Ciências ULisboa
Faculdade de Ciências da Universidade de Lisboa

BIOELECTRICIDADE E ELECTROFISIOLOGIA

Resposta de um axónio não mielinizado a um campo elétrico aplicado

21 de Dezembro de 2016

David Leitão, n°45301
João Oliveira, n°45306

Prof. Pedro Cavaleiro Miranda

Conteúdo

Sumário

Neste trabalho, numa primeira parte, foi implementado, com recurso ao software `Matlab v2013a`, o modelo de Hodgkin-Huxley. Foram simuladas posteriormente experiências *space-clamped*, visualizando-se a propagação temporal de potenciais de ação, as correntes associadas à membrana, as condutâncias dos diferentes iões e as probabilidades das diferentes subunidades estarem abertas. Ainda se demonstrou o efeito *anode break excitation* e calculou-se a intensidade limiar de geração de potenciais de ação para um estímulo de duração $1\,ms$.

Numa segunda parte foram implementadas as equações do cabo e simulou-se a propagação temporal e espacial de um potencial de ação, visualizando-se a evolução do potencial da membrana, da corrente axial intracelular, transmembranar, de potássio e de sódio. Determinou-se também o limiar da corrente de estimulação para um estímulo de duração $100\,\mu s$. Foi ainda analisada a influência do raio da fibra na velocidade de propagação de um potencial de ação.

1 Introdução

A membrana de células excitáveis contêm vários canais e bombas na sua estrutura que permitem o fluxo de iões entre os meios intracelular e extracelular. Este fluxo é regulado pelos gradientes de concentração e pelo potencial de membrana, definido como a diferença de potencial entre os meios intracelular e extracelular: $V_m = \Phi_i - \Phi_e$. A membrana apresenta então características resistivas e capacitivas, para além de gerarem correntes, o que é descrito pelo modelo matemático de Hodgkin-Huxley. Segundo o modelo de Hodgkin-Huxley os fluxos de cada ião são independentes, sendo apenas consideradas as correntes devido ao sódio e ao potássio. A corrente de cada um destes iões é dada por:

$$I_k\left(t,v_m\right) = g_K\left(t,v_m\right)\left[V_m\left(t\right) - E_K\right] \tag{1}$$
$$I_{Na}\left(t,v_m\right) = g_{Na}\left(t,v_m\right)\left[V_m\left(t\right) - E_{Na}\right] \tag{2}$$

Onde E_K e E_{Na} são os potenciais de Nernst para o potássio e sódio, respetivamente, e g_K e g_{Na} são as condutâncias específicas para o potássio e sódio, respetivamente. Estas condutâncias são dadas por:

$$g_K\left(t,v_m\right) = \bar{g}_K\, n^4\left(t,v_m\right) \tag{3}$$
$$g_{Na}\left(t,v_m\right) = \bar{g}_{Na}\, m^3\left(t,v_m\right)\, h\left(t,v_m\right) \tag{4}$$

Onde v_m é o potencial reduzido da membrana, $v_m = V_m - V_r$ ($V_r = -60\,mV$ é o potencial de repouso da membrana), $\bar{g}_K$ e $\bar{g}_{Na}$ são as condutâncias específicas máximas para o potássio e o sódio, respetivamente, e n, m e h definem a probabilidade das subunidades que constituem os canais estarem abertas. Os canais de potássio são então constituídos por quatro subunidades n, e os de sódio por três subunidades m e uma h, sendo que para um canal estar aberto todas as suas subunidades também têm de o estar. A sua dinâmica é dada assumindo uma cinética de primeira ordem:

$$\frac{dn\left(t,v_m\right)}{dt} = \alpha_n\left(v_m\right)\left(1-n\right) - \beta_n\left(v_m\right)\left(n\right) \tag{5}$$
$$\frac{dm\left(t,v_m\right)}{dt} = \alpha_m\left(v_m\right)\left(1-m\right) - \beta_m\left(v_m\right)\left(m\right) \tag{6}$$
$$\frac{dh\left(t,v_m\right)}{dt} = \alpha_h\left(v_m\right)\left(1-h\right) - \beta_h\left(v_m\right)\left(h\right) \tag{7}$$

Por sua vez, as taxas α e β apenas dependem de v_m e são dadas por:

$$\begin{cases} \alpha_n\left(v_m\right) = \frac{0.01(10-v_m)}{\left[\exp\left(\frac{10-v_m}{10}\right)-1\right]} & \text{se } v_m \neq 10 \\ \alpha_n\left(v_m\right) = 0.1 & \text{se } v_m = 10 \end{cases} \tag{8}$$

$$\beta_n\left(v_m\right) = 0.125\,\exp\left(-\frac{v_m}{80}\right) \tag{11}$$

$$\begin{cases} \alpha_m\left(v_m\right) = \frac{0.1(25-v_m)}{\left[\exp\left(\frac{25-v_m}{10}\right)-1\right]} & \text{se } v_m \neq 25 \\ \alpha_m\left(v_m\right) = 1 & \text{se } v_m = 25 \end{cases} \tag{9}$$

$$\beta_m\left(v_m\right) = 4\,\exp\left(-\frac{v_m}{18}\right) \tag{12}$$

$$\alpha_h\left(v_m\right) = 0.07\,\exp\left(-\frac{v_m}{20}\right) \tag{10}$$

$$\beta_h\left(v_m\right) = \frac{1}{\left[\exp\left(\frac{30-v_m}{10}\right)+1\right]} \tag{13}$$

O modelo de Hodgkin-Huxley reconhece a existência de outras correntes iónicas, definindo por isso ainda um terceira corrente iónica, a *leakage current*:

$$I_L\left(t,v_m\right) = g_L\left(t,v_m\right)\left[V_m\left(t\right) - E_L\right] \tag{14}$$

Onde g_L e E_L são os análogos das equações 1 e 2, sendo ajustados de forma a se obter os resultados observados para V_m durante o repouso.

A evolução temporal de V_m através do modelo de Hodgkin-Huxley é obtida considerando que corrente membranar I_m é dada por:

$$I_m = I_s = I_K + I_{Na} + I_L + I_C \tag{15}$$

Onde I_s é a corrente de estimulação injetada no meio intracelular e I_C a corrente devido às características capacitivas da membrana. Reescrevendo $I_{ion} = I_K + I_{Na} + I_L$, em que I_{ion} é a corrente iónica, e $I_C = C_m \frac{dV_m}{dt}$, onde C_m é a capacitância da membrana, vem:

$$\frac{dV_m}{dt} = \frac{I_s - I_{ion}}{C_m} = \frac{I_C}{C_m} \tag{16}$$

No entanto o modelo de Hodgkin-Huxley não permite descrever a evolução espacial de um potencial de ação. Tal é possível com o modelo do núcleo condutor, cujas equações do cabo resultam da aplicação das leis de Kirchhoff e de Ohm. Neste modelo a membrana é modelada através de, por exemplo, o modelo de Hodgkin-Huxley, e os meios intracelular e extracelular pelas suas resistências por unidade de comprimento, r_i e r_e respetivamente. Vem então que a correntes axial intracelular e transmembranar são dadas respetivamente por:

$$I_i = -\frac{1}{r_i + r_e}\frac{\partial V_m}{\partial x} \tag{17}$$

$$I_m = \frac{1}{2\pi a\,(r_i + r_e)}\left[\frac{\partial^2 V_m}{\partial x^2} - r_e i_p\right] + I_s \tag{18}$$

Onde a é o raio da fibra e $i_p = 2\pi a\, I_p$ é a corrente de estimulação injectada no meio extracelular.

Numa primeira parte foi implementada as equações de Hodgkin-Huxley para simular uma experiência *space clamped*, e numa segunda parte foram implementadas as equações do cabo para simular a propagação temporal e espacial de um potencial de ação. O relatório apresenta de forma separada os métodos e resultados obtidos para as duas partes do trabalho, mas a discussão está conjunta.

2 Materiais e Métodos

2.1 Materiais

Para a implementação deste projecto foi utilizado o software `Matlab v2013a`, e o código desenvolvido encontra-se no anexo 5.2.

2.2 Métodos

2.2.1 Parte 1 - Modelo Hodgkin-Huxley

Para a primeira parte foi utilizado o seguinte procedimento que está implementado no script *main_1.m* (anexo 5.2):

a) Com base nas equações 8 - 13 foi criada a função *HHrates.m*, anexo 5.2, que calcula estas taxas dado o valor de v_m e tendo em conta as singularidades das equações 8 e 9.

b) O código pede ao utilizador a intensidade da corrente de estímulo I_s (em $\mu A/cm^2$), a sua duração e o seu instante inicial, bem como a duração da simulação numérica (mínimo de $30\,ms$). A evolução temporal do potencial de membrana foi obtida pela implementação das equações 1 - 16. As discretizações das equações 5, 6, 7 e 16 foram obtidas através do método de Euler:

$$\Delta n = \Delta t\,[\alpha_n(1-n) - \beta_n\, n] \tag{19}$$

$$\Delta m = \Delta t\,[\alpha_m(1-m) - \beta_m\, m] \tag{20}$$

$$\Delta h = \Delta t\,[\alpha_h(1-h) - \beta_h\, h] \tag{21}$$

$$\Delta V_m = \Delta t\,\frac{I_C}{C_m} \tag{22}$$

A ordem de implementação foi primeiro o cálculo das condutâncias específicas (equações 3 e 4), seguido do cálculo das correntes (equações 1, 2, 14 e 15), das taxas α e β (equações 8 - 13), dos incrementos (equações 19 - 22), e por fim a atualização das variáveis de estado:

$$n^{i+1} = n^i + \Delta n \tag{23}$$

$$m^{i+1} = m^i + \Delta m \tag{24}$$

$$h^{i+1} = h^i + \Delta h \tag{25}$$

$$V_m^{i+1} = V_m^i + \Delta V_m \tag{26}$$

Estes passos estão implementados na função *HHmodel.m*. Os valores iniciais usados para as variáveis de estado encontram-se no anexo 5.1.

c) São produzidos os gráficos de I_m, V_m, n, m, h, g_Na, g_K, I_Na, I_K, I_L, I_{ion} e I_C em função do tempo através do script *graphics.m*.

d) Para calcular o valor da intensidade limiar de geração de potenciais de ação para um tempo de estimulação de $1\,ms$ e com um $\Delta t = 50\,\mu s$, foi usado o método da bissecção e a função *HHmodel.m*. O critério para definir a existência de um potencial de ação foi o potencial membranar reduzido subir acima do *threshold* $v_m > 30\,mV$ [1]. Caso o intervalo inicial definido para o método da bissecção não incluísse um potencial de ação, então o intervalo era expandido até incluir um.

e) Foi realizada a demonstração do efeito *anode break excitation*. Para tal foi usada a função *HHmodel.m* e aplicado um estímulo de $-5\,\mu A/cm^2$ durante os primeiros $30\,ms$ de forma a que as variáveis atingissem uma estado estacionário, numa simulação com uma duração de $60\,ms$. Foram ainda obtidos os mesmos gráficos que no ponto c) com o script *graphics.m*.

