} and w are defined on $(0, T) \times I_Z$.

In the following, we will refer to the solution of system (7.1) with fixed parameters as the deterministic solution as opposed to the solution of (7.2) where the parameters are considered as random variables.

The main idea of UQ is to draw samples from the parameter space according to the assumed distribution and compute the solution of problem (7.2) with these parameter values, which we will refer to as collocation points. These collocation points and their corresponding weights for the different distributions and methods are all obtained from the Multi-Element Probabilistic Collocation Method Package[1] (V1.01) developed by the Crunch group of Prof. Dr. George Em Karniadakis at Brown University in Providence, US (Foo and Karniadakis, 2010; Foo, Wan, and Karniadakis, 2008).

7.2.1 One-dimensional parameter space

Here, we work with the one-dimensional parameter spaces with uniformly distributed values and study separately the impact of each of the previously mentioned parameters while keeping the other three ones fixed.

If not stated differently, we assume one of the parameters $\eta_1(\omega)$, $\eta_2(\omega)$, $\eta_3(\omega)$ and $\eta_4(\omega)$ to be uniformly distributed on an interval ranging from -10% to +10% of their previously chosen value that we indicate as $\bar{\eta}_i$ (see Table 3.1), that is,

$$\eta_i \sim \mathcal{U}(0.9\,\bar{\eta}_i, 1.1\,\bar{\eta}_i), \qquad \text{for } i = 1, \ldots, 4. \tag{7.3}$$

By making this choice of distribution, we are assuming that our first guess of the parameter values was already reasonable but not perfect.

Monte Carlo method One of the most popular and well-developed methods for solving stochastic differential equations is the Monte Carlo method. Even though it converges relatively slowly, with a convergence rate of $1/\sqrt{M}$ for a simulation consisting of M samples, its performance is independent from the number of random variables, that is the dimension of the parameter space. The general procedure to apply Monte Carlo sampling is:

1. For a given number of samples M, generate independent, identically distributed random variables $\{Z_i^j\}_{i=1}^d = \{Z_i(\omega_j)\}_{i=1}^d$ for $j = 1, \ldots, M$, where d is the dimension of the parameter space.
2. For each $j = 1, \ldots, M$ solve the deterministic problem (7.1) with $Z^j = (Z_1^j, \ldots, Z_d^j)$ obtaining the solution $k_{\text{bath}}(t, Z^j)$.
3. Post-process the results to obtain the solution statistics, such as the solution mean

$$\mathbb{E}(k_{\text{bath}}(t)) \approx \frac{1}{M} \sum_{j=1}^{M} k_{\text{bath}}(t, Z^j)$$

or the solution variance

$$\mathbb{V}(k_{\text{bath}}(t)) \approx \frac{1}{M} \sum_{j=1}^{M} (k_{\text{bath}}(t, Z^j) - \mathbb{E}(k_{\text{bath}}(t)))^2.$$

It is important to take into account that for each realisation $j = 1, \ldots, M$, step 2 is a deterministic problem which has to be solved by a suitable scheme. Hence, the computational

---

[1] `www.cfm.brown.edu/crunch`

cost of the Monte Carlo method does not only depend on the number of samples M but also on the complexity of the solution of the deterministic problem. As the samples are drawn randomly this requires M to be a large number in order to obtain significant statistics. We plot the estimated mean and standard deviation computed with the Monte Carlo method for $M = 10, 100, 10^4$ samples of the variable $\eta_1(\omega) \sim \mathcal{U}(0.9\,\bar{\eta}_1, 1.1\,\bar{\eta}_1)$ in Figure 7.2. The black dotted line marks the solution corresponding to the original parameter set (see Table 3.1).

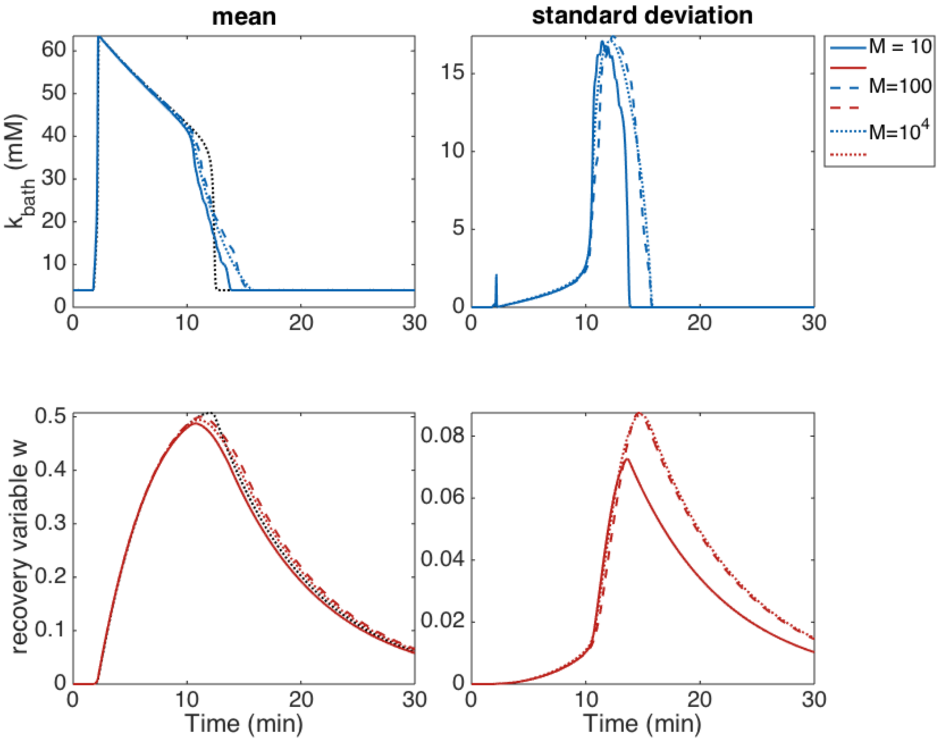


FIGURE 7.2: The solution statistics with the Monte Carlo method with $M = 10$, $M = 100$ and $M = 10^4$ realisations of the uniformly distributed parameter $\eta_1(\omega)$ on $[0.9\,\bar{\eta}_1, 1.1\,\bar{\eta}_1]$ with the other parameters (deterministic) as listed in Table 3.1, and the original solution (dotted black).

In Figure 7.2 we can observe two things: the Monte Carlo method provides a smoother mean solution for a higher number of samples and the parameter η_1 influences the extracellular potassium wave when values are returning to the resting state concentration.

Probabilistic collocation method Instead of drawing numerous samples from a random distribution with the Monte Carlo approach, other methods approximate the solution on a small number of (well-chosen) points in the parameter space. The probabilistic collocation method (PCM) is one of them and it relies on the basic concept of approximating the response of the model by a polynomial. For PCM the model equations are solved at a set of *collocation points*, also called *nodes*, in the parameter space.

Let $(Z^1, \dots, Z^m)$ be a set of collocation points and $\{L_j\}_{j=1}^m$ the corresponding Wiener-Askey scheme polynomials on the support I_Z. An example of polynomials of this scheme are the Legendre polynomials that are defined on $[-1, 1]$. We will see later on that these polynomials can be exploited when considering uniformly distributed parameters. We can approximate k_{bath} and w by

$$\hat{k}_{\text{bath}}(t) = \sum_{j=1}^{m} k_{\text{bath}}(t, Z^j) L_j(Z) \quad \text{and} \quad \hat{w}(t) = \sum_{j=1}^{m} w(t, Z^j) L_j(Z).$$

The collocation points and polynomials are chosen in correspondence to the assumed underlying distribution of the parameters. With these approximations the statistics of the solution can be evaluated, e.g.,

$$\begin{aligned} \mu(t) := \mathbb{E}(\hat{k}_{\text{bath}}(t)) &= \int_{I_Z} \hat{k}_{\text{bath}}(t)\rho(Z)\, dZ \\ &= \sum_{j=1}^{m} k_{\text{bath}}(t, Z^j) \int_{I_Z} L_j(Z)\rho(Z)\, dZ, \end{aligned}$$

where ρ is the probability density function (PDF) corresponding to the assumed distribution and

$$\begin{aligned} \sigma^2(t) := \mathbb{V}(\hat{k}_{\text{bath}}(t)) &= \mathbb{E}\left(\hat{k}^2_{\text{bath}}(t)\right) - \mu^2(t) \\ &= \sum_{j=1}^{m} k^2_{\text{bath}}(t, Z^j) \int_{I_Z} L_j(Z)\rho(Z)\, dZ - \mu^2(t). \end{aligned}$$

The same representations hold analogously for w. Consequently, applying PCM is equivalent to solving m deterministic problems – namely system (7.1) – at each node Z^j in the parameter space.

Assuming to work with a single parameter (i.e., $d = 1$) and assuming this parameter to be uniformly distributed on the interval $[a, b]$, the solution statistics can be simplified to

$$\begin{aligned} \mu(t) &\approx \frac{1}{b-a} \sum_{j=1}^{m} k_{\text{bath}}(t, Z^j) q_j, \\ \sigma^2(t) &\approx \frac{1}{b-a} \sum_{j=1}^{m} k^2_{\text{bath}}(t, Z^j) q_j - \mu^2(t). \end{aligned}$$

where Z^j for $j = 1, \dots, m$ are the roots of the Legendre polynomials and q_j the corresponding weights.

Assuming that the parameter η_1 is uniformly distributed, we apply PCM with $m = 3, 10, 20$ Legendre collocation nodes. The solution statistics are shown in Figure 7.3.

We analyse separately the impact of each of the four parameters $(\eta_1, \dots, \eta_4)$ on the solution by comparing the solution statistics for $\eta_1, \dots, \eta_4$ being uniformly distributed on different intervals. Choosing big intervals or intervals away from the values in Table 3.1 results in solutions that do not exhibit the typical behaviour of the potassium wave: some do not reach the maximum, others do not return to the baseline concentration. Assuming that our first guess on the parameters is a reasonable one, we define the parameters as uniformly distributed on the following intervals:

$$\begin{aligned} \eta_i(\omega) &\sim \mathcal{U}\left(0.8\,\bar{\eta}_i, \bar{\eta}_i\right), \\ \eta_i(\omega) &\sim \mathcal{U}\left(0.9\,\bar{\eta}_i, 1.1\,\bar{\eta}_i\right), \qquad \text{for } i = 1, \dots, 4 \\ \eta_i(\omega) &\sim \mathcal{U}\left(\bar{\eta}_i, 1.2\,\bar{\eta}_i\right). \end{aligned} \tag{7.4}$$

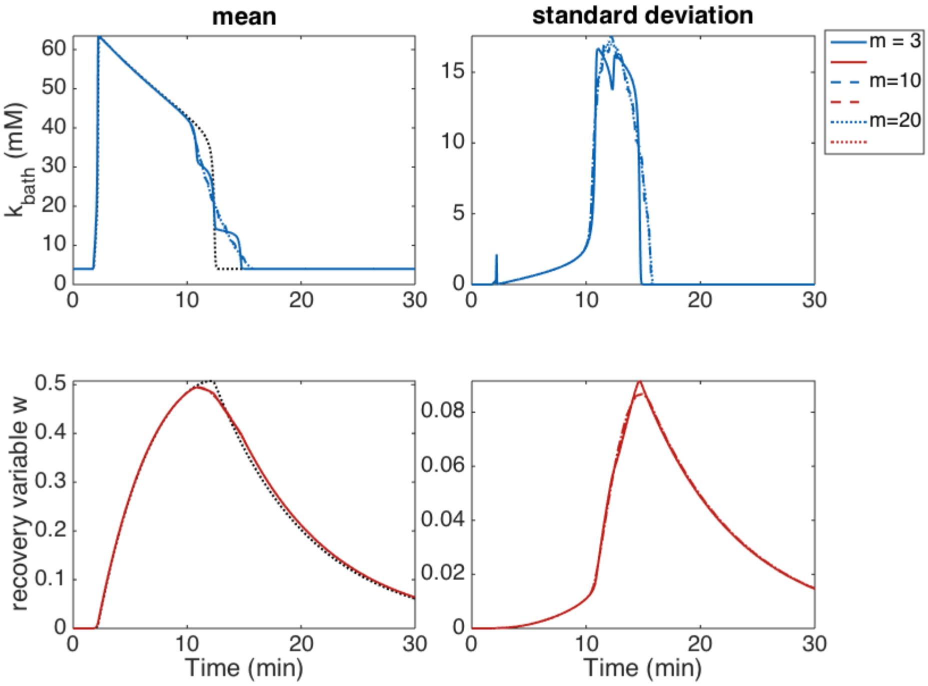


FIGURE 7.3: The solution statistics with PCM for $m = 3$, $m = 10$ and $m = 20$ Legendre collocation points with parameter as listed in Table 3.1 and η_1 uniformly distributed on $[0.9\,\bar{\eta}_1, 1.1\,\bar{\eta}_1]$, and the original solution (dotted black).

Figures 7.4 - 7.7 show the mean and standard deviation of the solution corresponding to PCM with $m = 10$ applied to the different parameters. In the case of η_1 and η_2 we can observe that sampling in an interval of larger values can result in a mean solution that does not return to the low potassium concentration once the wave has passed. These figures show that all the parameters have an impact on the time period of the high potassium concentration.

In the following we choose the parameters as uniformly distributed on the intervals $\eta_i \sim \mathcal{U}\,(0.9\,\bar{\eta}_i, 1.1\,\bar{\eta}_i)$ for $i = 1, \ldots, 4$ so that the parameter choice used in the previous chapters is the average value of this distribution.

In Figures 7.4 - 7.7 the higher number of collocation points results in a smoother mean of the solution but requires a higher computational costs. Even though in the one-dimensional case the difference between computing 10 or 20 collocation points is not significant, considering all four parameters of interest simultaneously would mean refining by a factor 2 in each dimension of a 4-dimensional tensor grid. Increasing the number of collocation points by a factor 2 increases the computation time 10-fold and causes memory issues due to big matrices that save the solution at every time step in all the collocation points.

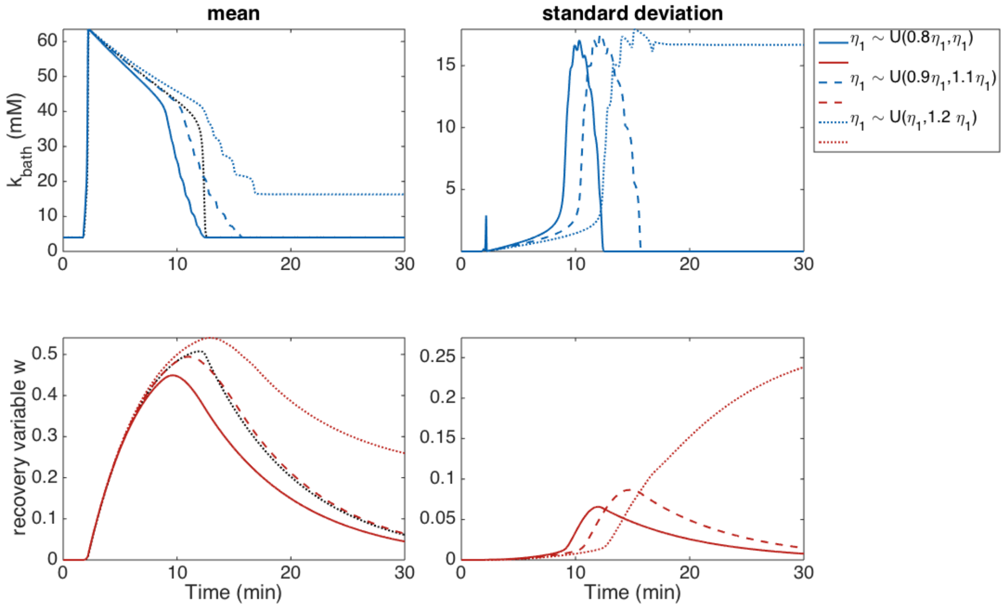


FIGURE 7.4: The solution statistics with PCM for $m = 10$ Legendre collocation points with parameter as listed in Table 3.1 and η_1 uniformly distributed on different intervals defined in (7.4), and the original solution (dotted black).

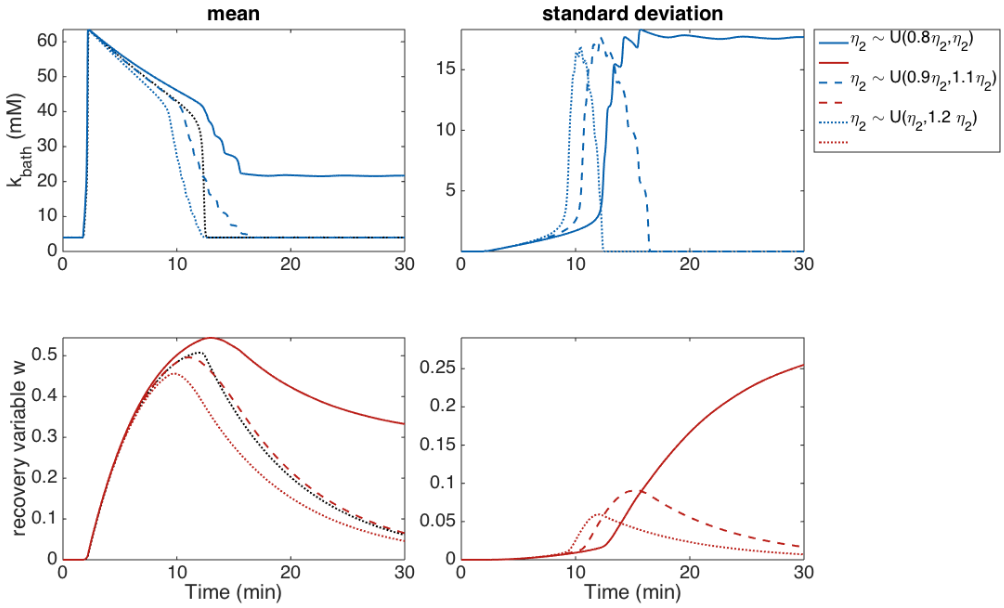


FIGURE 7.5: The solution statistics with PCM for $m = 10$ Legendre collocation points with parameter as listed in Table 3.1 and η_2 uniformly distributed on different intervals defined in (7.4), and the original solution (dotted black).

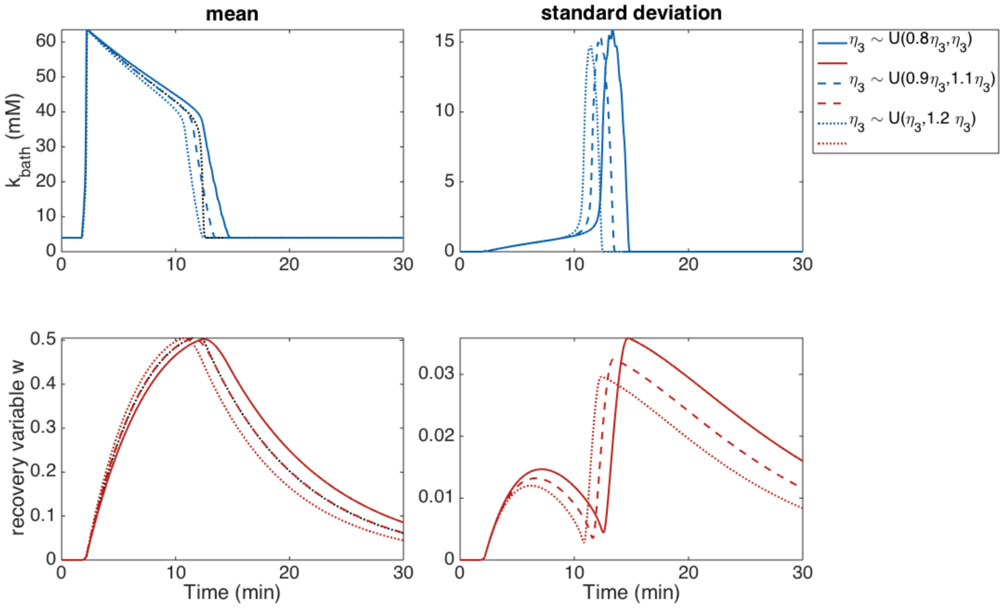


FIGURE 7.6: The solution statistics with PCM for $m = 10$ Legendre collocation points with parameter as listed in Table 3.1 and η_3 uniformly distributed on different intervals defined in (7.4), and the original solution (dotted black).

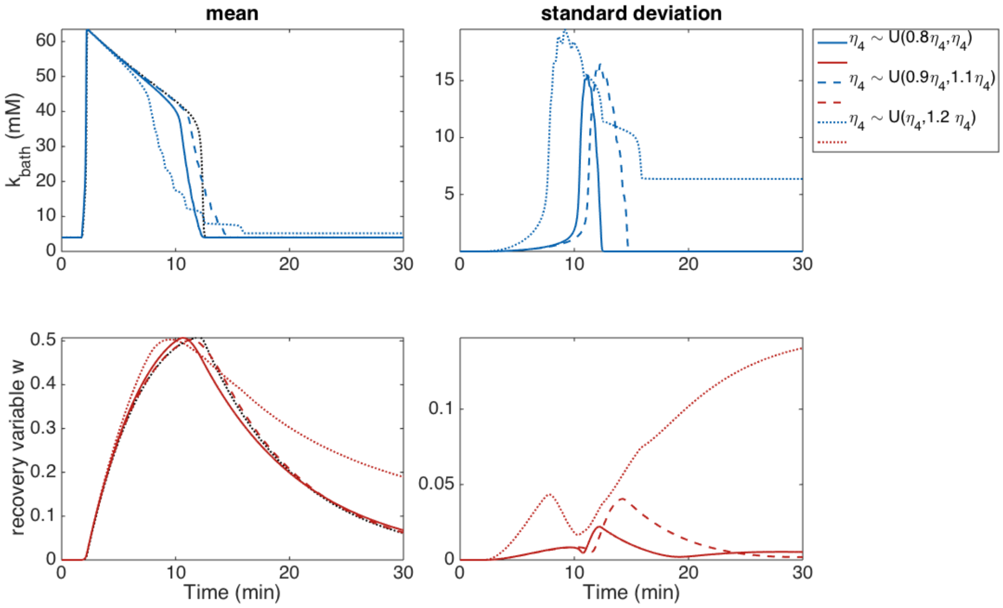


FIGURE 7.7: The solution statistics with PCM for $m = 10$ Legendre collocation points with parameter as listed in Table 3.1 and η_4 uniformly distributed on different intervals defined in (7.4), and the original solution (dotted black).

7.2.2 Four-dimensional parameter space

To investigate the joint impact of the parameters on the solution we consider now the four-dimensional parameter space. In dimensions higher than 1, the most natural choice of collocation points is the tensor product of the one-dimensional collocation set. For a general formulation in each of the d dimensions, we choose a set of m_i collocation points $(Z_i^1, \dots, Z_i^{m_i})$, $i = 1, \dots, d$ selected from the support I_{Z_i}. Then k_{bath} (and analogously w) can be approximated as

$$\hat{k}_{\text{bath}}(t) = \sum_{j_1=1}^{m_1} \dots \sum_{j_d=1}^{m_d} k_{\text{bath}}(t, Z_1^{j_1}, \dots, Z_d^{j_d}) L_{j_1} \dots L_{j_d}(Z_1^{j_1}, \dots, Z_d^{j_d}),$$

where $L_{j_1}, \dots, L_{j_d}$ are the multi-dimensional polynomials from the Wiener-Askey scheme. With this approximation the statistics of the solution can be evaluated, e.g.,

$$\begin{aligned}
&\mathbb{E}\left(\hat{k}_{\text{bath}}(t)\right) = \\
&\int_{I_{Z_1}} \dots \int_{I_{Z_d}} k_{\text{bath}}(t, Z_1^{j_1}, \dots, Z_d^{j_d}) \rho(Z)\, dZ = \\
&\sum_{j_1=1}^{m_1} \dots \sum_{j_d=1}^{m_d} k_{\text{bath}}(t, Z_1^{j_1}, \dots, Z_d^{j_d}) \int_{I_{Z_1}} \dots \int_{I_{Z_d}} L_{j_1} \dots L_{j_d}(Z_1^{j_1}, \dots, Z_d^{j_d}) \rho(Z)\, d(Z),
\end{aligned}$$

where ρ is the multi-dimensional PDF corresponding to the assumed distribution. The same representation holds analogously for w.

Assuming uniformly distributed parameters on the intervals $I_{Z_i} = [a_i, b_i] \subset \mathbb{R}$ the collocation points are the Legendre nodes with the corresponding weights and we can evaluate the solution statistics, as previously done in the one-dimensional case.

For a total of $M = m_1 \cdot \ldots \cdot m_d$ collocation points the expected value of the extracellular potassium concentration is given by

$$\mathbb{E}\left(\hat{k}_{\text{bath}}(t)\right) \approx \left(\prod_{i=1}^{d} (b_i - a_i)^{-1}\right) \sum_{j_1=1}^{m_1} \dots \sum_{j_d=1}^{m_d} k_{\text{bath}}(t, Z_1^{j_1}, \dots, Z_d^{j_d}) q_{j_1} q_{j_2} q_{j_3} q_{j_d},$$

where $d = 1, \dots, 4$ is the dimension of the parameter space, $Z_i^{j_i}$ for $i = 1, \dots, d$ are the roots of the Legendre polynomials and q_{j_i} are the corresponding weights. The variance can be approximated in a similar manner.

Assuming the parameters to be uniformly distributed on the intervals defined in (7.3), we increase the dimension of the parameter space step by step from η_1 to η_4, considering a tensor grid. In particular, with one dimension we just consider η_1 as a random variable, with two dimensions we consider η_1 and η_2 as random variables and so on. The mean and variance of the solution obtained with PCM on 10 Legendre points in each dimension are given in Figure 7.8. With increasing dimensions in the parameter space the mean and standard deviation of the solutions deviate more from the original solution. In dimensions higher than one the mean of the solutions of k_{bath} does not return to the low potassium concentration after the wave has passed.

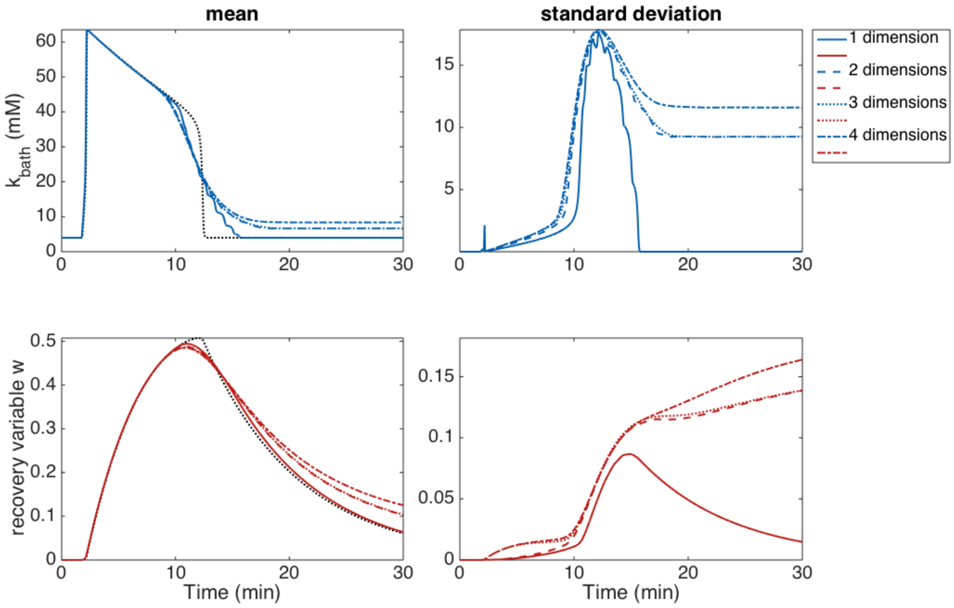


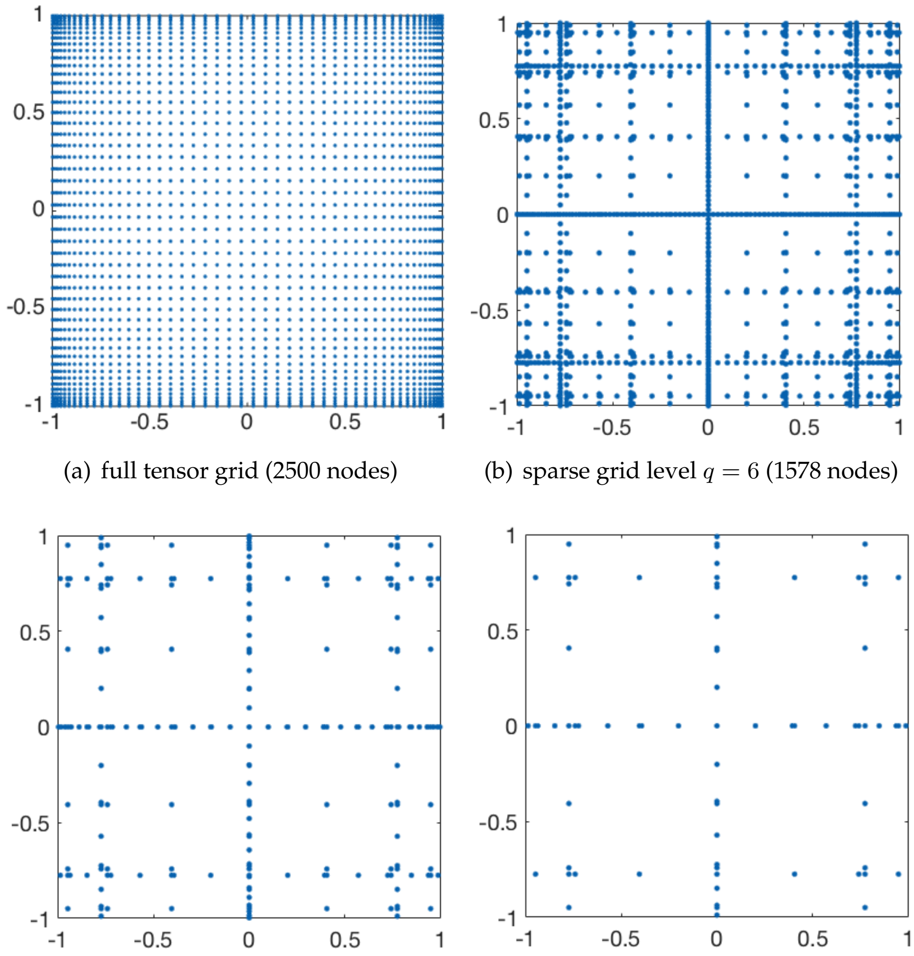
FIGURE 7.8: The solution statistics with PCM for $m = 10$ Legendre collocation points for different dimensions $d = 1, \ldots, 4$. The dimension of the parameter space is increased step by step: one dimension considers only η_1 as a random variable, two dimensions consider η_1 and η_2 as random variables and so on. The original solution is plotted in dotted black.

With increasing dimensions, the number of collocation points and thus the number of times problem (7.2) has to be solved increases which is also known as the curse of dimensionality. A possible way to cope with this issue is to use sparse grids, which reduce the number of collocation points in higher dimensional parameter spaces.

7.2.3 Sparse grids

For the sparse grids, we use again Legendre polynomials and compute the collocation points with the Smolyak algorithm. Details on sparse grids can be found in Xiu and Hesthaven, 2005 and in Appendix B.4. We visualise the full tensor grid and sparse grids at different levels in Figure 7.9. The level of a sparse grid decreases as the sparsity increases. In Figure 7.10 we show collocation points and the corresponding weights on sparse grids in 2D and 3D. The total number of points in the 2D case is 1578 whereas the number of nodes in the 3D case is 6507. The collocation points are coloured according to their weights. Note that some of the weights are negative.

In Figure 7.11 we show the solution statistics evaluated on sparse grids of different levels. Here, the solutions computed on a sparse grid exhibit an irregular behaviour rather than a smooth decrease in the potassium concentration. In contrast to the case of a full tensor grid, the collocation points of the sparse grid can be associated with negative weights. When several solutions assigned to negative weight in the parameter space return to the low potassium concentration, the impact of the solutions with positive weights increases and leads to a sudden increase in the mean value. To visualise the effect of the negative weights of the sparse grids on the solution statistics, we show in Figure 7.12 the mean and some solutions associated with negative weights on a four-dimensional sparse grid

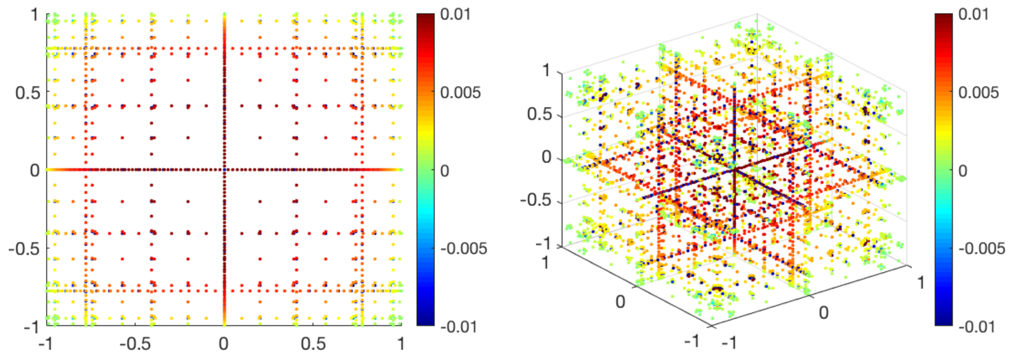


(a) full tensor grid (2500 nodes) (b) sparse grid level $q = 6$ (1578 nodes)

(c) sparse grid level $q = 4$ (224 nodes) (d) sparse grid level $q = 3$ (75 nodes)

FIGURE 7.9: The full tensor grid in two dimensions (a) and the sparse grid based on the extrema of the Legendre polynomials at level $q = 6$ (b), $q = 4$ (c) and $q = 3$ (d).

of level 4. In Figure 7.13 we compare the results of the sparse grid with the ones from the full tensor grid. We have seen that sparse grids reduce the computational effort by reducing the number of collocation points at which the solution has to be computed but they result in irregular solutions. Thus, there is a trade-off between fast computations and smooth solutions. As the problem at hand only features a four-dimensional parameter space, computations on a tensor grid are not very expensive and we opt for the smooth solution. For example, determining the time of high potassium concentration could be problematic with a solution that does not decrease smoothly after the potassium wave has passed.



(a) two-dimensional sparse grid (b) three-dimensional sparse grid

FIGURE 7.10: Two-dimensional and three-dimensional sparse grids based on the extrema of the Legendre polynomials at level $q = 6$. The total number of points in the 2D case is 1578 (a), whereas the number of nodes in the 3D case is 6507 (b). The collocation points are coloured according to their weights.

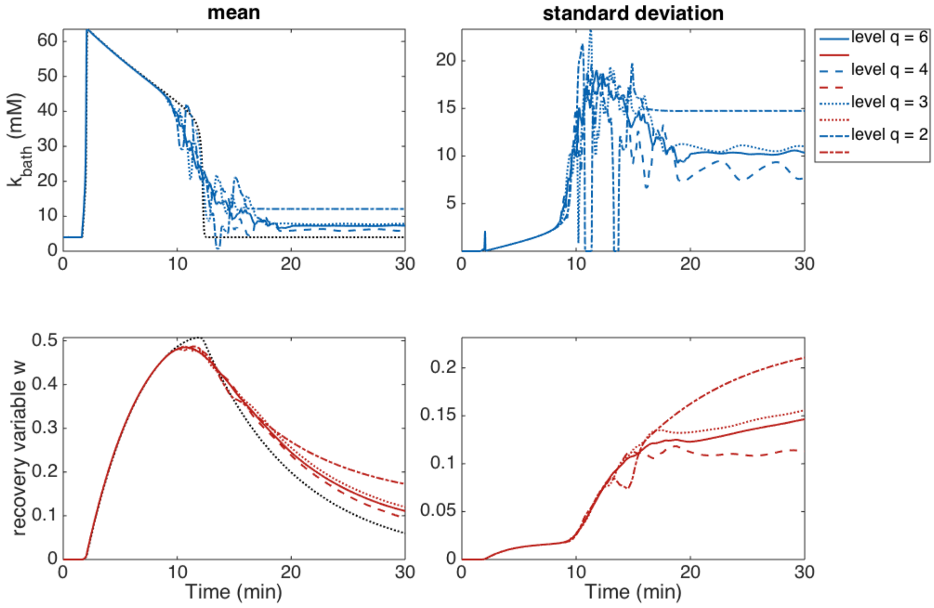


FIGURE 7.11: The solution statistics with PCM on a 4D sparse grid of level 6 (18976 collocation points), level 4 (1268 collocation points), level 3 (289 collocation points) and level 2 (57 collocation points) and $\eta_1, \ldots, \eta_4$ uniformly distributed as defined in (7.3). The original solution is plotted in black dots.

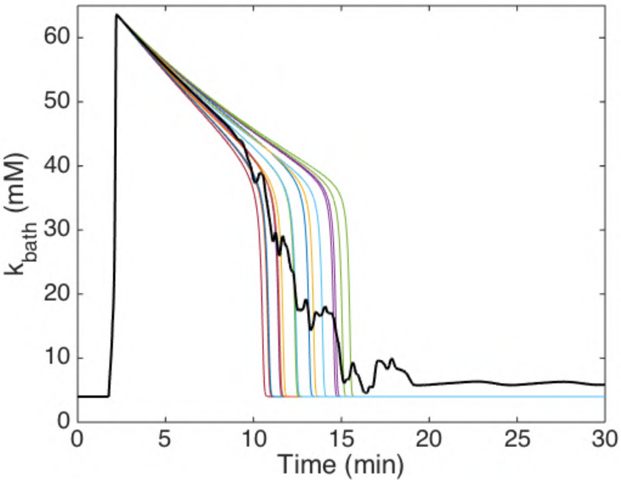


FIGURE 7.12: The mean solution (black) obtained with PCM on a 4D sparse grid of level 4, and 20 solutions corresponding to collocation points associated with negative weights (plotted in colour).

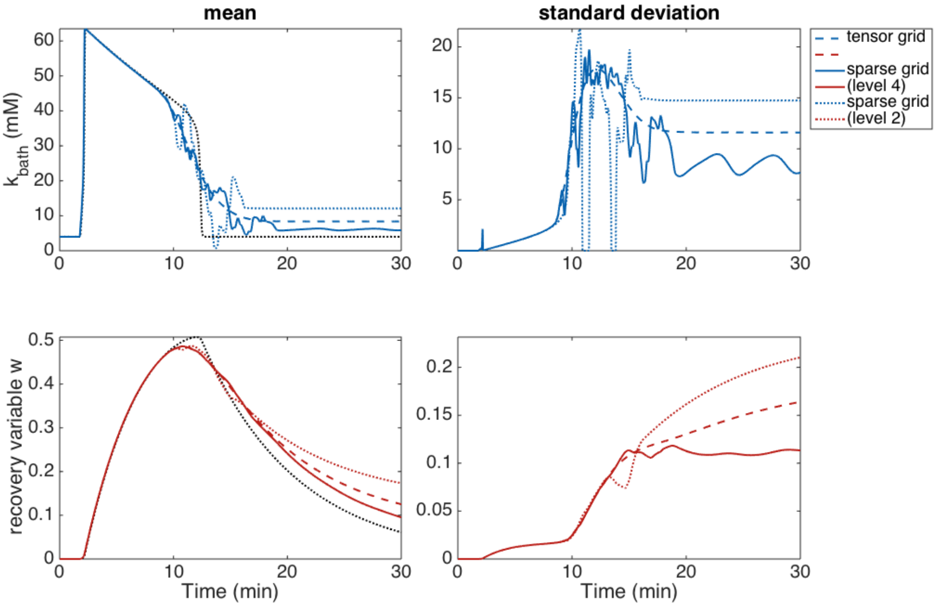


FIGURE 7.13: The solution statistics with the PCM on a 4D tensor grid with $M = 10^4$ collocation points and on a 4D sparse grid of level 4 (608 collocation points), and of level 2 (37 collocation points) and $\eta_1, \ldots, \eta_4$ uniformly distributed as defined in (7.3). The original solution is plotted in black dots.

7.2.4 Beta distribution

In addition to the uniform distribution, we also consider the parameters following a beta distribution. In the context of the Wiener-Askey scheme, the beta distribution corresponds to the Jacobi polynomials. Hence, instead of considering the Legendre polynomials (as for the uniform distribution), we now consider collocation points based on the roots of the Jacobi polynomials.

The beta distribution is (and thus also the Jacobi polynomials are) defined on $[0, 1]$ and depends on two positive shape parameters, denoted by α and β. For a better understanding and to visually explain the next steps we plot the beta distribution for different parameter combinations (α, β) in Figure 7.14. Specifically, we choose the parameter combinations: $(10, 10)$, $(2, 10)$ and $(10, 2)$. For $\alpha = \beta$ the beta distribution is very similar to a truncated Gauss distribution.

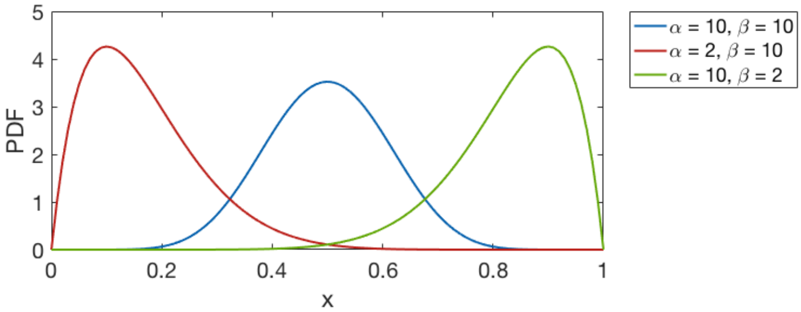


FIGURE 7.14: The probability density function of the beta distribution for the parameter combinations $(10, 10)$, $(2, 10)$ and $(10, 2)$.

