
PROSJEKT 5

Ei litt titt på pre- og postsynaptisk membran

Kandidat 72

Dato: 20. november 2015

Sammendrag

Introduksjon

Overgangen mellom to nerveceller kalles en synapse. For at det skal kunne overføres signaler fra den ene cella til den andre så må “noe” diffundere gjennom synapsen, dette “noe” er signalmolekyler som kalles nevrotransmittere. Disse molekylene kommer i en blokk som vil koble seg på den presynaptiske membranen, og så slippe molekylene fri slik at de kan diffundere over den synaptiske kløfta og komme fram til den postsynaptiske membranen der de vil bli tatt i mot av reseptorer som bringer signalet videre i cella.

For å modellere dette kan vi bruke diffusjonslikninga i én dimensjon,

$$D \frac{\partial^2 u(x, t)}{\partial x^2} = \frac{\partial u(x, t)}{\partial t}.$$

Hvor vi da har antatt visse ting om systemet vårt: den synaptiske kløfta er jevn og like bred overalt, og at konsentrasjonen av nevrotransmittere varierer kun på langs i bevegelsesretninga.

Vi skal løse denne likninga både analytisk og numerisk slik at vi kan teste programmet på høvelig vis og sammenlikne de ulike metodene. De tre numeriske metodene som testes ut er ekplisitt forover-Euler, implisitt bakover-Euler, og Crank-Nicholson-metoden. Resultatene fra disse metodene vil testes for forskjellige parametre og vi vil finne ut hvilken som gir best resultat.

Systemet beskrevet ovenfor kan også simuleres med Monte Carlo-metoder og tilfeldig gange. Vi vil bruke en algoritme som baserer seg på at nevrotransmitterne kan hoppe fram og tilbake og så sammenlikne resultatene fra denne metoden med de fra diffusjonslikningssimuleringa.

Teori

Å løse systemet analytisk er essensielt for å kunne sjekke at våre numeriske resultater stemmer. Vi setter diffusjonskonstanten lik 1 slik at vi får en dimensjonsløslikning. Diffusjonslikninga er en separabel likning, slik at vi kan skrive den

$$\begin{aligned} \mathcal{T}(t) \frac{\partial^2 \mathcal{X}(x)}{\partial x^2} &= \mathcal{X}(x) \frac{\partial \mathcal{T}(t)}{\partial t} \\ \Rightarrow \frac{1}{\mathcal{X}(x)} \frac{\partial^2 \mathcal{X}(x)}{\partial x^2} &= -\lambda^2 \quad \frac{1}{\mathcal{T}(t)} \frac{\partial \mathcal{T}(t)}{\partial t} = -\lambda^2, \end{aligned}$$

hvor λ er en konstant vi snart skal finne. Det viser seg imidlertid at det lønner seg å skrive løsninga som

$$u(x, t) = \mathcal{X}(x) \mathcal{T}(t) + u_s(x),$$

hvor $u_s(x)$ er likevektstilstanden løsninga vår skal konvergere mot. Første leddet i løsninga vil gå mot null ettersom tida øker, så vi står igjen med likevektsleddet som er $u_s(x) = 1 - x$. Vi finner så $\mathcal{X}(x)$,

$$\begin{aligned} \frac{\partial^2 \mathcal{X}(x)}{\partial x^2} &= -\lambda^2 \mathcal{X}(x) \\ \Rightarrow \mathcal{X}(x) &= A \sin(\lambda x) + B \cos(\lambda x). \end{aligned}$$

Tidsdelen av løsninga finner vi ved å løse,

$$\begin{aligned}\frac{\partial \mathcal{T}(t)}{\partial t} &= -\lambda^2 \mathcal{T}(t) \\ \Rightarrow \mathcal{T}(t) &= C e^{-\lambda^2 t}.\end{aligned}$$

Dette gir oss

$$u(x, t) = A' (\sin(\lambda x) + B \cos(\lambda x)) e^{-\lambda^2 t} + 1 - x,$$

hvor vi slo sammen de to integreringskonstantene fra den romlige og tids-avhengige løsninga, til A' . Initialbetingelsene er $u(x = d = 1, \text{alle } t) = 0$, $u(x = 0, t > 0) = u_0 = 1$ som gir oss

$$u(x, t) = A' \sin\left(\frac{2\pi x}{d}\right) e^{-\lambda^2 t} + 1 - x.$$

Denne løsninga har mange moduser, slik at vi må skrive den som en sum av alle modene,

$$u(x, t) = \sum_{n=1}^{\infty} \left(A' \sin\left(\frac{2\pi n x}{d}\right) \right) e^{-\lambda^2 t} + 1 - x. \quad \square$$

De tre metodene Som allerede nevnt skal vi løse diffusjonslikninga numerisk ved hjelp av tre forskjellige metoder, vi ser først på den eksplisitte forover-Euler-metoden.