2.2.2 Parte 2 - Equações do cabo

Para a segunda parte foi utilizado o seguinte procedimento que está implementado no script *main_2.m* (anexo 5.2):

a) Para realizar a discretização espacial e temporal foram escolhidos os valores $\Delta x = 0.05\, cm$ e $\Delta t = 2\, \mu s$. Com base nestes valores e nas constantes do anexo 5.2 foi calculada a *mesh ratio*:

$$Mesh\ ratio = \frac{\Delta t}{r_i c_m \Delta x^2} = \frac{a \Delta t}{2 R_i C_m \Delta x^2} \tag{27}$$

Onde c_m é a capacitância da membrana por unidade de comprimento $(c_m = 2\pi a C_m)$ e R_i é a resistividade $\left(R_i = \pi a^2 r_i\right)$. A *mesh ratio* é um indicador importante para a estabilidade numérica do algoritmo, em que deve $mesh\ ratio \ll 1$.

b) Para obter o limiar da corrente de estimulação i_p foram inicialmente implementadas as equações do cabo de forma a simular a propagação temporal e espacial de um potencial de ação. A discretização das equações 17 e 18 foram feitas usando o método de Euler:

$$I_i^{i,j} = -\frac{1}{r_i + r_e} \frac{V_m^{i,j+1} - V_m^{i,j}}{\Delta x} \tag{28}$$

$$I_m^{i,j} = \frac{1}{2\pi a \left(r_i + r_e\right)} \left[\frac{V_m^{i,j+1} - 2\, V_m^{i,j} + V_m^{i,j-1}}{\Delta x^2} - r_e i_p^{i,j}\right] + I_s^{i,j} \tag{29}$$

Onde os índices i e j representam os instantes temporal e espacial, respetivamente. Foram utilizadas *sealed boundary conditions* na sua implementação na função *equacao_cabo.m* (anexos 5.2). O critério utilizado para definir a existência de um potencial de ação foi o mesmo que na primeira parte, na alínea d) dos métodos 2.2.1, assim como também foi implementado o método da bissecção da mesma forma.

c) Foi simulado um potencial de ação aplicando uma corrente $i_p = -2\, mA/cm$ no primeiro nodo e $i_p = 2\, mA/cm$ no último nodo durante $100\, \mu s$, numa simulação com uma duração total de $20\, ms$, em que se considerou $I_s = 0$. Foram feitos os gráficos do potencial de membrana, das correntes de sódio, de potássio, transmembranar e axial intracelular em função da posição para o instante $t = 10\, ms$ através da função *graphics_2.m*.

d) Com os resultados obtidos na alínea c) foram feitos os gráficos das mesmas quantidades com a função *graphics_2.m*, mas desta vez em função do tempo para o nodo 300.

e) Analisou-se a influência do raio da fibra na velocidade de propagação de um potencial de ação. Para tal foram testados raios no intervalo $[3, 300]\ \mu m$, e era aplicada uma corrente de estimulação $I_p = \frac{i_p}{2\pi a} = -\frac{1}{0.03 \times \pi}\, mA/cm^2$ no primeiro nodo e o simétrico no último, durante os primeiros $100\, \mu s$. Foi usado $\Delta t = 2\, \mu s$ constante enquanto que Δx era adaptado de forma a que a *mesh ratio* usada tivesse sempre o valor de 0.4 para todos os raios. A duração da estimulação foi de $20\, ms$ para cada caso. Para calcular a velocidade foi registado o instante do máximo do potencial para cada nodo, e no fim foi calculado o declive entre o vetor das posições e o dos instantes de tempo correspondentes aos máximos do potencial através de uma regressão linear.

3 Resultados

3.1 Parte 1 - Modelo Hodgkin-Huxley

Na primeira parte do trabalho, na simulação duma experiência *space-clamped*, a membrana foi estimulada assumindo sempre que a corrente de estímulo I_s é uniforme no segmento considerado da membrana, usando sempre $\Delta t = 50\,ms$. Os valores iniciais usados para as diferentes variáveis de estado encontram-se no anexo 5.1.

No início a membrana foi estimulada com uma corrente cujas característica foram definidas pelo utilizador. Os valores obtidos através da simulação para as correntes, o potencial de membrana, as condutâncias específicas e as probabilidades das subunidades estarem abertas encontram-se na figura 1.

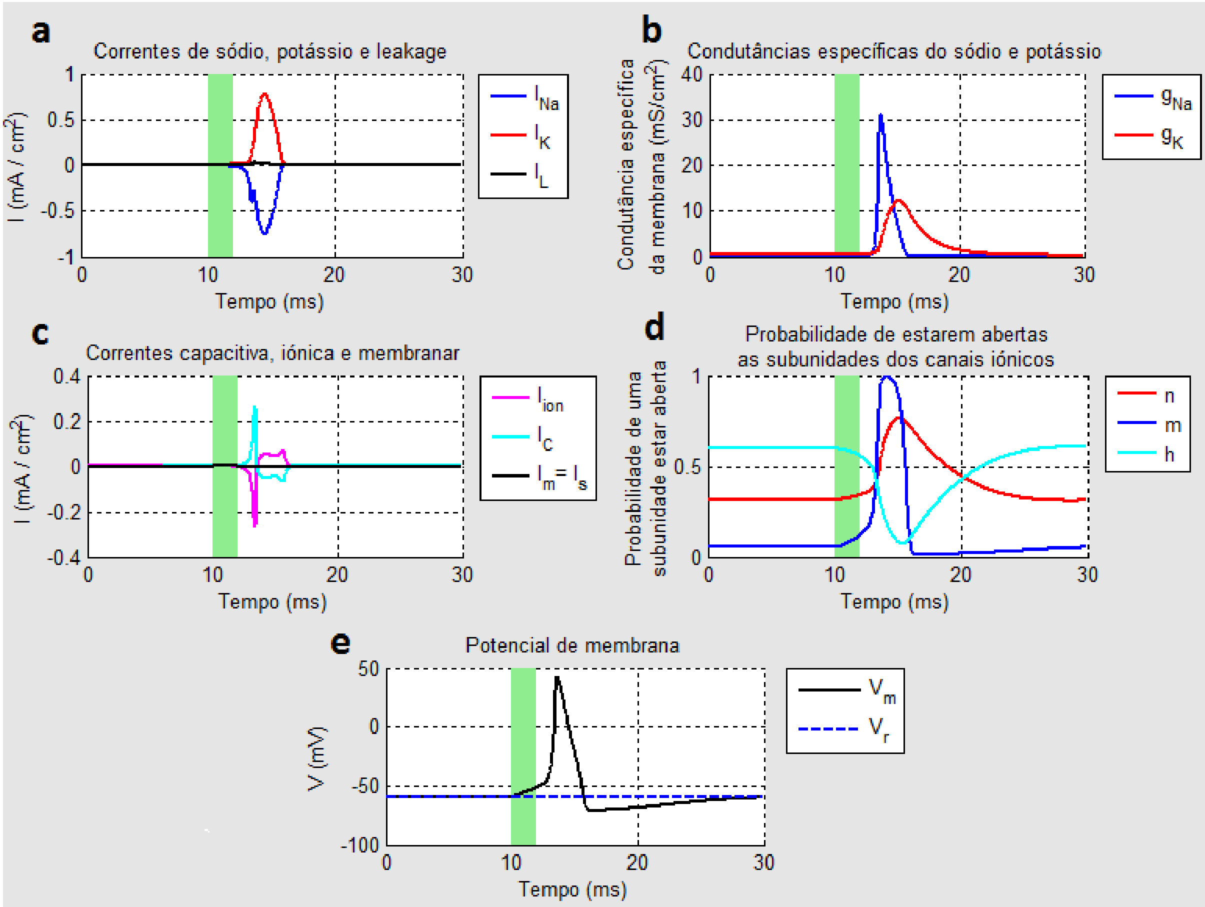

Figura 1: Simulação da estimulação da membrana com uma corrente $I_s = 5\,\mu A/cm^2$ durante $2\,ms$, com início no instante $t = 10\,ms$ - zonas a verde correspondem ao período de estimulação. A duração da simulação foi de $30\,ms$. a) Correntes (mA/cm^2) de sódio, potássio e *leakage*. b) Condutâncias específicas (mS/cm^2) da membrana para o sódio e potássio. c) Correntes (mA/cm^2) capacitiva, iónica e membranar. d) Probabilidades de as subunidades n, m e h estarem abertas. e) Potencial de membrana e de repouso (mV).

Posteriormente, para se calcular a intensidade limiar de geração de potenciais de ação para uma estimulação com uma duração de $1\,ms$ foi usado usado o procedimento descrito nos métodos 2.2.1, na alínea d). Mais uma vez, nas simulações foi usado um passo de tempo $\Delta t = 50\,\mu s$. O valor então obtido para o limiar de estimulação foi $7.092 \pm 0.001\,\mu A/cm^2$.