We assume the parameters $\eta_1, \ldots, \eta_4$ to be beta-distributed on the interval $[0.9\,\bar{\eta}_i, 1.1\,\bar{\eta}_i]$ for $i = 1, \ldots, 4$ so that the beta distribution with $\alpha = \beta$ is centred in $\bar{\eta}_i$.

In Figure 7.15 we visually compare the solution statistics with PCM on a 4D tensor grid assuming an underlying beta distribution to the solution statistics assuming uniformly distributed parameters. Depending on the choice of α and β the distribution is either centred around $\bar{\eta}_i$ or accumulates close to the endpoints of the interval. This reflects in the solution statistics: with the centred beta distribution the solution resembles the original solution computed with $\bar{\eta}_i$, while the beta distribution and $\alpha \neq \beta$ result in a solution mean that returns earlier or later to the low potassium concentration, depending on the choice of α and β.

7.3 Sensitivity analysis

Instead of computing and visually inspecting the solution statistics as seen in the previous section, sensitivity analysis offers the possibility to quantify the impact of the input parameters on the model output. Variance-based sensitivity analysis estimates the impact of input variability on the output variance, considered for either individual parameters or for combinations.

The computations of the classical sensitivity indices is a method for variance-based sensitivity analysis of the model outputs (Archer, Satelli, and Sobol', 1997; Sobol', 1993, 2001). This method is based on the decomposition of the variance of the model output into terms of increasing dimensionality, as in the classical analysis of variance (ANOVA).

Together with the classical sensitivity indices we define some problem-specific measures in Section 7.3.2. For more details on the classical indices, see Appendix B.4.6.

For the two sensitivity indices we use collocation points based on the ANOVA decomposition with 4 elements and 3 collocation points per element. This is just another way of selecting collocation points, similar to the approach of tensor grids or sparse grids in

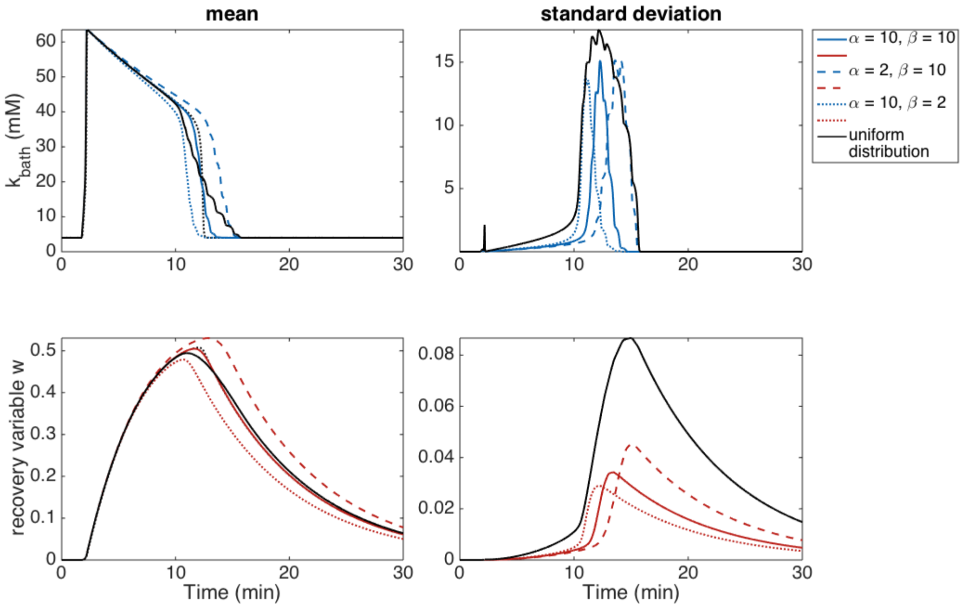


FIGURE 7.15: The solution statistics with PCM on a 4D tensor grid with $M = 10^4$ (Jacobi) collocation points with $\eta_1, \ldots, \eta_4$ being beta-distributed on the interval as defined in (7.3) with different combinations for the parameter set (α, β), namely $(10, 10)$, $(2, 10)$ and $(10, 2)$. The original solution is visualised in black dots and the solution statistics with PCM assuming uniformly distributed parameters is plotted in black.

the previous section. For more information on the multi-element collocation method, we refer to Appendix B.4.5.2 and the documentation of the Multi-Element Probabilistic Collocation Method Package (V1.01) developed by the Crunch group at Brown University.

Sensitivity analysis is based on the definition of collocation points and the computation of the solution with these parameter values. For the definition of the classical sensitivity indices, we need certain concepts that we introduce below.

Let $Z_1^1, \ldots, Z_1^m$ be collocation points. Then keeping k_{bath} fixed at a reference point $c = (c_1, \ldots, c_4)$, also known as the *anchor point*, and evaluating k_{bath} only in the collocation points of the first dimension of the parameter space, its value at collocation point Z_1^j for $j = 1, \ldots, m$ is

$$k^1_{\text{bath}}(Z_1^1) := k_{\text{bath}}(Z_1^1, c_2, c_3, c_4) - k^0_{\text{bath}},$$

with $k^0_{\text{bath}} = k_{\text{bath}}(c)$. The mean and variance of k_{bath} can be approximated by

$$\mu(k^1_{\text{bath}}) \approx \sum_{j=1}^{m} k^1_{\text{bath}}(Z_1^j) q_1^j, \qquad \sigma^2(k^1_{\text{bath}}) \approx \sum_{j=1}^{m} (k^1_{\text{bath}}(Z_1^j))^2 q_1^j - \left(\mu(k^1_{\text{bath}})\right)^2, \tag{7.5}$$

where $q_1^1, \ldots, q_1^m$ are the weights corresponding to the collocation points $Z_1^1, \ldots, Z_1^m$. This definition applies analogously to all first-order terms $k^2_{\text{bath}}, \ldots, k^4_{\text{bath}}$ with the general notation $k^i_{\text{bath}}(Z_i^j) = k_{\text{bath}}(c_1, \ldots, c_{i-1}, Z_i^j, c_{i+1}, \ldots, c_4) - k^0_{\text{bath}}$.

Similarly, we can compute the second-order terms. For example, considering k_{bath} at the collocation point $(Z_1^{j_1}, Z_2^{j_2})$ with $j_1, j_2 = 1, \dots, m$, is

$$k_{\text{bath}}^{12}(Z_1^{j_1}, Z_2^{j_2}) := k_{\text{bath}}(Z_1^{j_1}, Z_2^{j_2}, c_3, c_4) - k_{\text{bath}}^{1}(Z_1^{j_1}) - k_{\text{bath}}^{2}(Z_2^{j_2}) - k_{\text{bath}}^{0}.$$

Using the definition of the ANOVA decomposition we can approximate the mean and variance of k_{bath}^{12} in a similar manner as in the one-dimensional case, namely

$$\begin{aligned} \mu(k_{\text{bath}}^{12}) &\approx \textstyle\sum_{j=1}^{m} k_{\text{bath}}^{12}(Z_1^{j_1}, Z_2^{j_2}) q_1^{j_1} q_2^{j_2}, \\ \sigma^2(k_{\text{bath}}^{12}) &\approx \textstyle\sum_{j=1}^{m} (k_{\text{bath}}^{12}(Z_1^{j_1}, Z_2^{j_2}))^2 q_1^{j_1} q_2^{j_2} - \left(\mu(k_{\text{bath}}^{12})\right)^2. \end{aligned} \tag{7.6}$$

As obtaining the total variance is computationally expensive, we assume that higher order distributions to the variance are negligible. Thus, we use only the first-order terms as a proxy for the total variance. Nevertheless, other choices, like approximating the total variance by the first and second-order terms, are possible.

7.3.1 Classical sensitivity indices

With the above definitions, the first-order sensitivity indices for the function k_{bath} describing the extracellular potassium propagation are defined as

$$S_i := \frac{\sigma^2(k_{\text{bath}}^i)}{\sum_{i=1}^4 \sigma^2(k_{\text{bath}}^i)} \in [0,1], \qquad \text{for } i = 1, \dots, 4,$$

where $\sigma^2(k_{\text{bath}}^i)$ is defined as in (7.5). The second-order sensitivity indices are defined as

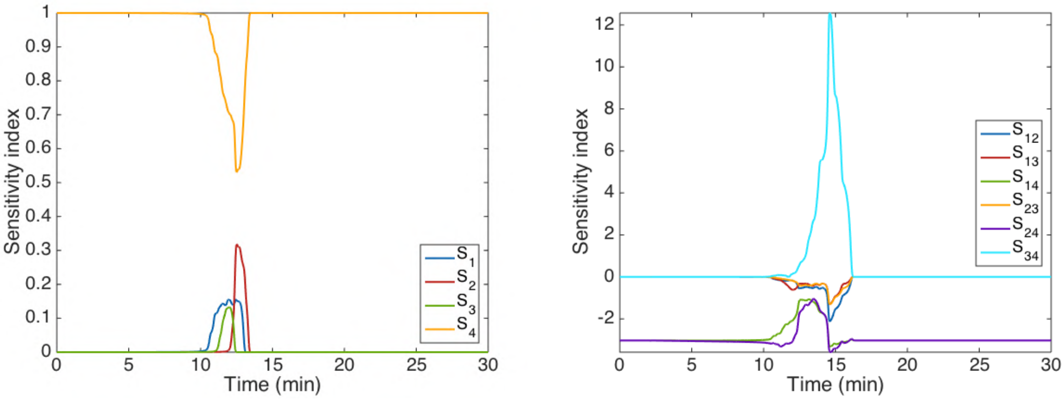
$$S_{il} \quad := \quad \frac{\sigma^2(k_{\text{bath}}^{il})}{\sum_{i=1}^4 \sigma^2(k_{\text{bath}}^i)}, \qquad \text{for } l = 1, \dots, 4,$$

where $\sigma^2(k_{\text{bath}}^{il})$ is defined as in (7.6). These indices describe the contribution of the variance (for each individual parameter and for combinations) to the total variance: the higher the index value, the more important is this parameter (or parameter combination) for the solution. Note that S_i and S_{il} are time-dependent.

As in most of Section 7.2 we assume uniformly distributed parameters on the interval $[0.9\,\bar{\eta}_i, 1.1\,\bar{\eta}_i]$, $i = 1, \dots, 4$. Though the same analysis can be performed for any other probability distribution that correlates to polynomials of the Wiener-Askey scheme. In Figure 7.16 we visualise the temporal evolution of the sensitivity indices S_i and S_{il} for $i, l = 1, \dots, 4$ with $i \neq l$.

From these figures we conclude that η_4 has the biggest contribution overall, but at the moment when the potassium concentration returns to the low concentration the other three parameters also contribute to the variance, though just for a brief period. In the second-order terms, especially the two parameter interactions, by which we mean the simultaneous variation of two parameters at a time, of the parameters η_3 and η_4 (S_{34}) is striking. Due to negative weights on some collocation points, some of the second-order sensitivity indices are negative.

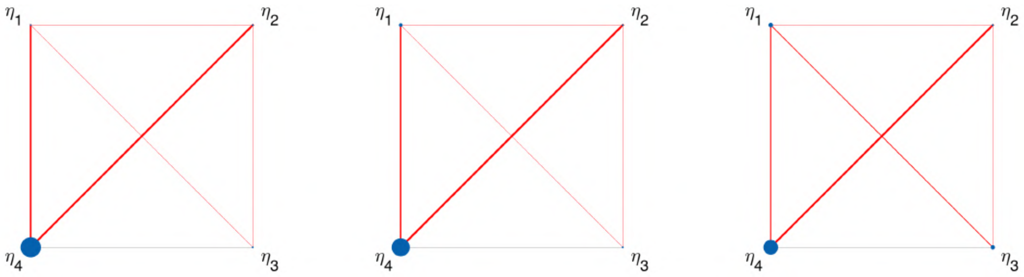
To inspect the sensitivity indices from another perspective, we plot in Figure 7.17 the first-order sensitivity indices S_i for each parameter and the second-order terms S_{il} for



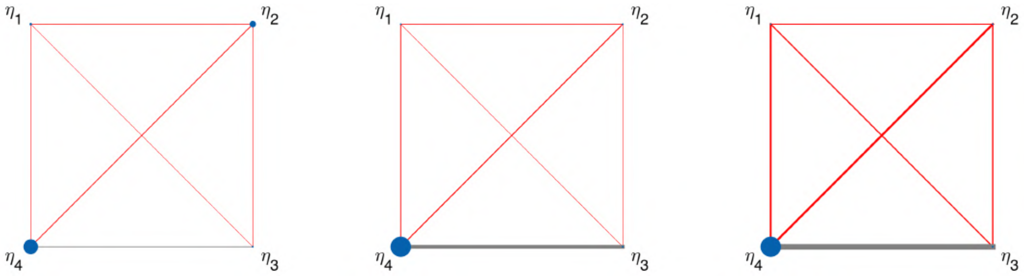
(a) sensitivity indices S_i (b) sensitivity indices S_{il}

FIGURE 7.16: The sensitivity indices S_i and S_{il} for $i, l = 1, 2, 3, 4$ and $i \neq l$.

the parameter interactions at different snapshots in time, namely $t = 10, 11, 12, 13, 14, 15$ minutes. The circles indicate the single parameter sensitivity and the thickness of the lines indicates the sensitivity of two-parameter interactions, red lines indicate a negative sensitivity.



(a) $t = 10$ min (b) $t = 11$ min (c) $t = 12$ min



(d) $t = 13$ min (e) $t = 14$ min (f) $t = 15$ min

FIGURE 7.17: The sensitivity indices S_i (circles) and S_{il} (lines) at the time steps $t = 10, 11, 12, 13, 14, 15$ min. The size of the circles in each vertex of the diagrams indicates the sensitivity of the model solution to the particular parameter considered and the thickness of the lines indicates the sensitivity of two-parameter interactions, red lines indicate a negative sensitivity values.

Both Figure 7.16 and Figure 7.17 show that even though η_4 contributes the most to the variance of the solution, the other parameters have an impact on the variance, especially when the potassium concentration decreases after the wave has passed.

7.3.2 Sensitivity measure tailored to CSD wave characteristics

Instead of measuring the contribution of every parameter to the variance, we can define more problem-specific measures. Therefore, we introduce certain quantities of interest, e.g., the maximum of the extracellular potassium concentration or the interval length of high potassium concentration. These measures depend on the system's solution at specific collocation points but are not directly dependent on time.

Let Z be a collocation point and θ a threshold. Then we define the function Φ as

$$\Phi(k_{\text{bath}}(Z)) := \max_{t\in[0,T]}(k_{\text{bath}}(t,Z) > \theta) - \min_{t\in[0,T]}(k_{\text{bath}}(t,Z) > \theta),$$

that is giving the interval length of the high potassium concentration for the function k_{bath} evaluated at the collocation point Z. The particular value of the threshold θ is not relevant as the solution features steep slopes in the increase and decrease of the potassium concentration. We choose $\theta = 11.8$ as the threshold value for the potassium concentration k_{th} in the model equation. Then we compute the expected value and the variance

$$\mu(\Phi(k^i_{\text{bath}})) \approx \sum_{j=1}^{m} \left(\Phi(k^i_{\text{bath}}(Z^j_i)) - \Phi(k^0_{\text{bath}})\right) q^j_i,$$

$$\sigma^2(\Phi(k^i_{\text{bath}})) \approx \sum_{j=1}^{m} \left(\Phi(k^i_{\text{bath}}(Z^j_i)) - \Phi(k^0_{\text{bath}})\right)^2 q^j_i - \left(\mu(\Phi(k^i_{\text{bath}})\right)^2,$$

where $q^1_i, \dots, q^m_i$ are the weights corresponding to the collocation points $Z^1_i, \dots, Z^m_i$ for $i = 1, \dots, 4$ and k^0_{bath} the initial condition. With the above definitions the sensitivity indices are

$$S_i = \frac{\sigma^2(\Phi(k^i_{\text{bath}}))}{\sum_{i=1}^{4} \sigma^2(\Phi(k^i_{\text{bath}}))} \in [0,1], \qquad \text{for } i = 1, \dots, 4$$

for the first-order terms. Note that these values do not depend directly on time. Based on the anchored ANOVA approach with 4 elements and 3 collocation points per element, the values are: $S_1 = 0.399$, $S_2 = 0.0998$, $S_3 = 0.1147$ and $S_4 = 0.3864$. Consequently, the second dimension in parameter space, that is η_2, contributes the least to the length of the high potassium concentration, whereas the first and fourth dimension in parameter space, that is η_1 and η_4, are very important parameters for the duration of the potassium wave.

Another important characteristic of the potassium wave is its maximum value. Even though the model still includes the peak constant k_p, certain parameters combined actions prevent the system from reaching this maximum value. Thus, we consider the function Ψ representing the maximum of the solution defined as

$$\Psi(k_{\text{bath}}(Z)) := \max_{t\in[0,T]} k_{\text{bath}}(t,Z).$$

Applying to Ψ the definition of the sensitivity indices above, the results are: $S_1 = 0.3553$, $S_2 = 0.3403$, $S_3 = 0.3044$ and $S_4 = 4.8 \cdot 10^{-5}$. Therefore, especially η_1, η_2 and η_3 contribute to the solution reaching the maximum potassium concentration, while η_4 does not contribute at all.

The results of the sensitivity analysis using problem-specific measures are different to those using the classical definition. This difference is due to the different quantities of interest that we are considering: the variance, the maximum of the extracellular potassium concentration and the interval length of high potassium concentration. Consequently, none of the parameters are negligible and to improve the predictive role of our model all of them would have to be further investigated, i.e., in experimental studies.

7.4 Uncertainties in spatially dependent parameters

So far we considered random processes in time, whose dynamics are not analogous to the dynamics of random processes in space. Time-dependent random processes are only influenced by past values, while random processes in space have a dependence in all directions.

Considering the diffusion tensor as a random process in space, we can represent it by the Karhunen-Loeve expansion, an infinite linear combination of orthogonal random variables and deterministic functions (Karhunen, 1947). The deterministic functions used in this representation are determined by the covariance function of the process. With an explicit knowledge about the covariance function, the Karhunen-Loeve expansion is a powerful tool to cope with random processes in spatial dimensions. However, in our model the covariance function corresponding to the diffusion process is unknown. Since the diffusion coefficients are based on real DTI data, it would be best to learn from this data and construct the covariance function based on the available DTI data. However, this is not a trivial step. Another problem is that most commonly used covariance function depends on the Euclidean distance between points in space. As we are dealing with a brain cortex, a folded surface in 3D, a geodesic distance measure would be advantageous and more accurate. The ultimate goal would be to include a DTI data-dependent covariance function that is based on geodesic distances, but this is still work in progress. Nevertheless, to give an idea of what can be done, we provide an illustrative example by approximating the diffusion tensor by a Karhunen-Loeve expansion assuming an exponential covariance function. For detailed definitions and properties of stochastic processes in space, we refer to Appendix B.4.7.

Assuming the diffusion coefficient to be a random process, the solution of system (7.1) depends on random inputs and we can rewrite the model as

$$\begin{aligned}
\frac{\partial k_{\text{bath}}}{\partial t}(\boldsymbol{x},t;\omega) &= \operatorname{div}(D(\boldsymbol{x},\omega)\nabla k_{\text{bath}}(t,\boldsymbol{x};\omega)) \\
&\quad -\eta_1(k_{\text{bath}}(\boldsymbol{x},t;\omega)-k_0(\boldsymbol{x}))\left(1-\frac{k_{\text{bath}}(\boldsymbol{x},t;\omega)}{k_{th}}\right)\left(1-\frac{k_{\text{bath}}(\boldsymbol{x},t;\omega)}{k_p}\right) \\
&\quad -\eta_2(k_{\text{bath}}(\boldsymbol{x},t;\omega)-k_0(\boldsymbol{x}))w(\boldsymbol{x},t;\omega), \\
\frac{\partial w}{\partial t}(\boldsymbol{x},t;\omega) &= \eta_3\left(k_{\text{bath}}(\boldsymbol{x},t;\omega)-k_0(\boldsymbol{x})-\eta_4 w(\boldsymbol{x},t;\omega)\right),
\end{aligned} \tag{7.7}$$

where k_{bath} and w are defined on $W \times (0,T) \times \Omega$ with the spatial domain $W \subset \mathbb{R}^r$ and the sample space Ω. Here, the parameters $\eta_1, \ldots, \eta_4$ are not considered random variables but simple scalars as defined in Table 3.1.

Let the diffusion tensor $D(\boldsymbol{x},\omega)$ be a random process with covariance function $C(\boldsymbol{x}_1,\boldsymbol{x}_2) = Cov(\boldsymbol{x}_1,\boldsymbol{x}_2)$, for $\boldsymbol{x}_1,\boldsymbol{x}_2 \in W$. Then its representation via the Karhunen-Loeve expansion (in space), reads

$$D(\boldsymbol{x},\omega) = \bar{D}(\boldsymbol{x}) + \gamma \sum_{n=1}^{\infty} \sqrt{\lambda_n}\psi_n(\boldsymbol{x})\xi_n(\omega),$$

where $\bar{D}(\boldsymbol{x})$ is the expected value of $D(\boldsymbol{x},\omega)$, γ is a scaling factor, λ_n and ψ_n are the eigenvalues and eigenvectors of the covariance function and $\xi_n(\omega)$ are random variables that satisfy

$$\mathbb{E}(\xi_n) = 0, \qquad \mathbb{E}(\xi_n\xi_j) = \delta_{nj}, \qquad \text{for } n, j \geq 1.$$

Thus, λ_n and ψ_n are are the solution to the eigenvalue problem

$$\int_W C(\boldsymbol{x}_1, \boldsymbol{x}_2)\psi_n(\boldsymbol{x}_1)\, d\boldsymbol{x}_1 = \lambda_n\psi_n(\boldsymbol{x}_2). \tag{7.8}$$

The usefulness of the Karhunen-Loeve expansion depends heavily on the ability to solve the integral equation (7.8). As the expansion as an infinite series is unwieldy, random processes are often approximated by a finite series expansion, namely

$$D(\boldsymbol{x},\omega) = \bar{D}(\boldsymbol{x}) + \gamma \sum_{n=1}^{n_{\max}} \sqrt{\lambda_n}\psi_n(\boldsymbol{x})\xi_n(\omega), \qquad n_{\max} \geq 1. \tag{7.9}$$

The truncation of the expansion, that is, the choice of $n_{\max}$, is closely related to the eigenvalues λ_i. A common approach is to study the decay of eigenvalues and keep the first $n_{\max}$ of them, such that the contribution of the remaining eigenvalues is negligible.

7.4.1 Exponential covariance function

As an example, we choose $W = [0, 2]$ and consider the exponential kernel $C(x_1, x_2) = \exp(-|x_1 - x_2|/l)$ with $x_1, x_2 \in W$ and parameter l that reflects the rate at which the correlation decays between different points in the space domain, also called the *correlation length*. To approximate the eigenvalues and eigenvectors we solve equation (7.8) with the Matlab function **eigs**. In Figure 7.18 we visualise the kernels for different correlation lengths and the corresponding eigenvalues. We define the finite Karhunen-Loeve expansion of D as in (7.9), namely

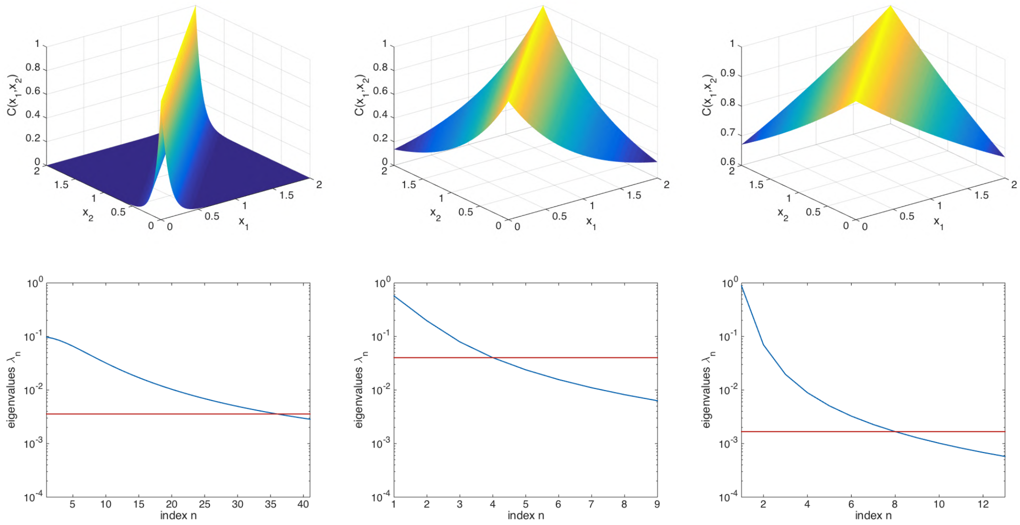
$$D(x,\omega) = \bar{D}(x) + \gamma \sum_{n=1}^{n_{\max}} \sqrt{\lambda_n}\psi_n(x)\xi_n(\omega), \tag{7.10}$$

where we define the cut-off as the maximum integer $n_{\max} \geq 1$ such that

$$\sum_{n=1}^{n_{\max}} \lambda_n < \theta$$

for a given threshold θ, here set either as $\theta = 0.9$ or $\theta = 0.99$. This $n_{\max}$ defines the dimensionality of the expansion in the probability space.

Similar to PCM in Section 7.2 we draw samples assuming that the random variables in (7.10) are uniformly distributed on $[-1, 1]^{n_{\max}}$. Like before we have a total of M collocation points corresponding o the Legendre polynomial. Setting $\bar{D} = 5 \cdot 10^{-4}$, $\gamma = 10^{-4}$, $l = 1$ and $\theta = 0.9$, we can compute D at each collocation point and with this compute the solution of k_{bath} in system (7.7). Post-processing the results we obtain the solution



(a) $l = 0.1, \theta = 0.9, n_{\max} = 36$ (b) $l = 1, \theta = 0.9, n_{\max} = 4$ (c) $l = 5, \theta = 0.99, n_{\max} = 8$

FIGURE 7.18: The exponential kernel $C(x_1, x_2)$ for different correlation lengths $l = 0.1, 1, 5$ (top row) and the corresponding eigenvalues (bottom row). The eigenvalues are marked with a blue line and the cut-off threshold 0.9 (for $l = 0.1, 1$) and 0.99 (for $l = 5$) of the total contribution is marked in red.

statistics, like the solution mean and variance

$$\begin{aligned} \mu(k_{\text{bath}}(x,t)) &\approx \textstyle\sum_{j=1}^{M} k_{\text{bath}}(x,t;\xi_j) q_j, \\ \sigma^2(k_{\text{bath}}(x,t)) &\approx \textstyle\sum_{j=1}^{M} k_{\text{bath}}^2(x,t;\xi_j) q_j - \mu^2(k_{\text{bath}}(x,t)), \end{aligned}$$

where q_j are the weights corresponding to the collocation points ξ_j for $j = 1, \ldots, M$.

As an example we choose the spatial domain $W = [0, 2]$ discretised by $N = 100$ mesh points with a mesh size $\Delta x = 2/(N-1)$. The collocation points in the probability space are chosen from a sparse grid (of level 2) based on Gauss-Legendre nodes. Figure 7.19 shows the mean and standard deviation at different time steps, while Figure 7.20 is a space-time plot of the mean and standard deviation to underline the propagative nature of the solution.

Assuming an exponential kernel for the covariance function, we obtain solution statistics that show the effect of uncertainties in the diffusion tensor on the spatio-temporal propagation of the potassium wave. Even though this kernel choice is a strong assumption on the correlation between different points in the space domain and the example is restricted to a one-dimensional space domain with a constant diffusion coefficient $\bar{D}$, the results in Figure 7.19 highlight the impact of uncertainties in the diffusion parameter. We plan to extend this analysis to the case of variable diffusion coefficients in a 2D and 3D domain, also including a covariance function based on geodesic distance to treat the case of a complex surface as the reconstructed brain cortex.

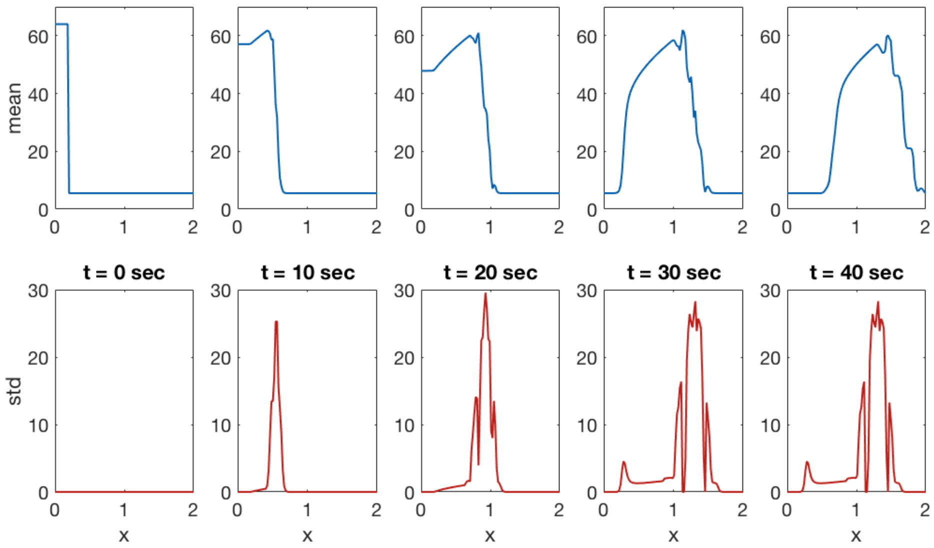
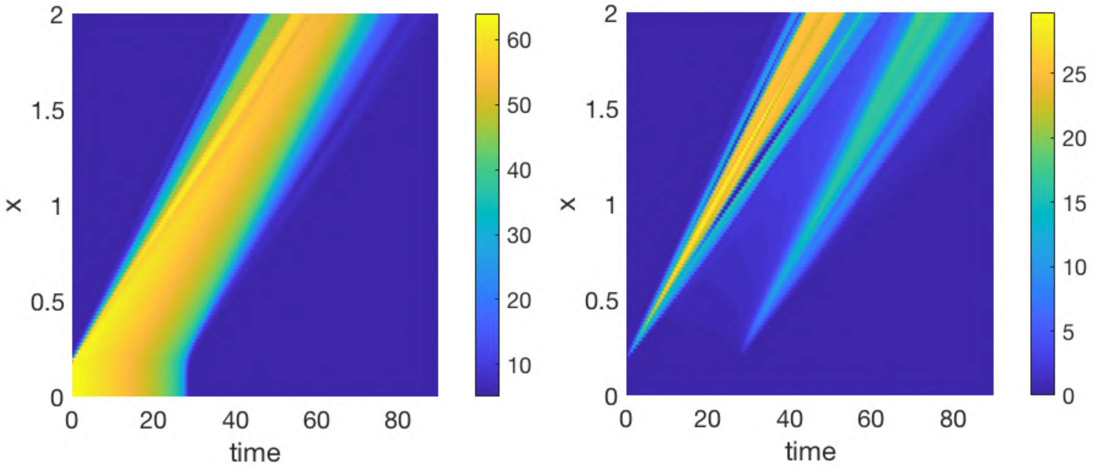


FIGURE 7.19: The mean and the standard deviation of the potassium wave k_{bath} at different time steps, namely $t = 0, 10, 20, 30, 40$ sec.



(a) mean (b) standard deviation

FIGURE 7.20: The mean and the standard deviation of the potassium wave k_{bath} in space and time.

7.5 Summary

In this chapter we treated the model parameters as random variables and processes (in space) and analysed the impact of their uncertainties on the model solution. Since there are different strategies to cope with uncertainties in the parameters that do not depend on space and in those that are spatially-dependent, we treated them separately.

Using PCM and other UQ techniques for uncertainties in the space-independent parameters we observed the impact of individual parameters and of their combined effect on the shape and characteristics of the potassium wave. We assumed either uniform or beta distributed parameters to ensure that their values are selected from finite intervals, as it is natural for biological parameters. With a sensitivity analysis based on the ANOVA approach we discovered the contribution of each parameter to the solution variance and to problem-specific measures, such as the time interval of high potassium concentration or the maximum potassium concentration. Despite some of the parameters contributing more than others to these characteristics, our results showed that none of them is negligible. Consequently, to improve the predictive role of the model all the parameters would need to be specified in experimental set-ups.

In principle all parameters could be space-dependent, but for this study we only considered the diffusion coefficient (or diffusion tensor) to depend on space and approximated it by the Karhunen-Loeve expansion. We considered the particular case of an exponential kernel and presented the solution statistics in a one-dimensional space domain. These results gave a first hint of what type of information could potentially be inferred from this type of analysis. Nevertheless, assuming a specific kernel requires a strong knowledge about the correlation between different points of the mesh discretisation. Hence, the next step of some more grounded research in this direction would be to construct a kernel based on the available DTI data and depending on geodesic distances.

Conclusions

"All endings are also beginnings. We just don't know it at that time."

Mitch Albom, in *The Five People you meet in Heaven* (2003)

This thesis focused on patient-specific modelling and simulation of the cortical spreading depression (CSD), a correlate of migraine with aura. We developed and implemented a multi-scale PDE-ODE model that couples the propagation of the depolarisation wave associated to CSD with a detailed electrophysiological model for neuronal activity in order to capture both macroscopic and microscopic dynamics. The model was combined with personalised brain geometries, obtained from magnetic resonance imaging (MRI) and with patient-specific conductivity tensors derived from diffusion tensor imaging (DTI) data. This DTI data provides detailed information about the anisotropy and the electrical conductivity properties of cortical tissue. To quantify the results and evaluate the propagation behaviour we identified various quantities of interest (QoI), such as the surface regularity index or asymmetry measures in the CSD propagation behaviour, for intra-patient statistics. This set-up was applied to a series of data sets of different patient groups for statistical evaluations. As the system parameters are unknown and generally tuned to reproduce main features of the CSD, we concluded this work by applying different techniques of uncertainty quantification to the developed model and study the effect of uncertainties in the parameter choice.

In this final chapter we summarise and discuss the results and conclusions drawn in this thesis, also highlighting some important limitations of the presented work. We then present some possible future research to extend and improve the work presented in this thesis.

Patient-specific CSD model

In Chapters 3 and Chapter 4 we introduced and described the implementation of a multi-scale model for the potassium diffusion that successfully reproduces the electrophysiological and propagative characteristics of CSD waves. We included patient-specific information about the cerebral cortex and diffusion data from DTI data representing the microstructure of the cerebral tissue. For computational reasons in the context of full cortex simulations, we only considered the propagative component of the model. However, as the model is coupled unidirectionally, the electrophysiology could be easily reproduced from the results of the propagating potassium wave on the cerebral cortex. Finally, we simulated the CSD propagation on a real brain geometry obtained from MRI data using diffusion data from DTI imaging. However, this model has several limitations, i.e., it is missing to account for inhibitory neurons, synaptic gaps, and large vessels. Especially the latter can potentially stop the propagation of the depolarisation wave (Ayata and Lauritzen, 2015). CSD is not only triggering ionic changes but also a vascular response, including vasodilation, that is, the widening of blood vessels. Accounting for their presence and effect could add to the accuracy of the model and its predictions.

Quantities of interest and case studies of migraine patients

To correct artefacts from the brain reconstruction, in Chapter 5 we tested several smoothing techniques, weight functions and parameter choices to smoothen the surface mesh of the cortex. Taubin smoothing with equal weights turned out to be the best option for cortical geometries and instead of limiting the iteration number we introduce a new stopping criterion, a threshold for the maximum volume loss, addressing a typical drawback of Taubin smoothing. For this choice of weights, the mesh quality of the two brain hemispheres improved with every step of the iterative smoothing algorithm. Another approach to improve the smoothing result could be to optimise the mesh quality, while satisfying the volume loss restriction. This is an approach we are currently working on with Franco Dassi from the University of Milan-Biccoca in Italy.

In the sixth chapter we considered a set of chronic migraine patients with aura and confirmed that the simulated CSD wavefront propagates according to the sequence of symptoms experienced by these patients during a migraine attack. In the second case study we computed and compared the QoI introduced in Chapter 5 for patients in three different groups. One limitation of this study is the fact that CSD is a neuronal correlate of migraine aura, but the migraine patients in this study do not experience aura. Even though we could not find a QoI that classifies a certain patient type or significantly distinguishes healthy individuals from migraineurs, we could show that some QoI, e.g., the time the CSD wave remains in specific regions of the left hemisphere, have the potential to differentiate between the patient groups to a certain extent. The main limitation of this chapter is the size of the data sets, especially in the first case study where considerations were made only for five patients. Bigger data sets and more sophisticated techniques for data mining could yield more informative results or drive the selection of appropriate QoI. In addition, the QoI we introduced (especially the geometric ones) could possibly be efficient indicators for other neurological diseases and a study on patient groups of different clinical backgrounds could be envisioned.

Uncertainty quantification

In Chapter 7 we applied different techniques in uncertainty quantification to the parameters of the CSD model. The impact of spatially independent parameters depends on the assumed underlying distribution and the choice of support for this distribution. By testing the effect on the model solution of different underlying distributions we showed that different aspects of the model outcome are sensitive to different parameters (considered either individually or jointly). We analysed the contribution of each parameter to the solution variance and to other problem-specific measures with a sensitivity analysis. Even though some of the parameters contribute more than others to the CSD characteristics, none of them is negligible and experimental studies would have to inform the selection of parameter values or parameter distributions.

For the sensitivity analysis we limited ourselves to considering uniform distributions only, though different distributions can potentially be considered. Another restriction we imposed on the parameters was that all of them had the same distribution. However, in reality, different parameters might be governed by different distributions. In this thesis, we only considered parameters defined on finite intervals, but distributions defined on $\mathbb{R}$ could be considered as well.

To approximate space-dependent parameter by a random process, strong assumptions on the correlation of points in the mesh discretisation and on the covariance kernel are necessary. We addressed this aspect only partially by showing the solution statistics in a one-dimensional space domain for a particular case of covariance kernel. A more grounded analysis in this setting could include the three-dimensional space domain of the cerebral cortex and a kernel based on DTI-data and was left as a possible topic for further research.

Future work

Even though spreading depression has initially been described in the cerebral cortex, it can also be induced in subcortical brain structures with sufficiently high neuronal and synaptic density, and with limited extracellular space (Ayata, 2010; Bures, Buresova, and Krivanek, 1974). Besides, CSD can propagate into subcortical structures via contiguous matter (Ayata, 2010; Bures, Buresova, and Krivanek, 1974; Eikermann-Haerter et al., 2009). On the basis of these two arguments we are planning to extend the model to brain volumes instead of limiting it to the cortical surface. The next step in this direction would be to include *fibre tracts* (computed from DTI data) that connect different cortical regions through inter-brain highways. These connections could contribute to the accuracy of CSD propagation and reveal features that could not be observed simply with a surface model.

In this thesis, we approximated the diffusion coefficients or diffusion tensors by the Karhunen-Loeve expansion in space when applying uncertainty quantification to the model parameters. However, for this expansion we needed the eigenvalues and eigenvectors of the corresponding covariance kernel. As an illustrative example of this strategy, we assumed an exponential kernel and presented the corresponding solution statistics in a one-dimensional spatial domain. However, assuming a specific kernel requires a strong knowledge about the correlation between different points of the mesh discretisation. To obtain an improved representation of the diffusion coefficients via the Karhunen-Loeve expansion, a better option would be to construct a kernel based on real DTI data. Another problem is that most commonly used kernels depend on the Euclidean distance between the points in space. As we are dealing with a brain cortex, a folded surface in 3D, a geodesic distance measure would provide much more accurate information. Including uncertainties of all parameters in the model and learning from the DTI measurements is an imperative step of future research for the improvement of the predictive role of this model.

Appendix A

Contributions and developed softwares

A.1 Publications

Kroos, J.M., Diez, I., Cortes, J.M., Stramaglia, S., Gerardo-Giorda, L. , ***Geometry shapes propagation: assessing the presence and absence of cortical symmetries through a computational model of cortical spreading depression*****, Front. Comput. Neurosci., 2016, vol. 10. ISSN: 1662-5188, DOI: 10.3389/fncom.2016.00006**

Abstract: Cortical spreading depression (CSD), a depolarization wave which originates in the visual cortex and travels towards the frontal lobe, has been suggested to be one neural correlate of aura migraine. To the date, little is known about the mechanisms which can trigger or stop aura migraine. Here, to shed some light on this problem and, under the hypothesis that CSD might mediate aura migraine, we aim to study different aspects favoring or disfavoring the propagation of CSD. In particular, by a using a computational neuronal model distributed throughout a realistic cortical mesh, we study the role that the geometry has in shaping CSD. Our results are two-fold: first, we found important differences in the propagation traveling patterns of CSD, both intra and inter-hemispherically, revealing important asymmetries in the propagation profile. Second, we identified the brain regions which play a key role in the CSD propagation. Our study reveals important dynamical aspects of CSD, which, if applied to subject-specific cortical geometry, might help to differentiate between healthy subjects and those suffering migraine.