$$u_t \approx \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t}$$

og

$$u_{xx} \approx \frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j))}{\Delta x^2} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2},$$

hvor vi har diskretisert tid og rom, $i, j = 0, 1, \dots$. Vi husker hvordan diffusjonslikninga ser ut og innser at vi kan skrive

$$u_{i,j+1} = u_{i,j} + \alpha \left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \right),$$

hvor $\alpha = \Delta t / h^2$. Vi kan så definere en vektor som inneholder alle funksjonsverdiene til u ,

$$V_j^T = [u_{0,j}, \dots, u_{n,j}].$$

Denne vektoren trenger noen venner, nemlig matriser som sier hvordan vi skal behandle funksjonsverdiene. Vi setter $\hat{A} \equiv \mathbb{1} + \alpha \hat{B}$, hvor vi har,

$$\begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & 0 & \vdots \\ 0 & \ddots & \ddots & \ddots & -1 \\ \vdots & \dots & \dots & -1 & 2 \end{pmatrix}. \quad (1)$$

Vi kan da skrive diffusjonslikninga på matriseform: $\hat{V}_{j+1} = \hat{A} \hat{V}_j$, akkurat som i prosjekt 1. Dette kan vi enkelt implementere i programmet vårt som en dobbel

for-løkke som går over tid og rom. Vi vil da ikke definere hele matriser, siden \hat{B} er tridiagonal så kan vi operere med vektorer og gange riktige elementer sammen slik at vi slipper å sløse FLOPS ved å gange med null mange ganger.

Vi kan så se på implisitt bakover-Euler-metoden. Til forskjell fra den eksplisitte metoden over, så bruker vi her det forrige tidssteget til å finne det nye. Vi har da,

$$u_t \approx \frac{u(x_i, t_j) - u(x_i, t_j - \Delta t)}{\Delta t} = \frac{u_{i,j} - u_{i,j-1}}{\Delta t},$$

mens u_{xx} er den samme som i stad. Dette skriver vi så sammen til å være

$$u_{i,j-1} = u_{i,j} - \alpha \left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \right).$$

På matrisiform får vi da $\hat{C} = \mathbb{1} - \alpha \hat{B}$ og dermed $\hat{C} \hat{V}_j = V_{j-1} \Rightarrow \hat{V}_{j+1} = \hat{C}^{-1} \hat{V}_j$. Vi kan også her bruke vår egenproduserte tridiagonale regnemaskin fra prosjekt 1.

Siste metode ut er den implisitte Crank-Nicolson-metoden. **Vi har et tidssentrert skjema???**, $(x, t + \Delta t/2)$

$$u_t \approx \frac{u(x_i, t_j + \Delta t) - u(x_i, t_j)}{\Delta t} = \frac{u_{i,j+1} - u_{i,j}}{\Delta t}.$$

Den andreordens deriverte ser nå slik ut,

$$\begin{aligned} u_{xx} &\approx \frac{1}{2} \left(\frac{u(x_i + \Delta x, t_j) - 2u(x_i, t_j) + u(x_i - \Delta x, t_j)}{\Delta x^2} + \right. \\ &\quad \left. \frac{u(x_i + \Delta x, t_j + \Delta t) - 2u(x_i, t_j + \Delta t) + u(x_i - \Delta x, t_j + \Delta t)}{\Delta x^2} \right) \\ &= \frac{1}{2} \left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}}{\Delta x^2} \right). \end{aligned}$$

Vi setter dette så sammen og får,

$$\begin{aligned} 2u_{i,j+1} - 2u_{i,j} &= \alpha \left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \right) + \alpha \left(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1} \right) \\ 2u_{i,j+1} - \alpha \left(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1} \right) &= 2u_{i,j} + \alpha \left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \right), \end{aligned}$$

Vi kjenner igjen høyre- og venstresidene som det vi fikk fra de henholdsvis eksplisitte og implisitte Euler-metodene, slik at vi kan skrive

$$(2\mathbb{1} + \alpha \hat{B}) \hat{V}_j = (2\mathbb{1} - \alpha \hat{B}) \hat{V}_{j-1},$$

hvor faktoren 2 kommer fra den halve vi ganget u_{xx} med. For å løse dette skriver vi

$$\hat{V}_j = (2\mathbb{1} + \alpha \hat{B})^{-1} (2\mathbb{1} - \alpha \hat{B}) \hat{V}_{j-1}.$$

Vi har nå det meste klart for å implementere disse metodene i programmet vårt. Men før vi gjør det så er det lurt å se litt nærmere på hvilke verdier av Δt og Δx som gir oss stabile løsninger. Spektralradien til en matrise er definert som

$$\rho(\hat{A}) = \max \{ |\lambda| : \det(\hat{A} - \lambda \mathbb{1}) = 0 \},$$

altså absoluttverdien til den største egenverdien. Dette kan tolkes som radien til den minste sirkelen i \mathbb{C}^2 , med senter i origo, som omslutter alle egenverdiene

til matrisa \hat{A} . Hvis vi har $\rho(\hat{A}) < 1$, så vil løsningen vår *alltid* konvergere mot en stabil løsning. Hvis matrisa \hat{A} er positiv definitt så er dette kravet alltid tilfredsstilt. Så for våre tre metoder og de tilhørende tre matriselikninger, så kan vi sjekke om dette kravet er tilfredsstilt.