Foi demonstrado o efeito *anode break excitation* através da descrição feita nos métodos 2.2.1, na alínea e). Os resultados obtidos para as correntes, o potencial de membrana, as condutâncias específicas e as probabilidades das subunidades estarem abertas encontram-se na figura 2.

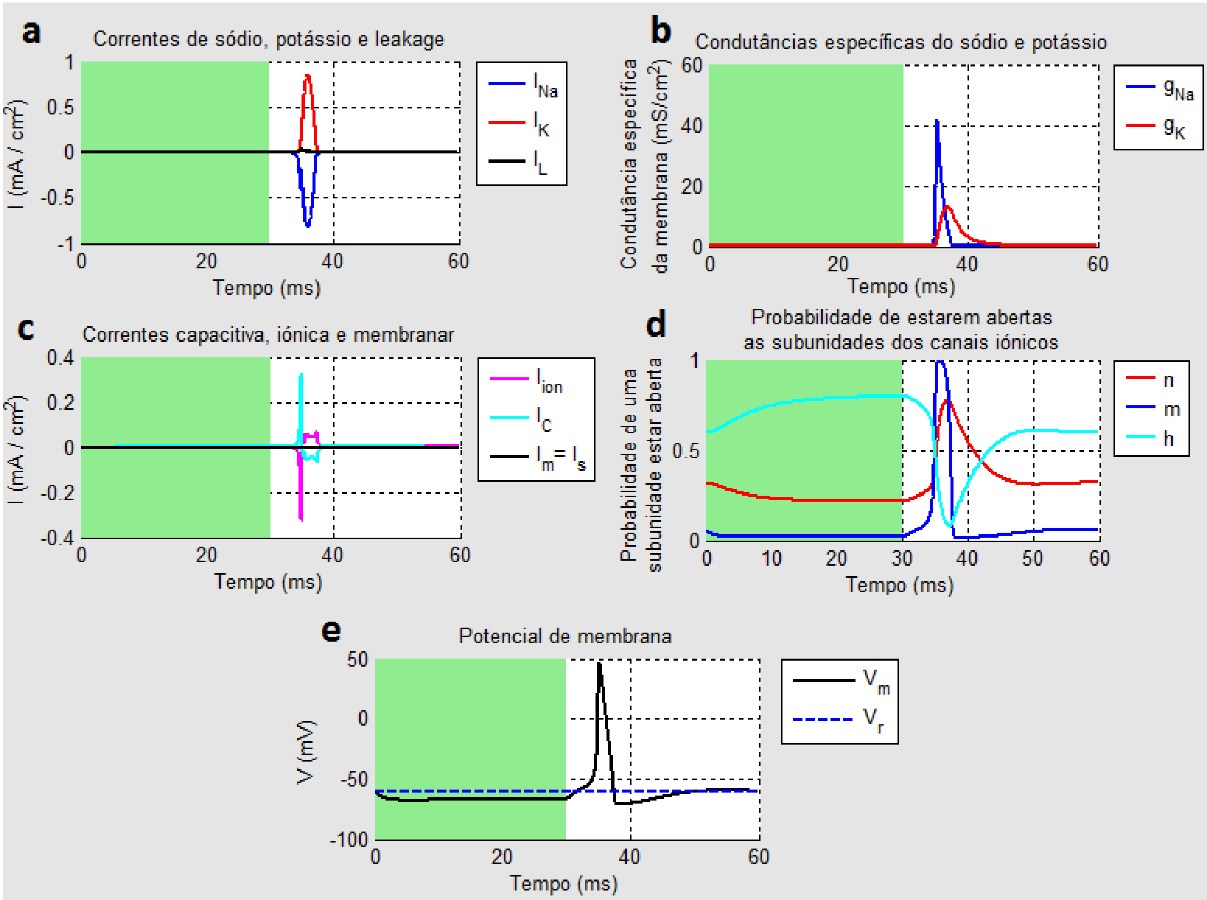

Figura 2: Estimulação da membrana com uma corrente $I_s = -5\,\mu A/cm^2$ durante os primeiros $30\,ms$ - zonas a verde correspondem ao período de estimulação. A duração da simulação foi de $60\,ms$. a) Correntes (mA/cm^2) de sódio, potássio e *leakage*. b) Condutâncias específicas (mS/cm^2) da membrana para o sódio e potássio. c) Correntes (mA/cm^2) capacitiva, iónica e membranar. d) Probabilidades de as subunidades n, m e h estarem abertas. e) Potencial de membrana e de repouso (mV).

3.2 Parte 2 - Equações do cabo

Na segunda parte do trabalho foi primeiro calculado a *mesh ratio* para $\Delta x = 0.05\,cm$ e $\Delta t = 2\,\mu s$, numa fibra com raio $a = 300\,\mu m$. O valor obtido para a *mesh ratio* foi 0.4.

Para se calcular o limiar da corrente de estimulação i_p para uma estimulação com uma duração de $100\,\mu s$ foi usado o procedimento descrito nos métodos 2.2.2, na alínea b). Além disso nas simulações foram usadas as discretizações espaciais e temporais acima referidas. O valor então obtido para o limiar da corrente de estimulação foi $-1.371 \pm 0.001\,mA/cm$.

Posteriormente a membrana foi estimulada de acordo com a descrição nos métodos 2.2.2, na alínea c). Foram produzidos os gráficos do potencial de membrana, correntes de potássio, sódio, transmembranar e axial intracelular, tudo em função da posição para o instante $t = 10\,ms$. Os resultados encontram-se na figura 3.

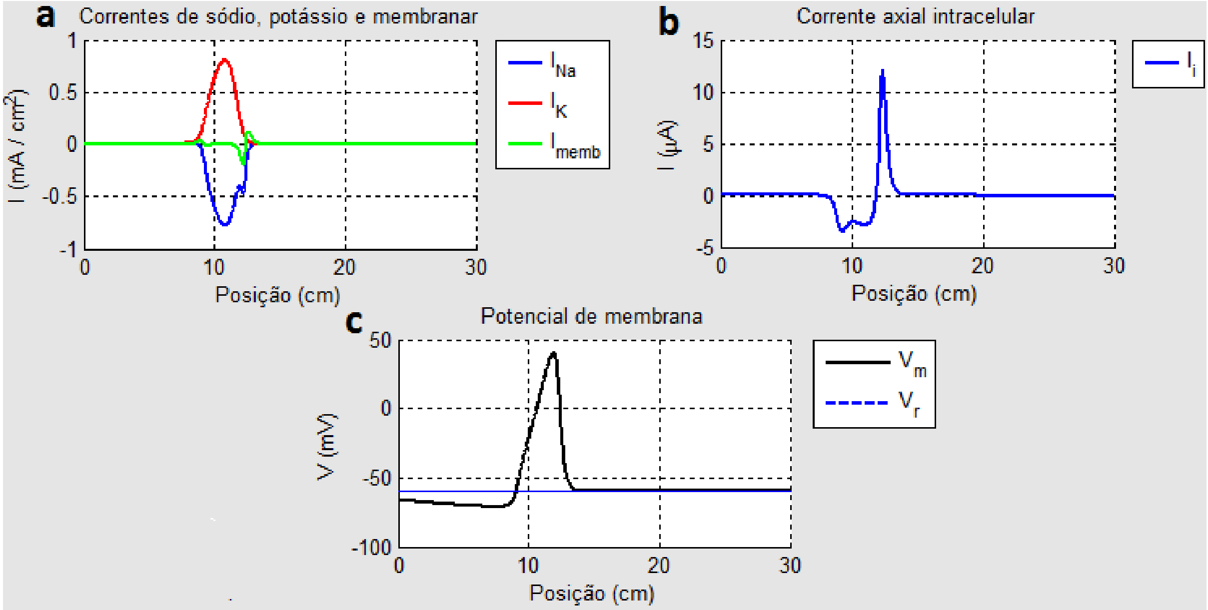

Figura 3: Simulação das correntes de potássio, sódio, transmembranar (a) e axial intracelular (b), e do potencial de membrana (c), todas para o instante $t = 10\,ms$, numa simulação de duração $20\,ms$. Foi usado $\Delta x = 0.05\,cm$, $\Delta t = 2\,\mu s$ e aplicada uma corrente $i_p = -2\,mA/cm$ no primeiro nodo e o simétrico no último, durante $100\,\mu s$.

Foram também produzidos os gráficos do potencial de membrana, correntes de potássio, sódio e axial intracelular em função do tempo para o nodo 300. Os resultados encontram-se na figura 4.