Kroos, J.M., Diez, I., Cortes, J.M., Stramaglia, S., Gerardo-Giorda, L. , ***Temporal excitation patterns on the cerebral cortex as a result of migraine modeling*****, In: Mantica G., Stoop R., Stramaglia S. (eds) Emergent Complexity from Nonlinearity, in Physics, Engineering and the Life Sciences, Springer Proceedings in Physics, vol 191. Springer, Cham March 2017, Doi: 10.1007/978-3-319-47810-4_13**

Abstract: The complex, highly individual, geometry of the cerebral cortex in humans presents a major challenge in studying the spreading of spontaneous neuronal activity. Recent computational advances allow to simulate the propagation of depolarization waves on the macroscale and for individual geometries, reconstructed from accurate medical imaging as MRI, with high levels of detail. In this paper, we take advantage of such technique to study the temporal excitation patterns that follow the passage of a depolarization wave on the cerebral cortex.

Kroos, J.M., Marinelli, I., Diez, I., Cortes, J.M., Stramaglia, S., Gerardo-Giorda, L. , ***Patient-specific computational modeling of Cortical Spreading Depression via Diffusion Tensor Imaging*****, Int. J. Numer. Method Biomed. Eng., 2017, vol. 33 (11), e2874, DOI: 10.1002/cnm.2874**

Abstract: Cortical spreading depression, a depolarization wave originating in the visual cortex and traveling towards the frontal lobe, is commonly accepted as a correlate of migraine visual aura. As of today, little is known about the mechanisms that can trigger or stop such phenomenon. However, the complex and highly individual characteristics of the brain cortex suggest that the geometry might have a significant impact in supporting or contrasting the propagation of cortical spreading depression. Accurate patient-specific computational models are fundamental to cope with the high variability in cortical geometries among individuals, but also with the conduction anisotropy induced in a given cortex by the complex neuronal organisation in the grey matter. In this paper, we integrate a distributed model for extracellular potassium concentration with patient-specific diffusivity tensors derived locally from diffusion tensor imaging data.

Gerardo-Giorda, L. , Kroos, J.M., ***Multiscale modeling and simulation of cortical spreading depression propagation*****, Computers and Mathematics with Applications, 2017, vol. 74 (5), p. 1076-1090, DOI: 10.1016/j.camwa.2017.05.013**

Abstract: Cortical Spreading Depression (CSD) is a disruption of the brain homeostasis that, originating in the visual cortex and traveling towards the frontal lobe, temporarily impairs the normal functioning of neurons. Although, as of today, little is known about the mechanisms that can trigger or stop such phenomenon, CSD is commonly accepted as a correlate of migraine visual aura. In this paper, we introduce a multi-scale PDE-ODE model that couples the propagation of the depolarization wave associated to CSD with a detailed electrophysiological model for the neuronal activity to capture both macroscopic and microscopic dynamics.

Kroos, J.M., Dassi, F., Perotto, S. and Gerardo-Giorda, L., *Curvature analysis of smoothed meshes from medical brain images*, in preparation 2019

Kroos, J.M., de Tommaso, M., Stramaglia, S., Gerardo-Giorda, L., *Single case study of migraine with aura patients on the basis of mathematical modelling for cortical spreading depression*, in preparation 2019

A.2 Presentations

eVita Winter School: Uncertainty Quatification, Geilo, Norway – poster presentation entitled "Numerical Simulation of Cortical Spreading Depression on a Real Brain Geometry", January 2015.

Zaferna Colloquium, Institute for Applied Mathematics, Westfälische Wilhelms-Universität Münster, Münster, Germany – presentation entitled "Mathematical Models for Excitable Media: An Application to the Cerebral Cortex", March 2015.

BCAM Workshop on Quantitative Biomedicine for Health and Disease, BCAM, Bilbao, Spain – poster presentation entitled "Personalised Simulation of the Cortical Spreading Depression", Februar 2016.

Stockholm Mathematics Centre, University of Stockholm, Sweden – presentation entitled "Personalised Simulation of the Cortical Spreading Depression – Computational Modelling via Diffusion Tensor Imaging", 10 June 2016.

SIAM Conference on the Life Sciences, Boston, Massachusetts, USA – presentation entitled "Personalised Simulation of the Cortical Spreading Depression – Computational Modelling via Diffusion Tensor Imaging", 14 July 2016.

International Conference on Systems Biology, Barcelona, Spain – poster presentation entitled "Patient-specific Modelling of Cortical Spreading Depression", September 2016.

2nd BCAM Workshop on Nonlinear Dynamics in Biological Systems, BCAM, Bilbao, Spain – poster presentation entitled "Patient-specific Modelling of Cortical Spreading Depression", September 2016.

3rd BCAM Workshop on Quantitative Biomedicine for Health and Disease, BCAM, Bilbao, Spain – presentation entitled "Patient-specific Modelling of Cortical Spreading Depression", 22 February 2017.

Crunch Group, The Division of Applied Mathematics, Brown University, Providence, Rhode Island, USA – presentation entitled "Patient-specific Modelling of Cortical Spreading Depression", 10 March 2017.

Stockholm Mathematics Centre, University of Stockholm, Sweden – presentation entitled "Patient-specific Modelling of Cortical Spreading Depression", 19 June 2017.

11th European Conference on Mathematical and Theoretical Biology, Lisbon, Portugal – presentation entitled "Patient-specific Modelling of Cortical Spreading Depression", 27 July 2018.

2nd International Workshop on Neurodynamics, Castro-Urdiales, Spain – presentation entitled "Patient-specific Modelling of Cortical Spreading Depression", 27 September 2018.

A.3 Developed software

Cortical spreading depression simulation

Program Name: compute_csd_whole_brain.m
Author: J.M. Kroos
License: N/A
Operating System: MacOS
Programming Language: Matlab
Description: The package numerically solves the CSD progression on the meshes of real brain geometries. It initiates the wave in each of the 34 regions of interest records the

arrival times in the other regions, respectively. The electrophysiology part on the microscopic level is coupled with the macroscopic propagative part. After the CSD computation, the QoI can be investigated by the user using a general user interface.

Diffusion data processing

Program Name: DTI_preprocessing.m
Author: J.M. Kroos
License: N/A
Operating System: MacOS
Programming Language: Matlab
Description: This package processes the DTI and DWI data and integrates it in the already existing code for the CSD progression.

Brain smoothing and curvature approximation

Program Name: Mesh smoothing package
Authors: F. Dassi, J.M. Kroos
License: N/A
Operating System: MacOS
Programming Language: C++
Description: This code was developed in C++ together with Franco Dassi. All in all we provided simulations of the CSD on a real brain geometry with diffusion coefficients obtained from DTI data.

Appendix B

Mathematical compendium

"If you can't understand it without an explanation,
you can't understand it with an explanation."

Haruki Murakami, in *1Q84* (2009)

In this appendix we gather some mathematical concepts that we just briefly referred to in the previous chapters. First, we comment on preconditioned iterative solvers especially for sparse linear systems. For a better understanding of the concepts of curvature used in Chapter 5, we give a basic introduction and an example in Appendix B.2. In Appendix B.3 we summarise concepts of graph theory that we use in Chapter 5 to confirm the connectivity of points considered for the initial condition. Finally, Appendix B.4 gives a brief introduction to the concepts, ideas and common techniques in uncertainty quantification in Chapter 7.

B.1 Preconditioned iterative solvers for sparse linear systems

For the computation of CSD propagation, we need to solve a linear system $Ax = b$ where A is a large sparse matrix. We give a short introduction to iterative solvers and numerical techniques for preconditioning to increase the efficiency and robustness of the solution. We mainly follow Saad, 2003 and Quarteroni, Sacco, and Saleri, 2007.

B.1.1 Iterative solvers for linear systems

Iterative methods yield the solution x of a linear system $Ax = b$ after an infinite number of steps. The basic idea is to construct a sequence of vectors x_k that *converges*

$$x = \lim_{k \to \infty} x_k.$$

In practice, the iterative process is stopped at a minimum value n, such that

$$||x_n - x|| < \varepsilon,$$

where ε is a tolerance and $||\cdot||$ is any suitable vector norm.

For a given linear system and a given initial datum x_0, iterative methods are of the form

$$x_{k+1} = x_k + M^{-1} r_k, \quad \text{for } k \geq 1,$$

where M is a non-singular matrix, also called *preconditioner*, and

$$r_k = b - Ax_k$$

denotes the *residual* vector at step k.

There are two main classes of iterative methods, stationary iterative methods and Krylov subspace methods. Here, we focus on Krylov subspace methods that approximate the solution by minimising the residual. Typical methods of this class are the conjugate gradient method that assumes matrix A to be symmetric positive-definite and the biconjugate gradient method for non-symmetric linear systems. We briefly present the *preconditioned conjugate gradient method* (PCG) and the *biconjugate gradient stabilised method* (BiCGSTAB), improvements of the afore mentioned methods.

B.1.1.1 Preconditioned conjugate gradient

PCG is one of the best known iterative techniques to solve linear systems with sparse symmetric positive-definite matrices. Denoting with $r_k = b - Ax_k$ the original residual and with $z_k = M^{-1} r_k$ the residual for the preconditioned system, the algorithm of the PCG is given in Algorithm B.1.

Algorithm B.1 PCG

```
compute $r_0 = b - Ax_0$, $z_0 = M^{-1} r_0$ and $p_0 = z_0$
for $k = 0, 1, \ldots$ until $r_k$ sufficiently small do
    $\alpha_k = \frac{(r_k, z_k)}{(Ap_k, p_k)}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k Ap_k$
    $z_{k+1} = M^{-1} r_{j+1}$
    $\beta_k = \frac{(r_{k+1}, z_{k+1})}{(r_k, z_k)}$
    $p_{k+1} = z_{k+1} + \beta_k p_k$
end for
```

B.1.1.2 Biconjugate gradient stabilised method

BiCGSTAB is a variant of the biconjugate gradient method and the conjugate gradient squared method. The BiCGSTAB supports a faster but less stable convergence than the other two methods. For a preconditioning with the matrix $M = M_1 M_2$ the BiCGSTAB method is listed in detail in Algorithm B.2.

Algorithm B.2 BiCGSTAB

```
compute $r_0 = b - Ax_0$
choose $\hat{r}_0$ s.t. $(\hat{r}_0, r_0) \neq 0$
set $p_0 = r_0$
for $k = 0, 1, \ldots$ until convergence do
    $\alpha_k = \frac{\hat{r}_0^T r_k}{\hat{r}_k^T A M^{-1} p_k}$
    $s_k = r_k + \alpha_k A M^{-1} p_k$
    $t = A M^{-1} s_k$
    $w_k = \frac{(M_1^{-1} t, M_1^{-1} s)}{(M_1^{-1} t, M_1^{-1} t)}$
    $x_{k+1} = x_k + \alpha_k M^{-1} p_k + w_k M^{-1} s_k$
    $r_{k+1} = s_k - w_k t$
    $\beta_k = \frac{(r_{k+1}, \hat{r}_0)}{(r_j, \hat{r}_0)} \frac{\alpha_k}{w_k}$
    $p_{k+1} = r_{k+1} + \beta_k (p_k - w_k A M^{-1} p_k)$
end for
```

B.1.2 Preconditioners for sparse matrices

Preconditioning is a technique to increase efficiency and robustness of linear solvers. It transforms the linear system into one that has the same solution but can be solved more easily and accurately. Considering the problem $Ax = b$, the preconditioner M is an approximation of A for which the system $M^{-1}Ax = M^{-1}b$ is easier to solve and M^{-1} is cheap to compute. Thus, we can solve the preconditioned system

$$M^{-1}Ax = M^{-1}b \qquad \text{or} \qquad AM^{-1}u = b, \quad x = M^{-1}u.$$

In the next subsection, we introduce some of the most common preconditioners.

B.1.2.1 Incomplete LU factorisation

For a sparse matrix $A = (a_{ij})_{i,j=1,\ldots,n}$, the *incomplete LU factorisation* (ILU) computes a sparse lower triangular matrix L and a sparse upper triangular matrix U, so that the residual $R = A - LU$ satisfies certain constraints. The elements to drop at each step can be chosen as some non-zero pattern in advance. For any zero pattern set P, such that

$$P \subset \{(i,j) \,|\, i \neq j, 1 \leq i, j \leq n\},$$

the ILU factorisation can be computed as follows.

Zero fill-in ILU In the incomplete LU factorisation with no fill-in, denoted by ILU(0), the pattern of the matrix A is taken as a template for the sparsity patterns of the triangular matrices. Thus the zero pattern P is defined as $P = \{(i,j)|a_{ij} \neq 0\}$. The matrix L has the sparsity pattern of the lower triangular part of A and the matrix U the one from the upper triangular part. An example of the sparsity pattern of a matrix A, the matrices L and U obtained with ILU(0) factorisation and their product are shown in Figure B.1.

Algorithm B.3 General ILU factorisation

```
for $i = 2, \dots, n$ do
    for $k = 1, \dots, i-1$ and if $(i,k) \notin P$ do
        $l_{ik} = \frac{a_{ik}}{a_{kk}}$
        for $j = k+1, \dots, n$ and if $(i,j) \notin P$ do
            $u_{ij} = a_{ij} - l_{ik} a_{kj}$
        end for
    end for
end for
```

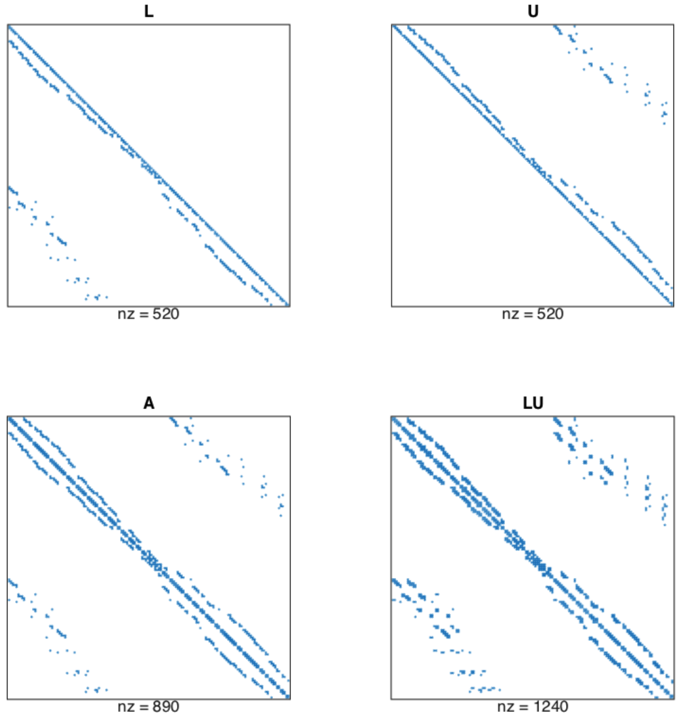


FIGURE B.1: The sparsity patterns of a matrix A, the preconditioning matrices L and U computed with the ILU(0) factorisation and the product of them. The numbers below the graphs refer to the number of non-zero entries.

Nevertheless, the product of the two matrices L and U does not express the same sparsity pattern as the original matrix A. This is due to the fact that it is impossible to find L and U with the same pattern as A so that their product matches A. The multiplication of L and U results in extra diagonal entries that are called fill-in elements. Ignoring these fill-in elements, it is possible to find L and U so that their product matches A in the other diagonals. The standards ILU(0) can be constructed with the general algorithm for ILU (Algorithm B.3) choosing the pattern P equal to the zero pattern of A.

The disadvantage of the ILU(0) is that the accuracy might not be sufficient to guarantee an adequate convergence of the iterative solver. More accurate are ILU factorisations that allow some fill-in, also called ILU(p). However, in these techniques, the fill-in only depends on the structure of the original matrix A. A more sensible approach is to determine the zero pattern P more dynamically and also depending on the values in A.

Threshold strategies and ILUT There are a few alternative methods to ILU(p) that base the dropping of elements in the Gaussian elimination process according to their magnitude instead of their location. With this, the zero pattern P is chosen more dynamically. The most common way to obtain an incomplete factorisation that ignores small entries below a threshold is to take a sparse direct solver and add the dropping conditions for the elements. An example is given in Algorithm B.4. Here v is a row vector collecting the linear combinations of the rows in the elimination and v_k is the k-th entry of v. The notation a_{i*} describes the i-th row of matrix A.

Algorithm B.4 ILUT

```
 1: for $i = 1, \dots, n$ do
 2:     $v := a_{i*}$
 3:     for $k = 1, \dots, i-1$ and if $v_k \neq 0$ do
 4:         $v_k := \frac{v_k}{a_{kk}}$
 5:         apply the dropping rule to $v_k$
 6:         if $v_k \neq 0$ then
 7:             $v := v - v_k u_{k*}$
 8:         end if
 9:     end for
10:     apply the dropping rule to row $v$
11:     $l_{ij} := v_j$ for $j = 1, \dots, i-1$
12:     $u_{ij} := v_j$ for $j = i, \dots, n$
13:     $v := 0$
14: end for
```

The ILU(0) can be considered as a special case of this algorithm with the rule to drop elements that are in the position of zero entries in the original matrix. In the factorisation ILUT(p,τ) the dropping rules are:

(i) In line 5, the element v_k is set to zero (e.g. by dropping it) if it is below the relative tolerance τ_i which is the product of τ and the norm of a_{i*}, the i-th row of the original matrix A.

(ii) In line 10, first all elements with values below the relative threshold τ_i are dropped. Then only the p largest elements in the L part and the p largest elements in the U part are kept.

Here, the first dropping rule reduces the computational costs, whereas the second one helps to control the memory usage.

ILUTP approach For certain matrices in real applications the ILUT approach fails and one way to make the ILUT more stable is to introduce column pivoting. The ILUTP (with "P" for pivoting) uses a permutation array to hold the new ordering of variables. A tolerance parameter τ can be introduced to determine whether to permute variables or not. A non-diagonal element a_{ij} is permuted when the condition $\tau \cdot |a_{ij}| > |a_{ii}|$ is fulfilled.

B.1.2.2 Incomplete Cholesky factorisation

The *incomplete Cholesky factorisation* is a sparse approximation of the Cholesky factorisation. The Cholesky decomposition of a symmetric, positive-definite matrix A is defined as $A = LL^T$, where L is a lower triangular matrix. The incomplete Cholesky factorisation uses the algorithm to find the exact Cholesky decomposition but drops every entry whose indices are in the zero pattern P, the exact procedure is given in Algorithm B.5. Usually P is chosen as the zero pattern of the matrix A.

Algorithm B.5 Incomplete Cholesky factorisation

```
for $i = 1, \dots, n$ do
    $l_{ii} := \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2\right)^{1/2}$
    for $j = i+1, \dots, n$ and if $(j, i) \notin P$ do
        $l_{ji} := \frac{1}{l_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk}\right)$
    end for
end for
```

Incomplete Cholesky factorisation with threshold dropping Like in the ILU factorisation we can increase the accuracy of the incomplete Cholesky factorisation in order to support an adequate convergence rate of the iterative solver. An option is to introduce a threshold for dropping elements instead of assuming the sparsity pattern of the original matrix. This method is called *incomplete Cholesky factorisation with threshold dropping* (ICT).

B.2 Differential geometry

Differential geometry is an important tool for studying of the local behaviour of a surface in the vicinity of certain points. In the following, we give a brief summary of the most important definitions and results of differential geometry with the aim to clarify the concept of Gaussian and mean curvature. In this section we refer to Kühnel, 2006 for notations and definitions, while further details and proofs can be found in Do Carmo, 1976.

To develop a local theory of surfaces, we assume that the surface is given by a differentiable map in two parameters and admits a geometric linearisation. This means that at every point there is a linear surface, i.e., a plane, that touches the surface at that point only. A map that satisfies this condition is called *immersion*.

Definition B.2.1. A map $f : U \to \mathbb{R}^m$ defined on an open set $U \subset \mathbb{R}^n$ is *differentiable* at a point $x \in U$, if there is a linear map $A_x : \mathbb{R}^n \to \mathbb{R}^m$ so that

$$f(x+\xi) = f(x) + A_x\xi + o\left(||\xi||\right),$$

is true in a neighbourhood $U_\epsilon(x)$. The symbol $o\left(||\xi||\right)$ indicates, that the term converges to zero for $\xi \to 0$ after dividing it by ξ. The matrix A_x is the *Jacobian* of f, namely

$$J_x f = \left(\frac{\partial f_i}{\partial x_j}\Big|_x \right)_{i,j}.$$

The *rank* of the map f in x is then defined as the rank of the Jacobian. If f is differentiable at every point $x \in U$ and the rank is maximal everywhere, the map f is called an *immersion* (if $n \leq m$) or a *submersion* (if $n \geq m$).

In the following, let $U \subset \mathbb{R}^2$ be an open subset and $f : U \to \mathbb{R}^3$, $(u,v) \mapsto f(u,v)$ an immersion. A *parametrised surface* is an immersion f and is also called *parametrisation*. The classical notation of a parametrisation is given by a triple of functions (x,y,z) in cartesian coordinates

$$f(u,v) = (x(u,v), y(u,v), z(u,v)) \in \mathbb{R}^3,$$

where the parameter (u,v) is mapped to the point (x,y,z). A two-dimensional submanifold of $\mathbb{R}^3$ can be locally described as a surface element. As an example, the upper cap of the unit sphere $\mathcal{S} = \{(x,y,z) \in \mathbb{R}^3 \,|\, x^2+y^2+z^2 = 1\}$ can be parametrised by

$$(u,v) \mapsto (u, v, \pm\sqrt{1-u^2-v^2}), \quad u^2+v^2 < 1.$$

Being an immersion is equivalent to the fact, that the vectors $\frac{\partial f}{\partial u}$ and $\frac{\partial f}{\partial v}$ are linearly independent at every point and span the *tangent plane*. We will introduce the following notations for $u \in U$ and $p \in f(u)$:

T_uU	is the *tangential space* of U at u	$T_uU = \{u\} \times \mathbb{R}^2$,
$T_p\mathbb{R}^3$	is the *tangential space* of $\mathbb{R}^3$ at p	$T_p\mathbb{R}^3 = \{p\} \times \mathbb{R}^3$,
T_uf	is the *tangential plane* of f at p	$T_uf := Df\vert_u(T_uU) \subset T_{f(u)}\mathbb{R}^3$

Definition B.2.2. We denote by $\langle\cdot,\cdot\rangle$ the Euclidean scalar product in $\mathbb{R}^3$ and in all the tangent spaces $T_p\mathbb{R}^3$. The *first fundamental form I* of a surface element is just the restriction of $\langle\cdot,\cdot\rangle$ to all the tangent planes T_uU, i.e.,

$$I(X,Y) := \langle X, Y\rangle$$

for two tangent vectors $X, Y \in T_uf$.

In the coordinates $f(u,v) = (x(u,v), y(u,v), z(u,v))$, the first fundamental form is described by the symmetric and positive-definite matrix

$$G = \begin{pmatrix} I(\frac{\partial f}{\partial u}, \frac{\partial f}{\partial u}) & I(\frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}) \\ I(\frac{\partial f}{\partial v}, \frac{\partial f}{\partial u}) & I(\frac{\partial f}{\partial v}, \frac{\partial f}{\partial v}) \end{pmatrix} = \begin{pmatrix} \langle \frac{\partial f}{\partial u}, \frac{\partial f}{\partial u}\rangle & \langle \frac{\partial f}{\partial u}, \frac{\partial f}{\partial v}\rangle \\ \langle \frac{\partial f}{\partial v}, \frac{\partial f}{\partial u}\rangle & \langle \frac{\partial f}{\partial v}, \frac{\partial f}{\partial v}\rangle \end{pmatrix}.$$

In the two-dimensional case the curvature is characterised by the changes of the tangents. Analogously, we expect the curvature of a surface in three dimensions to be defined by the changes in the tangent planes. As every plane is uniquely determined by its normal vector, we can study the changes in those vectors instead. This is the major motivation for the Gauss mapping.

Definition B.2.3. For a surface element $f : U \to \mathbb{R}^3$ the *Gauss map* $\nu : U \to \mathcal{S}$ with the unit sphere $\mathcal{S} = \{(x, y, z) \in \mathbb{R}^3 \,|\, x^2 + y^2 + z^2 = 1\}$, is defined as

$$\nu(u, v) := \frac{\frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v}}{\left\| \frac{\partial f}{\partial u} \times \frac{\partial f}{\partial v} \right\|}.$$

Definition B.2.4. Let $f : U \to \mathbb{R}^3$ be a surface element with Gauss map $\nu : U \to \mathcal{S} \subset \mathbb{R}^3$.

(i) For every $u \in U$ the image of the linear map

$$D\nu|_u : T_u U \to T_\nu \mathbb{R}^3$$

is parallel to the tangent plane $T_u f$. By canonically identifying $T_\nu \mathbb{R}^3$ with $\mathbb{R}^3$ and $T_{f(u)}\mathbb{R}^3$, we can take $D\nu$ at every point as a map

$$D\nu|_u : T_u U \to T_u f.$$

Furthermore, we can view the map $Df|_u$ as a linear isomorphism

$$Df|_u : T_u U \to T_u f,$$

by restricting the image. In this sense, the inverse map $(Df|_u)^{-1}$ is well-defined on the tangential plane $T_u f$ and is also an isomorphism.

(ii) The map $L := -D\nu \circ (Df)^{-1}$ that is point-wise defined by

$$L_u := -\left(D\nu|_u\right) \circ \left(Df|_u\right)^{-1} : T_u f \to T_u f,$$

is called *Weingarten map* or *shape operator* of f. Thus, for every parameter u this map is a linear endomorphism of the tangent plane at the corresponding point $f(u)$. This means that $L\left(\frac{\partial f}{\partial u_i}\right) = -\frac{\partial \nu}{\partial u_i}$ for $i = 1, 2$ and the rank of L is equal to the rank of $D\nu$.

(iii) L is independent of the parametrisation f and it is self-adjoint with respect to the first fundamental form *I*.

Definition B.2.5. Let $f : U \to \mathbb{R}^3$, $\nu : U \to \mathcal{S}$ and L be the Weingarten map of f as in Definition B.2.4. Then we define for the tangent vectors X and Y:

(i) the *second fundamental form II* of f by

$$\mathit{II}(X, Y) := \mathit{I}(LX, Y),$$

(ii) the *third fundamental form III* of f by

$$\mathit{III}(X, Y) := \mathit{I}(L^2 X, Y) = \mathit{I}(LX, LY).$$

As L is self-adjoint with respect to *I*, *II* and *III* are symmetric bilinear forms on $T_u f$ for every $u \in U$.

For the three fundamental forms, the following equation holds

$$III - \operatorname{Tr}(L)II + \operatorname{Det}(L)I = 0.$$

We consider an arbitrary curve $c = c_X$ on a surface through a fixed point p with unit tangent vector $c'(p) = X$. The curvature κ of the curve c on the surface is defined as the length of the vector c''. We decompose c'' into its tangential and normal parts:

$$c'' = \underbrace{(c'')^{Tang.}}_{\text{tangential component}} + \underbrace{\langle c'', \nu\rangle\nu}_{\text{normal component}} .$$

The normal component in p is given by

$$\langle c'', \nu\rangle\nu = \left\langle \frac{d^2c}{ds^2}, \nu \right\rangle = -\left\langle c', \frac{\partial \nu}{\partial s} \right\rangle = \langle X, LX\rangle\nu = II(X,X)\nu$$

and thus only depends on the tangent X at point p but not on the choice of the curve. Therefore, $II(X,X)$ is called *normal curvature* κ_ν of the curve c_X. It always holds that $\kappa^2 \geq \kappa_\nu^2$.

Definition B.2.6. Let $X \in T_u f$ denote the unit tangent vector, i.e., $I(X,X) = 1$. X is called *principal curvature direction* of f, if one of the following equivalent conditions is satisfied:

(i) $II(X,X)$ (the normal curvature κ_ν in the direction of X) has a stationary value among all X with $I(X,X) = 1$.

(ii) X is an eigenvector of the Weingarten map L.

The corresponding eigenvalue λ (for $LX = \lambda X$) is called the *principal curvature*.

For a two-dimensional surface the two principal curvatures are simply the maximum and minimum of the normal curvature. In this case we denote the principal curvatures of f by κ_1 and κ_2.

Definition B.2.7. (i) The determinant $K = \operatorname{Det}(L) = \kappa_1 \cdot \kappa_2$ is called the *Gaussian curvature* of f.

(ii) The average value $H = \frac{1}{2}\operatorname{Tr}(L) = \frac{1}{2}(\kappa_1 + \kappa_2)$ is called the *mean curvature* of f.

Example: As an example for the Gaussian and mean curvature, we consider the sphere $\mathcal{S} \subset \mathbb{R}^3$ with radius r. Therefore, let n be a normal vector pointing outwards. We fix a point $p \in \mathcal{S}$, choose an arbitrary tangent vector $X \in T_p\mathcal{S}$ and let $\sigma : (-\varepsilon, \varepsilon) \to \mathcal{S}$ be a curve on the sphere $\mathcal{S}$ so that $\sigma(0) = p$, $\frac{d\sigma}{dt}(0) = X$. Then the Weingarten map L is defined by

$$L(X) = -\nabla_X n = -\frac{d}{dt} n \circ \sigma(t)\bigg|_{t=0} .$$

But with $n \circ \sigma(t) = n(\sigma(t)) = \frac{\sigma(t)}{|\sigma(t)|} = \frac{\sigma(t)}{r}$ it follows that

$$L(X) = -\frac{d}{dt}\frac{\sigma(t)}{r}\bigg|_{t=0} = -\frac{1}{r}\frac{d\sigma}{dt}\bigg|_{t=0} = -\frac{1}{r}X$$

for any $X \in T_p\mathcal{S}$. Thus, the Weingarten map for a sphere $\mathcal{S}$ is defined as $L = -\frac{1}{r}\mathbb{I}$, where $\mathbb{I}$ is the identity matrix. With this and Definition B.2.7 we can determine the Gaussian curvature K and the mean curvature H of the sphere $\mathcal{S}$ as

$$K = \text{Det}(L) = \frac{1}{r^2}, \quad H = \frac{1}{2}\text{Tr}(L) = -\frac{1}{r}.$$

B.3 Graph theory

Let $P = \{p_1, \dots, p_n\}$ be a set of vertices and $E = \{e_1, \dots, e_m\}$ the set of edges connecting P, the resulting triangulation can also call *graph* Γ.

Definition B.3.1. If the edge $\overline{p_i p_j} \in E$, then we say p_i and p_j are *adjacent*. The *adjacency matrix* $A \in \mathbb{R}^{n \times n}$ with $A = (a_{ij})_{i,j=1,\dots,n}$ is then defined as

$$a_{ij} = \begin{cases} 1, & \text{if } p_i \text{ and } p_j \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases}.$$

The matrix A is a symmetric matrix with trace zero.

Definition B.3.2. A *walk* from vertex p_i to p_j of length k is defined as a finite sequence of vertices such that

$$p_i = u_0, u_1, \dots, u_k = p_j,$$

where u_{t-1} and u_t are adjacent for $1 \leq t \leq k$. A graph is called *connected* if every pair of vertices is joined by a walk.

Lemma B.3.1. A graph Γ with n vertices is connected if and only if $(A + \mathbb{I})^{n-1}$ has no zero entries, where A is the adjacency matrix of Γ (Biggs, 1993).

Lemma B.3.2. The number of walks of length k from p_i to p_j is the entry in position (i, j) of the matrix A^k (Biggs, 1993).

Consequently, if an entry a_{ij} of the adjacency matrix A^k is zero, there is no walk of length k from p_i to p_j.

Definition B.3.3. The number of edges traversed in the shortest way joining p and p' is called the *distance* between p and p' and denoted by $\partial(p, p')$. The diameter of a connected graph Γ, denoted by $diam(\Gamma)$ is the maximum distance between two vertices, namely

$$diam(\Gamma) := \max_{p \in P} \min_{p' \in P} \partial(p, p').$$

The *eccentricity* of a vertex p is the maximum distance from this point to any other vertex p', denoted by $ecc(p)$. The *radius*, denoted by $rad(\Gamma)$, is the minimum eccentricity among all vertices of Γ, namely

$$rad(\Gamma) := \min_{p \in P} \max_{p' \in P} \partial(p, p') = \min_{p \in P} ecc(p).$$

Definition B.3.4. The *degree* of $p \in P$, denoted by $deg(p)$, is the number of edges incident to p. The *maximum degree* of a graph Γ is denoted by $\Delta(\Gamma)$ and defined as

$$\Delta(\Gamma) := \max_{p \in P} deg(p).$$

The maximum number of edges in an undirected graph without loop is $n(n-1)/2$, but the following lemma gives a tighter upper boundary.

Lemma B.3.3. For a graph Γ of order n it holds (Goddard and Oellermann, 2011):

1. $diam(\Gamma) \leq n - \Delta(\Gamma) + 1$
2. $rad(\Gamma) \leq diam(\Gamma) \leq 2rad(\Gamma)$
3. $rad(\Gamma) \leq \frac{n-\Delta(\Gamma)}{2} + 1$

B.4 Uncertainty quantification

B.4.1 Orthogonal polynomials

Let $\mathcal{N} = \{0, 1, \ldots\}$ or $\mathcal{N} = \{0, 1, \ldots, N\}$ be a (possibly finite) index set of non negative integers. A system of polynomials $\{Q_n(x) \,|\, n \in \mathcal{N}\}$ with polynomials $Q_n(x)$ of exact degree n, is an *orthogonal system of polynomials* with respect to a real positive measure υ, if it satisfies the orthogonality condition

$$\int_S Q_m(x)\, Q_n(x)\, d\upsilon(x) = \gamma_n \delta_{mn}, \qquad m, n \in \mathcal{N}, \tag{B.1}$$

where S is the support of the measure υ, δ_{mn} is the Kronecker delta function with $\delta_{mn} = 1$ if $m = n$ and $\delta_{mn} = 0$ else, and γ_n is a positive constant, the so-called *normalisation constant*. The system is orthonormal, if $\gamma_n = 1$.

The measure υ normally has a density $w(x)$ or weights $w(i)$ at point i in the discrete case. With this position, equation (B.1) becomes

$$\int_S Q_m(x)\, Q_n(x) w(x)\, dx = \gamma_n \delta_{mn}, \qquad m, n \in \mathcal{N},$$

or in the discrete case

$$\sum_{i=1}^{N} Q_m(x_i)\, Q_n(x_i) w(x_i)\, dx = \gamma_n \delta_{mn}, \qquad m, n \in \mathcal{N}.$$

In the theory of orthogonal polynomials, the density $w(x)$, or the weights $w(i)$ in the discrete case, are often referred to as the *weighting functions*.

B.4.1.1 Hypergeometric series and the Askey scheme

An important class of orthogonal polynomials are the ones expressed in a unified way by using *hypergeometric series* belonging to the *Askey scheme* (Askey and Willson, 1985). This scheme classifies the hypergeometric orthogonal polynomials that satisfy certain types of differential (or difference) equations, and reveals their mutual relation. Each subset of orthogonal polynomials of the Askey scheme has a different weighting function and it has been discovered that some of these weighting functions are identical to the probability density functions (PDFs) of certain random distributions (Xiu and Karniadakis, 2002). These findings allow to represent stochastic processes with different orthogonal polynomials, according to the inherent properties of the process.

To start, we define the *Pochammer symbol* $(a)_n$ as

$$(a)_n = \begin{cases} 1, & \text{if } n = 0, \\ a(a+1)\cdots(a+n-1), & \text{if } n = 1, 2, 3, \ldots \end{cases}.$$

The *hypergeometrical series* ${}_rF_s$ is then defined as

$${}_rF_s(a_1, \ldots, a_r; b_1, \ldots, b_s; z) = \sum_{k=0}^{\infty} \frac{(a_1)_k \ldots (a_r)_k}{(b_1)_k \ldots (b_s)_k} \frac{z^k}{k!},$$

with $b_i > 0$ for $i = 1, \ldots, s$ and $r, s \in \mathbb{N}_0$. Orthogonal polynomials of this form that are linked to probability functions of certain random distributions are the Hermite, Legendre, Laguerre, Jacobi, Charlier, Meixner, Krawtchouk and Hahn polynomials. A detailed description of these polynomials and their limit relations can be found in Koekoek and Swarrouw, 1998 and Schoutens, 2000.

B.4.1.2 Examples of orthogonal polynomials

In the following, we present several orthogonal polynomials that will be used later on in the context of the generalised polynomial chaos (see Appendix B.4.3). The main focus here is on continuous polynomials and we present and discuss Hermite polynomials defined on $\mathbb{R}$, Legendre polynomials defined on $[-1, 1]$ and Laguerre polynomials defined on $[0, \infty)$.

Hermite polynomials The sequence of Hermite polynomials

$$H_n(x) = (2x)^n \, {}_2F_0 \left(-\frac{n}{2}, -\frac{n-1}{2}; \, ; -\frac{2}{x^2} \right) \tag{B.2}$$

satisfies the recursion

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), \qquad n > 0,$$

and the polynomials are orthogonal, that is,

$$\int_{-\infty}^{\infty} H_m(x)H_n(x)w(x)\,dx = n!\delta_{mn},$$

with respect to the weighting function

$$w(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}.$$

This weighting function is the same as the PDF of a standard Gaussian random variable with zero mean and unit variance.

Legendre polynomials The sequence of Legendre polynomials

$$P_n(x) = {}_2F_1\left(-n, n+1; 1; \frac{1-x}{2}\right) \tag{B.3}$$

satisfies the recursion

$$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x), \qquad n > 0,$$

and the polynomials are orthogonal, that is,

$$\int_{-1}^{1} \frac{1}{2} P_m(x) P_n(x)\, dx = \frac{1}{2n+1} \delta_{mn}.$$

Evidently, the weighting function in the orthogonality relation is constant, i.e., for the Legendre polynomials defined on $[-1, 1]$ it is $w(x) = 0.5$. In general, this is the PDF of a uniform distribution defined on a generic interval $[a, b]$ with the PDF of $f(x) = \frac{1}{b-a}$. In Figure B.2 we plot the Legendre polynomials on the interval $[-1, 1]$ up to the order $n = 5$.

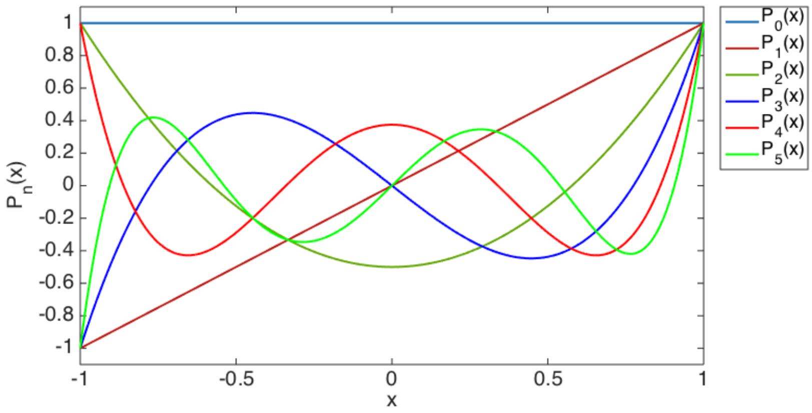


FIGURE B.2: Legendre polynomials on the interval $[-1, 1]$ up to the order $n = 5$.

Laguerre polynomials The sequence of Laguerre polynomials

$$L_n^{(\alpha)}(x) = \frac{(\alpha+1)_n}{n!} {}_1F_1\left(-n; \alpha+1; x\right) \tag{B.4}$$

satisfies the recursion

$$(n+1)L_{n+1}^{(\alpha)}(x) = (2x + \alpha + 1 - x)L_n^{(\alpha)}(x) - (n+\alpha)L_{n-1}^{(\alpha)}(x), \qquad n > 0,$$

and the polynomials are orthogonal, that is,

$$\int_0^{\infty} e^{-x} x^{\alpha} L_m^{(\alpha)}(x) L_n^{(\alpha)}(x)\, dx = \frac{\Gamma(n+\alpha+1)}{n!} \delta_{mn}.$$

Recalling that the PDF of the gamma distribution is defined as

$$f(x) = \frac{x^{\alpha} e^{-x/\beta}}{\beta^{\alpha+1}\Gamma(\alpha+1)}, \qquad \alpha > -1, \beta > 0, \text{ for } x > 0 \tag{B.5}$$

the weighting function of the Laguerre polynomial can be rescaled to the PDF of a gamma distribution.

B.4.2 Formulation of stochastic systems

We briefly describe the general aspects of formulating stochastic equations and how to properly set up the stochastic model to study the effect of uncertainties in the input to the system. It is essential to properly characterise the random inputs, which usually means to reduce the infinite-dimensional probability space to a finite one (if necessary). In general, the critical step in formulating a stochastic system is to characterise the probability space defined by the random inputs by a finite set of mutually independent random variables. Such a characterisation usually involves an approximation error.

B.4.2.1 Input parametrisation: random parameters

In the case of uncertainties in the system parameters, the randomness is in the input and the formulation of the stochastic system is straight forward. Under these circumstances the most important issue is to identify independent parameters in the set.

The problem can be formulated as follows: Let $Y = (Y_1, \ldots, Y_m)$, $m > 1$ be the system parameters with given probability distribution function $F_Y(y) = P(Y \leq y)$, $y \in \mathbb{R}^m$. Find a set of mutually independent random variables $Z = (Z_1, \ldots, Z_d) \in \mathbb{R}^d$, with $1 \leq d \leq m$, such that $Y = R(Z)$ for a suitable transformation function R.