Først eksplisitt forover-Euler, da har vi $\hat{A} = \mathbb{1} + \alpha \hat{B}$. Egenverdiene til identitetsmatrisa er 1, mens vi for \hat{B} bør faktisk regne det ut så vi er helt sikre (selv om vi vet at en **reell** tridiagonal matrise er positiv definitt). Vi kan skrive

$$b_{i,j} = 2\delta_{i,j} - \delta_{i+1,j} - \delta_{i-1,j}.$$

Eigenverdilikninga er gitt som

$$\hat{B}\hat{v} = \lambda\hat{v}$$

$$\begin{aligned}\Rightarrow (\hat{B}\hat{v})_i &= \lambda_i \hat{v}_i \\ &= \sum_{j=1}^n (2\delta_{i,j} - \delta_{i+1,j} - \delta_{i-1,j}) v_j \\ &= 2v_i - v_{i+1} - v_{i-1} = \lambda_i v_i.\end{aligned}$$

Vi kan så velge en basis, $\sin(i\theta)$, å uttrykke egenvektorene i slik at vi får

$$2\sin(i\theta) - \sin(i+1\theta) - \sin(i-1\theta) = \lambda_i \sin(i\theta).$$

Bruker vi så identiteten $\sin(x+y) + \sin(x-y) = 2\cos(x)\sin(y)$ så kan vi forenkle uttrykket over til,

$$\begin{aligned}2(1 - \cos(i\theta)) \sin(i\theta) &= \lambda_i \sin(i\theta) \\ \lambda_i &= 2(1 - \cos(i\theta)).\end{aligned}$$

Eigenverdiene for \hat{A} blir da $\Gamma = 1 - 2\alpha(1 - \cos(i\theta))$

$$\Rightarrow -1 < 1 - 2\alpha(1 - \cos(i\theta)) < 1 \Rightarrow \alpha < \frac{1}{2},$$

hvilket impliserer at vi har kravet,

$$\frac{\Delta t}{\Delta x^2} < \frac{1}{2},$$

som betyr at vi ikke kan velge tids- og lengdesteg som vi vil og fremdeles få et konvergerende resultat.

Ser vi derimot på det implisitte skjemaet så skal vi finne eigenverdiene til $\hat{C} = \mathbb{1} + \alpha \hat{B}$, som ved samme utregning som over gir,

$$\lambda_i = 1 + 2\alpha(1 - \cos(i\theta)).$$

Å nei, vi har en egenverdi som alltid er større enn 1! Frykt ei, det implisitte skjemaet finner neste tidssteg ved $V_j = \hat{C}^{-1} V_{j-1}$, vi må altså ha at $(1/\lambda_i) < 1$, som er tilfredsstilt her for alle verdier av Δx og Δt . Hurra!

For Crank-Nicolson blir eigenverdiene $\gamma = 1 - 2\alpha(1 - \cos(i\theta)) / (1 + 2\alpha(1 - \cos(i\theta)))$, som gir oss,

$$\begin{aligned}1 &> \frac{1 - 2\alpha(1 - \cos(i\theta))}{1 + 2\alpha(1 - \cos(i\theta))} \\ 1 - 2\alpha(1 - \cos(i\theta)) &> 1 + 2\alpha(1 - \cos(i\theta)) \\ 0 &< 2\alpha[(1 - \cos(i\theta)) + (1 + \cos(i\theta))] \\ &< 4\alpha,\end{aligned}$$

som åpenbart alltid er tilfredsstilt.

Metode

Numerisk stabilitet og presisjon

For å forstå mer av hvordan programmet vårt oppfører seg og hvor presise resultat det gir, så kan vi se på trunkeringsfeil og hvordan valg av tids- og lengdesteg virker inn på resultatene. Vi finner først trunkeringsfeilene for eksplisitt- og implisitt-Euler-metode, samt Crank-Nicolson. For å gjøre det så må vi Taylorutvikle $u(x - \Delta x, t)$, $u(x + \Delta x, t)$ og $u(x, t + \Delta t)$, sette sammen uttrykkene slik at vi får formlene for metodene vi bruker, og så lese av trunkeringsfeilen i leddet med $\mathcal{O}(sth)$

$$\begin{aligned}u(x + \Delta x, t) &= u(x, t) + \frac{\partial u(x, t)}{\partial x} \Delta x + \frac{\partial^2 u(x, t)}{2\partial x^2} \Delta x^2 + \mathcal{O}(\Delta x^3), \\u(x - \Delta x, t) &= u(x, t) - \frac{\partial u(x, t)}{\partial x} \Delta x + \frac{\partial^2 u(x, t)}{2\partial x^2} \Delta x^2 + \mathcal{O}(\Delta x^3), \\u(x, t + \Delta t) &= u(x, t) + \frac{\partial u(x, t)}{\partial t} \Delta t + \mathcal{O}(\Delta t^2).\end{aligned}\tag{2}$$

Figur 1

Resultat

Konklusjon

Kommentarer til prosjektet