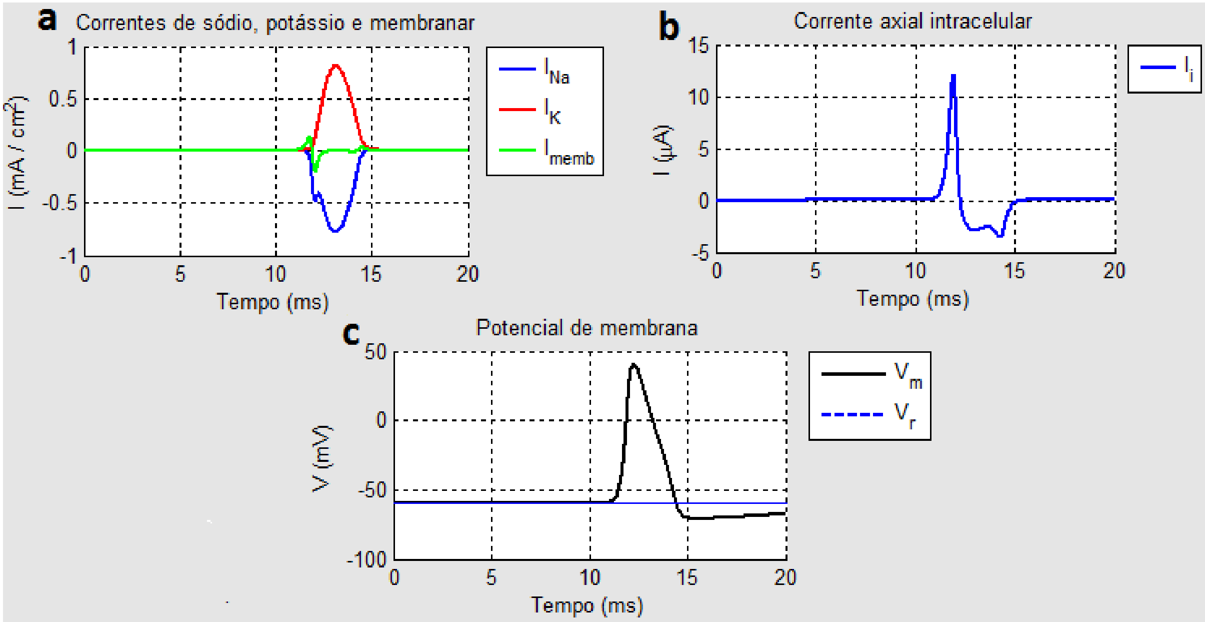

Figura 4: Simulação das correntes de potássio, sódio, transmembranar (a) e axial intracelular (b), e do potencial de membrana (c), todas para o nodo 300, numa simulação de duração $20\,ms$. Foi usado $\Delta x = 0.05\,cm$, $\Delta t = 2\,\mu s$ e aplicada uma corrente $i_p = -2\,mA/cm$ no primeiro nodo e o simétrico no último, durante $100\,\mu s$.

Por fim foi analisada a influência do raio da fibra na velocidade de propagação do potencial de ação de acordo com o procedimento nos métodos 2.2.2, na alínea e). Os resultados obtidos encontram-se na figura 5.

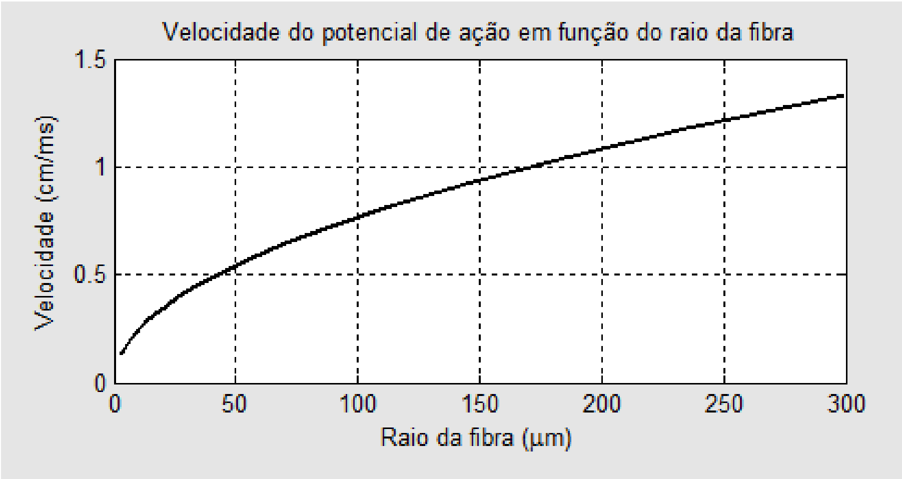

Figura 5: Velocidade de propagação do potencial de ação (cm/ms) em função do raio da fibra (μm). Foi aplicada uma corrente de estimulação $I_p = -\frac{1}{0.03\times\pi}\, mA/cm^2$ no primeiro nodo e o seu simétrico no último nodo, durante um período de $100\,\mu s$. O valor de $\Delta t = 2\,\mu s$ usado foi constante, bem como o da *mesh ratio*$= 0.4$, sendo adaptado o valor de Δx. A duração de cada simulação foi $20\,ms$. Simularam-se 20 valores de raio entre $3\,\mu m$ e $300\,\mu m$, inclusive, espaçados de forma logarítmica.

4 Discussão

Pela análise dos gráficos representados na figura 1, verifica-se pela análise de a) a deslocação do ião sódio do meio extracelular para o meio intracelular, visto a sua corrente assumir valores negativos. Quanto ao ião potássio, a sua corrente é positiva, mostrando assim a deslocação deste ião do meio intracelular para o extracelular. Além disto, verifica-se que os picos das correntes de sódio e potássio coincidem com os picos das respetivas condutâncias - gráfico b). Quanto à figura b) observa-se que os picos das condutâncias para além de estarem relacionados com as respetivas correntes, estão relacionados com as probabilidades das suas subunidades estarem abertas, sendo mais notório para o potássio, dado só depender de n. Pelo gráfico c) pode-se verificar que as correntes capacitiva e iónica são simétricas, à exceção da secção em que $I_m \neq 0$, tal como esperado. Verifica-se também que as oscilações de I_C estão associadas às variações de V_m - gráfico e) - tal como esperado. No gráfico d) observa-se que as probabilidades das subunidades de n e m estarem abertas aumentam no início do potencial de ação, confirmando o seu papel de parâmetros de ativação na ocorrência de um potencial de ação. Por sua vez, a probabilidade da subunidade h estar aberta diminui durante a ocorrência do potencial de ação, confirmando também o seu papel de parâmetro de inativação. Por fim, pela análise do gráfico e) verifica-se a ocorrência de um potencial de ação através de uma despolarização inicial da membrana até $V_m \approx 40\,mV$, seguido da relaxação deste, ocorrendo por fim uma hiperpolarização da mesma.

Na figura 2 observa-se por sua vez um potencial de ação resultante do *anode break excitation*. Por comparação com os respetivos gráficos da figura 1 verifica-se que estes são semelhantes, com a exceção do gráfico d) e com o facto de V_m hiperpolarizar no início, durante a estimulação, antes de se iniciar o potencial de ação. Relativamente ao gráfico d) observa-se neste caso que a probabilidade da subunidade h estar aberta aumenta consideravelmente devido ao estímulo, e que as probabilidades das subunidades n e m estarem abertas diminuem, mas baixam pouco comparativamente com a variação de h. Tal leva a que, antes do estímulo cessar, mais canais de potássio estejam abertos, e consequentemente a condutância específica e corrente do potássio também são maiores. Assim o fluxo de iões potássio é maior e a membrana inicialmente despolariza mais rápido, dado a corrente de potássio ser negativa, o que leva à ocorrência do potencial de ação - efeito *anode break*.

Pela análise do gráfico c) da figura 3 verifica-se que o potencial está hiperpolarizado nas regiões mais perto do primeiro nodo. A hiperpolarização confirma que nessa zona da membrana já ocorreu um potencial de ação, enquanto o potencial de membrana está a despolarizar do lado direito, indicando que o potencial de ação se está a deslocar para a direita, tal como esperado. No gráfico b) constata-se que a corrente axial intracelular é proporcional ao simétrico da primeira derivada do potencial de membrana, tal como esperado, dada a equação 17. Pela análise do gráfico a) constata-se que as correntes de potássio e sódio são semelhantes às já vistas para o caso das experiências *space-clamped* da primeira parte, enquanto a corrente transmembranar está relacionada com a segunda derivada do V_m.

Na figura 4, no gráfico c) verifica-se que o potencial de membrana está despolarizado do lado esquerdo do seu pico e hiperpolarizado do lado direito. Estes resultados são esperados, pois após o pico a membrana num determinado nodo começa a repolarizar com o tempo e consequentemente hiperpolariza. Já antes do pico a membrana ainda está a despolarizar, daí este perfil do potencial de membrana, que é semelhante aos das experiências *space-clamped* da primeira parte. Assim os resultados observados nos gráficos a) e b) são o simétrico do observado na figura 3, verificando-se as mesmas relações constatadas anteriormente.

Da análise da influência do raio da fibra na velocidade de propagação do potencial de ação, constata-se pela figura 5 que a velocidade é tanto maior quanto maior o raio da fibra, tal como esperado. Além disso, a variação aparenta ser diretamente proporcional à raiz quadrada do raio da fibra, tal como reportado na literatura [1].

Referências

[1] Robert Plonsey & Roger C. Barr, *Bioeletricity: A Quantitative Approach*; Springer 3ª edição (2007). ISBN-10:0387488642

5 Anexos

5.1 Anexo I

Nome	Unidades	Descrição
I.Na	mA/cm^2	Campo com o vetor dos valores da corrente de sódio
I.K	mA/cm^2	Campo com o vetor dos valores da corrente de potássio
I.L	mA/cm^2	Campo com o vetor dos valores da corrente de *leakage*
I.ion	mA/cm^2	Campo com o vetor dos valores da corrente iónica
I.memb	mA/cm^2	Campo com o vetor dos valores da corrente membranar
I.C	mA/cm^2	Campo com o vetor dos valores da corrente capacitiva
I.s	mA/cm^2	Campo com o vetor dos valores da corrente de estimulação
I.i	mA/cm^2	Campo com o vetor dos valores da corrente axial intracelular
ip	mA/cm^2	Vetor com a corrente de estimulação
Amp_Is	mA/cm^2	Amplitude da corrente de estimulação I_s
Amp_ip	mA/cm^2	Amplitude da corrente de estimulação i_p
Vm	mV	Vector com o potencial membranar
Vr	mV	Valor do potencial de repouso
vm	mV	Valor do potencial membranar reduzido
E.K	mV	Campo com o potencial de Nernst do potássio
E.Na	mV	Campo com o potencial de Nernst do sódio
E.L	mV	Campo com o potencial de Nernst associado ao *leakage*
nmh.n		Probabilidade da subunidade n estar aberta
nnmh.n(1)		0.31768 (Valor incial)
nmh.m		Probabilidade da subunidade m estar aberta
nmh.m(1)		0.05293 (Valor inicial)
nmh.h		Probabilidade da subunidade h estar aberta
nmh.h(1)		0.59612 (Valor inicial)
g.Na	mS/cm^2	Campo com a condutância específica do sódio
g.Na(1)	mS/cm^2	0.011 (Valor inicial)
g.K	mS/cm^2	Campo com a condutância específica do sódio
g.K(1)	mS/cm^2	0.367 (Valor inicial)
g.L	mS/cm^2	Campo com a condutância específica do *leakage* (=0.3)
g_max.Na	mS/cm^2	Campo com a condutância máx. espec. da memb. para o Na
g_max.K	mS/cm^2	Campo com a condutância máx. espec. da memb. para o K
dx	cm	Espaçamento (discretização espacial)
dt	ms	Passo de tempo (discretização temporal)
tmax	ms	Tempo total de simulação
t_i	ms	Tempo inicial da estimulação
t_estimulacao	ms	Tempo total de estimulação
Cm	$\mu F/cm^2$	Capacidade específica da membrana
a	cm	Raio da fibra
a_min	cm	Raio mínimo das fibras simuladas
a_max	cm	Raio máximo das fibras simuladas
mesh_ratio		Mesh ratio
Ri	$\Omega \cdot cm$	Resistividade do meio intracelular
Re	$\Omega \cdot cm$	Resistividade do meio extracelular
ri	Ω/cm	Resistência intracelular por unidade de comprimento
re	Ω/cm	Resistência extracelular por unidade de comprimento
alpha	ms^{-1}	Vetor com as taxas α: $[\alpha_n, \alpha_m, \alpha_h]$
beta	ms^{-1}	Vetor com as taxas β: $[\beta_n, \beta_m, \beta_h]$
N		Número de nodos