To illustrate the previous formulation, we consider the following example of an ordinary differential equation with two random variables:

$$\frac{du}{dt}(t,\omega) = -\alpha(\omega)u, \qquad u(0,\omega) = \beta(\omega),$$

where the growth rate α and the initial condition β are assumed to be random. Thus, the input random variables are $Y(\omega) = (\alpha, \beta) \in \mathbb{R}^2$ and the solution is $u(t,\omega)\colon [0,T] \times \Omega \to \mathbb{R}$.

If α and β are mutually independent, we can simply set $Z(\omega) = Y(\omega)$ and the solution $u(t,\omega)$ can be expressed as

$$u(t,\omega)\colon [0,T] \times \mathbb{R}^2 \to \mathbb{R},$$

with one dimension in time and two random dimensions.

In the case that α and β are not mutually independent, there exists a function g such that

$$g(\alpha,\beta) = 0.$$

Then we can find a random variable $Z(\omega)$ to parametrise the above relation so that

$$\alpha(\omega) = a(Z(\omega)), \qquad \beta(\omega) = b(Z(\omega)),$$

with $g(a, b) = 0$. Thus, the random inputs α and β can now be expressed in the form of a single random variable Z and the solution can be written as

$$u(t, Z)\colon [0, T] \times \mathbb{R} \to \mathbb{R},$$

which has only one random dimension.

B.4.2.2 Input parametrisation: random processes

In many cases, the random inputs are not random variables but stochastic processes. The random inputs can be time-dependent, which is a stochastic process in time, or they account for uncertain material properties, like conductivity or capacitance, which is a stochastic process in space.

The parametrisation problem can be formulated as: Let $(Y_t, t \in T)$ be a stochastic process modelling the random inputs, where t is the index belonging to a set of indices T. Then find a suitable transformation function R such that $Y_t = R(Z)$, with mutually independent $Z = (Z_1, \ldots, Z_d) \in \mathbb{R}^d$ with $d \geq 1$.

The index set T can be either describing the space domain or the time domain and is usually infinite-dimensional. Requiring d to be a finite integer means that the transformation is not exact, thus $Y_t \approx R(Z)$, where the accuracy is problem-dependent. Obviously, the finer the discretisation, the better the approximation of Y_t, but this also leads to a larger dimension of d and increases the computational effort. Therefore, dimension reduction techniques are required that keep the dimension of Z as low as possible, while at the same time maintain a suitable approximation accuracy. For more information on random processes in space and the Karhunen-Loeve expansion, we refer to Appendix B.4.6.

B.4.2.3 General formulation

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a complete probability space, where Ω is the sample space, $\mathcal{A}$ the σ-algebra over Ω and $\mathbb{P}$ the probability measure. Let us consider a system of PDEs defined on an n-dimensional bounded spatial domain $D \subset \mathbb{R}^n$, $n = 1, 2, 3$ with a boundary ∂D and a time domain $[0, T]$ with $T > 0$,

$$\begin{cases} u_t(\boldsymbol{x}, t, \omega) = \mathcal{L}(\boldsymbol{x}, t, \omega; u), & D \times (0, T] \times \Omega \\ \mathcal{B}(\boldsymbol{x}, t, \omega; u) = 0, & \partial D \times [0, T] \times \Omega \\ u = u_0, & D \times \{t = 0\} \times \Omega \end{cases}, \tag{B.6}$$

where $\mathcal{L}$ is a differential operator, $\mathcal{B}$ is the boundary condition operator, u_0 is the initial condition and $\omega \in \Omega$ denotes the random inputs of the system. The random inputs to the system (B.6) can be in the form of random parameters or random processes.

Strong and weak solution To solve system (B.6) numerically, it is necessary to reduce the infinite-dimensional probability space to a finite-dimensional one by rewriting the random variable as an approximation of a finite number of variables.

Assuming the existence of a proper parametrisation, like discussed in Appendix B.4.2.1 and Appendix B.4.2.2, let $Z = (Z_1, \ldots, Z_d) \in \mathbb{R}^d$, $d \geq 1$ be a set of mutually independent random variables describing (or in the case of infinite-dimensional random processes, approximating) the random input. Let us further assume that these random variables

$\{Z_i\}_{i=1}^d$ have probability density functions $\rho_i \colon I_{Z_i} \to \mathbb{R}^+$ and the images $I_{Z_i} = Z_i(\Omega)$ are bounded intervals in $\mathbb{R}$ for $i = 1, \dots, d$. The joint PDF is

$$\rho(Z) = \prod_{i=1}^{d} \rho_i(Z_i), \qquad \text{for all } Z_i \in \mathbb{R},$$

with the support $I_Z = \bigotimes_{i=1}^d I_{Z_i} \subset \mathbb{R}^d$. Then we can rewrite the system (B.6) as a $(d+n)$-dimensional differential equation in the strong form

$$\begin{cases} u_t(\boldsymbol{x}, t, Z) = \mathcal{L}(\boldsymbol{x}, t, Z; u), & D \times (0, T] \times I_Z \\ \mathcal{B}(\boldsymbol{x}, t, Z; u) = 0, & \partial D \times [0, T] \times I_Z \\ u = u_0, & D \times \{t = 0\} \times I_Z \end{cases} . \tag{B.7}$$

Thus, the solution is

$$u(\boldsymbol{x}, t, Z) \colon \bar{D} \times [0, T] \times I_Z \to \mathbb{R}^{n_u},$$

where d is the dimensionality of the random space, n is the dimensionality of the physical space D and $n_u \geq 1$ is the dimension of u.

Similar to the weak formulation in deterministic problems, the system (B.7) is often solved in a weak form. For that reason, we define a finite-dimensional subspace $V_{I_Z} \subset L_p^2(I_Z)$, the space of all square integrable functions in I_Z with respect to the measure $\rho(Z)dZ$. We are looking for a solution $u^V(\boldsymbol{x}, t, Z) \in L_p^2(I_Z)$, such that

$$\int_{I_Z} \rho(Z) u_t^V(\boldsymbol{x}, t, Z) \phi(Z)\, dZ = \int_{I_Z} \rho(Z) \mathcal{L}(\boldsymbol{x}, t, Z; u^V) \phi(Z)\, dZ, \quad \text{for } \boldsymbol{x} \in D, \tag{B.8}$$

$$\int_{I_Z} \rho(Z) \mathcal{B}(\boldsymbol{x}, t, Z; u^V) \phi(Z)\, dZ = 0, \quad \text{for } \boldsymbol{x} \in \partial D, \tag{B.9}$$

for $\phi(Z) \in V_{I_Z}$. The resulting problem is a deterministic one in the physical domain D and can be solved with standard discretisation techniques, like finite elements or finite volumes.

B.4.2.4 Traditional numerical methods

There are several numerical standard techniques that can be applied to the problem (B.7), or (B.9) - (B.9). Here, we shortly review the Monte Carlo method that has a relatively slow convergence rate, but is independent of the dimensionality of the random space.

For further explanations on traditional numerical methods, including the moment equation approach and the perturbation method, we refer to Xiu, 2010.

Monte Carlo method One of the most popular and well-developed methods for solving stochastic differential equations is the Monte Carlo method. Even though it converges relatively slowly, with a convergence rate of $1/\sqrt{K}$ for a simulation consisting of K samples, it is independent of the number of random variables $\{Z_i(\omega)\}_{i=1}^d$ (Fishman, 1996). The general procedure to apply Monte Carlo sampling to problem (B.7) is:

1. For a given number of realisations K, generate independent, identically distributed random variables $\{Z_i^j\}_{i=1}^d = \{Z_i(\omega_j)\}_{i=1}^d$ for $j = 1, \dots, K$.

2. For each $j = 1, \dots, K$ solve the deterministic problem (B.7) with $Z_j = (Z_1^j, \dots, Z_d^j)$ obtaining the solution $u_j(\boldsymbol{x}, t, Z_j)$.
3. Post-process the results to obtain the solution statistics, as the solution mean

$$\bar{u}(\boldsymbol{x}, t) = \frac{1}{K} \sum_{j=1}^{K} u_j \approx \mathbb{E}(u)$$

or the solution variance

$$\mathbb{V}(u) \approx \frac{1}{K} \sum_{j=1}^{K} (u_j - \bar{u})^2.$$

Note that for each realisation $j = 1, \dots, K$ step 2 is a deterministic problem which has to be solved with a suitable scheme.

B.4.3 Generalised polynomial chaos

In this subsection we focus on the generalised polynomial chaos (gPC) expansion based on globally smooth orthogonal polynomials, in particular the polynomials of the Askey scheme introduced in Appendix B.4.1, as in contrast to for e.g. piecewise polynomials.

Let Z be a random variable with a distribution function $F_Z(z) = P(Z \leq z)$ and let $\{Q_n(x), n \in \mathcal{N}\}$ be an orthogonal system of polynomials as defined in Appendix B.4.1, that fulfil the orthogonality condition (B.1). The *generalised polynomial chaos* basis functions are defined as these orthogonal polynomial functions, that additionally satisfy

$$\gamma_n = \mathbb{E}(Q_m^2(Z)), \qquad n \in \mathcal{N}$$

for the normalisation factor. If Z is continuous, the orthogonality can be rewritten in terms of the corresponding PDF ρ, like in equation (B.1), namely

$$\mathbb{E}(Q_m(Z) Q_n(Z)) = \int Q_m(z) Q_n(z) \rho(z)\, dz = \gamma_n \delta_{mn}, \qquad m, n \in \mathcal{N}. \tag{B.10}$$

We will now consider basis functions for different types of polynomials. Let Z be a standard Gaussian random variable, namely $Z \sim \mathcal{N}(0, 1)$ with PDF $\rho(z) = (2\pi)^{-1/2} e^{-z^2/2}$. Then the orthogonality condition (B.1), or rather (B.10), defines the Hermite orthogonal polynomials $\{H_n(Z) \,|\, n \geq 0\}$ like in (B.2). Employing the Hermite polynomials as the basis function, we have

$$H_0(Z) = 1, \quad H_1(Z) = Z, \quad H_2(Z) = Z^2 - 1, \quad H_3(Z) = Z^3 - 3Z, \quad \dots. \tag{B.11}$$

The basis in (B.11) is the classical Wiener-Hermite chaos.

Let Z be a random variable uniformly distributed on $[-1, 1]$, namely $Z \sim \mathcal{U}(-1, 1)$ with constant PDF $\rho(z) = 1/2$. Then the orthogonality condition (B.1), or rather (B.10), defines the Legendre orthogonal polynomials $\{P_n(Z) \,|\, n \geq 0\}$ like in (B.3), with

$$P_0(Z) = 1, \quad P_1(Z) = Z, \quad P_2(Z) = \frac{2}{3} Z^2 - \frac{1}{2}, \quad \dots.$$

Let Z be a random variable gamma-distributed on $[0, \infty)$ with shape parameter α and rate parameter β, namely $Z \sim \Gamma(\alpha, \beta)$ with PDF as defined in (B.5). Then the orthogonality condition (B.1), or rather (B.10), defines the Laguerre orthogonal polynomials $\{L_n(Z) \,|\, n \geq 0\}$ like in (B.4), with

$$L_0(Z) = 1, \quad P_1(Z) = -Z + 1, \quad L_2(Z) = \frac{1}{2}(Z^2 - 4Z + 2), \quad \dots . \tag{B.12}$$

In Table B.1 we list the established correspondences between the probability distribution of Z and its gPC basis polynomials.

	random variable	**Wiener-Askey chaos**	**support**
continuous	Gaussian	Hermite-chaos	$(-\infty, \infty)$
	gamma	Laguerre-chaos	$[0, \infty)$
	beta	Jacobi-chaos	$[a, b]$
	uniform	Legendre-chaos	$[a, b]$
discrete	Poisson	Charlier-chaos	$\{0, 1, 2, \dots, N\}$
	binomial	Krawtchouk-chaos	$\{0, 1, 2, \dots, N\}$
	negative binomial	Meixner-chaos	$\{0, 1, 2, \dots\}$
	hypergeometric	Hahn-chaos	$\{0, 1, 2, \dots, N\}$

TABLE B.1: The different types of Wiener-Askey scheme polynomials and their underlying random distribution.

In general, the homogenous chaos expansion employs orthogonal polynomials from the Askey scheme in terms of random variables. The resulting polynomial chaos provides a method for expanding second-order random processes in terms of orthogonal polynomials. The Wiener-Askey polynomial chaos is a generalisation of the original Wiener polynomial chaos, using Hermite polynomials for the expansion to account for Gaussian distributed random variables. Thus, the Wiener-Askey polynomial chaos allows accounting for a wider range of distributions of random variables and processes.

We can represent the general random process $Y(\omega)$ by

$$\begin{aligned} Y(\omega) \quad = \quad & a_0 \, Q_0 \\ & + \sum_{i_1=1}^{\infty} c_{i_1} \, Q_1(\xi_{i_1}(\omega)) \\ & + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{\infty} c_{i_1} c_{i_2} \, Q_2(\xi_{i_1}(\omega), \xi_{i_2}(\omega)) \\ & + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{\infty} \sum_{i_3=1}^{\infty} c_{i_1} c_{i_2} c_{i_3} \, Q_3(\xi_{i_1}(\omega), \xi_{i_2}(\omega), \xi_{i_3}(\omega)) \\ & + \dots , \end{aligned} \tag{B.13}$$

where $Q_n(\xi_{i_1}, \dots, \xi_{i_n})$ denotes the Wiener-Askey polynomial chaos of order n in terms of the random vector $\boldsymbol{\xi} = (\xi_{i_1}, \dots, \xi_{i_n})$. For notational convenience, we can summarise equation (B.13) as

$$Y(\omega) = \sum_{j=1}^{\infty} \widehat{c}_j \, Q_j(\boldsymbol{\xi}). \tag{B.14}$$

Then the orthogonality condition of the Wiener-Askey polynomial chaos can be written as

$$\langle Q_m, Q_n \rangle = \langle Q_m, Q_m \rangle \, \delta_{mn}, \tag{B.15}$$

where δ_{mn} is the Kronecker delta and $\langle \cdot, \cdot \rangle$ denotes the ensemble average, namely

$$\langle f(\boldsymbol{\xi}), g(\boldsymbol{\xi}) \rangle = \int f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) w(\boldsymbol{\xi}) \, d\boldsymbol{\xi},$$

where $w(\boldsymbol{\xi})$ denotes the weighting function of the corresponding polynomial chaos basis $\{Q_n\}_{n\geq 1}$. As previously mentioned, some of the polynomials from the Askey scheme have weighting functions corresponding to the distribution of certain types of random variables.

B.4.3.1 General procedure

Now, we introduce the general procedure of how to apply the Wiener-Askey polynomial chaos to solve stochastic differential equations. Let us consider the differential equation

$$\mathcal{L}(\boldsymbol{x}, t, \omega; u) = f(\boldsymbol{x}, t, \omega). \tag{B.16}$$

Then the solution $u = u(\boldsymbol{x}, t, \omega)$ can be expanded by the Wiener-Askey polynomial chaos as

$$u(\boldsymbol{x}, t, \omega) = \sum_{j=0}^{P} u_j(\boldsymbol{x}, t) \, Q_j(\boldsymbol{\xi}(\omega)). \tag{B.17}$$

The total number of expansion terms is depending on the dimension n of the random variable $\boldsymbol{\xi}$ and the highest order p of the polynomials $\{Q_n\}_{n\geq 1}$, i.e.,

$$P + 1 = (n!p!)^{-1}(n+1)! \, .$$

Substituting (B.17) into equation (B.16) results in

$$\mathcal{L}\left(\boldsymbol{x}, t, \omega; \sum_{j=0}^{P} u_j(\boldsymbol{x}, t) \, Q_j(\boldsymbol{\xi}\,(\omega)) \right) = f(\boldsymbol{x}, t, \omega). \tag{B.18}$$

To ensure that the error is orthogonal to the functional space spanned by the finite dimensional basis $\{Q_n\}_{n\geq 1}$, a Galerkin projection of equation (B.18) is applied to each polynomial basis element.

$$\left\langle \mathcal{L}\left(\boldsymbol{x}, t, \omega; \sum_{j=0}^{P} u_j(\boldsymbol{x}, t) \, Q_j(\boldsymbol{\xi}\,(\omega)) \right), Q_k \right\rangle = \langle f(\boldsymbol{x}, t, \omega), Q_k \rangle \,, \qquad k = 0, 1, \ldots, P. \tag{B.19}$$

From (B.19) and the orthogonality of the polynomial basis, we obtain a set of $P+1$ coupled equations for each random mode $u_j(\boldsymbol{x}, t)$. The Wiener-Askey chaos expansion transfers the randomness into the polynomial basis, so that the equations for the expansion coefficient u_j from (B.19) are deterministic and can be solved with any conventional deterministic technique.

B.4.3.2 Stochastic ordinary differential equations

To exemplify the above, we consider the following ordinary differential equation

$$\frac{dy(t)}{dt} = -ky, \qquad y(0) = y_0, \tag{B.20}$$

where the coefficient k is assumed to be a random variable $k(\omega)$ with a specific distribution and mean value $\bar{k}$. The PDF is defined as $f(k)$ and the deterministic solution of (B.20) is $y(t) = y_0 e^{-\bar{k}t}$, and

$$\bar{y}(t) = y_0 \int_S e^{-kt} f(k)\, dk$$

is the mean of the stochastic solution where S is the support of f. Applying the Wiener-Askey polynomial chaos expansion (B.14) to the solution y and the random input k, we can rewrite these accordingly as

$$y(t) = \sum_{i=0}^{P} y_i(t) Q_i, \qquad k = \sum_{i=0}^{P} k_i Q_i.$$

Substituting this into the governing equation (B.20), we obtain

$$\sum_{i=0}^{P} \frac{dy_i(t)}{dt} Q_i = -\sum_{i=0}^{P}\sum_{j=0}^{P} k_i y_j(t) Q_i Q_j.$$

Then we project the above equation onto the polynomial basis $\{Q_n\}_{n\geq 1}$, to ensure the orthogonality of the error to the space spanned by the orthogonal polynomials. Using the orthogonality condition (B.15), we obtain by applying $\langle \cdot, Q_l \rangle$ to the previous equation

$$\frac{dy_l(t)}{dt} = -\frac{1}{\langle Q_l, Q_l \rangle} \sum_{i=0}^{P}\sum_{j=0}^{P} k_i y_j(t) e_{ijl}, \qquad \text{for } l = 0, 1, \ldots, P,$$

with $e_{ijl} = \mathbb{E}(Q_i Q_j Q_l)$. From this point on, a standard solver for ordinary differential equations can be applied.

B.4.3.3 Definition in higher dimensions

In the case of more than one random variable, the multivariate gPC expansion is required. Therefore, let $Z = (Z_1, \ldots, Z_d)$ be a random vector with independent components with the distribution $F_Z(z_1, \ldots, z_d) = P(Z_1 \leq z_1, \ldots Z_d \leq z_d)$ with $z = (z_1, \ldots, z_d)$. For every $i = 1, \ldots, d$, let $F_{Z_i}(z_i) = P(Z_i \leq z_i)$ be the marginal distribution of Z_i on the support I_{Z_i}. The independence of the random variables Z_i implies that $F_Z(z) = \prod_{i=1}^{d} F_{Z_i}(z_i)$ and $I_Z = \bigotimes_{i=1}^{d} I_{Z_i}$. Then, let $\{Q_k(Z_i)\}_{k=0}^{N} \in \mathcal{P}_N(Z_i)$ be the univariate gPC basis functions in Z_i of degree smaller or equal to N. These functions satisfy the orthogonality condition, that is,

$$\mathbb{E}(Q_j(Z_i) Q_k(Z_i)) = \int Q_j(z_i) Q_k(z_i) dF_{Z_i} = \gamma_j \delta_{jk}, \qquad 0 \leq j, k \leq N. \tag{B.21}$$

Let $\boldsymbol{i} = (i_1, \ldots, i_d) \in \mathbb{N}_0^d$ be a multi-index with $|\boldsymbol{i}| = i_1 + \ldots + i_d$. Then, the d-variate N-th degree gPC basis functions are the product of gPC polynomials of the total degree less or equal to N, that is,

$$\boldsymbol{Q_i}(Z) = Q_{i_1}(Z_1)\cdots Q_{i_d}(Z_d), \qquad 0 \leq |\boldsymbol{i}| \leq N.$$

With this, it follows directly

$$\mathbb{E}\left(\boldsymbol{Q_i}(Z)\boldsymbol{Q_j}(Z)\right) = \int \boldsymbol{Q_i}(z)\boldsymbol{Q_j}(z)dF_Z = \gamma_{\boldsymbol{i}}\delta_{\boldsymbol{ij}},$$

where $\gamma_{\boldsymbol{i}} = \mathbb{E}(\boldsymbol{Q}_{\boldsymbol{i}}^2) = \gamma_{i_1}\cdots\gamma_{i_d}$ are the normalisation factors and $\delta_{\boldsymbol{ij}} = \delta_{i_1,j_1}\cdots\delta_{i_1,j_d}$ the d-variate Kronecker delta function. The span of these polynomials is $\mathbb{P}_N^d$, the linear space of polynomials of degree at most N in d variables, namely

$$\mathbb{P}_N^d(Z) = \left\{ p\colon I_Z \to \mathbb{R} \,\middle|\, p(Z) = \sum_{|\boldsymbol{i}|\leq N} c_{\boldsymbol{i}}\boldsymbol{Q_i}(Z) \right\},$$

whose dimension is $\binom{N+d}{N}$.

B.4.4 Stochastic collocation method

In the *probabilistic collocation method* (PCM) system (B.6) is evaluated at a discrete set of points, called nodes, in the random space. In this perspective, the classical sampling methods, like the Monte Carlo method are collocation methods. In the following, we describe this approach using the polynomial approximation theory with Legendre polynomials.

Let $\Theta_M = \{Z^{(i)}\}_{i=1}^M \subset I_Z$ be a set of nodes in the random space, where $M \geq 1$ is the number of nodes and I_Z the support of Z. Then in the collocation method we enforce (B.7) at the node $Z^{(i)}$, for all $i = 1, \ldots, M$. Thus, for every $Z^{(i)}$, the system is a deterministic problem as the value of the random parameter is fixed and reads

$$\begin{cases} u_t(\boldsymbol{x}, t, Z^{(i)}) = \mathcal{L}(\boldsymbol{x}, t, Z^{(i)}; u), & D \times (0, T] \times I_Z \\ \mathcal{B}(\boldsymbol{x}, t, Z^{(i)}; u) = 0, & \partial D \times [0, T] \times I_Z \\ u = u_0, & D \times \{t = 0\} \times \Omega \end{cases}.$$

Let $u^{(i)} = u(\cdot, \cdot, Z^{(i)})$ be the solution of the above system for $i = 1, \ldots, M$. The result is a set of deterministic solutions $\{u^{(i)}\}_{i=1}^M$.

In the *Legendre interpolation approach*, the solution u is approximated by a polynomial $v(Z) \in \Pi(Z)$ rewritten, such that $v(Z^{(i)}) = u^{(i)}$ for all $i = 1, \ldots, M$, namely

$$v(Z) = \sum_{i=1}^{M} u(Z^{(i)})P_j(Z), \tag{B.22}$$

where $P_j(Z^{(i)}) = \delta_{ij}$ for $1 \leq i, j \leq M$ are the Lagrangian polynomials and $u(Z^{(i)})$ is the solution at a given node $Z^{(i)}$. Consequently, the collocation method is equivalent to solving M deterministic problems, the deterministic counterpart of problem (B.6), at each node $Z^{(i)}$, for $i = 1, \ldots, M$. With this, the statistics of the random solution can be evaluated, e.g.,

$$\begin{aligned}\mathbb{E}(u) \approx \mathbb{E}(v) &= \sum_{i=1}^{M} u(Z^{(i)}) \int_a^b L_i(Z)\rho(Z)\,dZ \\ &\approx \sum_{i=1}^{M} u(Z^{(i)}) \frac{1}{b-a} \sum_{k=1}^{M} L_i(Z_k) w_k \\ &= \frac{1}{b-a} \sum_{i=1}^{M} u(Z^{(i)}) w_i,\end{aligned}$$

with the assumption of a uniform distribution with the PDF ρ on the interval $[a, b]$. The w_i are the weights of the collocation points depending on the choice of the quadrature rule.

B.4.4.1 Collocation points

The computational effort of the stochastic collocation method is M times that of the deterministic problem, where M is the total number of nodes. In order to make the procedure more effective and faster, we choose a nodal set with as few points as possible but maintaining a prescribed accuracy of the results. In the following, we present different choices of collocation points, with the main focus on multi-dimensional random spaces, that is $I_Z \subset \mathbb{R}^d, d > 1$. We assume that the bounded support of the random variable Z_i is $I_{Z_i} = [-1, 1]$, for $i = 1, \dots, d$, so that the random space is a d-dimensional hypercube, namely $I_Z = [-1, 1]^d$. Random variables with a bounded support $[a, b]$ can be mapped to $[-1, 1]$.

Tensor product collocation The most natural way to choose the nodal set is the tensor product of the one-dimensional sets. For any smooth function $f\colon [-1, 1]^d \to \mathbb{R}$, we can construct a good one-dimensional interpolation (for $d = 1$):

$$\mathcal{U}^i(f) = \sum_{k=1}^{m_i} f(Z_i^k) w_i^k, \qquad i = 1, \dots, d, \tag{B.23}$$

for a nodal set $\Theta_i^1 = (Z_i^1, \dots, Z_i^{m_i}) \subset [-1, 1]$ and interpolation weights $w_i^k = w^k(Z_i)$. In the multivariate case, with $d > 1$, the tensor product formula reads

$$\mathcal{I}(f) = (\mathcal{U}^{i_1} \otimes \dots \otimes \mathcal{U}^{i_d})(f) = \sum_{k_1=1}^{m_{i_1}} \dots \sum_{k_d=1}^{m_{i_d}} f(Z_{i_1}^{k_1}, \dots, Z_{i_d}^{k_d}) \cdot (w_{i_1}^{k_1} \otimes \dots \otimes w_{i_d}^{k_d}).$$

In the above product formula, we need $M = m_{i_1} \cdots m_{i_d}$ nodal points for the collocation method. Choosing the same interpolation function in all dimensions, i.e., $m_{i_1} = \ldots = m_{i_d} = m$, the total number of points is m^d which quickly grows in high dimensions $d \gg 1$.

Consequently, the tensor product interpolation is easy to construct but the number of interpolation nodes grows rapidly with the dimension of the random space. So for higher dimensions it is reasonably to construct the interpolation grid more efficiently.

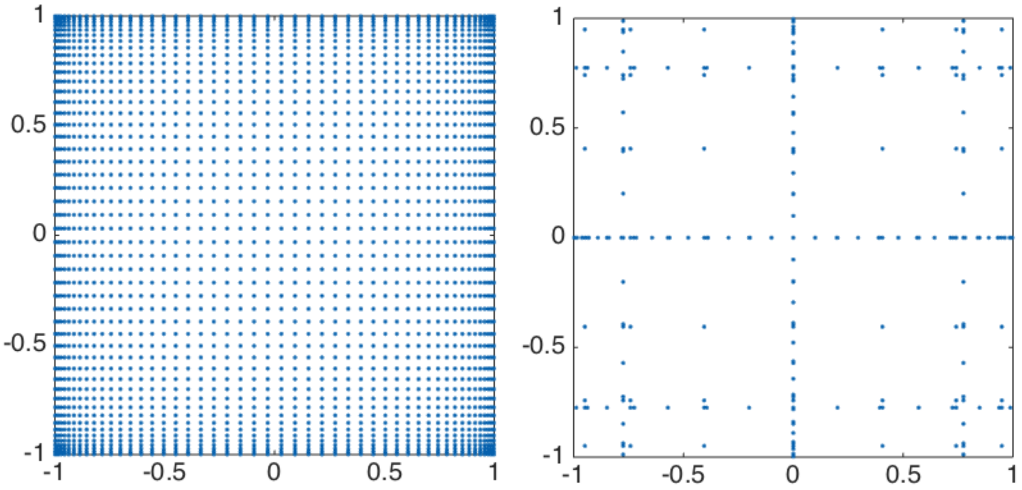
Sparse grids One way to reduce the total number of nodal points is to construct *sparse grids*, for example with the Smolyak algorithm (Smolyak, 1963). Considering the one-dimensional interpolation formula (B.23), the Smolyak algorithm is given by

$$\mathcal{I}(f) = A(q,d) = \sum_{q-d+1 \leq |\boldsymbol{i}| \leq q} (-1)^{q-|\boldsymbol{i}|} \binom{d-1}{q-|\boldsymbol{i}|} (\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_d}), \tag{B.24}$$

where $\boldsymbol{i} = (i_1, \ldots, i_d) \in \mathbb{N}^d$ is a multi-index and $q \geq d$ is an integer denoting the level of construction (Gerstner and Griebel, 1998; Xiu and Hesthaven, 2005). The result is a combination of subsets of the full tensor grid. To compute $A(q,d)$, we only need to evaluate the function on the sparse grid

$$\Theta_d = \bigcup_{q-d+1 \leq |\boldsymbol{i}| \leq q} (\Theta^1_{i_1} \otimes \cdots \otimes \Theta^1_{i_d}).$$

In Figure B.3 we visualise a comparison of the full tensor grid and a sparse grid in two dimensions ($d = 2$). The Smolyak formulas for this are based on one-dimensional polynomial interpolation at the extrema of the Legendre polynomials. The full grid is a tensor product of 50 nodes in each dimension, whereas the sparse grid has the level $q = 4$ of construction. Table B.2 lists the number of nodes of the Smolyak sparse grid for different dimensions d and levels q.



(a) full grid with 2500 points (b) sparse grid with 224 points

FIGURE B.3: Two-dimensional nodes ($d = 2$) based on the extrema of the Legendre polynomials at level $q = 4$. The total number of points is 2500 (a), whereas the number of nodes in the Smolyak sparse grid is 224 (b).

B.4.4.2 Multi-element probabilistic collocation method

The multi-element probabilistic collocation method (ME-PCM) is a generalised form of the PCM, where the parameter space is discretised and a collocation grid is prescribed on each element. The ME-PCM offers the advantages of domain decomposition in the parametric space and is especially useful to treat problems with discontinuous dependence

dimension d	**level** q	**degrees of freedom**
2	1	5
	2	22
	3	75
	4	224
	5	613
	6	1578
3	2	50
	3	161
	4	608
	5	2070
	6	6507
4	2	57
	3	289
	4	1268
	5	4994
	6	18976

TABLE B.2: The degrees of freedom of the Smolyak grid for different dimensions d and levels q.

on parameters. Using ME-PCM only requires a wrapper around the deterministic solver of the PDE system for every sample calculation.

We consider problem (B.6) and its fully deterministic counterpart (B.7) for a finite-dimensional probability space. The main idea of the ME-PCM is to discretise the space I_Z into non-overlapping elements and perform a standard PCM on each element. Then the resulting approximation of the local moment statistics on each element can be used to assemble the global statistics.

We assume $I_Z \subset \mathbb{R}^d$ to be bounded and define $\{B^i\}_{i=1}^{N_e}$ as a finite collection of non-overlapping subsets of I_Z, such that $\bigcup_{i=1}^{N_e} \overline{B^i} = I_Z$ with $B^i \cap B^j = \emptyset$, for $i \neq j$. For the sake of simplicity, we assume that B^i are rectangular, that is, $B^i = \prod_{j=1}^{d} B^i_j$ with $B^i_j \subset I_{Z_j}$. We refer to these sets as *elements* on the parameter space I_Z and thus N_e refers to the number of elements in a certain mesh discretisation. For a given mesh, we define a set of collocation points $\{\boldsymbol{q}^i_j\}_{j=1}^{r}$ on each element B^i, where r denotes the number of collocation points. Usually, the collocation points are chosen in a way that they coincide with the points of the quadrature rule on B^i with integration weights $\{w^i_j\}_{j=1}^{r}$.

Let u_k denote the semi-discrete solution (in space and/or time) of problem (B.7), then this solution is collocated on the set of points $\bigcup_{i=1}^{N_e} \bigcup_{j=1}^{r} \boldsymbol{q}^i_j$. This means that in each of these points $\boldsymbol{q}^i_j$ we find a finite element solution of the deterministic problem, which we denote by $u_k(\cdot, \boldsymbol{q}^i_j)$. The fully discrete approximation $\mathcal{I}_{B^i} u_k(\boldsymbol{x}, t; Z)$ using the set of solutions on each element B^i, can for example be chosen as the tensor product Legendre interpolant, namely

$$\mathcal{I}_{B^i} u_k(\boldsymbol{x}, t; Z) = \sum_{j=1}^{r} u_k(\boldsymbol{x}, t; \boldsymbol{q}^i_j) P^i_j(Z),$$

where $P^i_j(Z)$ is the Legendre polynomial corresponding to the point $\boldsymbol{q}^i_j$. Another choice of the operator $\mathcal{I}_{B^i}$ can also be the Smolyak sparse grid operator introduced in Appendix B.4.4.1.

Then we can define the global approximation of the solution as

$$\tilde{u}(\boldsymbol{x},t;Z) = \sum_{i=1}^{N_e} \mathcal{I}_{B^i} u_k(\boldsymbol{x},t;Z)\mathbb{1}_{B^i},$$

where $\mathbb{1}_{B^i}$ denotes the characteristic function of the set B^i.

For the statistics, i.e., the approximate global mean, we define the PDF on each element as

$$\eta^i(Z) = \frac{\rho(Z)}{\int_{B^i} \rho(Z)\, dZ}.$$

We assume the PDF ρ to be a tensor product form, namely $\rho(Z) = \prod_{j=1}^{d} \rho_j(Z_j)$, which is inherited by the local density $\eta^i(Z) = \prod_{j=1}^{d} \eta_j(Z_j)$. Then the local mean of a function $v\colon \overline{D} \times [0,T] \times B_i \to \mathbb{R}$ on an element B^i is given by

$$\mathbb{E}^i(v(\boldsymbol{x},t;\cdot)) = \mathbb{E}(v(\boldsymbol{x},t;\cdot)|Z(\omega) \in B^i) = \int_{B^i} v(\boldsymbol{x},t;Z)\eta^i(Z)\, dZ.$$

Using a quadrature rule, we can approximate the local mean of $\tilde{u}$ on each element B^i as

$$\mathbb{E}^i(\tilde{u}) \approx \sum_{j=1}^{r} u_k(\boldsymbol{x},t;\boldsymbol{q}_j^i)w_j^i.$$

With this, we can approximate the global mean of $\tilde{u}$ over all the elements, as

$$\mathbb{E}(\tilde{u}) \approx \sum_{i=1}^{N_e} \mathbb{E}^i(\tilde{u})(\boldsymbol{x},t)P(Z(\omega) \in B^i).$$

B.4.5 ANOVA approach and sensitivity measures

The analysis of variance (ANOVA) expansion offers a way to represent multivariate functions $f(\boldsymbol{x})$ in high dimensions when estimating the sensitivity indices with a variance-based approach. In general, the standard ANOVA expansion consists of a unique orthogonal decomposition of the function $f(\boldsymbol{x})$ for independent random inputs. The drawback of this expansion is the need for the computation of the high dimensional integrals when dealing with low-order component functions. The anchored ANOVA decomposition, on the other hand offers a computationally more efficient way for numerical evaluations of the component functions in the ANOVA expansion (Tang, Congedo, and Abgrall, 2015).

B.4.5.1 Standard ANOVA expansion

Let $y = f(\boldsymbol{x})$ be a function describing the model under investigation defined on the n-dimensional unit cube $I^n = [0,1]^n$ with $\boldsymbol{x} = (x_1,\ldots,x_n) \in I^n$. Then we can represent $f(\boldsymbol{x})$ as

$$f(\boldsymbol{x}) = f_0 + \sum_{s=1}^{n} \sum_{i_1<\ldots<i_s}^{n} f_{i_1\ldots i_s}(x_{i_1},\ldots,x_{i_s}), \tag{B.25}$$

with $1 \leq i_1 < \ldots < i_s \leq n$, or equivalently

$$f(\boldsymbol{x}) = f_0 + \sum_{1 \leq i \leq n} f_i(x_i) + \sum_{1 \leq i < j \leq n} f_{ij}(x_i, x_j) + \ldots + f_{12\ldots n}(x_1, \ldots, x_n).$$

The total number of summands in these formulas is 2^n. Alternatively, using a more compact representation based on multi-indices s_j, the ANOVA decomposition can be written as:

$$f(\boldsymbol{x}) = f_{s_0} + \sum_{j=1}^{2^n-1} f_{s_j}(x_{s_j}).$$

In the above expression, the multi-indices s_j are defined as

$$\begin{aligned}
s_0 &= (0, 0, 0, \ldots, 0) \\
s_1 &= (1, 0, 0, \ldots, 0) \\
s_2 &= (0, 1, 0, \ldots, 0) \\
&\vdots \\
s_n &= (0, 0, 0, \ldots, 1) \\
s_{n+1} &= (1, 1, 0, \ldots, 1) \\
s_{n+2} &= (1, 0, 1, \ldots, 1) \\
&\vdots \\
s_{\mathcal{N}} &= (1, 1, 1, \ldots, 1)
\end{aligned}$$

with $\mathcal{N} = 2^n - 1$.

The expression (B.25) is called ANOVA-representation of $f(\boldsymbol{x})$ if f_0 is constant and the integral of every summand over any of its own variables is zero, namely

$$\int_0^1 f_{i_1\ldots i_s}(x_{i_1}, \ldots, x_{i_s}) dx_k = 0, \qquad \text{for } k = i_1, \ldots, i_s \text{ and } s = 1, \ldots, n. \tag{B.26}$$

This equality can also be interpreted as an orthogonality condition under which every additional dimension considered in the function f is orthogonal to the preceding ones. From equations (B.25) and (B.26) it follows that $\int_{I^n} f(x)\, dx = f_0$ and that the summands in equation (B.25) are orthogonal and can be expressed as integrals of $f(\boldsymbol{x})$, that is

$$f_0 + f_i(x_i) = \int_0^1 \cdots \int_0^1 f(x)\, dx_{-\{i\}}, \tag{B.27}$$

$$f_0 + f_i(x_i) + f_j(x_j) + f_{ij}(x_i, x_j) = \int_0^1 \cdots \int_0^1 f(x)\, dx_{-\{ij\}}, \quad \ldots,$$

where $dx_{-\{i\}}$ denotes the integration over all variables except for those within the subscript parenthesis, namely $x_{-\{i\}} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$.

Then the variance of f is the sum of the variances of all the decomposition terms:

$$\sigma^2(f) = \sum_{s=1}^{n} \sum_{i_1<\ldots<i_s}^{n} \sigma^2(f_{i_1\ldots i_s}(x_{i_1},\ldots,x_{i_s})),$$
$$\sigma^2(f_{i_1\ldots i_s}(x_{i_1},\ldots,x_{i_s})) = \int_{I^s} f^2_{i_1\ldots i_s}(x_{i_1},\ldots,x_{i_s})\,dx_{i_1}\cdots dx_{i_s},$$

or equivalently

$$\sigma^2(f) = \sum_{1\leq j_1\leq n} \sigma^2(f_{j_1}) + \sum_{1\leq j_1\leq j_2\leq n} \sigma^2(f_{j_1 j_2}) + \ldots + \sigma^2(f_{12\ldots n}).$$

For the above equality, it is important that the measure used is the same as in the ANOVA decomposition (B.27). The value of $\sigma(f)$ is likely to be different from the exact value computed by the standard definition of the variance, that is, the integral with the Lebesgue measure.

B.4.5.2 Anchored ANOVA expansion

The construction of the standard ANOVA expansion requires the evaluation of high-dimensional integration, which can be very expensive. An alternative formulation, the anchored (or Dirac) ANOVA expansion requires no integrals, but has an accuracy that depends heavily on the choice of the so-called anchor point.

Instead of the Lebesgue measure we can use the Dirac measure in the integrations, i.e., $d\mu(\boldsymbol{x}) = \delta(\boldsymbol{x}-c)\,d\boldsymbol{x}$, $\boldsymbol{x} \in I^n$. The point c is called *anchor point* and this method is called the anchored ANOVA expansion. Consequently, we have

$$f_0 = f(c) \tag{B.28}$$

for the constant term. Different choices of the anchor point may lead to different approximations of the ANOVA decomposition of a function. For strategies on the selection of the anchor point, we refer for example to Gao and Hesthaven, 2010.

Applying the anchored ANOVA method to stochastic problems, we denote the highest dimension for the component function with ν and approximate $f(x)$ with

$$\begin{aligned} f(\boldsymbol{x}) \approx \quad & f_0 + \sum_{j_1\leq n} f_{j_1}(x_{j_1}) + \sum_{j_1<j_2\leq n} f_{j_1 j_2}(x_{j_1},x_{j_2}) + \ldots \\ & + \sum_{j_1<j_2<\ldots<j_\nu\leq n} f_{j_1 j_2\ldots j_\nu}(x_{j_1},x_{j_2},\ldots,x_{j_\nu}). \end{aligned} \tag{B.29}$$

Here, n is called the *nominal dimension*, ν the *truncation* or *effective dimension* and we use μ in the following to denote the number of collocation points in each dimension.

After obtaining the ANOVA decomposition of function f, we can approximate the integration of the component functions with the values of these functions at the quadrature points with the corresponding weights. These quadrature points can also be used as collocation points for PCM.

B.4.5.3 Adaptive ANOVA expansion

Even though the ANOVA method reduces the computational effort, it still requires a high number of sampling to compute all the terms for high nominal dimensions n. An effective way to reduce the computational effort is the adaptive ANOVA decomposition, where the nominal dimension is replaced by an active dimension D_i for each subgroup. In this approach, approximation (B.29) is modified to be

$$\begin{aligned} f(\boldsymbol{x}) \approx \quad & f_0 + \sum_{j_1 \leq D_1} f_{j_1}(x_{j_1}) + \sum_{j_1 < j_2 \leq D_2} f_{j_1 j_2}(x_{j_1}, x_{j_2}) + \dots \\ & + \sum_{j_1 < j_2 < \dots < j_\nu \leq D_\nu} f_{j_1 j_2 \dots j_\nu}.(x_{j_1}, x_{j_2}, \dots, x_{j_\nu}). \end{aligned}$$

In most cases, the decomposition is truncated with $\nu = 2$ and $D_1 = n$. Adaptive criteria to determine the active dimension were, i.e., developed in Yang et al., 2012.

B.4.5.4 Sensitivity analysis

Assuming that $f(\boldsymbol{x})$ is square integrable, the total variance of $f(\boldsymbol{x})$ can be written as $D = \int_{I^n} f^2(\boldsymbol{x})\, d\boldsymbol{x} - f_0^2$ and all the $f_{i_1 \dots i_s}$ are square integrable. Squaring equation (B.25) and integrating over I^n, we obtain

$$\int_{I^n} f^2(\boldsymbol{x})\, d\boldsymbol{x} - f_0^2 = \sum_{s=1}^{n} \sum_{i_1 < \dots < i_s}^{n} \int_{I^s} f^2_{i_1 \dots i_s}(x_{i_1}, \dots, x_{i_s})\, dx_{i_1} \cdots dx_{i_s},$$

with $I^s = [0,1]^s$. Defining the contribution to the total variance from the term $f_{i_1 \dots i_s}$ in the expansion as

$$D_{i_1 \dots i_s} = \int_{I^s} f^2_{i_1 \dots i_s}(x_{i_1}, \dots, x_{i_s})\, dx_{i_1} \cdots dx_{i_s},$$

the total variance can also be written as

$$D = \sum_{s=1}^{n} \sum_{i_1 < \dots < i_s}^{n} D_{i_1 \dots i_s}. \tag{B.30}$$

Then the global sensitivity indices are defined as the ratio

$$S_{i_1 \dots i_s} = \frac{D_{i_1 \dots i_s}}{D}, \tag{B.31}$$

where s is often called the *order* or the *dimension of the index* $S_{i_1 \dots i_s}$. From definition (B.31) and (B.30), it follows that all $S_{i_1 \dots i_s}$ are non-negative and

$$\sum_{s=1}^{n} \sum_{i_1 < \dots < i_s}^{n} S_{i_1 \dots i_s} = 1.$$

Exemplarily, we present how to obtain the mean and variance D_i of the first- and second-order terms in the ANOVA decomposition. Therefore, let $q_1^1, \dots, q_1^\mu$ be the quadrature points for the one-dimensional approximation. As we consider the Dirac measure when computing the component terms, this means that for example the value of f_1 in the quadrature point q_1^1 is

$$f_1(q_1^1) = f(q_1^1, c_2, c_3, \dots, c_n) - f_0,$$

where $c = (c_1, \dots, c_n)$ is the anchor point and $f_0 = f(c)$, see equation (B.28). Then the mean and variance of f_1 can be approximated by

$$\mathbb{E}(f_1) \approx \sum_{i=1}^{\mu} f_1(q_1^i) w_1^i, \qquad \sigma^2(f_1) \approx \sum_{i=1}^{\mu} f_1^2(q_1^i) w_1^i - (\mathbb{E}(f_1))^2\,,$$

where $w_1^1, \dots, w_1^\mu$ are quadrature weights corresponding to collocation points $q_1^1, \dots, q_1^\mu$. This definition applies analogously to all first-order terms $f_2, \dots, f_n$ with the general notation $f_k(q_k^i) = f(c_1, \dots, c_{k-1}, q_k^i, c_{k+1}, \dots, c_n) - f_0$.

Similarly, we can compute the second-order terms. For example, we consider f_{12} at the quadrature point (q_1^i, q_2^j) with $i, j = 1, \dots, \mu$. Using equation (B.27), we obtain

$$f_{12}(q_1^i, q_2^j) = f(q_1^i, q_2^j, c_3, c_4, \dots, c_n) - f_1(q_1^i) - f_2(q_2^j) - f_0,$$

and

$$\begin{aligned} \mathbb{E}(f_{12}) &\approx \sum_{i=1}^{\mu} \sum_{j=1}^{\mu} f_{12}(q_1^i, q_2^j) w_{12}^{ij} \\ \sigma^2(f_{12}) &\approx \sum_{i=1}^{\mu} \sum_{j=1}^{\mu} f_{12}^2(q_1^i, q_2^j) w_{12}^{ij} - (\mathbb{E}(f_{12}))^2\,, \end{aligned}$$

where w_{12}^{ij} with $i, j = 1, \dots, \mu$ are the quadrature weights corresponding to the tensor grid of the collocation points (q_1^i, q_2^j). Here, we use the tensor product rule to obtain the quadrature rule in higher dimensions, but other methods such as sparse grids can be used.

Generally speaking, for the higher-order terms of the component functions, we have

$$f_{s_i} = f(c|x_{s_i}) - \sum_{s_j \subset s_i} f_{s_i}(x_{s_i}),$$

where $f(c|x_{s_i})$ describes the value of $f(\boldsymbol{x})$ evaluated at the anchor point c except for the variables involved in x_{s_i}.

As an illustrative example, we consider a function with separated variables that was considered in Archer, Satelli, and Sobol', 1997 and Sobol', 2001, namely

$$g(\boldsymbol{x}) = \prod_{i=1}^{n} \varphi_i(x_i), \qquad \text{for } x_i \in [-2, 1], \qquad \text{with } \varphi_i(x_i) = \frac{|4x_i + 2| + a_i}{1 + a_i},$$

where $a_i \geq 0$ for $i = 1, \dots, n$ are fixed deterministic coefficients. If $a_i = 0$ this implies that φ_i varies between 0 and 6 and thus the variable x_i is important, if a_i is big, i.e., $a_i = 99$, then φ_i varies between 0.99 and 1.77 and thus the variable x_i is insignificant.

The contribution to the total variance of the functions g_{i_1} and $g_{i_1 i_2}$ for $i_1, i_2 = 1, \dots, n$ can be reformulated as

$$\begin{aligned} D_i &= \mathbb{V}\left(\int_{I^{n-1}} g(\boldsymbol{x})\, d\boldsymbol{x}_{-i}\right) = \frac{1}{3(1+a_i)^2}, \\ D_{ij} &= \mathbb{V}\left(\int_{I^{n-2}} g(\boldsymbol{x})\, d\boldsymbol{x}_{-\{ij\}}\right) - D_i - D_j = D_i D_j \end{aligned}$$

and the total variance can we written as a function of first-order terms only

$$D = -1 + \prod_{i=1}^{n} (1 + D_i).$$

Assuming that $n = 8$, and $a_1 = a_2 = a_6 = 0$, $a_3 = 1$ and $a_i = 3$ for $i = 4, 5, 7, 8$. Then the first, second and sixth variable are the most important ones, variable x_3 is not very important, whereas the other remaining variables are insignificant. This can be verified by the Sobol' indices: $S_1 = S_2 = S_6 = 0.333$, $S_3 = 0.083$ and $S_i = 0.021$ for the remaining $i = 4, 5, 7, 8$. The second-order indices are: $S_{ij} = 0.0621$ for $i, j \in 1, 2, 3$, $S_{ij} = 0.0155$ if one of the indices is 3 and the other one 1,2 or 6, $S_{ij} = 0.0039$ for one index being from the set $\{1, 2, 6\}$ and the other one from $\{4, 5, 7, 8\}$. All other indices of second-order are smaller than $9 \cdot 10^{-4}$. We visualise these sensitivity indices of first-order and second-order for the variables $x_1, \dots, x_8$ in Figure B.4. The size of the blue spots indicates the magnitude of the sensitivity index of first-order, whereas the connections between the spots mark the strength of their connectivity or rather the sensitivity indices of order 2.

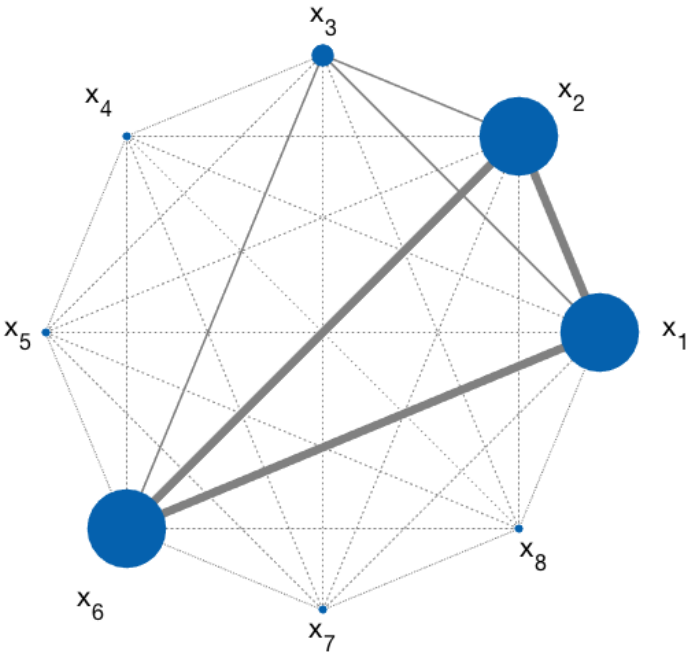


FIGURE B.4: The Sobol' indices for the above example.

B.4.6 Stochastic processes in space

So far we considered stochastic processes in time. However, the dynamics of stochastic processes in space are not analogous to the dynamics of the ones in the time domain. Due to the unilateral nature of the time series, the stochastic process is only influenced by past values, whereas the spacial process has a dependence in all directions.

In general random fields are used to model spatial data, that is, observed in environmental or biomedical sciences. These real life problems in 1, 2 or even 3 spatial dimensions can be modelled by multi-dimensional stochastic processes. Realistic models of random processes in space require autoregressive schemes with dependence in all directions, often referred to as multi-dimensional second-order stochastic processes (Su and Lucor, 2006).

In general, a continuous random process is formally defined as an indexed set of random variables, where the index belongs to a continuous uncountable set. Such process can be approximated by restricting the index to a subset of the indexing set.

In the most general form, the spectral representation of a random process, can be expressed as

$$Y(\boldsymbol{x},\theta) = \int g(\boldsymbol{x}) d\mu(\theta),$$

where $Y(\boldsymbol{x},\theta)$ is a stochastic process with a covariance function $C(x_1,x_2) = \mathrm{cov}(x_1,x_2)$ that admits the decomposition

$$C(x_1,x_2) = \int g(x_1)g(x_2)d\mu_1(\theta)d\mu_2(\theta),$$

where $g(\boldsymbol{x})$ is a deterministic function, and $d\mu(\theta)$ is an orthogonal set function, also called orthogonal stochastic measure (Ghanem and Spanos, 1991).

An alternative formulation of the spectral representation is the Karhunen-Loeve expansion based on a Fourier-type series expansion. Let $Y(\boldsymbol{x},\theta)$ be a random process, namely a function of the position vector $\boldsymbol{x}$ defined over the space domain D, with θ belonging to the space of random events Ω. Then the *Karhunen-Loeve expansion* reads

$$Y(\boldsymbol{x},\theta) = \bar{Y}(\boldsymbol{x}) + \sum_{n=0}^{\infty} \sqrt{\lambda_n}\xi_n(\theta) f_n(\boldsymbol{x}),$$

where $\bar{Y}(\boldsymbol{x})$ is the expected value of $Y(\boldsymbol{x},\theta)$, $\xi_n(\theta)$ is a set of random variables, and λ_n and $f_n(\boldsymbol{x})$ are the eigenvalue and eigenvector of the covariance kernel. That means they are the solution to the integral equation

$$\int_D C(x_1,x_2) f_n(x)\, dx_1 = \lambda_n f_n(x_2). \tag{B.32}$$

Truncating the series at the m-th term gives

$$Y(\boldsymbol{x},\theta) = \bar{Y}(\boldsymbol{x}) + \sum_{n=0}^{m} \sqrt{\lambda_n}\xi_n(\theta) f_n(\boldsymbol{x}).$$

The Karhunen-Loeve expansion described above has some interesting and useful properties that make it a good choice for some applications. Here, we briefly summarise them, but for a more detailed approach and proofs we refer to Ghanem and Spanos, 1991.

1. **Error minimising property**
   Choosing as an orthonormal set of functions the eigenfunctions of the covariance kernel is optimal to minimise the mean-square error resulting from the finite representation of the process.

2. **Uniqueness of the expansion**
   The random variables in the expansion are orthonormal if and only if the orthonormal functions $f_n(\boldsymbol{x})$ and the constants λ_n are the eigenfunctions and eigenvalues of the covariance kernel as given in (B.32), respectively.

The usefulness of the Karhunen-Loeve expansion depends heavily on the ability to solve the integral equation of the form

$$\int_D C(x_1, x_2) f(x_2)\, dx_2 = \lambda f(x_1), \tag{B.33}$$

where $C(x_1, x_2)$ is the covariance function. This also implies that $C(x_1, x_2)$ must be bounded, symmetric and positive-definite. These characteristics of the kernel ensure a variety of properties for the eigenvalues and eigenfunctions that are the solution to equation (B.33):

1. The set $f_i(\boldsymbol{x})$ of eigenfunctions is orthogonal and complete.
2. To each eigenvalue λ_i corresponds a finite number of linearly independent eigenfunctions.
3. There are at most a countable infinite set of eigenvalues.
4. The eigenvalues are all positive real numbers.
5. The kernel $C(x_1, x_2)$ allows the following uniform convergent expansion

$$C(x_1, x_2) = \sum_{k=1}^{\infty} \lambda_k f_k(x_1) f_k(x_2).$$

All these properties are independent of the stochastic nature of the process and allow a wide range of applications.

One important covariance kernel is the *exponential kernel* representing the first-order Markovian process. It is given by the equation

$$C(x_1, x_2) = e^{-|x_1 - x_2|/b}, \tag{B.34}$$

where b is a parameter often termed the *correlation length*, as it reflects the rate at which the correlation decays between two points of the process. Assuming that the process is defined on an interval $[-a, a]$, the eigenvalues and eigenfunctions are the solutions of the integral equation

$$\int_{-a}^{a} e^{-c|x_1 - x_2|} f(x_2)\, dx_2 = \lambda f(x_1),$$

with $c = 1/b$ can be explicitly computed. The eigenfunctions of this problem for even and odd indices are

$$f_{2n}(x) = \frac{\cos(\omega_{2n}x)}{\sqrt{a + \frac{\sin(2\omega_{2n}a)}{2\omega_{2n}}}}, \qquad f_{2n+1}(x) = \frac{\sin(\omega_{2n+1}x)}{\sqrt{a - \frac{\sin(2\omega_{2n+1}a)}{2\omega_{2n+1}}}}, \tag{B.35}$$

with ω being the solution of

$$\begin{cases} c - \omega_{2n}\tan(\omega_{2n}a) = 0 \\ \omega_{2n+1} + c\tan(\omega_{2n+1}a) = 0. \end{cases}$$

The corresponding eigenvalues are

$$\lambda_{2n} = \frac{2c}{\omega_{2n}^2 + c^2}, \qquad \lambda_{2n+1} = \frac{2c}{\omega_{2n+1}^2 + c^2}. \tag{B.36}$$

Consequently, a process with the covariance kernel defined as in (B.34), can be expanded as

$$Y(x,\theta) = \sum_{n=1}^{\infty} \left(\xi_n \sqrt{\lambda_n} f_n(x) + \xi_{2n+1}\sqrt{\lambda_{2n+1}} f_{2n+1}(x) \right).$$

Figure B.5 shows the first four eigenfunction as defined in equation (B.35), for $a = 0.5$ and a correlation length $b = 1$. Figure B.6 shows the eigenvalues as given in equation (B.36) for different correlation lengths b. The smaller the value of b, the more contribution can be expected from the terms associated to the smaller eigenvalues.

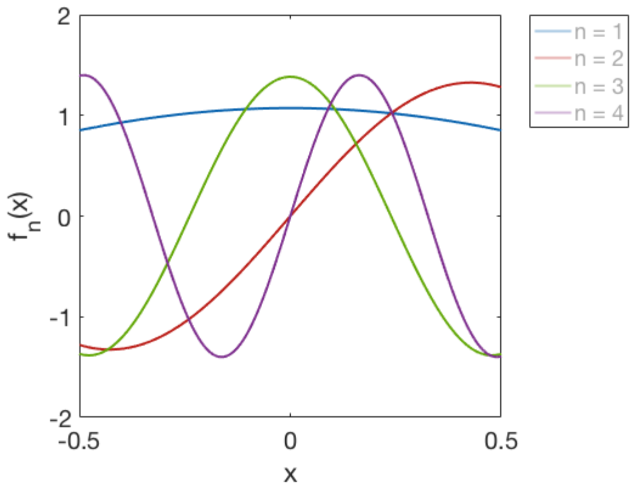


FIGURE B.5: The eigenfunctions $f_n(x)$ for $x \in [-0.5, 0.5]$, $n = 1,2,3,4$ and a correlation length of $b = 1$.

Other kernels, with an explicit solution of the integral equation are, i.e., the triangular kernel given by

$$C(x_1, x_2) = 1 - d|x_1 - x_2|, \quad |x_1 - x_2| \in [0, 1/d],$$

or the kernel of the Wiener process that is given by the equation

$$C(x_1, x_2) = min(x_1, x_2), \quad (x_1, x_2) \in [0, T] \times [0, T].$$

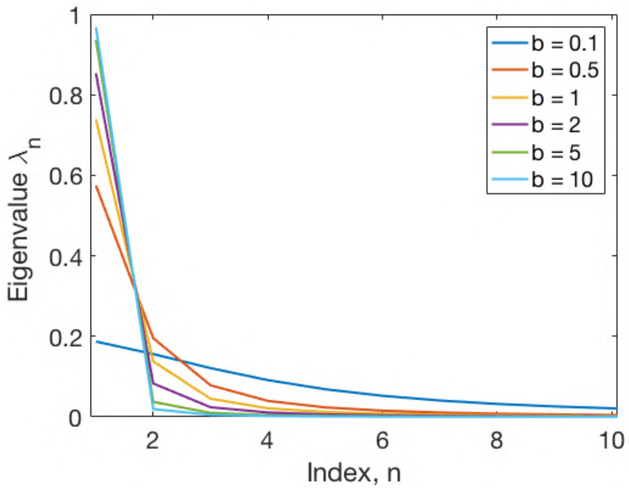


FIGURE B.6: The eigenvalues λ_n for various values of the correlation length b for $n = 1, 2, \ldots, 10$.

Consequently, the Karhunen-Loeve expansion requires the knowledge about the covariance function of the process, to compute the corresponding eigenfunctions and eigenvalues. Without the knowledge of the covariance function this expansion cannot be implemented. An alternative expansion that circumvents this problem is the homogenous chaos expansion. For more details on the homogenous chaos expansion we refer Ghanem and Spanos, 1991.

Appendix C

A hitchhiker's guide to MRI and DTI brain imaging data

"Running in circles, coming up tails; Heads on a science apart, [...]
No one ever said it would be this hard, oh take me back to the start."

Coldplay, in *The Scientist* (2002)

This chapter is a brief guide to MRI and DTI or DWI data processing to extract the geometry of the cerebral cortex and the corresponding diffusion parameters from brain imaging data. This appendix is meant as an introduction to the topic, providing the essential pieces of information and code, references and websites for in-depth studies of this topic and the available techniques and tools. We are commenting on the problems we encountered and their solution.

We mention the different ways that MRI and DTI or DWI data is available and how to deal with these file formats. For the extraction of the cortical geometry, we use the FreeSurfer[1] image analysis suit. FreeSurfer is an open-source software for processing and analysing (human) brain MRI images developed by the Laboratory for Computational Neuroimaging in Charlestown, MA, for further details, see Fischl, 2012. For the extraction and processing of the diffusion data, we use the FSL[2] (FMRIB Software Library) toolkit developed by the FMRIB Group in Oxford, UK. FSL is a library of analysis tools for functional MRI (fMRI), MRI and DTI brain imaging data and offers different tools to process DTI and DWI data (Jenkinson et al., 2012; Smith et al., 2004; Woolrich et al., 2009). Another commonly used tool for the automated processing of diffusion-weighted MRI images is the TRACULA[3] (tracts constrained by underlying anatomy) toolbox that is mainly used for the automatic reconstruction of major white matter pathways (Yendiki et al., 2011, 2013). However, this guide is focussing on the diffusion data processing with the FSL toolkit.

---

[1] `http://surfer.nmr.mgh.harvard.edu/`

[2] `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki`

[3] `https://surfer.nmr.mgh.harvard.edu/fswiki/Tracula`

C.1 Data

The MRI raw data can on one hand be provided in the form of NIfTI (neuroimaging informatics technology initiative) files, as single files (.nii) or dual files including a header and an image (.hdr and .img) file for the anatomical information, and gradient direction (.bvecs) and b-value (.bvals) files for the diffusion data. The NIfTI image format is a simple, compact and versatile standard format that was designed for scientific analysis of brain images. The .bvecs file is a text file containing the list of gradient directions applied during the acquisition of the diffusion-weighted image. The format is

$$\begin{array}{ccccc} x_1 & x_2 & x_3 & \dots & x_n \\ y_1 & y_2 & y_3 & \dots & y_n \\ z_1 & z_2 & z_3 & \dots & z_n. \end{array}$$

The list of b-values applied during each volume acquisition is assumed to be in sec/mm^2 and the order of entries in this file has to match the order of volumes in the input data and the entries in the gradient directions text file. The format is

$$b_1 \quad b_2 \quad b_3 \quad \dots \quad b_n.$$

On the other hand, the same information can be provided in the form of DICOM (digital imaging and communication in medicine) images compressing the information of MRI and DTI or DWI data.

C.2 Processing T1 images before using FreeSurfer

Before applying any structural imaging method (e.g., FreeSurfer, ANTs, etc.) it is important to follow a general pipeline[4] of preprocessing to prepare the data in a standard file format. This mainly includes the conversion to the NIfTI image format, which can then be used as the input data format of the FreeSurfer software.

In order to convert DICOM images to NIfTI format, we use the dicom2nii[5] software package integrated in MATLAB. To open the GUI window, simply run

```
dicm2nii;
```

and introduce the source and result folder for the NIfTI files. In the case of dual files of NIfTI images, we use the **mri_convert**[6] command from FreeSurfer:

```
mri_convert  T1.hdr T1.nii.gz
mri_convert  DTI.hdr DTI.nii.gz
```

Another important step in the pre-processing to obtain consistent results in the alignment of MRI and DTI data is to check the orientation of the images and correct it if necessary. Information on the orientation can be obtained by opening the images with FSLeyes[7], an

---

[4] `https://biabl.github.io/tutorials/part-1-topics/preprocessing-t1-weighted-images/`

[5] `https://es.mathworks.com/matlabcentral/fileexchange/42997-xiangruili-dicm2nii`

[6] `https://surfer.nmr.mgh.harvard.edu/fswiki/mri_convert`

[7] `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FSLeyes`

image viewer from the FSL toolbox, and by opening the image information window. An example of the orientation of T1 and DTI data is provided in Table C.1.

	T1	**DTI**
image storage order	radiological	radiological
X voxel orientation	anterior - posterior	right - left
Y voxel orientation	inferior - superior	posterior - anterior
Z voxel orientation	left - right	inferior - superior

TABLE C.1: An example of the orientation of MRI and DTI data.

If the two images are not aligned it is important to fix their orientation before processing the T1 images with the FreeSurfer suit. In the example of Table C.1 the two images are not aligned and the dimensions of the MRI images have to be rearranged in order to be in line with the ones of the DTI image. For the change of dimensions, we use the **fslswapdim**[8] command from the FSL toolbox, which reads for the above example:

```
fslswapdim T1.nii.gz -z -x y T1_reoriented.nii.gz
```

C.3 Cortical reconstruction with FreeSurfer

The input for the FreeSurfer software is a T1 contrast image of the whole brain, where the white matter is brighter than grey matter. Usually one acquisition with 1.5 Tesla or 3 Tesla of patients above the age of 5 and without major brain problems, i.e., tumours or missing parts is sufficient for the reconstruction.

To start the program and set the working directory and the directory of the FreeSurfer application itself, we use the following code in the terminal:

```
setenv MY_DATA "path to data"
setenv SUBJECTS_DIR $MY_DATA
setenv FREESURFER_HOME "/Applications/freesurfer"
source ${FREESURFER_HOME}/SetUpFreeSurfer.csh
```

The variables SUBJECTS_DIR and FREESURFER_HOME are environment variables that have to be defined in the beginning before running any commands in FreeSurfer. If they are not defined FreeSurfer sets them as default values and hence, it will not find the data.

For a given patient with identification name CON01 and MRI data CON01_T1.nii, the **recon-all** command creates a folder "CON01" in the SUBJECTS_DIR and uses the specified input path. The command -all runs the whole processing. Note that FreeSurfer automatically creates the folder with the patient ID and saves the results in this folder.

```
recon-all -s CON01 -i $SUBJECTS_DIR/T1data.nii -all
```

The individual steps of the acquisition are:

Volumetric processing stages (subjid/mri):

1. Motion cor, avg, conform (orig.mgz)
2. Non-uniform inorm (nu.mgz)
3. Talairach transform computation (talairach.xfm)

---

[8] `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Fslutils`

4. Intensity normalization 1 (T1.mgz)
5. Skull strip (brainmask.mgz)
6. EM register (linear volumetric registration)
7. CA intensity normalization (norm.mgz)
8. CA nonlinear volumetric registration
9. CA label (volumetric labeling) (aseg.mgz)
10. Intensity normalization 2 (T1.mgz)
11. White matter segmentation (wm.mgz)
12. Edit WM with aseg
13. Fill and cut (filled.mgz)

Surface processing stages (subjid/surf):

14. Tessellate (?h.orig.nofix)
15. Smooth1
16. Inflate1
17. Sphere (?h.qsqhere)
18. Automatic topology fixer (?h.orig)
19. Final surfs (?h.white, ?h.pial, ?.thickness)
20. Smooth 2 (?h.smoothwm)
21. Inflate 2 (?h.inflated)
22. Aseg statistics (stats/aseg.stats)
23. Cortical ribbon mask (?h.ribbon.mgz)
24. Spherical morph
25. Spherical registration (?h.sphere.reg)
26. Map average curvature to subject
27. Cortical parcellation (labeling)
28. Cortical parcellation statistics
29. Cortical parcellation mapped to aseg
30. White matter parcellation (wmparc.mgz)

where **?h.orig** denotes lh.orig or rh.orig relating to the left and right hemisphere. These results are saved in the CON01 folder. The overall structure of the folders is shown in Figure C.1.

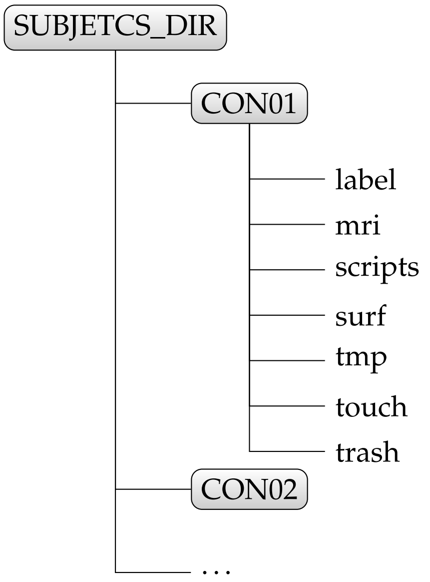


FIGURE C.1: The folder and subfolder structure created with the recon-all command of FreeSurfer.

C.3.1 Checking the results

Even if the recon-all command finishes without error, it is crucial to check for several problems that can occur without causing a failure of the procedure. We just give two examples: the Talairach registration and the pial surface check. When checking the outputs and the results always set up the FreeSurfer in the terminal so that the tools included in FreeSurfer work properly.

```
setenv FREESURFER_HOME "/Applications/freesurfer"
source ${FREESURFER_HOME}/SetUpFreeSurfer.csh
```

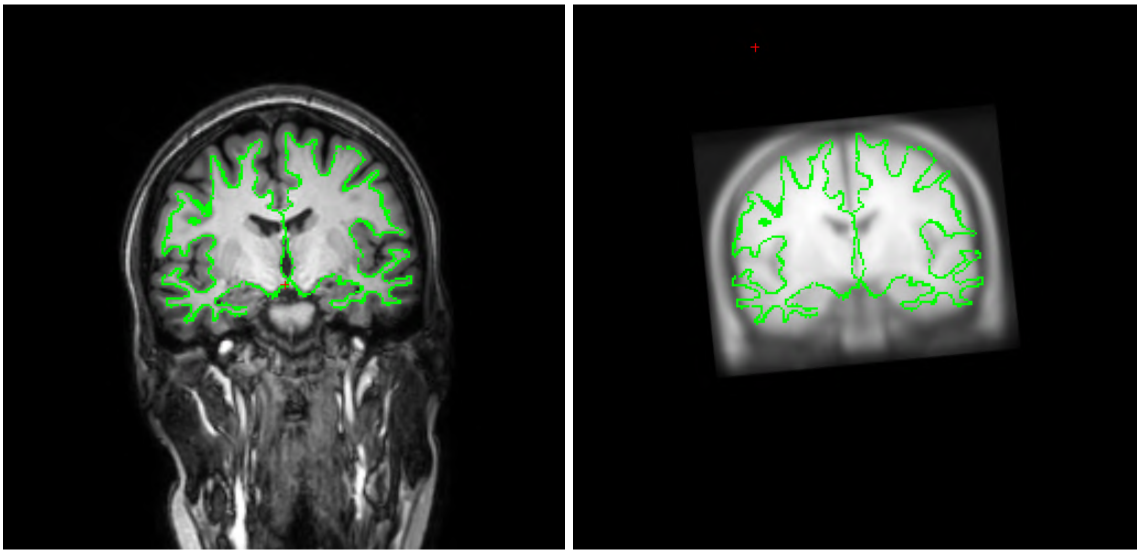
Output from the Talairach registration Sometimes the registration of the Talairach volume with the subject's volume does not properly work[9]. The best way to check the transform is by loading it in FreeView, a visualisation tool from FreeSurfer,

```
freeview -v $SUBJECTS_DIR/CON01/mri/T1.mgz
  -v $SUBJECTS_DIR/CON01/mri/brainmask.mgz:
  reg=$SUBJECTS_DIR/CON01/mri/transforms/talairach.xfm
```

and visually checking the registration of the subject's volume and the Talairach volume by

```
tkregister2 --mgz --s CON01 --fstal --surf orig
```

The goal is to have the two volumes aligned at least along the key anatomical points (anterior/posterior commissures, the temporal lobes in the coronal plane, and the midline cut). If the visual inspection shows a misalignment, the movable Talairach volume can be stretched, translated and rotated until the two volumes are aligned. In Figure C.2 we give an example of a Talairach registration.



(a) subject's volume (b) Talairach volume

FIGURE C.2: The subject's volume (a) and the Talairach volume (b), the green lines are the original surface from the subject.

---

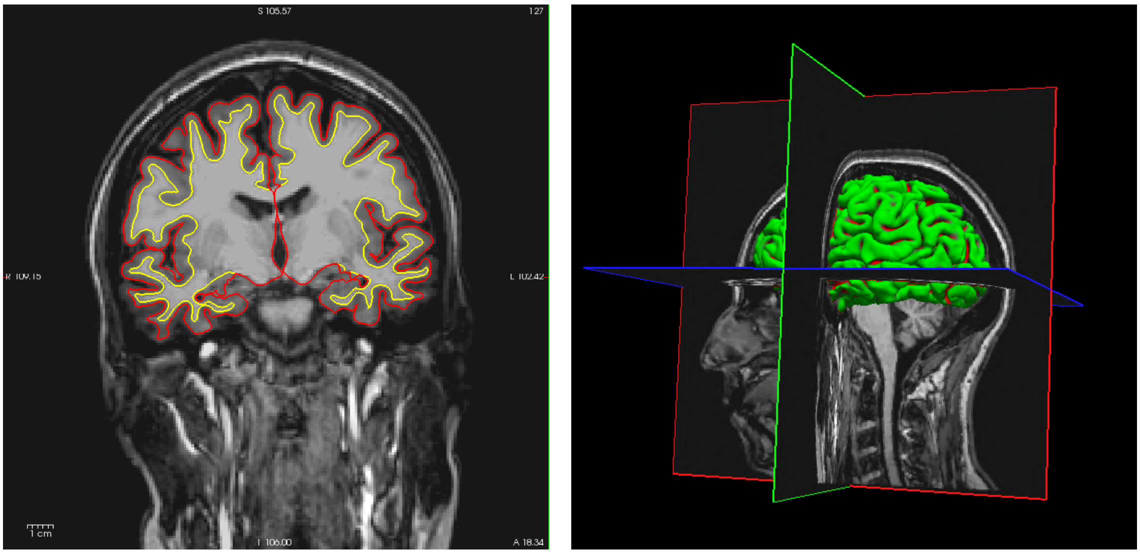
[9] http://freesurfer.net/fswiki/FsTutorial/Talairach_freeview

Correcting the pial surface After the pial surface, the boundary between grey matter and cerebrospinal fluid, has been generated, it is a good idea to visually check it for defects that may have been created during automatic topology fixing[10]. To check the pial surface, it may be loaded into FreeView and viewed along with the **brainmask.mgz** volume. If the surface appears not to follow the boundary between the grey matter and the cerebrospinal fluid in the volume, edits may be required.

Therefore, we load the pial surface with FreeSurfer:

```
freeview -v $SUBJECTS_DIR/CON01/mri/T1.mgz \
$SUBJECTS_DIR/CON01/mri/brainmask.mgz \
-f $SUBJECTS_DIR/CON01/surf/lh.white:edgecolor=yellow \
$SUBJECTS_DIR/CON01/surf/lh.pial:edgecolor=red \
$SUBJECTS_DIR/CON01/surf/rh.white:edgecolor=yellow \
$SUBJECTS_DIR/CON01/surf/rh.pial:edgecolor=red
```

In Figure C.3 we give the visual output of an example. The pial surface should not include non-cortex material within its boundaries.



(a) pial and white matter surface

(b) pial surface in 3D view

FIGURE C.3: The pial surface (red), the boundary of the white matter (yellow) and the pial surface visualised as a 3D surface (b).

C.3.2 Conversion to Matlab files

To make the reconstructed cortical surface accessible for our purpose, we process the pial surface and the annotation files defining the regions of interest (ROI) with Matlab functions provided by the FreeSurfer software. This code reads

---

[10] `https://surfer.nmr.mgh.harvard.edu/fswiki/FsTutorial/PialEdits_freeview`

```
% move to directory of matlab files in Freesurfer
cd '/Applications/freesurfer/matlab'

% load data for points, vertices and curvature
[data,vert]=read_surf('$SUBJECTS_DIR/CON01/surf/lh.pial');
C = data';

vert = vert+1;
L = vert;

[vertices,label,colortable]=read_annotation...
   ('$SUBJECTS_DIR/CON01/label/lh.aparc.annot');

for i = 1:36
    a = find(label == colortable.table(i,5));
    label(a) = i;
end
label_names = colortable.struct_names;

save CON01_left_hemisphere.mat C L label label_names
```

The variable **colourtable** is a structure including the number of entries, the path to the original Desikan-Killiany atlas, the structure names (label names) and a table including colour values and the ID for the different regions. For example, the label '6553700' can be found in the **colortable.table** structure and its label name is the one with the same index in the **colortable.struct_name** structure, in this case the 'frontalpole'. However, after applying the above code the regions are associated with a number between 1 and 36 (the regions of the Desikan-Killiany atlas including the corpus callosum plus a label for unknown regions - usually empty) and their name corresponds to the respective entry in **label_names**.

C.4 Processing the DTI or DWI data with FSL

In the following, we differentiate between DTI data that provides information about the diffusion strength and diffusion direction, and apparent diffusion coefficient (ADC) data that provides an average diffusion coefficient but no diffusion directions.

In general, the processing of diffusion-weighted data needs a little bit more care than the processing of T1 images and involves results from the FreeSurfer recon-all function applied to the T1 image. The processing includes:

1. Correction of eddy current distortion and simple head motion
2. Brain extraction of the DTI or DWI and T1 data, respectively.
3. Application of the function **DTIFIT** in the case of DTI data.
4. Registration of the DTI or DWI data with T1 space.
5. Application of the inverse transformation to the DWI data and check for alignment.
6. Transformation from T1 space into FreeSurfer space.

7. Transformation from diffusion space to FreeSurfer space.
8. Projection of the diffusion data to the mesh.
9. Loading and saving the data in MATLAB format.

In other words, we correct the data, extract the brain, register the diffusion and T1 space, and register the T1 and FreeSurfer space and apply the composition of transformations to the diffusion data. Then the resulting data is in the FreeSurfer space, where we project it to the previously computed surface mesh. Except for step 3 the DTI data and ADC data is handled in the same manner. For an overview of all those transformations between the different spaces, see Figure C.4 and Figure C.8.

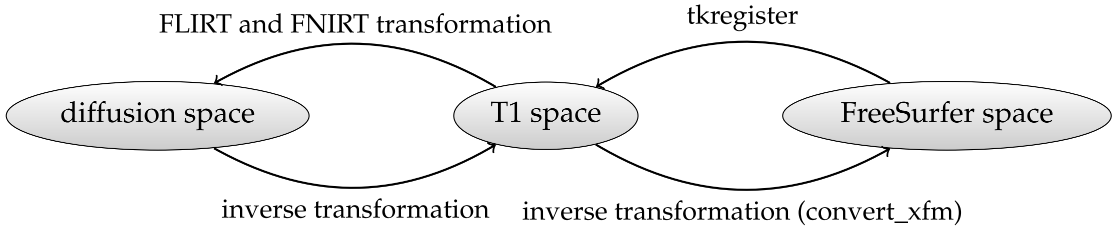


FIGURE C.4: An overview of the transformations between the different spaces.

For the introduction and the set-up we start the bash/tcsh file as follows:

```
#!/bin/csh

# process the DTI data in the folder
set DATA_NAME=CON01

setenv MY_DATA "pathotodata/$DATA_NAME/t1"
setenv SUBJECTS_DIR $MY_DATA

setenv FS_DATA_DIR "/pathtodata/$DATA_NAME/Freesurfer/$DATA_NAME"
setenv FREESURFER_HOME "/Applications/freesurfer"
source ${FREESURFER_HOME}/SetUpFreeSurfer.csh

set diffusion_file=$SUBJECTS_DIR/diffusion.nii.gz
```

C.4.1 Eddy current correction

The eddy current-induced distortions and subject movements are routinely fixed with **eddy_correct**, an FSL toolbox function (Jenkinson et al., 2012).

```
eddy --imain=$dti_file --out=${SUBJECTS_DIR}/eddydiffusion.nii.gz
```

However, this old **eddy_correct** tool has now been superseded by the new function **eddy**[11] that offers substantially better distortion correction (Andersson and Sotiropoulos, 2016). The new tool allows to work with higher b-value data than previously.

---

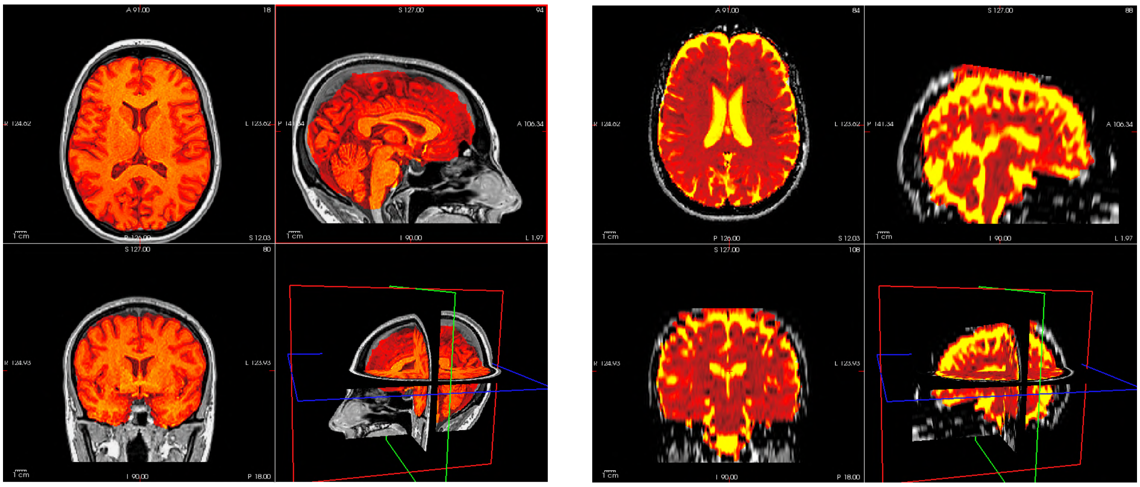
[11] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/eddy

C.4.2 Brain extraction tool

FSL's brain extraction tool[12] (BET) is used to extract the brain tissue from the rest of the image (Smith, 2002). For registration purposes, the brain extraction may contain some small errors without affecting the final result. Varying the fractional intensity threshold parameter controls the threshold that distinguishes brain from non-brain. The fractional intensity threshold varies between 0 and 1 with a default value of 0.5, smaller values giving larger brain outline estimates. The brain extraction has to be executed for the DTI or DWI data as well as for the T1 structural image data.

```
bet ${SUBJECTS_DIR}/eddydiffusion.nii.gz
  ${SUBJECTS_DIR}/diffusion_data_brain.nii.gz -m -f 0.3
bet ${SUBJECTS_DIR}/T1.nii.gz
  ${SUBJECTS_DIR}/T1_brain.nii.gz -m -f 0.3
```

Here, the option -m generates a binary brain mask to be used later on for the **DTIFIT** (of step 3), while the option -f sets the fractional intensity threshold. An example of a brain extraction from T1 and DWI data is shown in Figure C.5.