5.2 Anexo II

Script *main_1.m*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                      TRABALHO PRATICO - 1 PARTE                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Simulacao de uma experiencia voltage clamp, em que e aplicado um     %
%   estimulo com caracteristicas definidas pelo utilizador (alinea c).   %
%   Determinacao da intensidade do limiar de estimulacao de um potencial %
%   de accao atraves do criterio vm > 30mV para um potencial de accao,   %
%   onde vm e o potencial reduzido da membrana. Justificacao do criterio %
%   junto a alinea d). Aplicacao de um estemulo para demonstrar o efeito %
%   anode break (alinea e.i)                                             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% limpa o workspace; fecha handles abertos; limpa a consola;
clear; close all; clc;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                             CONSTANTES                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Vr=-60; % potencial de repouso (mV)
E.K=-72.1; % potencial de Nernst do potassio (mV)
E.Na=52.4; % potencial de Nernst do sodio (mV)
E.L=-49.187; % potencial de Nernst associado ao leakage (mV)

Cm=1; % capacidade especifica da membrana (uF/cm^2)

g_max.Na=120; % condutancia max. espec. da membrana para o sodio (mS/cm^2)
g_max.K=36; % condutancia max. espec. da membrana para o potassio (mS/cm^2)
g.L=0.3; % condutancia espec. da membrana associada ao leakage (mS/cm^2)

%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                             SIMULACOES                           %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
               %            ALINEA C)           %
               %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     PARAMETROS DO UTILIZADOR   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% pede ao utilizador o passo temporal (us), a duracao da estimulacao (ms),
% a intensidade da estimulacao (uA/cm^2), o instante inicial da estimulacao
% (ms) e a duracao da estimulacao (ms)
prompt={'Passo temporal \Deltat (\mus)',...
    'Duracao da simulacao (ms) [ minimo 30 ms ]',...
    ['';'I_{estimulacao} (\muA / cm^2)'],...
    'Instante inicial da estimulacao (ms)','Duracao da estimulacao (ms)'};
name = 'Input';
defaultans = {'','30','','',''};
options.Interpreter = 'tex';
user_input = inputdlg(prompt,name,[1 50;1 50;1 50;1 50;1 50],defaultans,...
options);

dt=str2num(user_input{1}); % passo de tempo (us)
dt=dt/10^3; % passo de tempo (ms)
```

```
tmax=str2num(user_input{2}); % tempo total da simulacao (ms)
if tmax<30
    tmax=30; % se o utilizador introduzir um tempo de simulacao inferior a
    % 30ms este e alterado para 30ms
    w=warndlg({'Tempo maximo de simulacao inferior ao minimo de 30 ms',...
        '     Tempo maximo de simulacao assumido como 30 ms'},'Warning');
    uiwait(w); clear w;
end
n_iter=tmax/dt; % numero de iteracoes realizadas na simulacao

Amp_Is=str2num(user_input{3});
Amp_Is=Amp_Is/1e3; % intensidade do estimulo em mA/cm^2

t_i=str2num(user_input{4}); % instante inicial da estimualacao (ms)
t_i=round(t_i/dt)*dt; % instante inicial da estimulacao em funcao de dt
t_estimulacao=str2num(user_input{5}); % duracao da estimulacao (ms)
t_estimulacao=round(t_estimulacao/dt)*dt; % duracao da estimulacao em
% funcao de dt
dialog_box_tempos(t_i,t_estimulacao); % informa o utilizador dos instantes
% em funcao de dt

Is = stimulus_current(dt,t_i,t_estimulacao,Amp_Is,n_iter); % cria o vector
% com a corrente de estimulacao (mA/cm^2) com os parametros dados pelo
% utilizador

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      INICIALIZACAO DE VARIAVEIS       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

t=0:dt:tmax-dt; % vector do tempo da simulacao (ms)

nmh.n=zeros(1,n_iter); % vector para a probabilidade de estar aberta uma
% subunidade n
nmh.n(1)=0.31768; % probabilidade de estar aberta uma subunidade n em t=0
nmh.m=zeros(1,n_iter); % vector para a probabilidade de estar aberta uma
% subunidade m
nmh.m(1)=0.05293; % probabilidade de estar aberta uma subunidade m em t=0
nmh.h=zeros(1,n_iter); % vector para a probabilidade de estar aberta uma
% subunidade h
nmh.h(1)=0.59612; % probabilidade de estar aberta uma subunidade h em t=0

g.Na=zeros(1,n_iter); % vector para a condutancia especifica da membrana
% para o sodio
g.Na(1)=0.011; % condutancia especifica da membrana para o sodio em t=0
g.K=zeros(1,n_iter); % vector para a condutancia especifica da membrana
% para o potassio
g.K(1)=0.367; % condutancia especifica da membrana para o potassio em t=0

Vm=zeros(1,n_iter); % vector para o potencial de membrana
Vm(1)=Vr; % potencial de membrana em t=0

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                SIMULACAO              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% simula as correntes, condutancias, potencial de membrana e probabilidades
% de abertura das subunidades dos canais durante o periodo de simulacao
% definido pelo utilizador
[I,nmh,Vm,g]=HHmodel(n_iter,dt,Cm,Is,Vm,Vr,E,g,g_max,nmh);

% cria graficos das correntes, condutancias, potencial de membrana e
% probabilidades de abertura das subunidades dos canais em funcao do tempo
run graphics.m
```

```
%%

                    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                    %                 ALINEA D)               %
                    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      PARAMETROS DO UTILIZADOR         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% pede ao utilizador a precisao (nA/cm^2) para o calculo da intensidade
% limiar
prompt={'Precisao para a intensidade limiar (nA / cm^2)'};
name = 'Input';
defaultans = {''};
options.Interpreter = 'tex';
user_input = inputdlg(prompt,name,[1 50],defaultans,options);
precisao=str2num(user_input{1});
precisao=precisao/10^6; % a constante 10^6 garante que fica a mesma escala
% da corrente Is (mA/cm^2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      INICIALIZACAO DE VARIAVEIS       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dt=0.05; % passo de tempo (ms)
t_estimulacao=1; % duracao da estimulacao (ms)
t_i=5; % instante inicial da estimualacao (ms)
tmax=30; % tempo total da simulacao (ms)
n_iter=tmax/dt; % numero de iteracoes realizadas na simulacao

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%             SIMULACAO                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Amp_Is=[0 1]; % valores iniciais de intensidade da corrente de estimulacao
cond=1; % garante que o ciclo while se inicia

while(cond)
Is = stimulus_current(dt,t_i,t_estimulacao,Amp_Is(2),n_iter); % vector cuja
% intensidade do estimulo (mA/cm^2) e dada pelo max(Amp_Is)
[~,~,Vm,~]=HHmodel(n_iter,dt,Cm,Is,Vm,Vr,E,g,g_max,nmh); % simula os
% resultados com Is definido anteriormente e apenas guarda o potencial da
% membrana para analise

% caso em algum instante o potencial da membrana reduzido ultrupasse 30mV
% entao um potencial de accao ocorre
if any((Vm-Vr)>30)
    % define-se uma intensidade de estimulo intermedia
    Is = stimulus_current(dt,t_i,t_estimulacao,sum(Amp_Is)/2,n_iter);
    % simula-se se com a intensidade de estimulo intermedia
    [~,~,Vm,~]=HHmodel(n_iter,dt,Cm,Is,Vm,Vr,E,g,g_max,nmh);

    % verifica-se se com a intensidade de estimulo intermedia ocorre algum
    % potencial de accao
    if any((Vm-Vr)>30)
        % se ocorrer entao o limiar de excitabilidade encontra-se entre o
        % valor mais baixo e o valor intermedio
        Amp_Is(2)=sum(Amp_Is)/2;
    else
        % se nao ocorrer entao o limiar de excitabilidade encontra-se entre
        % o valor intermedio e o valor mais alto
```

```
            Amp_Is(1)=sum(Amp_Is)/2;
        end
        cond=(diff(Amp_Is)/2)>precisao; % avalia-se se metade do tamanho do
        % intervalo ja e inferior a precisao desejada pelo utilizador
    else
        % caso nao ocorra potencial de accao com max(Amp_Is) entao esse valor e
        % duplicado
        Amp_Is(2)=Amp_Is(2)*2;
    end
end
fprintf('Estimativa da intensidade limiar = %f %c %f uA/cm^2 \n',...
    sum(Amp_Is)*1e3/2,char(177),precisao*1e3); % imprime na consola a
% estimativa do limiar de excitabilidade