(a) T1 image (b) DWI image

FIGURE C.5: The original image of an example dataset (grey scale) and the result of the BET (heat colormap) for a T1 image (a) and a DWI image (b).

C.4.3 Fitting the diffusion tensor model

The FSL function **DTIFIT**[13] assigns a diffusion tensor model to each voxel by fitting the tensors to the previously processed diffusion-weighted images. Typically **DTIFIT** is run on data that has been pre-processed and eddy current corrected. For **DTIFIT** the .bvecs and .bvals files have to be specified.

---

[12] `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide`

[13] `https://users.fmrib.ox.ac.uk/~behrens/fdt_docs/`

```
set bvecs=$SUBJECTS_DIR/*bvecs
set bvals=$SUBJECTS_DIR/*bvals
dtifit --data={$DTI_COMP_DIR}/{$DATA_NAME}_eddy.nii.gz
--out={$DTI_COMP_DIR}/{$DATA_NAME}_dti
--mask={$DTI_COMP_DIR}/{$DATA_NAME}_dtidata_brain.nii.gz
--bvecs=$bvecs --bvals=$bvals
```

The main output files of this procedure are:

- dtifit_FA.nii.gz - fractional anisotropy image
- dtifit_MD.nii.gz - mean diffusivity image
- dtifit_L1...L3 - primary, secondary, tertiary eigenvalues
- dtifit_V1...V3 - primary, secondary, tertiary eigenvectors

Some of the DTI image files (e.g., V1, the vector image showing the principal diffusion tensor direction) are multi-volume (4D) image files. To view such images in DTI Mode of the FslViewer, select the image in the Image List and press the Image Info button. This will open the Image Info dialog. In the DTI Display options section, it is possible, for example, to change the value from None to Lines or RGB. When Lines is set the DTI vectors are expressed by the direction of many small lines. Each layer set to display as Lines is rendered in a different colour. When RGB is set the DTI vectors are expressed by colours where red, green and blue represent the x, y and z diffusion directions, respectively. Colours such as purple are mixtures of red, green and blue and therefore express vectors that have significant non-zero components in more than one of the x, y and z directions. An example of visualising DTI images including the fractional anisotropy (FA), V1, V1 intensity modulated by FA and the V1 direction visualised as lines is given in Figure C.6.

C.4.4 Registration of diffusion space with T1 space

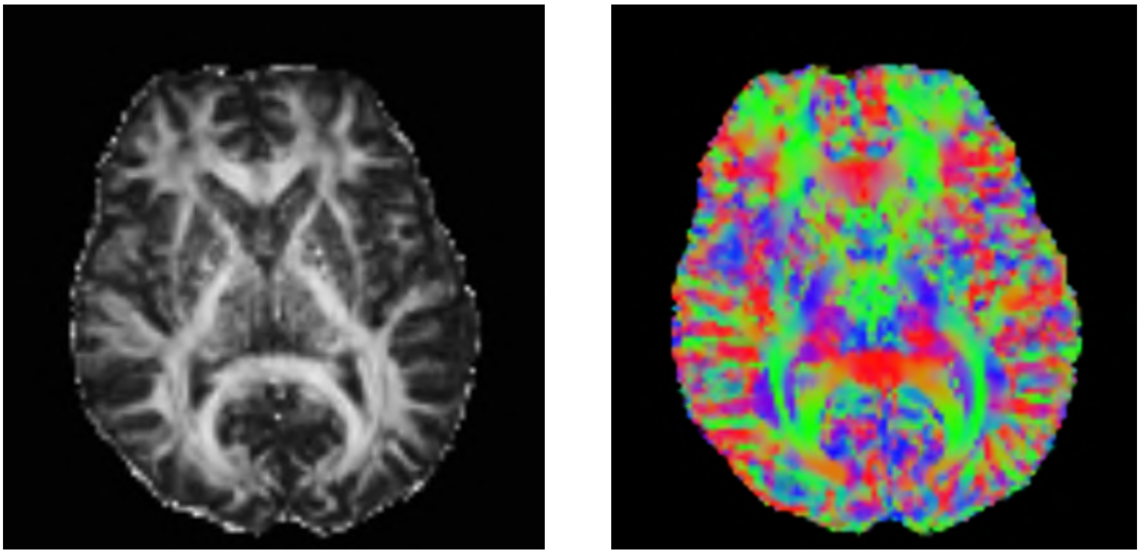
To register the diffusion and T1 image, we perform a linear or non-linear transformation. FLIRT[14] (FMRIB's Linear Image Registration Tool) is a fully automated robust and accurate tool for linear (affine) intra- and inter-modal brain image registration (Jenkinson and Smith, 2001; Jenkinson et al., 2002). To run the FSL non-linear registration tool FNIRT[15], it is required to perform an initial affine registration first using FLIRT (Greve and Fischl, 2009; Jenkinson and Smith, 2001; Jenkinson et al., 2002).

One of the simplest applications of FLIRT is to register two single volumes together. The main options are the input (-in) and the reference (-ref) volume; the output includes the calculated affine transformation that registers the input to the reference and it is saved as a 4×4 affine matrix (-omat), and the final output volume (-out). Important features for the registration are:

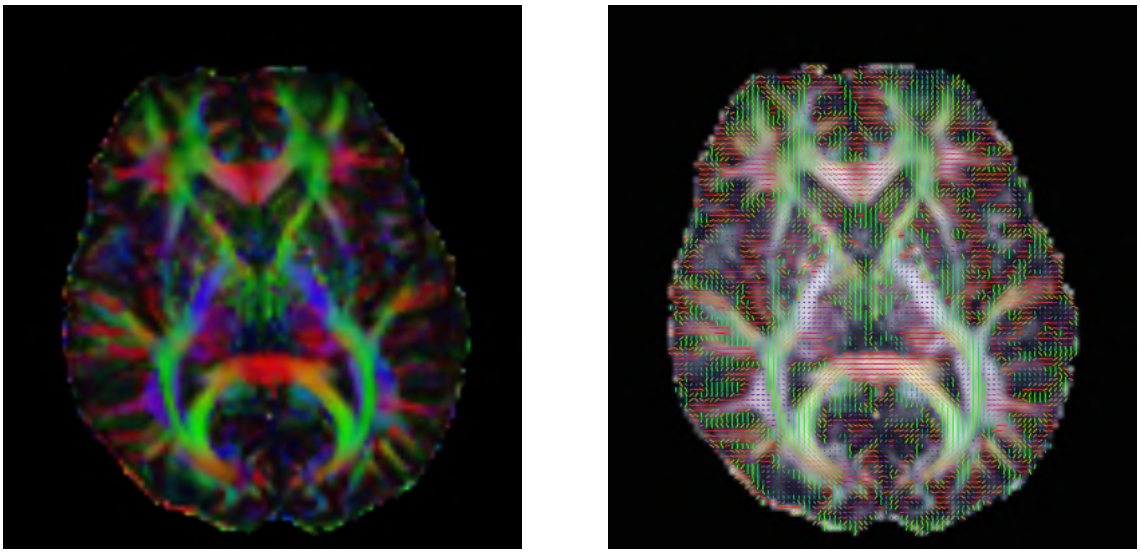
1. *Degrees of freedom*: For 3D to 3D mode the degrees of freedom (DOF) can be set to 12 (affine), 9 (traditional), 7 (global rescale) or 6 (rigid body). In 2D to 2D mode only 3 DOF can be set.

---

[14] `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FLIRT/UserGuide`
[15] `https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FNIRT/UserGuide`



(a) fractional anisotropy (b) principal diffusion tensor direction V1



(c) V1 intensity modulated by FA (d) V1 direction visualised as lines

FIGURE C.6: Visualisation of DTI images: fractional anisotropy (FA) (a), the principal diffusion tensor direction V1 displayed in RGB colours (b), V1 intensity modulated by FA (c) and V1 direction visualised as lines (d).

2. *Cost functions*: The options include the within-modality functions least squares and normalised correlation, as well as the between-modality functions correlation ratio (the default), mutual information and normalised mutual information.

Below we list the minimal example of the usage of the FLIRT[16] toolbox, including a choice of options and parameter choices.

---

[16] `https://wiki.gacrc.uga.edu/wiki/Fsl`

```
Syntax:  flirt [options] -in <inputvol> -ref <refvol>
Options are:
 -init <matrix-filname>              (input 4x4 affine matrix)
 -omat <matrix-filename>             (output 4x4 ascii format)
 -out, -o <outputvol>                (default is none)
 -cost (mutualinfo,corratio,         (default is corratio)
  normcorr,normmi,leastsq,labeldiff)
 -dof <number of transform dofs>     (default is 12)
```

The default and basic options of the non-linear transformation with FNIRT are not very likely to give good results. Thus, FNIRT has a large set of parameters but we will just briefly mention a few of them. Adding, for example, information about the affine transformation and the reference brain mask can improve the transformation.

```
Syntax:  fnirt -in=<inputvol> -ref=<refvol>
Options are:
 -aff      file containing affine transform
 -inwarp   file containing initial non-linear warps
 -cout     output file with field coefficients
 -refmask  binary mask that determines interesting field of view
```

The function **invwarp** is used to reverse a nonlinear mapping. Even though there are some optional arguments, we list only the basic command here.

```
Syntax: invwarp -w warpvol -o invwarpvol -r refvol
```

The overall registration process results in the following code for our example:

```
flirt -in ${SUBJECTS_DIR}/T1_brain.nii.gz
-ref ${SUBJECTS_DIR}/diffusion_data_brain.nii.gz
-cost mutualinfo -omat ${SUBJECTS_DIR}/struct2ADC.mat

fnirt --ref={$SUBJECTS_DIR}/diffusion_data_brain.nii.gz
--in={$SUBJECTS_DIR}/T1_brain.nii.gz
--aff={$SUBJECTS_DIR}/struct2diffusion.mat
--cout=${SUBJECTS_DIR}/struct2diffusion_warp

invwarp --warp=${SUBJECTS_DIR}/struct2diffusion_warp
--ref=${SUBJECTS_DIR}/T1_brain.nii.gz
--out=${SUBJECTS_DIR}/struct2diffusion_warp_inv
```

C.4.5 Transformation of DTI or DWI data to T1 space

The function **applywarp** is used to apply the transformations estimated by FNIRT (or some other software) to the diffusion-weighted image to align it with the structural scan.

```
Syntax:  applywarp -i invol -o outvol -r refvol -w warpvol
Optional arguments:
 -w,-warp  filename of warp/coefficient (volume)
 -interp   interpolation method (nn,trilinear,sinc,spline)
 -postmat  filename for post-transform (affine matrix)
```

We apply the previously computed transformation to the diffusion-weighted image with

```
applywarp --in=$SUBJECTS_DIR/diffusion_data_brain.nii.gz
--out=$SUBJECTS_DIR/T1_diffusion.nii.gz
--ref=$SUBJECTS_DIR/T1_brain.nii.gz
--warp=$SUBJECTS_DIR/struct2diffusion_warp_inv
--interp=trilinear
```

This step is somehow superfluous for the following computations, but is essential to check the registration. Applying the transformation to the DWI data, it is now aligned with the T1 image in the T1 space. In Figure C.7 we show the original T1 image and the registered DWI data of an example dataset.

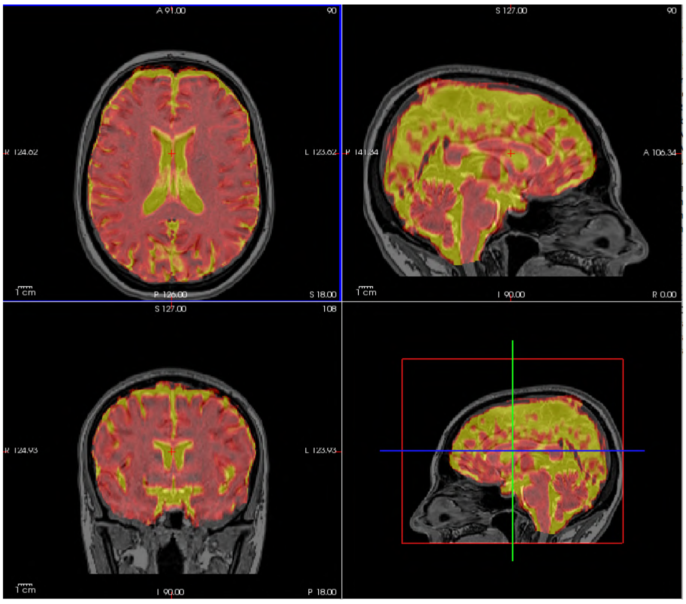


FIGURE C.7: The original image of an example dataset (grey scale) and the registered diffusion data (heat colormap).

C.4.6 Transformation from T1 native space to FreeSurfer structure

The function **tkregister2** is a tool to assist in the manual tuning of the linear registration between two volumes, mainly for the purpose of interacting with the FreeSurfer anatomical stream.

The volumes used for the registration are results from running the FreeSurfer recon - all command, in the specific mri/rawavg.mgz and mri/orig.mgz. The rawavg.mgz file is an unconformed volume file and it is generated after averaging the volumes in the FreeSurfer result folder mri/orig/ (if there are more than 1 run). If there is just one run, the rawavg.mgz file will be the same as 001.mgz. The orig.mgz is a conformed output of rawavg.mgz as a result of running **mri_convert**.

```
Syntax:  tkregister2 -mov vol -regheader -noedit -reg register.dat
Options are:
 -mov                 movable volume
 -targ                target volume
 -reg register.dat    input/output registration file
 -regheader           compute registration from headers
 -fsl file            FSL-style registration input matrix
 -fslregout           FSL-Style registration output matrix
 -noedit              do not open edit window
```

The function **convert_xfm** is used to convert between different transformation file formats. It can read and write ascii 4×4 matrices. In addition, it can be used to concatenate two transformations (using -concat with the second transform) or to find the inverse transformation (using -inverse).

```
Syntax: convert_xfm -omat refvol2invol.mat -inverse invol2refvol.mat
```

All in all, the code for the transformation from T1 space to FreeSurfer structure reads:

```
tkregister2 --mov ${FREESURFER_DATA_DIR}/mri/orig.mgz
--targ ${FREESURFER_DATA_DIR}/mri/rawavg.mgz
--regheader --reg $SUBJECTS_DIR/junk
--fslregout $SUBJECTS_DIR/freesurfer2struct.mat --noedit
convert_xfm  -omat $SUBJECTS_DIR/struct2freesurfer.mat
-inverse $SUBJECTS_DIR/freesurfer2struct.mat
```

C.4.7 Transformation from diffusion space to FreeSurfer space

The function **mri_convert** is a general purpose utility for converting between different file formats.

```
mri_convert [options] <inputvol> <outputvol>
```

The input volume here is a result from the FreeSurfer processing of the T1 image, in the specific mri/brain.mgz. We transform the diffusion by applying **applywarp** to the FreeSurfer space and the overall code reads:

```
mri_convert  ${FREESURFER_DATA_DIR}/mri/brain.mgz
$SUBJECTS_DIR/data_brain.nii.gz
applywarp -i $SUBJECTS_DIR/diffusion_data_brain.nii.gz
-o $SUBJECTS_DIR/diffusion_fr.nii.gz
-r $SUBJECTS_DIR/data_brain.nii.gz --interp=trilinear
-w ${SUBJECTS_DIR}/struct2diffusion_warp_inv
--postmat=$SUBJECTS_DIR/struct2freesurfer.mat
```

C.4.8 Projecting diffusion data to the mesh

The FreeSurfer function **mri_vol2surf**[17] assigns values from a volume to each vertex of a discretised surface.

```
Syntax:  mri_vol2surf [options] -src inputfile -out outputfile
-srcreg registrationfile -hemi hemisphere
Options are:
 -mov          input volume path (or -src)
 -ref          reference volume name (default=orig.mgz)
 -reg          source registration
 -hemi         hemisphere (lh or rh)
 -projfrac     fraction (0,1) of the cortical thickness at each
               vertex to project along the surface normal.
               Default=0.  When set at 0.5 with the white
               surface, this should sample in the middle of the
               cortical surface.
```

To project the diffusion data that has already been transformed to the FreeSurfer space to the surface, we use the following code:

```
printf "$DATA_NAME\n1\n1\n1\n1 0 0 0\n0 1 0 0\n0 0 1 0\n0 0 0 1\n"
> $SUBJECTS_DIR/reg.dat
setenv SUBJECTS_DIR "pathtodata/$DATA_NAME/Freesurfer"
setenv WORK_DIR "pathtodata/$DATA_NAME/t1"
mri_vol2surf --src  ${WORK_DIR}/diffusion_fr.nii.gz
--reg ${WORK_DIR}/reg.dat --projfrac 0.5 --hemi lh
--out ${WORK_DIR}/diffusion_Left.mgh
mri_vol2surf --src  ${WORK_DIR}/diffusion_fr.nii.gz
--reg ${WORK_DIR}/reg.dat --projfrac 0.5 --hemi rh
--out ${WORK_DIR}/diffusion_Right.mgh
```

C.4.9 Load and save the DTI or DWI data in Matlab format

To save the diffusion data on the surface in an adequate format we process the data in Matlab using the .mgh files and Matlab functions provided by the FreeSurfer package. This program reads the volume files and combines them with the data from the FreeSurfer processing.

---

[17] `https://surfer.nmr.mgh.harvard.edu/fswiki/mri_vol2surf`

```
function [] = createDWIparams(DWIpath,FSresultpath,dataname)
% DWIpath: the path to the end results of the DWI processing,
% namely adc_left.mgh and adc_Right.mgh
% FSresultpath: the path to the end results of the Freesurfer
% recon_all, namely
% *_left_hemsiphere.mat and *_right_hemsiphere.mat

addpath(genpath('/Applications/freesurfer/matlab'))

cd(DWIpath) % change to DWI path and load the .mgh file
f = MRIread('diffusion_Left.mgh');
Diffusion = f.vol;

cd(FSresultpath)
s = strcat(dataname,'_left_hemisphere_sorted.mat');
load(s)

name = strcat(dataname,'_DWI_left_hemisphere.mat');
save(name,'L', 'C', 'label_names_sorted', 'label_sorted','Diffusion')
end
```

In the case of DTI data, we apply the **MRIread** function to l1_left.mgh, l2_left.mgh, l3_left.mgh, v1_left.mgh, v2_left.mgh, v3_left.mgh, FA_left.mgh and MD_left.mgh. An overview of all the steps and files involved is given in Figure C.8.

The exact code for DTI and DWI processing with FSL is given in the next two sections. The procedures are very similar for the two different diffusion data types. The only difference is that DTI data requires the application of **DTIFIT** and the following steps have to be applied to the results from **DTIFIT** instead of just to the ADC value, like in the case of DWI data.

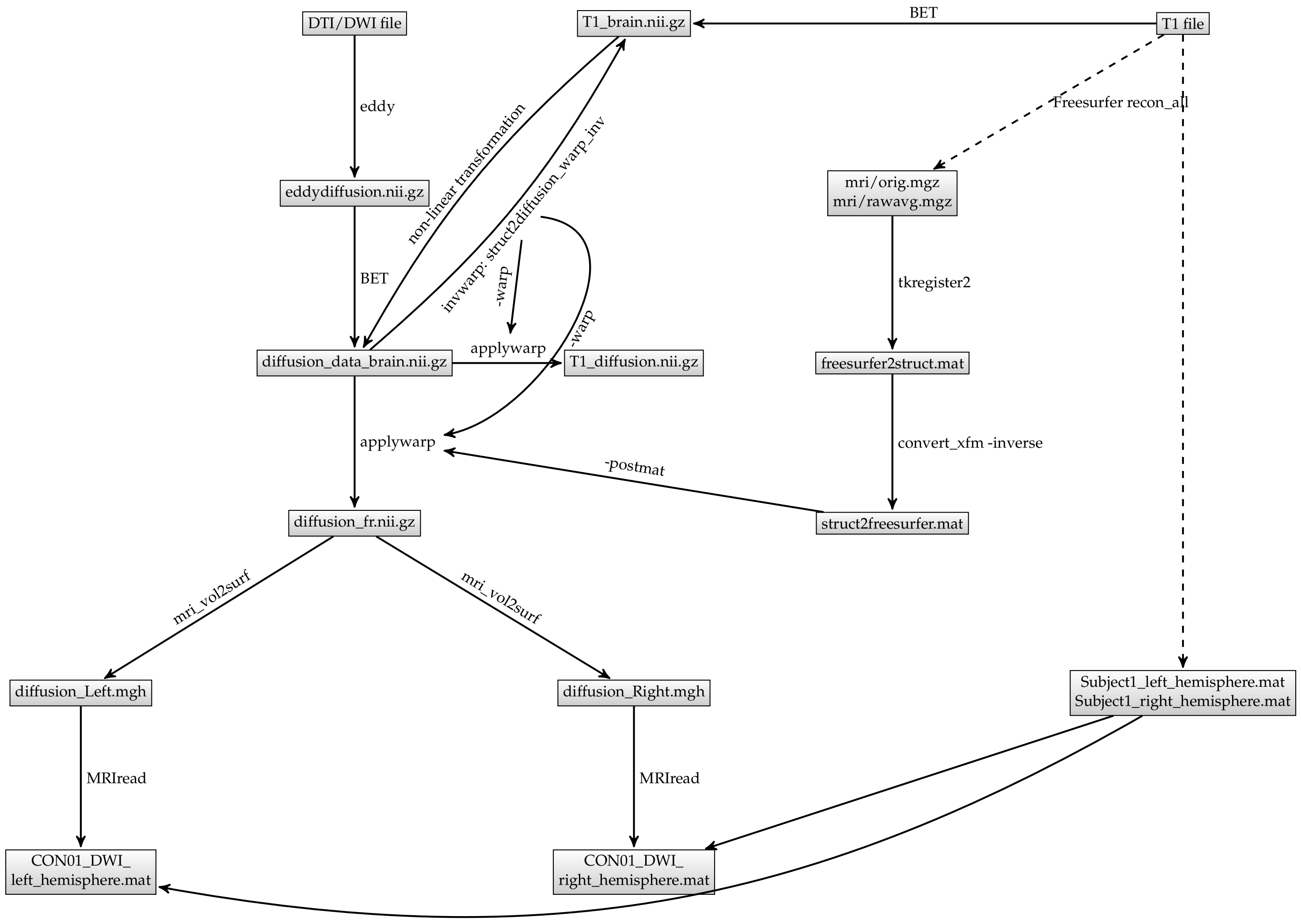


FIGURE C.8: An overview of the whole DWI conversion procedure.

C.4.10 Code for DWI processing

In the following, we provide the whole code to process diffusion-weighted data in the case that only ADC values are provided.

```
# process the DTI data in the folder
set DATA_NAME=Subject1

setenv MY_DATA "pathtodata/$DATA_NAME/t1"
setenv SUBJECTS_DIR $MY_DATA

setenv FREESURFER_DATA_DIR
  "pathtodata/$DATA_NAME/Freesurfer"
setenv FREESURFER_HOME "/Applications/freesurfer"
source ${FREESURFER_HOME}/SetUpFreeSurfer.csh

set diffusion_file=$SUBJECTS_DIR/ep2d_diffusione_ADC.nii

#1. Correct for eddy current distortions and head motion
eddy_correct $diffusion_file ${SUBJECTS_DIR}/eddyADC.nii.gz 0

#2. B0: If necessary, you can specify the fractional intensity
# threshold using the "-f" option. Different values of this
# parameter control the size of the mask. (Smaller values result
# in larger brain outline estimates.) The default value of 0.5
# probably works for your data.
bet ${SUBJECTS_DIR}/eddyADC.nii.gz
  ${SUBJECTS_DIR}/adc_data_brain.nii.gz -m -f 0.3

#3. T1 brain extraction
bet ${SUBJECTS_DIR}/t1_vibe_tra_p2_burdi2.nii
  ${SUBJECTS_DIR}/T1_brain.nii.gz -m -f 0.3

#4. Registration of DTI with T1
mkdir ${SUBJECTS_DIR}/transformations
flirt -in ${SUBJECTS_DIR}/adc_data_brain.nii.gz -ref
${SUBJECTS_DIR}/T1_brain.nii.gz -cost mutualinfo -dof 6
-omat ${SUBJECTS_DIR}/transformations/ADC2struct.mat
-out ${SUBJECTS_DIR}/transformations/adc_in_t1_lin

#5. transformation from T1 native space to FreeSurfer struct
tkregister2 --mov ${FREESURFER_DATA_DIR}/mri/orig.mgz
--targ ${FREESURFER_DATA_DIR}/mri/rawavg.mgz --regheader
--reg $SUBJECTS_DIR/transformations/junk
--fslregout $SUBJECTS_DIR/transformations/freesurfer2struct.mat
--noedit
convert_xfm
-omat $SUBJECTS_DIR/transformations/struct2freesurfer.mat
-inverse $SUBJECTS_DIR/transformations/freesurfer2struct.mat
```

```
#6. transformation from diffusion space to FreeSurfer space
mri_convert  ${FREESURFER_DATA_DIR}/mri/brain.mgz
$SUBJECTS_DIR/data_brain.nii.gz
convert_xfm -omat $SUBJECTS_DIR/transformations/ADC2freesurfer.mat
-concat $SUBJECTS_DIR/transformations/struct2freesurfer.mat
$SUBJECTS_DIR/transformations/ADC2struct.mat
flirt -ref $SUBJECTS_DIR/data_brain.nii.gz
-in ${SUBJECTS_DIR}/adc_data_brain.nii.gz
-init $SUBJECTS_DIR/transformations/ADC2freesurfer.mat
-applyxfm -out $SUBJECTS_DIR/adc_fr.nii.gz

#7. Project diffusion data to mesh
printf "$DATA_NAME\n1\n1\n1\n1 0 0 0\n0 1 0 0\n0 0 1 0\n0 0 0 1\n"
> $SUBJECTS_DIR/reg.dat
setenv SUBJECTS_DIR "pathtodata/$DATA_NAME/Freesurfer"
setenv WORK_DIR "pathtodata/$DATA_NAME/t1"
mri_vol2surf --src  ${WORK_DIR}/adc_fr.nii.gz
--reg ${WORK_DIR}/reg.dat --projfrac 0.5 --hemi lh
--out ${WORK_DIR}/adc_Left.mgh
mri_vol2surf --src  ${WORK_DIR}/adc_fr.nii.gz
--reg ${WORK_DIR}/reg.dat --projfrac 0.5 --hemi rh
--out ${WORK_DIR}/adc_Right.mgh

# copy all dwi files to a folder
setenv DWI_DATA_DIR "pahtodata/$DATA_SET_NAME"
mkdir $DWI_DATA_DIR/dwi_data
mv ${WORK_DIR}/eddyADC.* $DWI_DATA_DIR/dwi_data
mv ${WORK_DIR}/adc_data_brain* $DWI_DATA_DIR/dwi_data
mv ${WORK_DIR}/T1_brain* $DWI_DATA_DIR/dwi_data
mv ${WORK_DIR}/transformations $DWI_DATA_DIR/dwi_data
mv ${WORK_DIR}/T1_ADC.nii.gz $DWI_DATA_DIR/dwi_data
mv ${WORK_DIR}/adc_fr.nii.gz $DWI_DATA_DIR/dwi_data
mv ${WORK_DIR}/*.dat $DWI_DATA_DIR/dwi_data
mv ${WORK_DIR}/adc_*.mgh $DWI_DATA_DIR/dwi_data
mv ${WORK_DIR}/*ADC_*.vtk $DWI_DATA_DIR/dwi_data

#8. Load and save the DTI data in MATLAB format
setenv MATLAB_DIR "pathtodata/Matlab files"
matlab -nodisplay -nodesktop -nosplash -r "cd(’$MATLAB_DIR’);
createDWIparams(’$DWI_DATA_DIR/dwi_data’,
’$SUBJECTS_DIR’,’$DATA_NAME’),exit"

mv ${SUBJECTS_DIR}/*ADC_*.vtk $DWI_DATA_DIR/dwi_data
```

C.4.11 Code for DTI processing

In the following, we provide the code to process DTI diffusion data. This code is written in a more general form designed to be run for groups of data sets and saves a log file of the procedures and outputs. For DTI data we need to run **DTIFIT** in addition to the steps for the DWI data in Appendix C.4.10 and all transformations have to be applied to the resulting diffusion files.

```
set arr=(CON01 CON02 CON03 CON04)

foreach DATA_NAME ($arr)
    set logfile="pathtodata/$DATA_NAME/{$DATA_NAME}_dtilog.txt"
    date +"%T" | tee -a $logfile
    mkdir "pathtodata/$DATA_NAME"
    setenv MY_DATA "pathtodata/$DATA_NAME"
    setenv SUBJECTS_DIR $MY_DATA
    setenv FREESURFER_DATA_DIR "pathtodata/$DATA_NAME/Freesurfer"
    setenv DTI_COMP_DIR "/pathtoDTIdata/{$DATA_NAME}"
    setenv FREESURFER_HOME "/Applications/freesurfer"
    source ${FREESURFER_HOME}/SetUpFreeSurfer.csh

    set dti_file=$SUBJECTS_DIR/{$DATA_NAME}_dtidata.nii.gz

    #1. Correct for eddy current distortions and head motion
    eddy_correct $dti_file {$DTI_COMP_DIR}/{$DATA_NAME}_eddy.nii.gz
      0 | tee -a $logfile

    #2. B0: if necessary, you can also specify the fractional
    # intensity threshold using the "-f" option. Different values
    # of this parameter control the size of the mask. (Smaller
    # values result in larger brain outline estimates.) The default
    # value of 0.5 probably works for your data.
    bet {$DTI_COMP_DIR}/{$DATA_NAME}_eddy.nii.gz
    {$DTI_COMP_DIR}/{$DATA_NAME}_dtidata_brain.nii.gz
    -m -f 0.3 | tee -a $logfile

    # 3. DTIFIT
    set bvecs=$SUBJECTS_DIR/*bvecs
    set bvals=$SUBJECTS_DIR/*bvals
    dtifit --data={$DTI_COMP_DIR}/{$DATA_NAME}_eddy.nii.gz
    --out={$DTI_COMP_DIR}/{$DATA_NAME}_dti
    --mask={$DTI_COMP_DIR}/{$DATA_NAME}_dtidata_brain.nii.gz
    --bvecs=$bvecs --bvals=$bvals | tee -a $logfile

    #4. T1 brain extraction
    bet $SUBJECTS_DIR/{$DATA_NAME}_T1.nii.gz
    {$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
    -m -f 0.3 | tee -a $logfile

    #5. Registration of DTI with T1
    mkdir ${DTI_COMP_DIR}/transformations
```

```
flirt -ref {$DTI_COMP_DIR}/{$DATA_NAME}_dti_FA.nii.gz
-in {$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz -dof 6
-omat ${DTI_COMP_DIR}/transformations/struct2FA.mat
-o {$DTI_COMP_DIR}/{$DATA_NAME}_dtiT1.nii.gz
-cost mutualinfo | tee -a $logfile
fnirt --ref={$DTI_COMP_DIR}/{$DATA_NAME}_dti_FA.nii.gz
--in={$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
--aff=${DTI_COMP_DIR}/transformations/struct2FA.mat
--cout=${DTI_COMP_DIR}/transformations/struct2FA_warp
--refmask={$DTI_COMP_DIR}/{$DATA_NAME}_dtidata_brain_mask.nii.gz
--inmask={$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain_mask.nii.gz |
tee -a $logfile
invwarp --warp=${DTI_COMP_DIR}/transformations/struct2FA_warp
--ref={$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
--out=${DTI_COMP_DIR}/transformations/struct2FA_warp_inv |
tee -a $logfile

# Apply the transformation to the data
applywarp -i {$DTI_COMP_DIR}/{$DATA_NAME}_dti_FA.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_T1_FA.nii.gz
-r {$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--interp=trilinear | tee -a $logfile
applywarp -i {$DTI_COMP_DIR}/{$DATA_NAME}_dti_MD.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_T1_MD.nii.gz
-r {$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--interp=trilinear | tee -a $logfile
applywarp -i {$DTI_COMP_DIR}/{$DATA_NAME}_dti_l1.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_T1_l1.nii.gz
-r {$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--interp=trilinear | tee -a $logfile
applywarp -i {$DTI_COMP_DIR}/{$DATA_NAME}_dti_l2.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_T1_l2.nii.gz
-r {$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--interp=trilinear | tee -a $logfile
applywarp -i {$DTI_COMP_DIR}/{$DATA_NAME}_dti_l3.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_T1_l3.nii.gz
-r {$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--interp=trilinear | tee -a $logfile
applywarp -i {$DTI_COMP_DIR}/{$DATA_NAME}_dti_v1.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_T1_v1.nii.gz
-r {$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--interp=nn | tee -a $logfile
applywarp -i {$DTI_COMP_DIR}/{$DATA_NAME}_dti_v2.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_T1_v2.nii.gz
-r {$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
```

```
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--interp=nn | tee -a $logfile
applywarp -i {$DTI_COMP_DIR}/{$DATA_NAME}_dti_v3.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_T1_v3.nii.gz
-r {$DTI_COMP_DIR}/{$DATA_NAME}_T1_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--interp=nn | tee -a $logfile

#6. transformation from T1 native space to FreeSurfer struct
tkregister2 --mov ${FREESURFER_DATA_DIR}/mri/orig.mgz
--targ ${FREESURFER_DATA_DIR}/mri/rawavg.mgz
--fslregout {$DTI_COMP_DIR}/transformations/freesurfer2struct.mat
--reg {$DTI_COMP_DIR}/transformations/junk
--regheader --noedit | tee -a $logfile
convert_xfm
-omat {$DTI_COMP_DIR}/transformations/struct2freesurfer.mat
-inverse {$DTI_COMP_DIR}/transformations/freesurfer2struct.mat |
 tee -a $logfile

#7. transformation from diffusion space to FreeSurfer space
mri_convert  ${FREESURFER_DATA_DIR}/mri/brain.mgz
{$DTI_COMP_DIR}/{$DATA_NAME}_data_brain.nii.gz | tee -a $logfile
applywarp -i ${DTI_COMP_DIR}/{$DATA_NAME}_dti_FA.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_FA_fr.nii.gz
-r ${DTI_COMP_DIR}/{$DATA_NAME}_data_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--postmat=${DTI_COMP_DIR}/transformations/struct2freesurfer.mat
--interp=trilinear | tee -a $logfile
applywarp -i ${DTI_COMP_DIR}/{$DATA_NAME}_dti_MD.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_MD_fr.nii.gz
-r ${DTI_COMP_DIR}/{$DATA_NAME}_data_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--postmat=${DTI_COMP_DIR}/transformations/struct2freesurfer.mat
--interp=trilinear | tee -a $logfile
applywarp -i ${DTI_COMP_DIR}/{$DATA_NAME}_dti_l1.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_l1_fr.nii.gz
-r ${DTI_COMP_DIR}/{$DATA_NAME}_data_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--postmat=${DTI_COMP_DIR}/transformations/struct2freesurfer.mat
--interp=trilinear | tee -a $logfile
applywarp -i ${DTI_COMP_DIR}/{$DATA_NAME}_dti_l2.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_l2_fr.nii.gz
-r ${DTI_COMP_DIR}/{$DATA_NAME}_data_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--postmat=${DTI_COMP_DIR}/transformations/struct2freesurfer.mat
--interp=trilinear | tee -a $logfile
applywarp -i ${DTI_COMP_DIR}/{$DATA_NAME}_dti_l3.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_l3_fr.nii.gz
-r ${DTI_COMP_DIR}/{$DATA_NAME}_data_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--postmat=${DTI_COMP_DIR}/transformations/struct2freesurfer.mat
```

```
--interp=trilinear | tee -a $logfile
applywarp -i ${DTI_COMP_DIR}/{$DATA_NAME}_dti_v1.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_v1_fr.nii.gz
-r ${DTI_COMP_DIR}/{$DATA_NAME}_data_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--postmat=${DTI_COMP_DIR}/transformations/struct2freesurfer.mat
--interp=nn | tee -a $logfile
applywarp -i ${DTI_COMP_DIR}/{$DATA_NAME}_dti_v2.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_v2_fr.nii.gz
-r ${DTI_COMP_DIR}/{$DATA_NAME}_data_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--postmat=${DTI_COMP_DIR}/transformations/struct2freesurfer.mat
--interp=nn | tee -a $logfile
applywarp -i ${DTI_COMP_DIR}/{$DATA_NAME}_dti_v3.nii.gz
-o ${DTI_COMP_DIR}/{$DATA_NAME}_v3_fr.nii.gz
-r ${DTI_COMP_DIR}/{$DATA_NAME}_data_brain.nii.gz
-w ${DTI_COMP_DIR}/transformations/struct2FA_warp_inv
--postmat=${DTI_COMP_DIR}/transformations/struct2freesurfer.mat
--interp=nn | tee -a $logfile

#8. Project diffusion data to mesh
printf "$DATA_NAME\n1\n1\n1\n1 0 0 0\n0 1 0 0\n0 0 1 0\n0 0 0 1\n"
> ${DTI_COMP_DIR}/reg.dat
setenv SUBJECTS_DIR "/pathtodata"

mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_FA_fr.nii.gz
--srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi lh
--out ${DTI_COMP_DIR}/{$DATA_NAME}_FA_left.mgh | tee -a $logfile
mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_MD_fr.nii.gz
--srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi lh
--out ${DTI_COMP_DIR}/{$DATA_NAME}_MD_left.mgh | tee -a $logfile
mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_l1_fr.nii.gz
--srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi lh
--out ${DTI_COMP_DIR}/{$DATA_NAME}_l1_left.mgh | tee -a $logfile
mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_l2_fr.nii.gz
--srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi lh
--out ${DTI_COMP_DIR}/{$DATA_NAME}_l2_left.mgh | tee -a $logfile
mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_l3_fr.nii.gz
--srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi lh
--out ${DTI_COMP_DIR}/{$DATA_NAME}_l3_left.mgh | tee -a $logfile
mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_v1_fr.nii.gz
--srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi lh
--out ${DTI_COMP_DIR}/{$DATA_NAME}_v1_left.mgh | tee -a $logfile
mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_v2_fr.nii.gz
--srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi lh
--out ${DTI_COMP_DIR}/{$DATA_NAME}_v2_left.mgh | tee -a $logfile
mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_v3_fr.nii.gz
--srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi lh
--out ${DTI_COMP_DIR}/{$DATA_NAME}_v3_left.mgh | tee -a $logfile

mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_FA_fr.nii.gz
```

```
    --srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi rh
    --out ${DTI_COMP_DIR}/{$DATA_NAME}_FA_right.mgh | tee -a $logfile
    mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_MD_fr.nii.gz
    --srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi rh
    --out ${DTI_COMP_DIR}/{$DATA_NAME}_MD_right.mgh | tee -a $logfile
    mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_l1_fr.nii.gz
    --srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi rh
    --out ${DTI_COMP_DIR}/{$DATA_NAME}_l1_right.mgh | tee -a $logfile
    mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_l2_fr.nii.gz
    --srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi rh
    --out ${DTI_COMP_DIR}/{$DATA_NAME}_l2_right.mgh | tee -a $logfile
    mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_l3_fr.nii.gz
    --srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi rh
    --out ${DTI_COMP_DIR}/{$DATA_NAME}_l3_right.mgh | tee -a $logfile
    mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_v1_fr.nii.gz
    --srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi rh
    --out ${DTI_COMP_DIR}/{$DATA_NAME}_v1_right.mgh | tee -a $logfile
    mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_v2_fr.nii.gz
    --srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi rh
    --out ${DTI_COMP_DIR}/{$DATA_NAME}_v2_right.mgh | tee -a $logfile
    mri_vol2surf --src  ${DTI_COMP_DIR}/{$DATA_NAME}_v3_fr.nii.gz
    --srcreg ${DTI_COMP_DIR}/reg.dat --projfrac 0.5 --hemi rh
    --out ${DTI_COMP_DIR}/{$DATA_NAME}_v3_right.mgh | tee -a $logfile

    # 9. Load and save the DTI data in MATLAB format
    matlab -nodisplay -nodesktop -nosplash -r
    "createDTIparams('$SUBJECTS_DIR/$DATA_NAME','$DATA_NAME'),exit" |
    tee -a $logfile

    echo "* copy the results to the folder *" | tee -a $logfile
    cp {$DTI_COMP_DIR}/{$DATA_NAME}_DTI_left_hemisphere.mat
    $SUBJECTS_DIR/$DATA_NAME/{$DATA_NAME}_DTI_left_hemisphere.mat
    cp {$DTI_COMP_DIR}/{$DATA_NAME}_FA_left.vtk
    $SUBJECTS_DIR/$DATA_NAME/{$DATA_NAME}_FA_left.vtk
    cp {$DTI_COMP_DIR}/{$DATA_NAME}_MD_left.vtk
    $SUBJECTS_DIR/$DATA_NAME/{$DATA_NAME}_MD_left.vtk
    cp {$DTI_COMP_DIR}/{$DATA_NAME}_DTI_right_hemisphere.mat
    $SUBJECTS_DIR/$DATA_NAME/{$DATA_NAME}_DTI_right_hemisphere.mat
    cp {$DTI_COMP_DIR}/{$DATA_NAME}_FA_right.vtk
    $SUBJECTS_DIR/$DATA_NAME/{$DATA_NAME}_FA_right.vtk
    cp {$DTI_COMP_DIR}/{$DATA_NAME}_MD_right.vtk
    $SUBJECTS_DIR/$DATA_NAME/{$DATA_NAME}_MD_right.vtk

    date +"%T" | tee -a $logfile
    echo "*END*" | tee -a $logfile

end

date +"%T" | tee -a $logfile
echo "*END OF PROCESSING*" | tee -a $logfile
```

Appendix D

Case study data

"We are not going in circles, we are going upwards.
The path is a spiral; we have already climbed many steps."

Hermann Hesse, in *Siddhartha* (1951)

We give here detailed information about the data provided by Dr. Marina de Tommaso from the Policlinico di Bari in Italy. We list the observed symptoms for each patient, show the results of the cortical reconstruction and the ADC values mapped to the cortical mesh. Finally, we visualise the CSD propagation and just plot it on the hemispheres that show symptoms during a migraine attack.

For each patient we have MRI and DWI data of the brain and the symptoms and their order of appearance during a migraine attack with aura. These symptoms can be related to certain areas of the cerebral cortex, also called eloquent regions. For a complete overview, we list the symptoms and information about the patient's clinical history in Table D.1, while Table D.2 provides the list of eloquent areas for each patient.

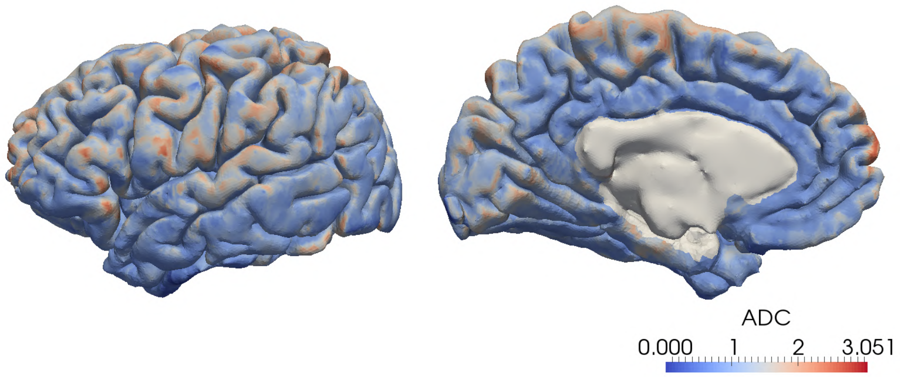
In Figures D.1 to D.8 we show the ADC values on the smoothed reconstructed cortical mesh, the sequence of eloquent areas during a migraine attack and the activation times of the CSD simulation for each patient.

patient	age of illness (years)	migraine with aura frequency (days with migraine/month)	migraine without aura frequency (days with migraine/month)	diagnosis	symptoms
Subject 1	9	5	–	• migraine with aura	• luminous scotoma in central visual field • right hemifield scotoma • right and left paresthesias • aphasia, apraxia
Subject 2	25	2	12	• migraine with aura (typical) • migraine without aura	• visual symptoms • 1 hour
Subject 3	10	0.5	–	• migraine with aura (typical) • migraine without aura • 1 single episode with prolonged atypical aura	• usually visual symptoms • one episode • visual symptoms, followed by headache, sensory symptoms and aphasia
Subject 4	10	–	1	• migraine aura (typical) with non-migraine headache	• visual symptoms • right arm motor symptoms • right arm sensory symptoms, aphasia
Subject 5	15	0.5	7	• migraine aura • migraine without aura	• black scotoma in full visual field • luminous scotoma in peripheral left visual hemifield

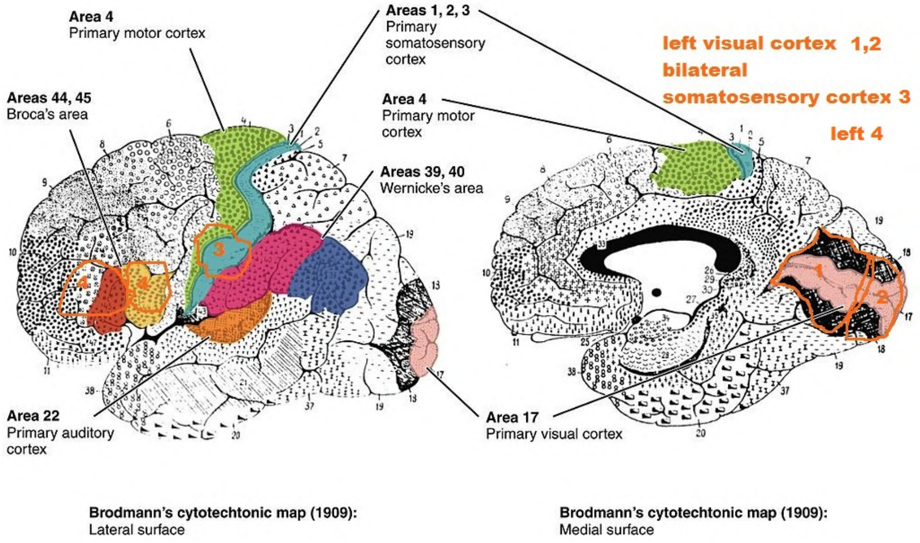
TABLE D.1: The clinical details of the symptoms during a migraine attack of Subjects 1-5.

patients	hemisphere	eloquent regions	sequence
Taranto 1	left	visual cortex, somatosensory cortex, frontal area	BA17/18, BA1-4, BA44/45
	right	somatosensory cortex	BA1-4
Subject 2	left	visual cortex, primary somatosensory cortex, frontal area	BA17/18, BA1-4, BA44
	right	-	-
Subject 3	left	visual cortex, one episode with visual cortex, primary somatosensory cortex, frontal area	BA17/18, BA1-4, BA44
	right	visual cortex	BA17/18
Subject 4	left	somatosensory cortex, frontal area	BA1-4, BA44
	right	visual cortex	BA17/18
Subject 5	left	-	-
	right	visual cortex	BA17/18

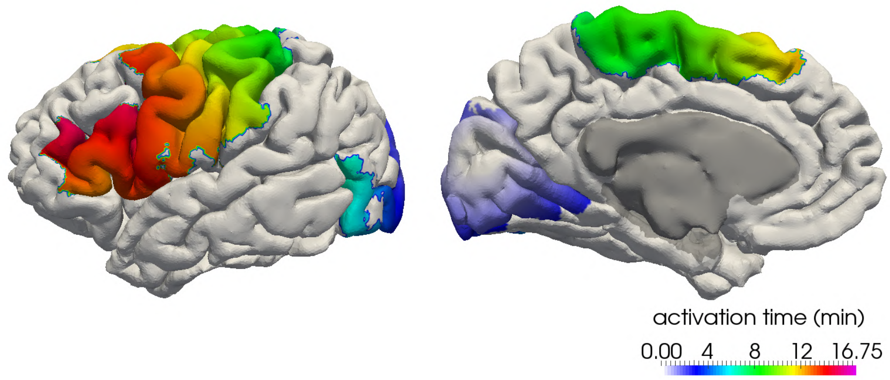
TABLE D.2: The eloquent regions of Subjects 1-5.



(a) normalised ADC values on the smoothed cortical surface

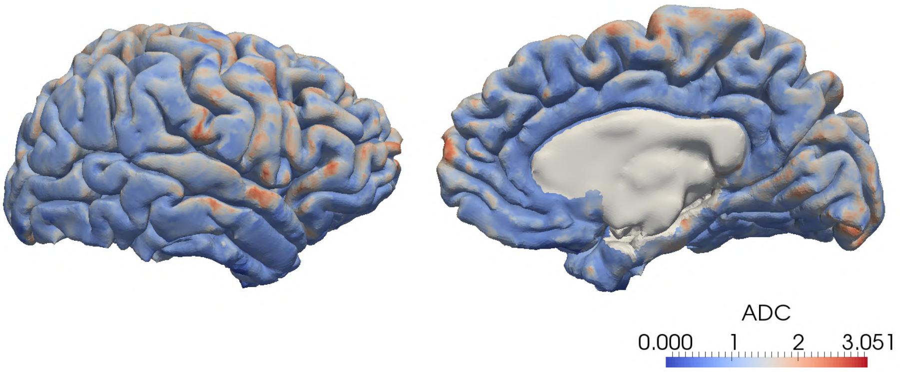


(b) sequence of eloquent areas during a migraine attack

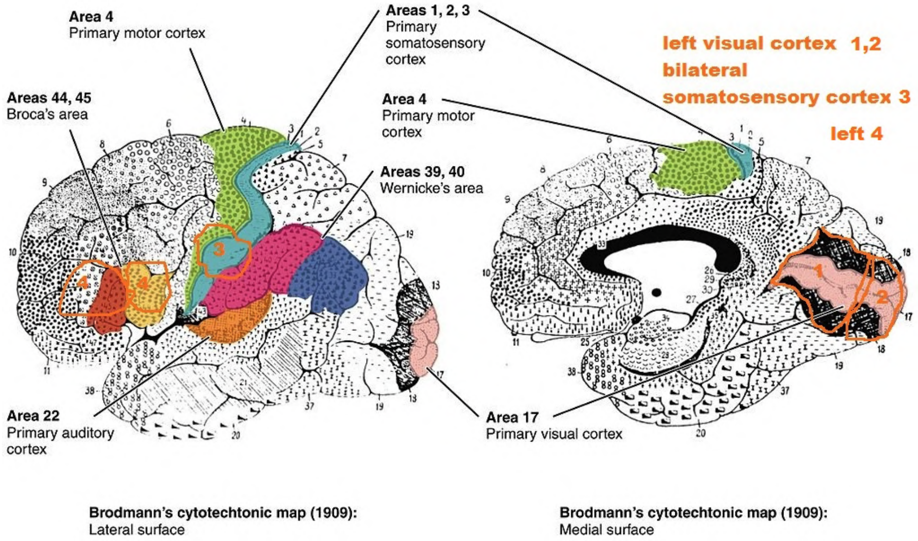


(c) activation times of the CSD simulation

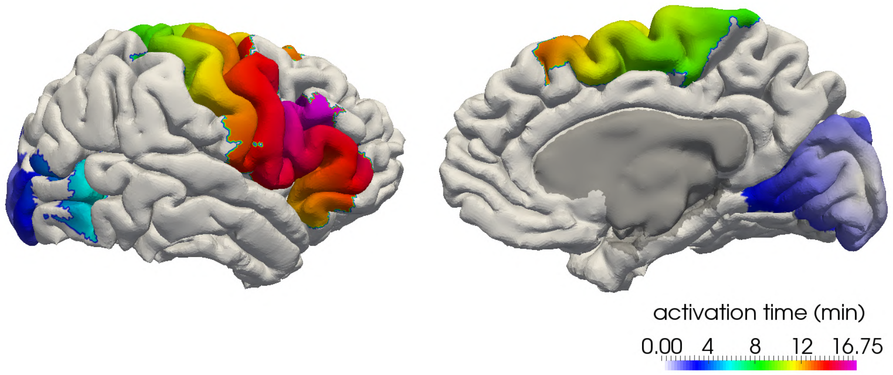
FIGURE D.1: Left hemisphere of Subject 1: The ADC values on the smooth surface mesh after taking out the corpus callosum (a), the sequence of eloquent areas during a migraine attack (b) and the activation times of the CSD simulation (c).



(a) normalised ADC values on the smoothed cortical surface

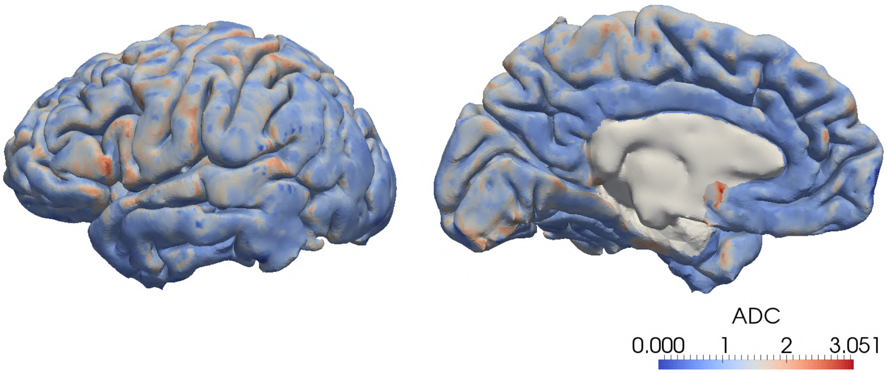


(b) sequence of eloquent areas during a migraine attack

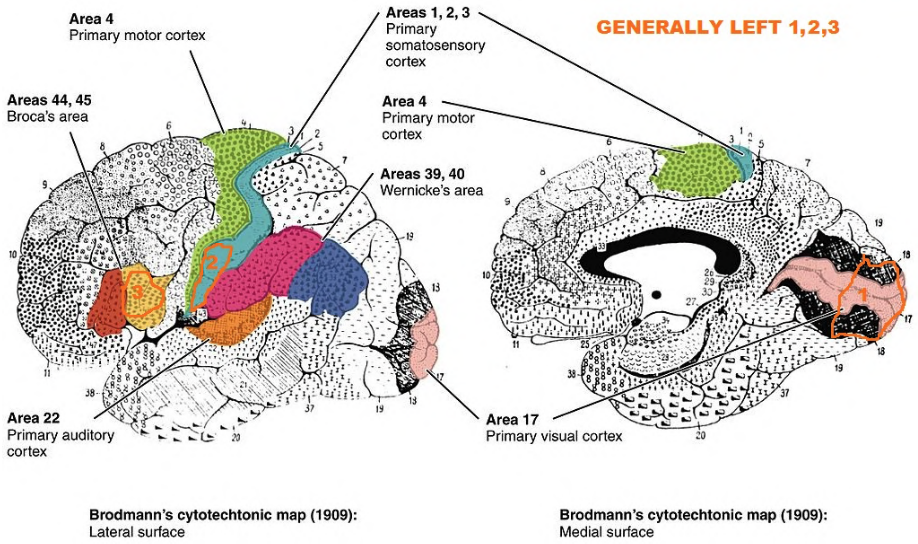


(c) activation times of the CSD simulation

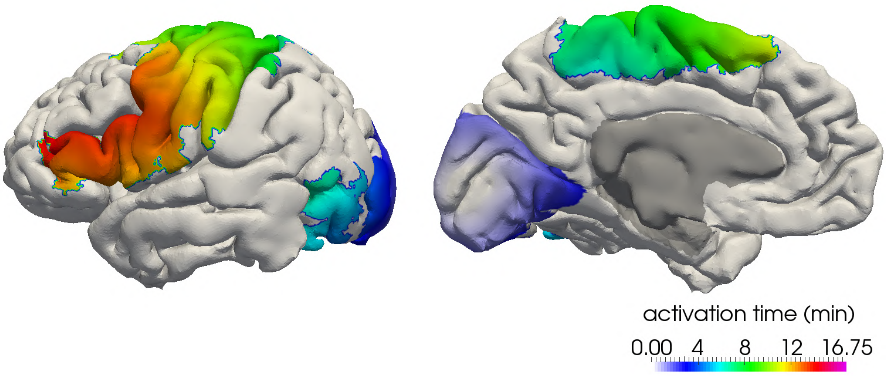
FIGURE D.2: Right hemisphere of Subject 1: The ADC values on the smooth surface mesh after taking out the corpus callosum (a), the sequence of eloquent areas during a migraine attack (b) and the activation times of the CSD simulation (c).



(a) normalised ADC values on the smoothed cortical surface

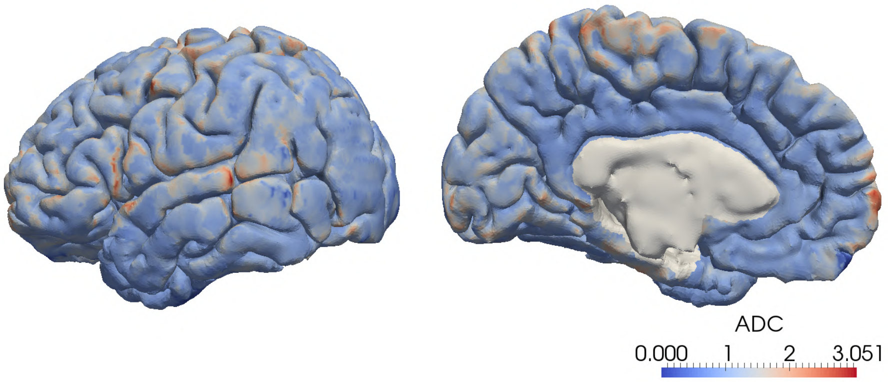


(b) sequence of eloquent areas during a migraine attack

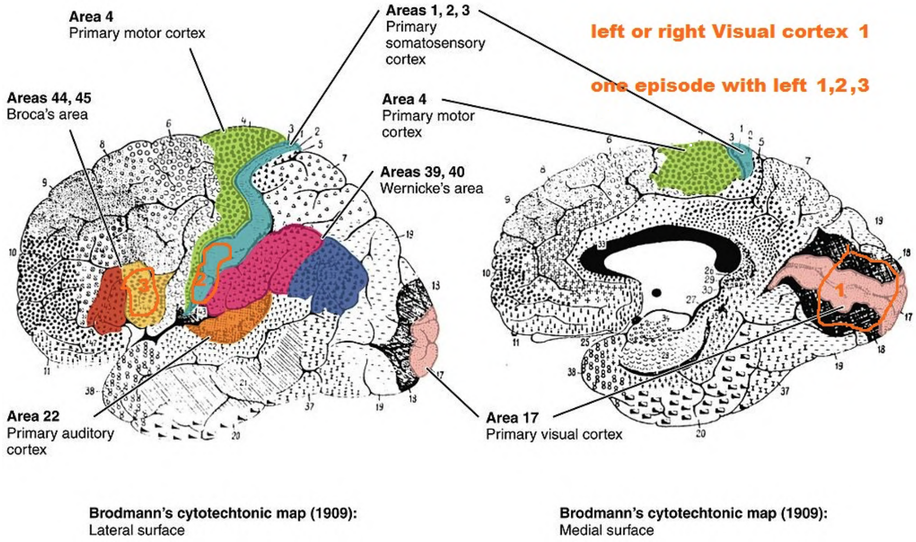


(c) activation times of the CSD simulation

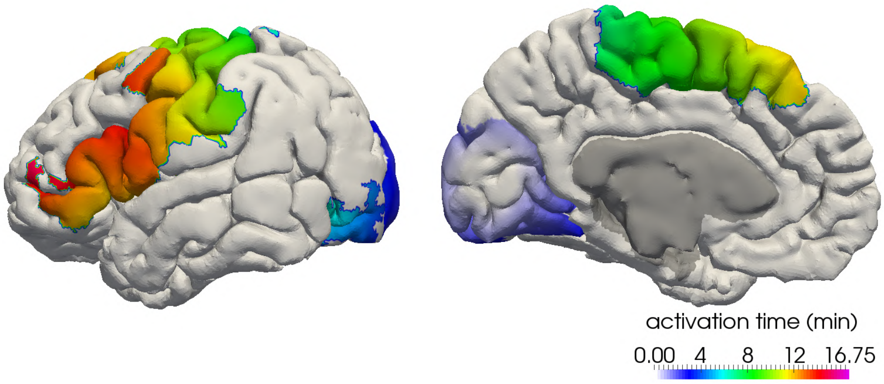
FIGURE D.3: Left hemisphere of Subject 2: The ADC values on the smooth surface mesh after taking out the corpus callosum (a), the sequence of eloquent areas during a migraine attack (b) and the activation times of the CSD simulation (c).



(a) normalised ADC values on the smoothed cortical surface

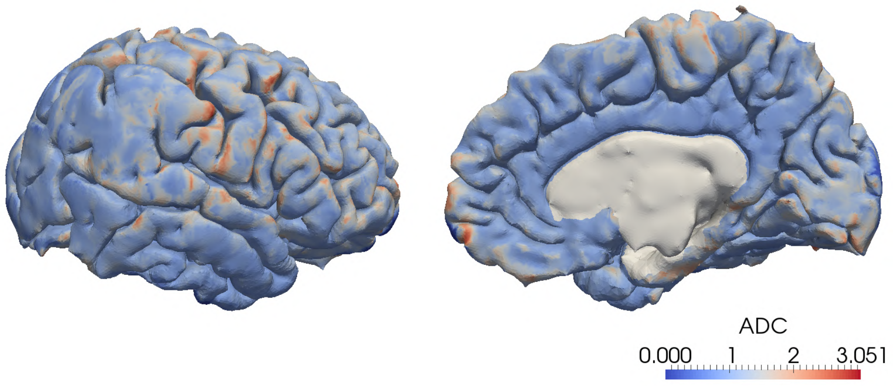


(b) sequence of eloquent areas during a migraine attack

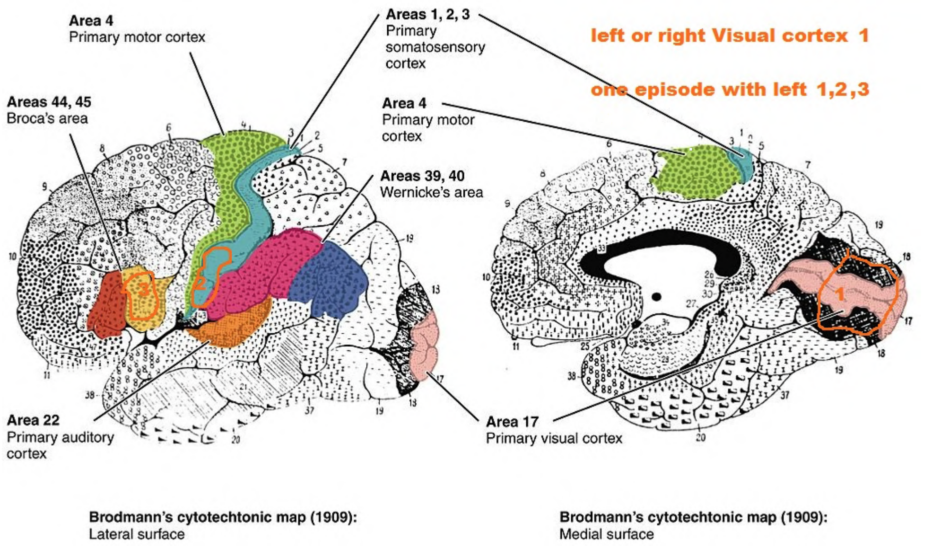


(c) activation times of the CSD simulation

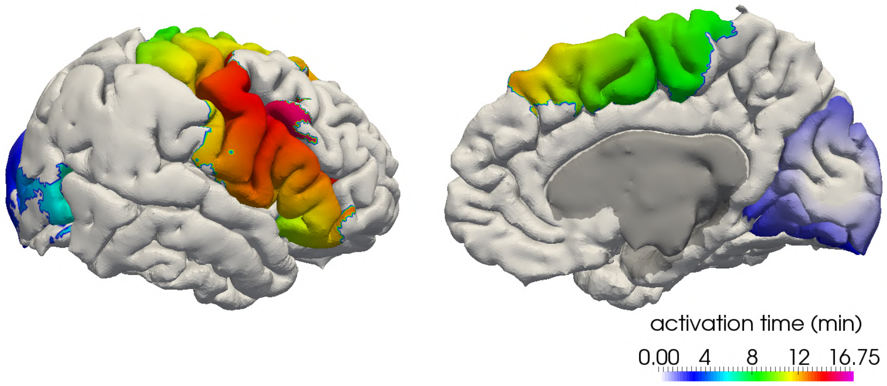
FIGURE D.4: Left hemisphere of Subject 3: The ADC values on the smooth surface mesh after taking out the corpus callosum (a), the sequence of eloquent areas during a migraine attack (b) and the activation times of the CSD simulation (c).



(a) normalised ADC values on the smoothed cortical surface

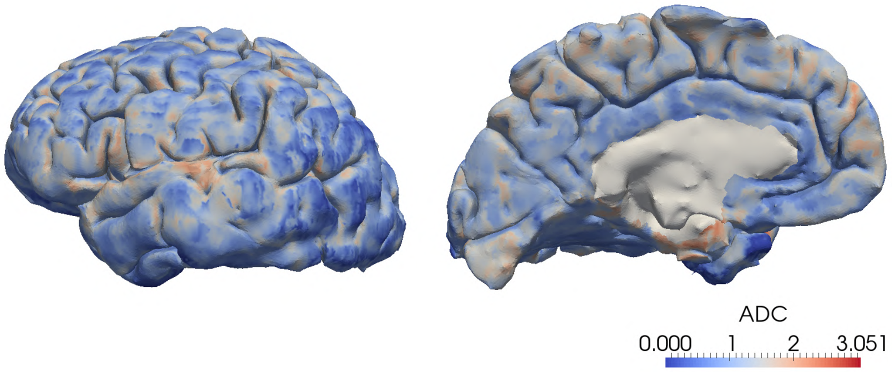


(b) sequence of eloquent areas during a migraine attack

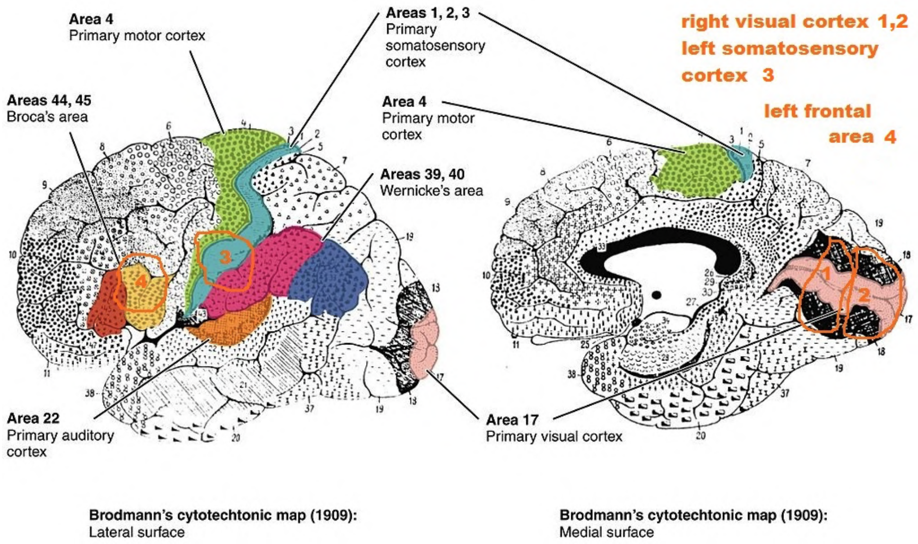


(c) activation times of the CSD simulation

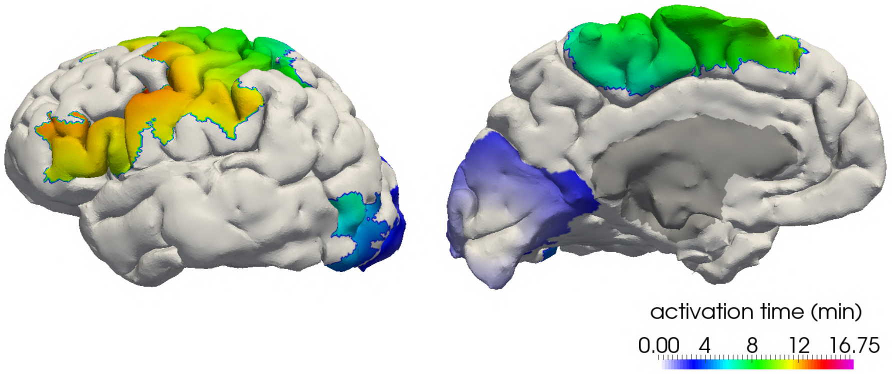
FIGURE D.5: Right hemisphere of Subject 3: The ADC values on the smooth surface mesh after taking out the corpus callosum (a), the sequence of eloquent areas during a migraine attack (b) and the activation times of the CSD simulation (c).



(a) normalised ADC values on the smoothed cortical surface

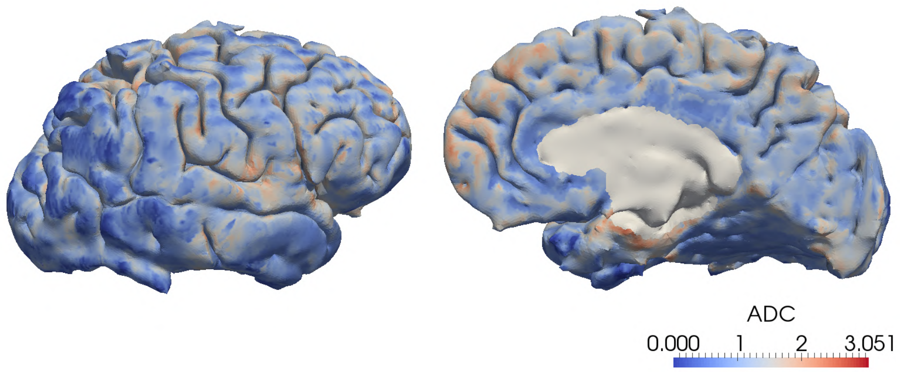


(b) sequence of eloquent areas during a migraine attack

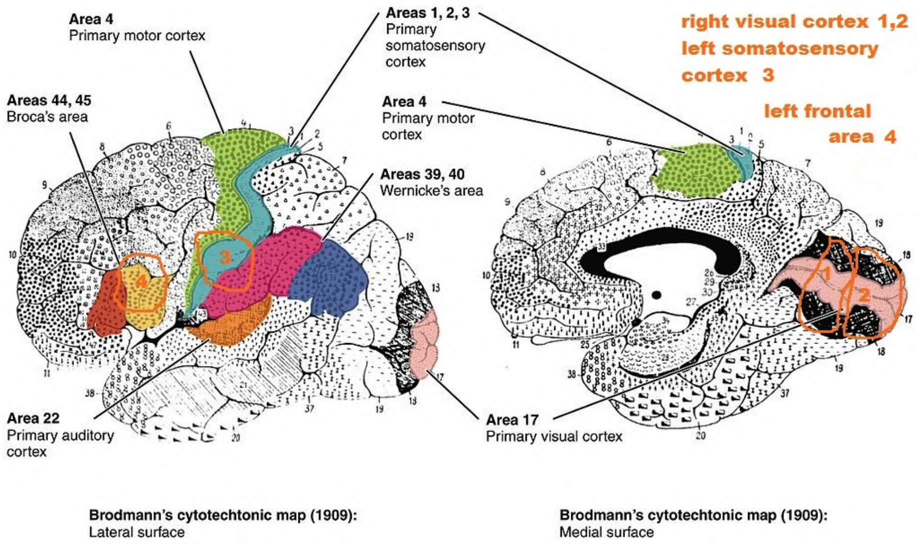


(c) activation times of the CSD simulation

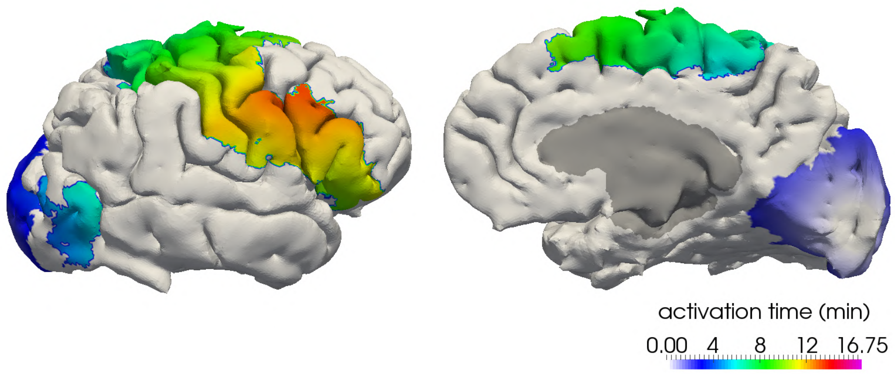
FIGURE D.6: Left hemisphere of Subject 4: The ADC values on the smooth surface mesh after taking out the corpus callosum (a), the sequence of eloquent areas during a migraine attack (b) and the activation times of the CSD simulation (c).



(a) normalised ADC values on the smoothed cortical surface

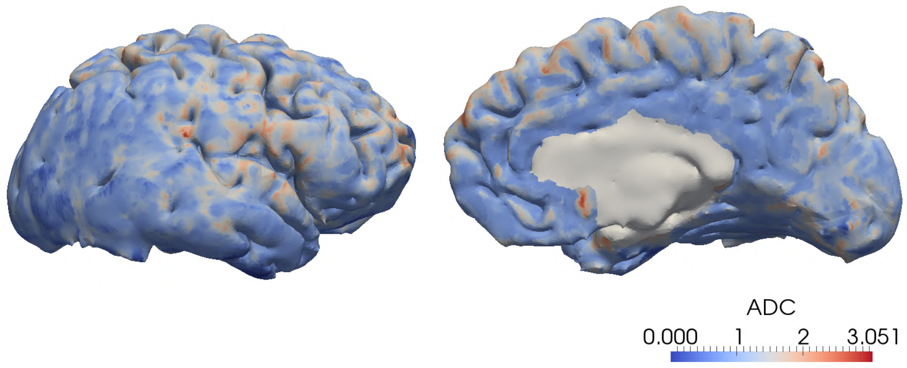


(b) sequence of eloquent areas during a migraine attack

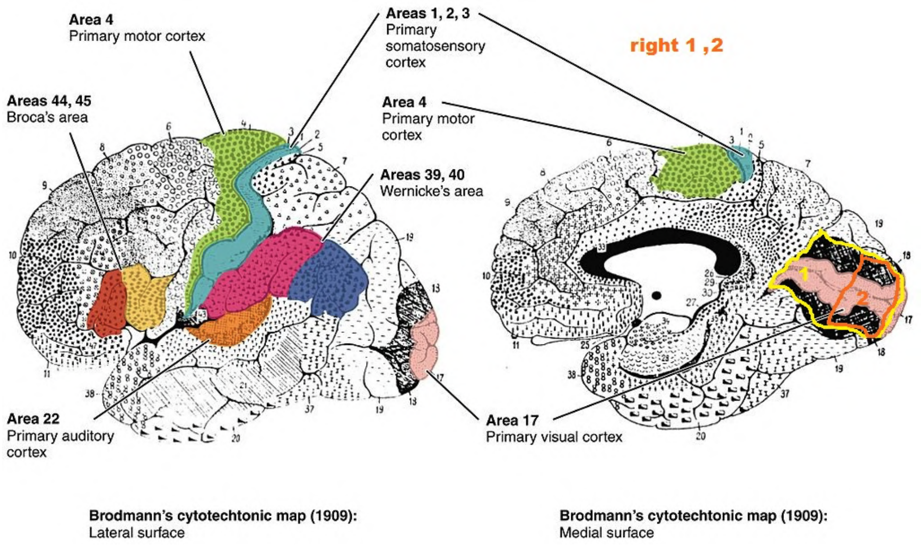


(c) activation times of the CSD simulation

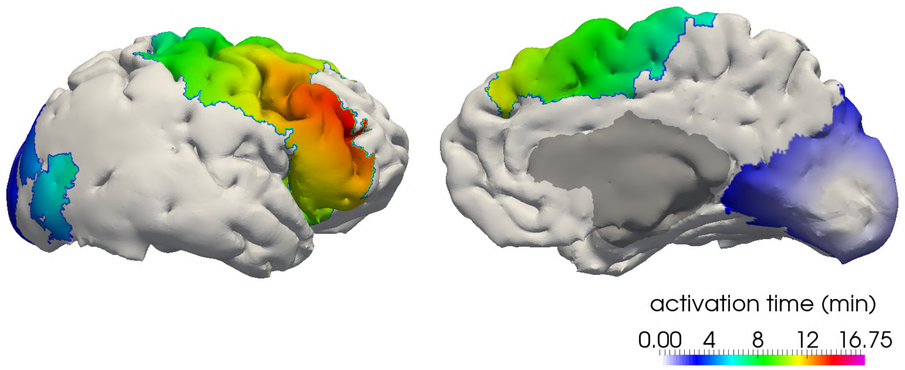
FIGURE D.7: Right hemisphere of Subject 4: The ADC values on the smooth surface mesh after taking out the corpus callosum (a), the sequence of eloquent areas during a migraine attack (b) and the activation times of the CSD simulation (c).



(a) normalised ADC values on the smoothed cortical surface



(b) sequence of eloquent areas during a migraine attack



(c) activation times of the CSD simulation

FIGURE D.8: Right hemisphere of Subject 5: The ADC values on the smooth surface mesh after taking out the corpus callosum (a), the sequence of eloquent areas during a migraine attack (b) and the activation times of the CSD simulation (c).

ROI number	ROI name	ROI number	ROI name
1	fusiform	18	superiorfrontal
2	parahippocampal	19	frontalpole
3	entorhinal	20	paracentral
4	temporalpole	21	precuneus
5	inferiortemporal	22	superiorparietal
6	middletemporal	23	inferiorparietal
7	superiortemporal	24	postcentral
8	transversetemporal	25	supramarginal
9	bankssts	26	cuneus
10	lateralorbitofrontal	27	pericalcarine
11	medialorbitofrontal	28	lingual
12	parsorbitalis	29	lateraloccipital
13	parstriangularis	30	isthmuscingulate
14	rostralmiddlefrontal	31	posteriorcingulate
15	parsopercularis	32	caudalanteriorcingulate
16	caudalmiddlefrontal	33	rostralanteriorcingulate
17	precentral	34	insula

TABLE D.3: The region of interest (ROI) number and the corresponding region name for the 34 ROI defined by the Desikan-Killiany atlas.

List of abbreviations

ABU	**Abu**sive painkiller user group
ADC	**A**pparent **d**iffusion **c**oefficient
ANOVA	**An**alysis **o**f **va**riance
ANTs	**A**dvanced **n**ormalization **t**ools
ATP	**A**denosine **t**ri**p**hosphate
BA	**B**rodmann **a**rea
BET	**B**rain **e**xtraction **t**ool
BiCGSTAB	**Bi**conjugate **g**radient **stab**ilised method
c.c.	**C**ingulate **c**ortex
CG	**C**onjugate **g**radient method
CON	**Con**trol group
CSD	**C**ortical **s**preading **d**epression
DICOM	**D**igital **i**maging and **com**munication in **m**edicine
DTI	**D**iffusion **t**ensor **i**maging
DOF	**D**egrees **o**f **f**reedom
DWI	**D**iffusion-**w**eighted **i**maging
EEG	**E**lectro**e**ncephalo**g**raphy
FA	**F**ractional **a**nisotropy
FAD	**F**lavin **a**denine **d**inucleotide
FHM	**F**amilial **h**emiplegic **m**igraine
FLIRT	**F**MRIB **l**inear **i**mage **r**egistration **t**ool
fMRI	**F**unctional **m**agnetic **r**esonance **i**maging
FNIRT	**F**MRIB **n**onlinear **i**mage **r**egistration **t**ool
FSL	**F**MRIB **s**oftware **l**ibrary
GAI	**G**lobal **a**symmetry **i**ndex
gPC	**G**eneralised **p**olynomial **c**haos
ICT	**I**ncomplete **C**holesky factorisation with drop **t**olerance
ILU	**I**ncomplete **LU** factorisation
IMEX	**Im**plicit **ex**plicit scheme
l.t.	**L**ateral **t**emporal lobe
MD	**M**ean **d**iffusivity
MEG	**M**agneto**e**ncephalo**g**raphy
ME-PCM	**M**ulti-**e**lement **p**robabilistic **c**ollocation **m**ethod
MIG	**Mig**raine patients without aura group
MRI	**M**agnetic **r**esonance **i**maging
m.t.	**M**edial **t**emporal lobe
NAD	**N**icotinamide **a**denine **d**inucleotide
NIfTI	**N**euroimaging **inf**ormatics **t**echnology **i**nitiative
ODE	**O**rdinary **d**ifferential **e**quation
o.l.	**O**ccipital **l**obe
PCG	**P**reconditioned **c**onjugate **g**radient method
PCM	**P**robabilistic **c**ollocation **m**ethod
PDE	**P**artial **d**ifferential **e**quation

PDF	**P**robability **d**ensity **f**unction
pH	**P**otential of **h**ydrogen
QoI	**Q**uantity/quantities **o**f **i**nterest
RA	**R**elative **a**nisotropy
RI	**R**etention **i**ndex
ROI	**R**egion/regions **o**f **i**nterest
SRI	**S**urface **r**egularity **i**ndex
std	**St**andard **d**eviation
TRACULA	**TRA**cts **c**onstrained by **u**nder**l**ying **a**natomy
UQ	**U**ncertainty **q**uantification
VR	**V**olume **r**atio

List of units

A	ampere	electric current
kg	kilogram	mass
m	metre	length
mol	mole	amount of substance
sec	second	time
C	coulomb	electric charge
F	farad	capacitance
g	gram	mass
GB	gigabyte	digital information
Hz	hertz	frequency
L	liter	volume
M	molar	molar concentration
min	minute	time
mm	millimetres	length
msec	millisecond	time
mV	millivolt	electrical potential
S	siemens	electric conductance
T	tesla	magnetic flux density
V	volt	electric potential

List of figures

List of tables

Bibliography

Ahrens, J., B. Geveci, and C. Law (2005). "ParaView: An end-user tool for large data visualization". In: *Elsevier*. DOI: `978-0123875822` (see p. 93).