%%

                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %            ALINEA E.i            %
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%    INICIALIZACAO DE VARIAVEIS    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dt=0.05; % passo de tempo (ms)
t_estimulacao=30; % duracao da estimulacao (ms)
t_i=0; % instante inicial da estimualacao (ms)
tmax=60; % tempo total da simulacao (ms)
n_iter=tmax/dt; % numero de iteracoes realizadas na simulacao
t=0:dt:tmax-dt; % vector do tempo da simulacao

nmh.n=zeros(1,n_iter); % vector para a probabilidade de estar aberta uma
% subunidade n
nmh.n(1)=0.31768; % probabilidade de estar aberta uma subunidade n em t=0
nmh.m=zeros(1,n_iter); % vector para a probabilidade de estar aberta uma
% subunidade m
nmh.m(1)=0.05293; % probabilidade de estar aberta uma subunidade m em t=0
nmh.h=zeros(1,n_iter); % vector para a probabilidade de estar aberta uma
% subunidade h
nmh.h(1)=0.59612; % probabilidade de estar aberta uma subunidade h em t=0

g.Na=zeros(1,n_iter); % vector para a condutancia especifica da membrana
% para o sodio
g.Na(1)=0.011; % condutancia especifica da membrana para o sodio em t=0
g.K=zeros(1,n_iter); % vector para a condutancia especifica da membrana
% para o potassio
g.K(1)=0.367; % condutancia especifica da membrana para o potassio em t=0

Vm=zeros(1,n_iter); % vector para o potencial de membrana
Vm(1)=Vr; % potencial de membrana em t=0

Amp_Is=-5; % intensidade do estimulo em uA/cm^2 para causa o efeito anode
% break
Amp_Is=Amp_Is/1e3; % intensidade do estimulo em mA/cm^2

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%            SIMULACAO            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% cria o vector com a corrente de estimulacao (mA/cm^2) com os parametros
% para demonstrar o efeito anode break
Is = stimulus_current(dt,t_i,t_estimulacao,Amp_Is,n_iter);
```

```
% simula as correntes, condutancias, potencial de membrana e probabilidades
% de abertura das subunidades dos canais com Is para o anode break
[I,nmh,Vm,g]=HHmodel(n_iter,dt,Cm,Is,Vm,Vr,E,g,g_max,nmh);

% cria graficos das correntes, condutancias, potencial de membrana e
% probabilidades de abertura das subunidades dos canais em funcao do tempo
run graphics.m
```

Função *HHmodel.m*

```
function [I,nmh,Vm,g] = HHmodel(n_iter,dt,Cm,Is,Vm,Vr,E,g,g_max,nmh)
%[I,nmh,Vm,g] = HHmodel(n_iter,dt,Cm,Is,Vm,Vr,E,g,g_max,nmh)
%  Com base no numero de iteracoes n_iter e no passo temporal dt simula a
%  evolucao dos parametros I (mA/cm^2) (correntes de sodio I.Na, potassio
%  I.K, capacitiva I.C, leakage I.L, ionica I.ion, estimulacao I.s e
%  membranar I.memb), Vm (mV) (potencial de membrana), g (mS/cm^2)
%  (condutancias do sodio g.Na e do potassio g.K) e das probabilidades n,
%  m e h das subunidades dos subunidades dos canais ionicos estarem
%  abertas (nmh.n, nmh.m, nmh.h)

I.memb=zeros(1,n_iter); % vector para a corrente transmembranar (mA/cm^2)
I.Na=zeros(1,n_iter); % vector para a corrente de sodio (mA/cm^2)
I.K=zeros(1,n_iter); % vector para a corrente de potassio (mA/cm^2)
I.L=zeros(1,n_iter); % vector para a corrente de leakage (mA/cm^2)
I.ion=zeros(1,n_iter); % vector para a corrente ionica (mA/cm^2)
I.C=zeros(1,n_iter); % vector para a corrente capacitiva (mA/cm^2)
I.s=Is; % (mA/cm^2)

for i=1:n_iter

    if(i~=1)
        g.Na(i)=(nmh.m(i)^3)*nmh.h(i)*g_max.Na; % condutancia especifica do
        % sodio (mS/cm^2)
        g.K(i)=(nmh.n(i)^4)*g_max.K; % condutancia especifica do potassio
        % (mS/cm^2)
    end

    % calculo das correntes
    I.memb(i)=I.s(i); % corrente membranar (mA/cm^2) e igual a de
    % estimulacao
    I.Na(i)=g.Na(i)*(Vm(i)-E.Na(1))/1e3; % corrente do sodio (mA/cm^2)
    I.K(i)=g.K(i)*(Vm(i)-E.K(1))/1e3; % corrente do potassio (mA/cm^2)
    I.L(i)=g.L*(Vm(i)-E.L(1))/1e3; % corrente leakage (mA/cm^2)
    I.ion(i)=I.K(i)+I.Na(i)+I.L(i); % corrente ionica (mA/cm^2)
    I.C(i)=I.memb(i)-I.ion(i); % corrente capacitiva (mA/cm^2)

    % incrementos
    dVm=I.C(i)*(1e3*dt)/Cm; % (mV)
    [alpha,beta]=HHrates(Vm(i)-Vr); % output:(ms^-1); intput:(mV)
    dn=dt*(alpha(1)*(1-nmh.n(i))-beta(1)*nmh.n(i));
    dm=dt*(alpha(2)*(1-nmh.m(i))-beta(2)*nmh.m(i));
    dh=dt*(alpha(3)*(1-nmh.h(i))-beta(3)*nmh.h(i));

    if(i~=n_iter) % actualizacao dos valores futuros
        Vm(i+1)=Vm(i)+dVm; % (mV)
        nmh.n(i+1)=nmh.n(i)+dn;
        nmh.m(i+1)=nmh.m(i)+dm;
        nmh.h(i+1)=nmh.h(i)+dh;
    end

end
```

```

end
```

Função *HHrates.m*

```
function [alpha, beta] = HHrates(vm)
%[alpha, beta] = HHrates(vm)
%   Retorna dois vectores 1x3 com as constantes alpha e beta para as
%   subunidades n, m, h respectivamente; vm tem de estar em mV e as
%   constantes tem as unidades de ms^(-1);

alpha=zeros(1,3);
beta=zeros(1,3);

if(vm==10)
    % caso da singularidade de alpha_n
    alpha(1)=0.1;
else
    alpha(1)=0.01*(10-vm)/(exp((10-vm)/10)-1);
end

beta(1)=0.125*exp(-vm/80);

if(vm==25)
    % caso da singularidade de alpha_m
    alpha(2)=1;
else
    alpha(2)=0.1*(25-vm)/(exp(0.1*(25-vm))-1);
end

beta(2)=4*exp(-vm/18);

alpha(3)=0.07*exp(-vm/20);
beta(3)=1/(exp((30-vm)/10)+1);
```

Função *stimulus_current.m*

```
function [Is] = stimulus_current(dt,t_i,t_estimulacao,Amp_Is,n_iter)
%[Is] = stimulus_current(t_i,t_estimulacao,Amp_Is)
%   Cria o vector da corrente de estimulacao

Is=[zeros(1,floor(t_i/dt)), ones(1,floor(t_estimulacao/dt))*Amp_Is,...
    zeros(1,floor(n_iter-(t_estimulacao+t_i)/dt))]; % vector da corrente de
% estimulacao

end
```

Função *dialog_box_tempos.m*

```
function [] = dialog_box_tempos(t_i,t_estimulacao)
%[] = dialog_box_tempos(t_i,t_estimulacao)
%   Cria uma box de dialogo para avisar o utilizador dos valores de tempo
%   do inicio e da duracao da estimulacao apos discretizacao pelo passo
%   temporal

```

```
d = dialog('Position',[550 350 250 120],'Name','Variaveis');

message = {['Valor de tempo inicial = ', num2str(t_i),' ms'],...
    ['Duracao da estimulacao = ', num2str(t_estimulacao),' ms']};
uicontrol('Parent',d,'Style','text','Position',[20 60 210 40],...
    'String',message);

uicontrol('Parent',d,'Position',[85 20 70 25],'String','Fechar',...
    'Callback','delete(gcf)'); % cria botao apra fechar a janela

uiwait(d) % espera ate o utilizador fechar a janela

end
```

Script *graphics.m*

```
% Cria um novo handle com uma descricao das propriedades da corrente de
% estimulacao e graficos das correntes, condutancias especificas, potencial
% de membrana e probabilidades de abertura das subunidades dos canais,
% tudo em funcao do tempo

figure,
set(gcf,'Position',[3 40 1380 645],'color',[223 223 223]/248)

% descricao das caracteristicas da estimulacao
h = subplot(3, 2, 1);
str={'Caracteristicas da corrente de estimulacao I_s',' ',...
    ['Instante inicial de estimulacao: ',num2str(t_i),' ms'],...
    ['Duracao da estimulacao: ',num2str(t_estimulacao),' ms'],...
    ['Amplitude da estimulacao: ',num2str(Amp_Is),' mA/cm^2'],' ',...
    'Regioes a verde correspondem ao periodo de estimulacao'};
text(0.05,0.5,str);
set ( h, 'visible', 'off')

% plot do potencial membranar em funcao do tempo
subplot(3,2,2), hold on,
plot(t,Vm,'k','Linewidth',2)
plot(t,repmat(Vr,1,n_iter),'--b','Linewidth',2)
y=ylim;
h_rect=rectangle('Position',[t_i,y(1),t_estimulacao,y(2)-y(1)],...
    'FaceColor',[144,238,144]/256,'Linestyle','none');
uistack(h_rect,'bottom')
grid on
title('Potencial de membrana')
ylabel('V (mV)')
xlabel('Tempo (ms)')
legend('V_m','V_r',-1)