Andersson, J.L.R. and S.N. Sotiropoulos (2016). "An integrated approach to correction for off-resonance effects and subject movement in diffusion MR imaging". In: *NeuroImage* 125, pp. 1063–1078. DOI: `10.1016/j.neuroimage.2015.10.019` (see p. 200).

Archer, G.E.B., A. Satelli, and I.M. Sobol' (1997). "Sensitivity measures, ANOVA-like techniques and the use of bootstrap". In: *J. of Statist. Comput. Simul.* 58, pp. 99–120. DOI: `10.1080/00949659708811825` (see pp. 141, 187).

Askey, R. and J. Willson (1985). *Some Basic Hypergeometric Orthogonal Polynomials that Generalize Jacobi Polynomials*. 319. Mem. Amer. Math. Soc. DOI: `978-0-8218-2321-7` (see p. 169).

Ayata, C. (2010). "Cortical spreading depression triggers migraine attack: pro". In: *Headache* 50.4, pp. 725–30. DOI: `10.1111/j.1526-4610.2010.01647.x` (see p. 153).

Ayata, C. and M. Lauritzen (2015). "Spreading depression, spreading depolarizations, and the cerebral vasculature". In: *Physiol Rev.* 95.3, pp. 953–993. DOI: `10.1152/physrev.00027.2014` (see p. 151).

Ayata, C. et al. (2006). "Suppression of cortical spreading depression in migraine prophylaxis". In: *Ann Neurol.* 59.4, pp. 625–61. DOI: `10.1002/ana.20778` (see p. 11).

Bade, R., J. Haase, and B. Preim (2006). "Comparison of fundamental mesh smoothing algorithms for medical surface models". In: *SimVis*. Vol. 6, pp. 289–304 (see p. 80).

Barreto, E. and J.R. Cressman (2011). "Ion concentration dynamics as a mechnism for neuronal bursting". In: *J. Biol. Phys.* 37.3, pp. 361–373. DOI: `10.1007/s10867-010-9212-6` (see pp. xv, 14, 20, 30).

Biggs, N. (1993). *Algebraic Graph Theory*. 2nd ed. Cambridge University Press. DOI: `10.1017/CBO9780511608704` (see p. 168).

Bolay, H. and M.A. Moskowitz (2005). "The emerging importance of cortical spreading depreassion in migraine headache". In: *Revue Neurologique* 161.6-7, pp. 655–7. DOI: `10.1016/s0035-3787(05)85108-2` (see p. 11).

Borgdorff, P. (2018). "Arguments against the role of cortical spreading depression in migraine". In: *Neurological Research* 40.3, pp. 173–181. DOI: `10.1080/01616412.2018.1428406` (see p. 12).

Bossen, F.J. and P.S. Heckbert (1996). "A pliant method for anisotropic mesh generation". In: *5th Intl. Meshing Roundtable*, pp. 63–74 (see p. 80).

Bowyer, S.M. et al. (1999). "Analysis of MEG signals of spreading cortical depression with propagation constrained to a rectangular cortical strip: II. Gyrencephalic swine model". In: *Brain Res.* 843.1-2, pp. 79–86. DOI: `10.1016/S0006-8993(99)01893-4` (see p. 11).

Bowyer, S.M. et al. (2001). "Magnetoencephalographic fields from patients with spontaneous and induced migraine aura". In: *Ann Neurol.* 50.5, pp. 582–587. DOI: `10.1002/ana.1293` (see p. 11).

Brander, A. et al. (2010). “Diffusion tensor imaging of the brain in a healthy adult population: Normative values and measurement reproducibility at 3 T and 1.5 T”. In: *Acta Radiologica* 51.7, pp. 800–807. DOI: `10.3109/02841851.2010.495351` (see p. 108).

Broca, P. (1861). “L’aphasie de Broca”. In: *Bulletin de la société française d’anthropologie* (see p. 7).

Brodmann, K. (2006). *Brodmann’s Localisation in the Cerebral Cortex - The Principles of Comparative Localisation in the Cerebral Cortex Based on Cytoarchitectonics*. Ed. by L.J. Garey. Springer. DOI: `10.1007/b138298` (see pp. 7, 110).

Brötzner, C.P. et al. (2014). “Resting state alpha frequency is associated with menstrual cycle phase, estradiol and use of oral contraceptives”. In: *Brain Res.* 1577, pp. 36–44. DOI: `10.1016/j.brainres.2014.06.034` (see p. 9).

Bures, J., O. Buresova, and J. Krivanek (1974). *Mechanism and Applications of Leao’s Spreading Depression of Electroencephalographic Activity*. Academic Press Inc.,U.S. DOI: `10.1002/neu.480060608` (see p. 153).

Colli Franzone, P. and G. Savare (2002). “Evolution Equations, Semigroups and Functional Analysis”. In: Progress in Nonlinear Differential Equations and Their Applications. Birkäuser Basel. Chap. Degenerate Evolution Systems Modeling the Cardiac Electric Field at Micro- and Macroscopic Level, pp. 49–78. DOI: `10.1007/978-3-0348-8221-7_4` (see p. 40).

Cressman, J.R. Jr. et al. (2009). “The influence of sodium and potassium dynamics on excitability, seizures, and the stability of persistent states: I. Single neuron dynamics”. In: *J. Comput. Neurosci.* 26.2, pp. 159–170. DOI: `10.1007/s10827-008-0132-4` (see pp. ix, xv, 14, 20, 23, 26, 30).

— (2011). “Erratum to: The influence of sodium and potassium dynamics on excitability, seizures, and the stability of persistent states: I. Single neuron dynamics”. In: *J Comput Neurosci* 30.3, pp. 781–781. DOI: `10.1007/s10827-011-0333-0` (see pp. 14, 20).

Dahlem, M.A. et al. (2015). “Cortical hot spots and labyrinths: why cortical neuromodulation for episodic migraine with aura should be personalized”. In: *Front. Comput. Neurosci.* DOI: `10.3389/fncom.2015.00029` (see pp. vii, xiii, 2, 79).

Dassi, F. et al. (2015). “Anisotropic finite element mesh adaptation via higher dimensional embedding”. In: *Procedia Engineering* 124, pp. 265–277. DOI: `j.proeng.2015.10.138` (see p. 80).

De Maesschalck, R., D. Jouan-Rimbaud, and D.L. Massart (2000). “The Mahalanobis distance”. In: *Chemometrics and Intelligent Laboratory Systems* 50.1, pp. 1–18. DOI: `S0169-7439(99)00047-7` (see p. 103).

Deco, G. et al. (2008). “The dynamic brain: from spiking neurons to neural masses and cortical fields”. In: *PLoS Comput Biol.* 4.8, e1000092. DOI: `10.1371/journal.pcbi.1000092` (see p. 34).

D’Elia, M. et al. (2011). “Applications of variational data assimilation in computational hemodynamics”. In: *Modeling of Physiological Flows*, pp. 363–394. DOI: `10.1007/978-88-470-1935-5_12` (see p. 64).

Delingette, H. et al. (2012). “Personalization of cardiac motion and contractility from images using variational data assimilation”. In: *IEEE Transactions on Biomedical Engineering, Institute of Electrical and Electronics Engineers* 59.1, pp. 20–24. DOI: `10.1109/TBME.2011.2160347` (see p. 64).

Demšar, J. et al. (2013). “Orange: Data mining toolbox in Python”. In: *Journal of Machine Learning Research* 14, pp. 2349–2353. URL: `http://jmlr.org/papers/v14/demsar13a.html` (see p. 123).

Deppe, M. et al. (2014). “Increased cortical curvature reflects white matter atrophy in individual patients with early multiple sclerosis”. In: *NeuroImage: Clinical* 6, pp. 475–487. ISSN: 2213-1582. DOI: `https://doi.org/10.1016/j.nicl.2014.`

02.012. URL: `http://www.sciencedirect.com/science/article/pii/S2213158214000308` (see p. 79).

Desikan, R.S. et al. (2006). "An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest". In: *NeuroI* 31, pp. 968–980. DOI: `10.1016/j.neuroimage.2006.01.021` (see pp. 7, 64, 117).

Diez, I. et al. (2015). "A novel brain partition highlights the modular skeleton shared by structure and function". In: *Sci Rep* 5.10532. DOI: `10.1038/srep10532` (see pp. 52, 54).

Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". In: *Numer. Math.* 1.1, pp. 269–271. DOI: `10.1007/BF01386390` (see p. 95).

Do Carmo, M.P. (1976). *Differential Geometry of Curves and Surfaces*. 1st ed. Prentice-Hall Englewood Cliffs. DOI: `9780132125895` (see pp. 87, 164).

Dunavant, D. A. (1985). "High degree efficient symmetrical Gaussian quadrature rules for the triangle". In: *Internat. J. Numer. Meth. Engrg* 21.6, pp. 1129 –1148. DOI: `10.1002/nme.1620210612` (see p. 41).

Eikermann-Haerter, K. et al. (2009). "Genetic and hormonal factors modulate spreading depression and transient hemiparesis in mouse models of familial hemiplegic migraine type 1". In: *J Clin Invest* 119.1, pp. 99–109. DOI: `10.1172/JCI36059` (see p. 153).

Enger, R. et al. (2015). "Dynamics of Ionic Shifts in Cortical Spreading Depression". In: *Cereb. Cortex* 25.11, pp. 4469–76. DOI: `10.1093/cercor/bhv054` (see pp. xv, 10).

Engnew, T.R. (2009). "Suffering, meaning, and healing: Challenges of contemporary medicine". In: *Ann Fam Med* 7.2, pp. 170–175. DOI: `10.1370/afm.943` (see p. 51).

Fischl, B. (2012). "FreeSurfer". In: *NeuroImage* 62.2, pp. 774–781. DOI: `10.1016/j.neuroimage.2012.01.021` (see pp. 52, 193).

Fishman, G.S. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. Springer New York. DOI: `10.1007/978-1-4757-2553-7` (see p. 174).

FitzHugh, R. (1955). "Mathematical models of threshold phenomena in the nerve membrane". In: *The Bulletin of Mathematical Biology* 17.4, pp. 257–278. DOI: `10.1007/BF02477753` (see pp. 14, 17).

— (1961). "Impulses and physiological states in theoretical models of nerve membrane". In: *Biophysical Journal* 1.6, pp. 445–466. DOI: `10.1007/s10867-010-9212-6` (see pp. 17, 37).

Foo, J. and G.E. Karniadakis (2010). "Multi-element probabilistic collocation method in high dimensions". In: *Journal of Computational Physics* 229.5, pp. 1536–1557. DOI: `10.1016/j.jcp.2009.10.043` (see p. 130).

Foo, J., X. Wan, and G.E. Karniadakis (2008). "The multi-element probabilistic collocation method (ME-PCM): Error analysis and applicatio". In: *Journal of Computational Physics* 227.22, pp. 9572–9595. DOI: `10.1016/j.jcp.2008.07.009` (see p. 130).

Frey, P.J (2004). "Generation and adaptation of computational surface meshes from discrete anatomical data". In: *Int. J. Numer. Meth. Eng.* 60.6, pp. 1049–1074. DOI: `10.1002/nme.992` (see p. 85).

Frey, P.J. and H. Borouchaki (1998). "Geometric surface mesh optimization". In: *Comput. Vis. Sci.* 1.3, pp. 113–121. DOI: `10.1007/s007910050011` (see p. 85).

— (2003). "Surface meshing using a geometric error estimate". In: *Int. J. Numer. Methods Engng.* 58.2, pp. 227–245. DOI: `10.1002/nme.766` (see p. 85).

Gao, Z. and J.S. Hesthaven (2010). "On ANOVA expansion and strategies for choosing the anchor point". In: *Applied Mathematica and Computations* 217.7, pp. 3274–3285. DOI: `10.1016/j.amc.2010.08.061` (see p. 185).

Gerardo-Giorda, L. and J.M. Kroos (2017). “Multiscale modeling and simulation of cortical spreading depression propagation”. In: *Computers and Mathematics with Applications* 74.5, pp. 1076–1090. DOI: `10.1016/j.camwa.2017.05.013` (see p. 50).

Gerstner, T. and M. Griebel (1998). “Numerical integration using sparse grids”. In: *Numerical Algorithms* 18, pp. 209–232. DOI: `10.1023/A:1019129717644` (see p. 181).

Geuzaine, C. and J.F. Remacle (2009). “Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities”. In: *Int. J. Numer. Meth. Engng.* 79, pp. 1309–1331. DOI: `10.1002/nme.2579` (see p. 85).

Ghanem, R.G. and P.D. Spanos (1991). *Stochastic Finite Elements: A Spectral Approach*. Springer New York. DOI: `10.1007/978-1-4612-3094-6` (see pp. 189, 190, 192).

Goddard, W. and O.R. Oellermann (2011). “Distance in Graphs”. In: *Structural Analysis of Complex Networks*. Ed. by M. Dehmer. Boston: Birkhäuser Boston, pp. 49–72. ISBN: 978-0-8176-4789-6. DOI: `10.1007/978-0-8176-4789-6_3` (see p. 169).

Goldman, R. (2005). “Curvature formulas for implicit curves and surfaces”. In: *Comput. Aided Geom. Des.* 22.7, pp. 632–658. DOI: `10.1016/j.cagd.2005.06.005` (see p. 87).

Grafstein, B. (1956). “Mechanism of spreading cortical depression”. In: *J. Neurophysiol.* 19.2, pp. 154–71. DOI: `10.1152/jn.1956.19.2.154` (see p. 9).

Greve, D.N. and B. Fischl (2009). “Accurate and robust brain image alignment using boundary-based registration”. In: *NeuroImage* 48.1, pp. 63–72. DOI: `10.1016/j.neuroimage.2009.06.060` (see p. 202).

Gronenschild, E.H.B.M. et al. (2012). “The effects of FreeSurfer version, workstation type, and Macintosh operating system version on anatomical volume and cortical thickness measurements”. In: *PLOS ONE* 7.6, e38234. DOI: `10.1371/journal.pone.0038234` (see p. 108).

Hadjikhani, N. et al. (2001). “Mechanisms of migraine aura revealed by functional MRI in human visual cortex”. In: *Proceedings of the National Academy of Science of the USA* 98.8, pp. 4687–92. DOI: `10.1073/pnas.071582498` (see pp. xiii, 1, 11).

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning*. 2nd ed. Springer-Verlag New York. DOI: `10.1007/978-0-387-84858-7` (see p. 123).

Hodgkin, A.L. and A.F. Huxley (1952). “A quantitative description of membrane current and its application to conduction and excitation in nerve”. In: *J. Physiol.* 117, pp. 500–544. DOI: `10.1113/jphysiol.1952.sp004764` (see pp. 14–16).

— (1953). “Propagation of electrical signals along giant nerve fibres”. In: *Proceedings of the Royal Society of London. Series B, Biological Sciences* 140.899, pp. 177–183. DOI: `10.1098/rspb.1952.0054` (see pp. 14, 15).

Huisman, T.A.G.M. (2010). “Diffusion-weighted and diffusion tensor imaging of the brain, made easy”. In: *Cancer Imaging* 10.1A, pp. 163–171. DOI: `10.1102/1470-7330.2010.9023` (see p. 54).

Huisman, T.A.G.M. et al. (2006). “Quantitative diffusion tensor MR imaging of the brain: field strength related variance of apparent diffusion coefficient (ADC) and fractional anisotropy (FA) scalars”. In: *European Radiology* 16.8, p. 1651. DOI: `10.1007/s00330-006-0175-8` (see p. 108).

Hyder, F. et al. (2013). “Glutamatergic function in the resting awake human brain is supported by uniformly high oxidative energy”. In: *J Cereb Blood Flow Metab.* 33.3, pp. 339–347. DOI: `10.1038/jcbfm.2012.207` (see p. 9).

Izhikevich, E.M. (2003). “Simple model of spiking neurons”. In: *IEEE Transaction on neural networks* 14.6, pp. 1569–1572. DOI: `10.1109/TNN.2003.820440` (see pp. 14, 19).

— (2004). “Which model to use for cortical spiking neurons?” In: *IEEE Transactions on Neural Networks* 15.5, pp. 1063–1070. DOI: `10.1109/TNN.2004.832719` (see p. 14).

— (2010). *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. The MIT Press. 528 pp. ISBN: 9780262514200. DOI: `10.7551/mitpress/2526.001.0001` (see pp. 14, 19).

Jenkinson, M. and S.M. Smith (2001). "A global optimisation method for robust affine registration of brain images". In: *Medical Image Analysis* 5.2, pp. 143–156. DOI: `10.1016/S1361-8415(01)00036-6` (see p. 202).

Jenkinson, M. et al. (2002). "Improved optimisation for the robust and accurate linear registration and motion correction of brain images". In: *NeuroImage* 17.2, pp. 825–841. DOI: `10.1006/nimg.2002.1132` (see p. 202).

Jenkinson, M. et al. (2012). "FSL". In: *NeuroImage* 62.2, pp. 782–790. DOI: `10.1016/j.neuroimage.2011.09.015` (see pp. 193, 200).

Jia, Z. and S. Yu (2017). "Grey matter alterations in migraine: A systematic review and meta-analysis." In: *Neuroimage Clin.* 14, pp. 130–140. DOI: `10.1016/j.nicl.2017.01.019` (see p. 12).

Jovicich, J. et al. (2009). "MRI-derived measurements of human subcortical, ventricular and intracranial brain volumes: Reliability effects of scan sessions, acquisition sequences, data analyses, scanner upgrade, scanner vendors and field strengths". In: *NeuroImage* 46.1, pp. 177–192. DOI: `10.1016/j.neuroimage.2009.02.010` (see p. 108).

Karhunen, K. (1947). *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Annales Academiae scientiarum Fennicae. Series A. 1, Mathematica-physica 37, pp. 1–79 (see p. 146).

Kellogg, O.D. (1929). "Foundations of Potential Theory. Die Grundlehren der Mathematischen Wissenschaften (In Einzeldarstellungen mit Besonderer Berücksichtigung der Anwendungsgebiete)". In: Springer, Berlin, Heidelberg. Chap. The Divergence Theorem, pp. 84–121. DOI: `10.1007/978-3-642-90850-7_4` (see p. 84).

Kingsely, P.B. (2006). "Introduction to diffusion tensor imaging mathematics: Part I. Tensor, rotation and eigenvectors". In: *Concepts in Magnetic Resonance Part A* 28A.2, pp. 101–122. DOI: `10.1002/cmr.a.20048` (see p. 54).

Kingsley, P.B. (2006). "Introduction to diffusion tensor imaging mathematics: Part II. Anisotropy, diffusion-weighting factors, and gradient encoding schemes". In: *Concepts in Magnetic Resonance Part A* 28 A.2, pp. 123–154. DOI: `10.1002/cmr.a.20049` (see p. 57).

Kneer, F., E. Schöll, and M.A. Dahlem (2014). "Nucleation of reaction-diffusion waves on curved surfaces". In: *New Journal of Physics* 16, p. 053010. DOI: `10.1088/1367-2630/16/5/053010` (see p. 105).

Koekoek, R. and R. Swarrouw (1998). *The Askey-scheme of hypergeometric orthogonal polynomials and its q-analogue*. Tech. rep. 98-17. Faculty of Technical Mathematics and Informatics, Delft University of Technology. URL: `http://homepage.tudelft.nl/11r49/askey.html` (see p. 170).

Kors, E.E. et al. (2004). "Recent findings on headache genetics". In: *Curr Opin Neurol* 17.3, pp. 283–288. DOI: `10.1097/01.wco.0000130295.54572.be` (see p. 11).

Kraio, R.P. and C. Nicholson (1978). "Extracellular ionic variations during spreading depression". In: *Neuroscience* 3.11, pp. 1045–1059. DOI: `10.1016/0306-4522(78)90122-7` (see pp. xv, 10, 13).

Kroos, J.M. et al. (2016). "Geometry shapes propagation: Assesssing the presence and absence of cortical symmetries through a computational model of cortical spreading depression". In: *Front. Comput. Neurosci.* 10.6. DOI: `10.3389/fncom.2016.00006` (see pp. 52, 77, 93).

Kroos, J.M. et al. (2017). "Patient-specific computational modeling of cortical spreading depression via diffusion tensor imaging". In: *Int J Numer Meth Biomed Engng* 33. DOI: `10.1002/cnm.2874` (see pp. 54, 77).

Kühnel, W. (2006). *Differential Geometry: Curves - Surfaces - Manifolds*. 2nd ed. Vol. 16. American Mathematical Society. DOI: `10.1007/978-3-658-00615-0` (see pp. 87, 164).

Lanczos, C. (1950). "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators". In: *J. Res. Natl. Bur. Stand. B* 45.4, pp. 255–282. DOI: `10.6028/jres.045.026` (see p. 70).

Langtangen, H.P. (2003). *Computational Partial Differential Equations*. 2nd ed. Springer-Verlag Berlin Heidelberg. DOI: `10.1007/978-3-642-55769-9` (see p. 41).

Lashley, K.S. (1941). "Patterns of cerebral integration indicated by the scotomas". In: *Arch NeurPsych* 46.2, pp. 331–339. DOI: `10.1001/archneurpsyc.1941.02280200137007` (see p. 11).

Lauritzen, M. (1994). "Pathophysiology of the migraine aura: The spreading depression theory". In: *Brain* 117.1, pp. 199–210. DOI: `10.1093/brain/117.1.199` (see pp. ix, xv, 11).

Leão, A.A.P. (1944). "Spreading depression of activity in the cerebral cortex". In: *J. Neurophysiol.* 7, pp. 391–396 (see pp. 1, 8).

— (1947). "Further observations on the spreading depression of activity in the cerebral cortex". In: *J. Neurophysiol.* 10.6, pp. 409–14. DOI: `10.1152/jn.1947.10.6.409` (see p. 8).

Le Bihan, D. (2003). "Looking into the functional architecture of the brain with diffusion MRI". In: *Nature Reviews Neuroscience* 4.2, pp. 469–480. DOI: `10.1038/nrn1119` (see p. 54).

— (2013). "Apparent diffusion coefficient and beyond: What diffusion MR imaging can tell us about tissue structure". In: *Radiology* 268.2, pp. 318–322. DOI: `10.1148/radiol.13130420` (see p. 54).

Le Bihan, D. et al. (2001). "Diffusion tensor imaging: Concepts and applications". In: *Journal of Magnetic Resonance Imaging* 13.4, pp. 534–546. DOI: `10.1002/jmri.1076` (see p. 57).

Lipton, R.B. et al. (2004). "Classification of primary headaches". In: *Neurology* 63.3, pp. 427–35. DOI: `10.1212/WNL.0000133301.66364.9B` (see p. 10).

Magid, E., O. Soldea, and E. Rivlin (2007). "A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data". In: *Computer Vision and Image Understanding* 107.3, pp. 139–159. DOI: `10.1016/j.cviu.2006.09.007` (see pp. xvi, 88).

Mahalanobis, P.C. (1936). "On the generalised distance in statistics". In: *Proceedings of the National Institute of Science of India*. URL: `http://library.isical.ac.in:8080/jspui/bitstream/123456789/6765/1/Vol02_1936_1_Art05-pcm.pdf` (see p. 103).

Marchandise, E., J.F. Remacle, and C. Geuzaine (2014). "Optimal parametrizations for surface remeshing". In: *Engineering with Computers* 30.3, pp. 383–402. DOI: `10.1007/s00366-012-0309-3` (see p. 87).

McNab, J.A. et al. (2013). "Surface based analysis of diffusion orientation for identifying architectonic domains in the in vivo human cortex". In: *Neuroimage* 69, pp. 87–100. DOI: `10.1016/j.neuroimage.2012.11.065` (see p. 51).

Meyer, M. et al. (2003). "Discrete differential-geometry operators for triangulated 2-manifolds". In: *Hege HC., Polthier K. (eds) Visualization and Mathematics III. Mathematics and Visualization, Springer, Berlin, Heidelberg*, pp. 35–57. DOI: `10.1007/978-3-662-05105-4_2` (see p. 88).

Nagumo, J. (1962). "An active pulse transmission line simulating nerve acon". In: *Proceedings of the IRE* 50.10, pp. 2061–2070. DOI: `10.1109/JRPROC.1962.288235` (see pp. 14, 17, 37).

Pearson, K. (1895). “Notes on regression and inheritance in the case of two parents”. In: *Proceedings of the Royal Society of London* 58, pp. 240–242. DOI: `10.1098/rspl.1895.0041` (see p. 97).

Peng, W. et al. (2012). “A fast algorithm to find all-pairs shortest paths in complex networks”. In: *Preocedia Computer Science* 9, pp. 557–566. DOI: `10.1016/j.procs.2012.04.060` (see p. 95).

Pérez-Beteta, J et al. (2018). “Tumor surface regularity at MR imaging predicts survival and response to surgery in patients with glioblastoma”. In: *Radiology* 288.1, pp. 218–225. DOI: `10.1148/radiol.201171051` (see pp. 79, 92).

Pietrobon, D. (2005). “Migraine: New molecular mechanisms”. In: *The Neuroscientist* 11.4, pp. 373–386. DOI: `10.1177/1073858405275554` (see pp. 11, 45).

Pietrobon, D. and M.A. Moskowitz (2014). “Chaos and commotion in the wake of cotical spreading depression and spreading depolarization”. In: *Nat. Rev. Neurosci.* 15.6, pp. 379–393. DOI: `10.1038/nrn3770` (see pp. xiii, 8–10).

Pocci, C. et al. (2010). “Numerical study of the stopping of aura during migraine”. In: *ESAIM: Proceedings* 30, pp. 44–52. DOI: `10.1051/proc/2010005` (see pp. vii, xiii, 2, 79).

Porooshani, H. et al. (2004). “Speed of progression of migrainous visual aura measured by sequential field assessment”. In: *Neuro-Ophthalmology* 28.2, pp. 101–105. DOI: `10.1076/noph.28.2.101.23739` (see p. 10).

Quarteroni, A., R. Sacco, and F. Saleri (2007). *Numerical Mathematics*. 2nd ed. Vol. 37. Texts in Applied Mathematics. Springer-Verlag Berlin Heidelberg. DOI: `10.1007/b98885` (see pp. 41, 69, 70, 159).

Quarteroni, A. and A. Valli (1994). *Numerical Approximation of Partial Differential Equations*. Springer-Verlag Berlin Heidelberg. DOI: `10.1007/978-3-540-85268-1` (see p. 40).

Reggia, J.A. and D. Montgomery (1996). “A computational model of visual hallucinations in migraine”. In: *Comput. Biol. Med.* 26.2, pp. 133–41. DOI: `10.1016/0010-4825(95)00051-8` (see p. 13).

Ribas, G.C. (2010). “The cerebral sulci and gyri”. In: *Neurosurgical Focus* 28.2. DOI: `10.3171/2009.11.FOCUS09245` (see p. 7).

Richter, F. and A. Lehmenkühler (2008). “Cortical spreading depression (CSD): A neurophysiological correlate of migraine aura”. In: *Der Schmerz* 22.5, pp. 544–50. DOI: `10.1007/s00482-008-0653-9` (see pp. xiii, 1).

Rodrigues, T.B., J. Valette, and A.K. Bouzier-Sore (2013). “13C NMR spectroscopy applications to brain energy metabolism”. In: *Front Neuroenergetics* 5.9. DOI: `10.3389/fnene.2013.00009` (see p. 6).

Rogers, J.M. and A.D. McCulloch (1994). “A collocation-Galerkin finite element model of cardiac action potential propagation”. In: *IEEE Trans Biomed Eng.* 41.8, pp. 743–757. DOI: `10.1109/10.310090` (see pp. xv, 14, 17, 37).

Ronan, L. et al. (2010). “Cortical curvature analysis in MRI-negative temporal lobe epilepsy: A surrogate marker for malformations of cortical development”. In: *Epilepsia* 52.1, pp. 28–34. DOI: `j.1528-1167.2010.02895.x` (see p. 79).

Rousseeuw, P. and B.C Van Zomeren (1990). “Unmasking multivariant outliers and leverage points”. In: *Journal of the American Statistical Association* 85.411, pp. 633–639. DOI: `10.1080/01621459.1990.10474920` (see p. 104).

Rousseeuw, P. and B.C. Van Zomeren (1991). “Robust distances: Simulations and cutoff values”. In: *Springer-Verlag New York* 34, pp. 195–203. DOI: `10.1007/978-1-4612-4444-8_11` (see p. 104).

Rousseeuw, P.J. and K. Van Driessen (1999). “A fast algorithm for the minimum covariance determinant estimator”. In: *Technometrics* 41.3, pp. 212–223. DOI: `10.1080/00401706.1999.10485670` (see p. 104).

Saad, Y. (2003). *Iterative methods for sparse linear systems*. 2nd ed. DOI: `10.1137/1.9780898718003` (see pp. 45, 70, 159).

Sadava, D. et al. (2006). *Life: The Science of Biology*. 8th ed. Sinauer Associates (see p. 2).

Sasaki, M. et al. (2008). “Variability in absolute apparent diffusion coefficient values across different platforms may be substantial: A multivendor, multi-institutional comparison study”. In: *Radiology* 249.2, pp. 624–630. DOI: `10.1148/radiol.2492071681` (see p. 108).

Sawat-Pokam, P.M. et al. (2016). “Mechanisms of neuronal silencing after cortical spreding depression”. In: *Cereb. Cortex*. DOI: `10.1093/cercor/bhv328` (see pp. 2, 9, 10, 45).

Schoutens, W. (2000). *Stochastic Processes and Orthogonal Polynomials*. Springer New York. DOI: `10.1007/978-1-4612-1170-9` (see p. 170).

Semenova, I.B., V.V. Savchenko, and I. Hagiwara (2004). “Two techniques to improve mesh quality and preserve surface characteristics”. In: *13th International Meshing Roundtable*, pp. 277–288 (see p. 80).

Shewchuk, J. R. (2002). “What is a good linear element? Interpolation, conditioning, and quality measures”. In: *11th International Meshing Roundtable*, pp. 115–126 (see p. 82).

Silberstein, S.D. (2004). “Migraine”. In: *The Lancet* 363.9406, pp. 381–391. DOI: `10.1016/S0140-6736(04)15440-8` (see p. 10).

Smith, S.M. (2002). “Fast robust automated brain extraction”. In: *Human Brain Mapping* 17.3, pp. 143–155. DOI: `10.1002/hbm.10062` (see p. 201).

Smith, S.M. et al. (2004). “Advances in functional and structural MR image analysis and implementation as FSL”. In: *NeuroImage* 23, pp. 208–219. DOI: `10.1016/j.neuroimage.2004.07.051` (see p. 193).

Smolyak, S.A. (1963). “Quadrature and interpolation formulas for tensor products of certain classes of functions”. In: *Soviet Mathematics Doklady* 4, pp. 240–243 (see p. 181).

Soares, J.M. et al. (2013). “A hitchhiker’s guide to diffusion tensor imaging”. In: *Front Neurosci.* 7.31. DOI: `10.3389/fnins.2013.00031` (see p. 55).

Sobol’, I.M. (1993). “Sensitivity estimates for nonlinear mathematical models”. In: *Math. Model. Comput. Exp.* 1, pp. 407–417. DOI: `1061-7590/93/04407-008` (see p. 141).

— (2001). “Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates”. In: *Mathematics and Computers in Simulation* 55.1-3, pp. 271–280. DOI: `10.1016/S0378-4754(00)00270-6` (see pp. 141, 187).

Somjen, G.G. (2001). “Mechanisms of spreading depression and hypxic spreading depression-like depolarisation”. In: *Physiol. Rev.* 81.3, pp. 1065–96. DOI: `PMID:11427692` (see pp. 9, 10).

Souza, R. et al. (2018). “An open, multi-vendor, multi-field-strength brain MR dataset and analysis of publicly available skull stripping methods agreement”. In: *NeuroImage* 170, pp. 482–494. DOI: `10.1016/j.neuroimage.2017.08.021` (see p. 108).

Steward, W.F. et al. (2008). “Cumulative lifetime migraine incidence in women and men”. In: *Cephalalgia* 28.11, pp. 1170–1178. DOI: `10.1111/j.1468-2982.2008.01666.x` (see p. 10).

Su, C.H. and D. Lucor (2006). “Covariance kernel representations of multidimensional second-order stochastic processes”. In: *J. Comput. Phys.* 217.1, pp. 82–99. DOI: `10.1016/j.jcp.2006.02.006` (see p. 189).

Tang, K., P.M. Congedo, and R. Abgrall (2015). “Sensitivity analysis using anchored ANOVA expansion and high-order moments computation”. In: *Int. J. Numer. Meth. Engng.* 102.9, pp. 1554–1584. DOI: `10.1002/nme.4856` (see p. 183).

Taubin, G. (1995). “Curve and surface smoothing without shrinkage”. In: *Proceedings of the Fifth International Conference on Computer Vision*. ICCV ’95, pp. 852–857. DOI: `10.1109/ICCV.1995.466848` (see pp. 81, 83).

— (2000). “Geometric signal processing on polygonal meshes”. In: *Eurographics 2000 – STARs*. ISSN: 1017-4656. DOI: `10.2312/egst.20001029` (see p. 81).

Taubin, G., T. Zhang, and G. Golub (1996). “Optimal surface smoothing as filter design”. In: *Computer Vision – ECCV ’96* 1064, pp. 283–292. DOI: `10.1007/BFb0015544` (see p. 85).

Thompson, P.M. et al. (1996). “Three-dimensional statistical analysis of sulcal variability in the human brain”. In: *Journal of Neuroscience* 16.13, pp. 4261–4274. DOI: `0.1523/JNEUROSCI.16-13-04261.1996` (see p. 79).

Toga, A.W. and P.M. Thompson (2003). “Mapping brain asymmetry”. In: *Nat Rev Neurosci.* 4.1, pp. 37–48. DOI: `10.1038/nrn1009` (see p. 105).

Tommaso, M. de et al. (2014). “Altered processing of sensory stimuli in patients with migraine”. In: *Nat. Rev. Neurol.* 10.3, pp. 144–55. DOI: `10.1038/nrneurol.2014.14` (see p. 11).

Tuch, D.S. et al. (1999). “Conductivity mapping of biological tissue using diffusion MRI”. In: *Annals New York Academy of Science* 888, pp. 314–316. DOI: `10.1111/j.1749-6632.1999.tb07965.x` (see p. 54).

— (2001). “Conductivity tensor mapping of the human brain using diffusion tensor MRI”. In: *Proceedings of the National Academy of Sciences* 98.20, pp. 11697–11701. DOI: `10.1073/pnas.171473898` (see p. 54).

Tuckwell, H.C. and R.M. Miura (1978). “A mathematical model for spreading cortical depression”. In: *Biophys. J.* 23.2, pp. 257–276. DOI: `10.1016/S0006-3495(78)85447-2` (see p. 13).

Vos, T. et al. (2012). “Years lived with disability (YLDs) for 1160 sequelae of 289 diseases and injuries 1990-2010: a systematic analysis for the Global Burden of Disease Study 2010”. In: *The Lancet* 380.9859, pp. 2163–2196. DOI: `10.1016/S0140-6736(12)61729-2` (see p. 1).

Walz, W. (2000). “Role of astrocytes in the clearance of exess extracellular potassium”. In: *Neurochemistry International* 36.4-5, pp. 291–300. DOI: `10.1016/S0197-0186(99)00137-0` (see p. 6).

Wei, Y., G. Ullah, and S.J. Schiff (2014). “Unification of neuronal spikes, seizures, and spreading depression”. In: *J. Neurosci.* 34.35, pp. 11733–11743. DOI: `10.1523/JNEUROSCI.0516-14.2014` (see pp. xv, 14, 23, 26, 28, 30, 33–35, 50).

Welch, B.L. (1947). “The generalization of ’student’s’ problem when several different population variances are involved”. In: *Biometrika* 34.1-2, pp. 28–35. DOI: `10.1093/biomet/34.1-2.28` (see p. 112).

WHO (2017). “World Health Report”. In: URL: `www.who.int/whr/index.htm` (see p. 10).

Woolrich, M.W. et al. (2009). “Bayesian analysis of neuroimaging data in FSL”. In: *NeuroImage* 45.1, pp. 173–186. DOI: `10.1016/j.neuroimage.2008.10.055` (see p. 193).

Xiu, D. (2010). *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press. DOI: `10.2307/j.ctv7h0skv` (see pp. 128, 174).

Xiu, D. and G.E. Karniadakis (2002). “The Wiener–Askey polynomial chaos for stochastic differential equations”. In: *SIAM J. Sci. Comput.* 24.2, pp. 619–644. DOI: `10.1137/S1064827501387826` (see p. 169).

Xiu, D.B. and J.S. Hesthaven (2005). “High-order collocation methods for differential equations with random inputs”. In: *SIAM Journal on Scientific Computing* 27.3, pp. 1118–1139. DOI: `10.1137/040615201` (see pp. 137, 181).

Yang, X. et al. (2012). “Adaptive ANOVA decomposition of stochastic incompressable and compressible flows”. In: *J. Comput. Phys.* 231.4, pp. 1587–1614. DOI: `10.1016/j.jcp.2011.10.028` (see p. 186).

Yendiki, A. et al. (2011). "Automated probabilistic reconstruction of white-matter pathways in health and disease using an atlas of the underlying anatomy". In: *Front. Neuroinform.* 5.23. DOI: `10.3389/fninf.2011.00023` (see p. 193).

Yendiki, A. et al. (2013). "Spurious group differences due to head motion in a diffusion MRI study". In: *NeuroImage* 88, pp. 79–90. DOI: `10.1016/j.neuroimage.2013.11.027` (see p. 193).

Zandt, B., B. ten Haken, and M.J.A.M van Putten (2013). "Diffusing substances during spreading depolarisation: Analytic expression for the propagation speed, triggering and concentration time courses". In: *Journal of Neuroscience* 33.14, pp. 5915–5923. DOI: `10.1523/JNEUROSCI.5115-12.2013` (see pp. xiii, 8).

Zandt, B. et al. (2015). "How does Spreading depression spread? - Physiology and modeling". In: *Rev. Neurosci.* 22.2, pp. 183–98. ISSN: 2191-0200. DOI: `10.1515/revneuro-2014-0069` (see pp. xiii, xv, 2, 9, 10, 14, 33, 37).

Acknowledgements

"Es gibt zwei Arten sein Leben zu leben: entweder so, als wäre nichts ein Wunder, oder so, als wäre alles eines. Ich glaube an Letzteres."

Albert Einstein

I would like to express my gratitude to all those who have supported me during my graduate years in making this work possible and these years unforgettable. My special thanks goes to:

My supervisor Dr. Luca Gerado-Giorda for introducing me to the thrilling field of computational neuroscience, for your guidance, patience and support during the last years.

George Karniadakis for hosting me at the Brown University and introducing me to the exciting topic of uncertainty quantification.

Franco Dassi for interesting discussions on mesh improvements and curvature estimations, Ibai Diez and Iñigo Gabilondo for their support in the data processing with the FreeSurfer Analysis suit, and Ivi Tsantili for inspiring discussions on the ANOVA approach and sensitivity analysis.

Silvia Daun, Gemma Huguet and Fabio Cavaliere for serving as members of my thesis committee.

Gabriela, Daniel, Martina and Ralf for their valuable comments on the thesis manuscript, and Daniel and Carmen-Ana for supporting me with the Basque and Spanish passages. I am especially grateful for Nicole supporting me in the final phase of my thesis not only with the manuscript revision but also with valuable scientific and academic advice.

The MMB group members, Ana, Argyrios, Gabriela, Isabella, Martina, and Nicole for always being available for friendly chats, either technical or non-technical.

Dedicated staff of BCAM, especially Miguel Benítez, for being available and resolving all the administrative issues appearing in my way, concerning the thesis and every other administrative aspect of living in a foreign country.

All present and former members of BCAM, especially Carmen-Ana, Marina, Inderpreet, Aitziber, Diana, Simone, Fabio, Mario, Ariel, Thomas, Biagio, Umberto, Antsa, Stefano, Daniel, Javi and David for many coffee breaks, lunches, and after work drinks, interesting discussions, and for creating such a pleasant team atmosphere, at work as well as in the bars and restaurants of Bilbao.

My office mates Mario, Simone, Tijana, Ander and Tamara for providing such a pleasant working atmosphere.

My second little family in the Bowen Street apartment in Providence: Antoine, Cecilia, Esben, Olga and Seok. For all those nice dinners, after-work wines and my very first scientific boat race.

Katie for being at my side through all those numerous Spanish classes, for the encouragement, the support and for making me feel at home away from home. Andrew for being my gym, beach and party buddy through all those years. Raúl for being the best tandem partner, not only helping me patiently to learn Spanish but also introducing me to the Basque culture. Raúl for the excursions around Bilbao and the Basque country. Mila esker!

Kristina, Sarah and Theresa for always being there for me, numerous phone calls, and visits all over Europe. Insbesondere für die Unterstützung, aufmunternden Worte und alternativen Sichtweisen in schwierigen Lebenslagen. Bis die Wolken wieder lila werden.

My brother Johannes, my mum Gerlinde and my dad Georg, for their unconditional love, patience, and advice in every situation and supporting me in every way possible not matter what. Ohne euch hätte ich das niemals geschafft. Vielen Dank!

Julia M. Kroos
Bilbao, May 2019