% plot das correntes de sodio, potassio e leakage em funcao do tempo
subplot(3,2,3), hold on,
plot(t,I.Na,'b','Linewidth',2)
plot(t,I.K,'r','Linewidth',2)
plot(t,I.L,'k','Linewidth',2)
y=ylim;
h_rect=rectangle('Position',[t_i,y(1),t_estimulacao,y(2)-y(1)],'FaceColor',[144,238,144]/256,'Linestyle','none');
uistack(h_rect,'bottom')
grid on
title('Correntes de sodio, potassio e leakage')
ylabel('I (mA / cm^2)')
xlabel('Tempo (ms)')
```

```
legend('I_{Na}','I_K','I_L',-1)

% plot das correntes ionica, capacitiva, e membranar (= estimulacao)
subplot(3,2,5), hold on,
plot(t,I.ion,'m','Linewidth',2)
plot(t,I.C,'c','Linewidth',2)
plot(t,I.memb,'k','Linewidth',2)
y=ylim;
h_rect=rectangle('Position',[t_i,y(1),t_estimulacao,y(2)-y(1)],'FaceColor'
    ,[144,238,144]/256,'Linestyle','none');
uistack(h_rect,'bottom')
grid on
title('Correntes capacitiva, ionica e membranar')
ylabel('I (mA / cm^2)')
xlabel('Tempo (ms)')
legend('I_{ion}','I_C','I_m= I_s',-1)

% plot das condutancias especificas do sodio e do potassio em funcao do
% tempo
subplot(3,2,4), hold on,
plot(t,g.Na,'b','Linewidth',2)
plot(t,g.K,'r','Linewidth',2)
y=ylim;
h_rect=rectangle('Position',[t_i,y(1),t_estimulacao,y(2)-y(1)],'FaceColor'
    ,[144,238,144]/256,'Linestyle','none');
uistack(h_rect,'bottom')
grid on
title('Condutancias especificas do sodio e potassio')
ylabel({'Condutancia especifica' ,'da membrana (mS/cm^2)'})
xlabel('Tempo (ms)')
legend('g_{Na}','g_{K}',-1)

% plot das probabilidades das subunidades n, m e h dos canais ionicos
% estarem abertas em funcao do tempo
subplot(3,2,6), hold on,
plot(t,nmh.n,'r','Linewidth',2)
plot(t,nmh.m,'b','Linewidth',2)
plot(t,nmh.h,'c','Linewidth',2)
y=ylim;
h_rect=rectangle('Position',[t_i,y(1),t_estimulacao,y(2)-y(1)],'FaceColor'
    ,[144,238,144]/256,'Linestyle','none');
uistack(h_rect,'bottom')
grid on
title({'Probabilidade de estarem abertas','as subunidades dos canais ionicos'})
ylabel({'Probabilidade de uma','subunidade estar aberta'})
xlabel('Tempo (ms)')
legend('n','m','h',-1)
```

Script *main_2.m*

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                          TRABALHO PRATICO - 2 PARTE                            %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Simulacao da propagacao espacial e temporal de um potencial de acao   %
%   atraves da implementacao das equacoes do cabo. Determinacao do        %
%   limiar da corrente de estimulacao para um estimulo de 100us numa      %
%   fibra com um raio de 300um ( alinea b ). Analise visual da            %
%   propagacao dos potenciais ( alineas c e d ). Analise da influencia    %
%   do raio da fibra na velocidade de propagacao dos potenciais de acao   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
% limpa o workspace; fecha handles abertos; limpa a consola;
clear; close all; clc;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                 CONSTANTES                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ri=30;             % ohm.cm
Re=20;             % ohm.cm

Vr=-60; % potencial de repouso (mV)
E.K=-72.1; % potencial de Nernst do potassio (mV)
E.Na=52.4; % potencial de Nernst do sodio (mV)
E.L=-49.187; % potencial de Nernst associado ao leakage (mV)

Cm=1; % capacidade especifica da membrana (uF/cm^2)

g_max.Na=120; % condutancia max. espec. da membrana para o sodio (mS/cm^2)
g_max.K=36; % condutancia max. espec. da membrana para o potassio (mS/cm^2)
g.L=0.3; % condutancia espec. da membrana associada ao leakage (mS/cm^2)

%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                 SIMULACOES                              %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %             ALINEA A)             %
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dx=0.05;    % discretizacao espacial (cm)
dt=2e-3;    % discretizacao temporal (ms)
a=0.03;     % raio da fibra (cm)

% valor de mesh ratio para os parametros definidos
mesh_ratio=dt*1e3/((Ri/a)*(Cm*2)*dx^2);
fprintf('Mesh ratio = %f \n',mesh_ratio);

%%

                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %             ALINEA B)             %
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     PARAMETROS DO UTILIZADOR      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

prompt={'Precisao para o limiar da corrente de estimulacao (\muA/cm)'};
name = 'Input';
defaultans = {''};
options.Interpreter = 'tex';
user_input = inputdlg(prompt,name,[1 60],defaultans,options);
precisao=str2num(user_input{1});
precisao=precisao/1e3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     INICIALIZACAO DE VARIAVEIS    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

t_estimulacao=0.1; % duracao da estimulacao (ms)
```

```
t_i=0; % instante inicial da estimulacao (ms)
tmax=10; % tempo total da simulacao (ms)

N=100;
n_iter=tmax/dt+1; % numero de iteracoes realizadas na simulacao

nmh.n=zeros(n_iter,N); % vector para a probabilidade de estar aberta uma
% subunidade n
nmh.m=zeros(n_iter,N); % vector para a probabilidade de estar aberta uma
% subunidade m
nmh.h=zeros(n_iter,N); % vector para a probabilidade de estar aberta uma
% subunidade h

nmh.n(1,:)=0.31768; % probabilidade de estar aberta uma subunidade n em t=0
nmh.m(1,:)=0.05293; % probabilidade de estar aberta uma subunidade m em t=0
nmh.h(1,:)=0.59612; % probabilidade de estar aberta uma subunidade h em t=0

g.Na=zeros(n_iter,N); % vector para a condutancia especifica da membrana
% para o sodio
g.K=zeros(n_iter,N); % vector para a condutancia especifica da membrana
% para o potassio

ip=zeros(n_iter,N); % corrente de estimulacao (mA/cm)

Vm=zeros(n_iter,N); % potencial de membrana
Vm(1,:)=Vr; % potencial de membrana em t=0

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%              SIMULACAO             %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Amp_ip=[-2, -1.25, 0]; % valores iniciais de intensidade da corrente de
% estimulacao (mA/cm^2)
get_max=1; % confirma se dentro do intervalo esta um potencial de acao
cond=1; % garante que o ciclo while se inicia

while(cond)

    if(get_max)
        ip(1:t_estimulacao/dt,1)=Amp_ip(1); % vector cuja intensidade
        % do estimulo (mA/cm) e dada o maximo absoluto de Amp_ip
        ip(1:t_estimulacao/dt,N)=-Amp_ip(1);
        [~,~,Vm,~]=equacao_cabo(Vm,Vr,E,dx,dt,a,Re,Ri,Cm,ip,n_iter,N,g,...
            g_max,nmh);
        % simula os resultados com ip definido anteriormente e apenas
        % guarda o potencial da membrana para analise
    end

    % caso em algum instante o potencial da membrana reduzido ultrupasse
    % 30mV entao um potencial de acao ocorre, ignorando os instantes a
    % seguir a estimulacao
    if (any(any((Vm(5*t_estimulacao/dt:end,:)-Vr)>30)) || get_max==0)
        get_max=0;
        % define-se uma intensidade de estimulo intermedia
        ip(1:t_estimulacao/dt,1)=Amp_ip(2);
        ip(1:t_estimulacao/dt,N)=-Amp_ip(2);
        % simula-se com a intensidade de estimulo intermedia
        [~,~,Vm,~]=equacao_cabo(Vm,Vr,E,dx,dt,a,Re,Ri,Cm,ip,n_iter,N,g,...
            g_max,nmh);

        % verifica-se se com a intensidade de estimulo intermedia ocorre
        % algum potencial de acao
        if any(any((Vm(5*t_estimulacao/dt:end,:)-Vr)>30))
```

```
            % se ocorrer, entao o limiar de excitabilidade encontra-se
            % entre o valor mais baixo e o valor intermedio
            Amp_ip(1)=Amp_ip(2);
            Amp_ip(2)=mean(Amp_ip(2:3));
        else
            % se nao ocorrer, entao o limiar de excitabilidade encontra-se
            % entre o valor intermedio e o valor mais alto
            Amp_ip(3)=Amp_ip(2);
            Amp_ip(2)=mean(Amp_ip(1:2));
        end
        cond=(abs(Amp_ip(1)-Amp_ip(3))/2)>precisao; % avalia-se se metade
        % do tamanho do intervalo ja e inferior a precisao desejada pelo
        % utilizador
    else
        % caso nao ocorra potencial de acao com max(Amp_Is) entao esse
        % valor e duplicado
        Amp_ip=Amp_ip*2;
    end
end

fprintf('Estimativa da intensidade limiar = %f %c %f mA/cm \n',...
    Amp_ip(2),char(177),precisao); % imprime na consola a estimativa do
% limiar de excitabilidade

%%

                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %               ALINEA C)            %
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     INICIALIZACAO DE VARIAVEIS     %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

t=20; % tempo total de simulacao (ms)
N=601;          % numero de nodos
n_iter=t/dt+1;  % numero de iteracoes temporais

nmh.n=zeros(n_iter,N); % vector para a probabilidade de estar aberta uma
% subunidade n
nmh.m=zeros(n_iter,N); % vector para a probabilidade de estar aberta uma
% subunidade m
nmh.h=zeros(n_iter,N); % vector para a probabilidade de estar aberta uma
% subunidade h

nmh.n(1,:)=0.31768; % probabilidade de estar aberta uma subunidade n em t=0
nmh.m(1,:)=0.05293; % probabilidade de estar aberta uma subunidade m em t=0
nmh.h(1,:)=0.59612; % probabilidade de estar aberta uma subunidade h em t=0

g.Na=zeros(n_iter,N); % vector para a condutancia especifica da membrana
% para o sodio
g.K=zeros(n_iter,N); % vector para a condutancia especifica da membrana
% para o potassio

Vm=zeros(n_iter,N); % vector do potencial de membrana

Vm(1,:)=Vr; % potencial de membrana em t=0

ip=zeros(n_iter,N); % vector com a corrente de estimulacao (mA/cm)
Amp_ip=-2; % amplitude da corrente de estimulacao
t_estimulacao=0.1; % duracao da estimulacao (ms)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%                 SIMULACAO                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

ip(1:(t_estimulacao/dt),1)=Amp_ip; % vector com a corrente de estimulacao
% (mA/cm)
ip(1:(t_estimulacao/dt),N)=-Amp_ip; % vector com a corrente de estimulacao
% (mA/cm)

% simulacao com os parametros definidos anteriormente
[I,nmh,Vm,g] = equacao_cabo(Vm,Vr,E,dx,dt,a,Re,Ri,Cm,ip,n_iter,N,g,...
    g_max,nmh);

% instante de tempo para fazer os graficos em funcao da posicao
t_graph=10;
graphics_2(0:dx:(N-1)*dx,Vm(t_graph/dt,:),I.Na(t_graph/dt,:),...
    I.K(t_graph/dt,:),I.memb(t_graph/dt,:),I.i(t_graph/dt,:),t_i,...
    t_estimulacao,Amp_ip,'posicao')

%%

                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %                ALINEA D)                     %
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% nodo para fazer os graficos em funcao do tempo
N_graph=300;
graphics_2(0:dt:t,Vm(:,N_graph),I.Na(:,N_graph),I.K(:,N_graph),...
    I.memb(:,N_graph),I.i(:,N_graph),t_i,t_estimulacao,Amp_ip,'tempo')

%%

                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                %               ALINEA E.ii                    %
                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     INICIALIZACAO DE VARIAVEIS               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a_min=0.0003;    % raio minimo da fibra testado (cm)
a_max=0.03;      % raio maximo da fibra testado (cm)
n=20;            % pontos
a=logspace(log10(a_min),log10(a_max),n); % lista de raios de fibra
% testados (cm)

dx_2=sqrt(1e3*a*dt/(2*mesh_ratio*Ri*Cm)); % discretizacoes espaciais para
% cada raio de fibra (cm)
N=ceil(30./dx_2)+1; % numero de nodos para cada raio de fibra
t=20; % tempo total de simulacao (ms)
n_iter_total=round(t./dt)+1; % numero de iteracoes temporais por cada raio
% de fibra

t_estimulacao=0.1; % tempo de estimulacao
Ip=-1/(pi*a(end));   % intensidade da corrente de estimulacao (mA/cm^2)

v=zeros(1,n); % vector para as velocidades

n_max=1.4e7; % LIMITE MAXIMO DE RAM: 100MB por matriz

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                 SIMULACAO                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```
for i=1:n

    % vectores para as posicoes e os instantes correspondentes aos maximos
    % do potencial em cada nodo
    x=0:dx_2(i):(N(i)-1)*dx_2(i);
    y=zeros(1,N(i));

    % potencial maximo em cada nodo
    max_V=zeros(1,N(i));

    n_col=0; % numero de instantes de tempo acumulados por segmentacao do
    % codigo
    break_down=0; % numero de segmentacoes do codigo

    n_elementos=N(i)*n_iter_total; % numero de elementos necessarios para
    % cada matriz para um determinado raio
    breaks=ceil(n_elementos/n_max)-1; % numero de vezes que se tem de
    % segmentar as matrizes/codigo

    while(break_down<=breaks)

        if(break_down==breaks)
            n_iter=n_iter_total-n_col;
        else
            n_iter=round(n_max/N(i));
        end

        % inicializacao de matrizes
        nmh.n=zeros(n_iter,N(i));
        nmh.m=zeros(n_iter,N(i));
        nmh.h=zeros(n_iter,N(i));
        g.Na=zeros(n_iter,N(i));
        g.K=zeros(n_iter,N(i));
        Vm=zeros(n_iter,N(i));
        ip=zeros(n_iter,N(i));

        if(break_down==0)
            % condicoes iniciais
            nmh.n(1,:)=0.31768;
            nmh.m(1,:)=0.05293;
            nmh.h(1,:)=0.59612;
            Vm(1,:)=Vr;
            ip(1:round(t_estimulacao/dt),1)=Ip*(2*pi*a(i));
            ip(1:round(t_estimulacao/dt),N(i))=-Ip*(2*pi*a(i));
        else
            % continuacao com os valores da ultima segmentacao
            nmh.n(1,:)=old.n;
            nmh.m(1,:)=old.m;
            nmh.h(1,:)=old.h;
            Vm(1,:)=old.Vm;
        end

        % simulacao para cada raio de fibra
        [~,nmh,Vm,~] = equacao_cabo(Vm,Vr,E,dx_2(i),dt,a(i),Re,Ri,Cm,ip,n_iter,N(
            i),g,g_max,nmh);

        % calculo do maximo do potencial de membrana por cada nodo e
        % registo dos instantes
        for j=1:N(i)
            if (max(Vm(2:end,j))-Vr)>max([30,max_V(j)-Vr])
                max_V(j)=max(Vm(2:end,j));
                y(j)=dt*(find(Vm(2:end,j)==max_V(j))-1+n_col);
            end
```

```
        end

        n_col=n_col+n_iter; % numero de instantes de tempo acumulados com
        % a segmentacao

        % guarda os ultimos valores da segmentacao anterior
        old.n=nmh.n(end,:);
        old.m=nmh.m(end,:);
        old.h=nmh.h(end,:);
        old.Vm=Vm(end,:);

        disp([num2str(break_down),' / ',num2str(breaks),'     a=',num2str(a(i)*1e4
            ),'um'])
        break_down=break_down+1;

    end

    % exclusao dos nodos em que nao houve potencial e regressao linear para
    % o calculo da velocidade
    x=x(y~=0);
    y=y(y~=0);
    v(i)=(x(50:end-50)-x(50))*(pinv(y(50:end-50)-y(50)));
    disp(['v(a=', num2str(a(i)*1e4),'um)= ', num2str(v(i)), ' cm/ms']);

end

% grafico da velocidade de propagacao em funcao do raio da fibra
figure,
set(gcf,'Position',[350 250 600 300],'color',[223 223 223]/248)
plot(a*1e4,v,'k','Linewidth',2)
grid on
title('Velocidade do potencial de acao em funcao do raio da fibra')
ylabel('Velocidade (cm/ms)')
xlabel('Raio da fibra (\mum)')
```

Função *graphics_2.m*

```
function [] = graphics_2(x,Vm,INa,IK,Imemb,Ii,t_i,t_estimulacao,Amp_ip,option)
%[] = graphics_2(x,Vm,INa,IK,Imemb,Ii,t_i,t_estimulacao,Amp_ip,option)
% Faz o plot do potencial membranar, das correntes axial intracelular,
% de sodio, de potassio e membranar, todos em funcao da option, que pode
% ser tempo ou posicao

Vr=-60; % potencial de repouso (mV)

figure,
set(gcf,'Position',[200 200 1000 400],'color',[223 223 223]/248)

% descricao das caracteristicas da estimulacao
h = subplot(2, 2, 1);
str={'Caracteristicas da corrente de estimulacao i_p',' ',...
    ['Instante inicial de estimulacao: ',num2str(t_i),' ms'],...
    ['Duracao da estimulacao: ',num2str(t_estimulacao),' ms'],...
    ['Amplitude da estimulacao: ',num2str(Amp_ip),' mA/cm']};
text(0.05,0.5,str);
set ( h, 'visible', 'off')

% plot do potencial membranar em funcao da option
subplot(2,2,3), hold on,
plot(x,Vm,'k','Linewidth',2)
plot(x,Vr,'--b','Linewidth',2)
```

```
grid on
title('Potencial de membrana')
ylabel('V (mV)')
if(strcmp(option,'tempo'))
    xlabel('Tempo (ms)')
elseif(strcmp(option,'posicao'))
    xlabel('Posicao (cm)')
else
    error('option nao reconhecida: insira tempo ou posicao')
end
legend('V_m','V_r',-1)

% plot das correntes de sodio, potassio e transmembranar em funcao da
% option
subplot(2,2,2), hold on,
plot(x,INa,'b','Linewidth',2)
plot(x,IK,'r','Linewidth',2)
plot(x,Imemb,'g','Linewidth',2)
grid on
title('Correntes de sodio, potassio e membranar')
ylabel('I (mA / cm^2)')
if(strcmp(option,'tempo'))
    xlabel('Tempo (ms)')
elseif(strcmp(option,'posicao'))
    xlabel('Posicao (cm)')
else
    error('option nao reconhecida: insira tempo ou posicao')
end
legend('I_{Na}','I_K','I_{memb}',-1)

% plot da corrente axial intracelular em funcao da option
subplot(2,2,4), hold on,
plot(x,Ii*1e3,'b','Linewidth',2)
grid on
title('Corrente axial intracelular')
ylabel('I (\muA)')
if(strcmp(option,'tempo'))
    xlabel('Tempo (ms)')
elseif(strcmp(option,'posicao'))
    xlabel('Posicao (cm)')
else
    error('option nao reconhecida: insira tempo ou posicao')
end
legend('I_i',-1)


end
```

Função *equacao_cabo.m*

```
function [I,nmh,Vm,g] = equacao_cabo(Vm,Vr,E,dx,dt,a,Re,Ri,Cm,ip,n_iter,N,g,g_max
    ,nmh)
%[I,nmh,Vm,g] = equacao_cabo(Vm,Vr,E,dx,dt,a,Re,Ri,Cm,ip,n_iter,N,g,g_max,nmh)
%   Implementa as equacoes do cabo, seguindo a mesma logica que a funcao
%   HHmodel.m, alterando apenas I.memb e introduzindo I.i, enquanto
%   introduz tambem sealed boundary conditions

ri=Ri/(pi*a^2);      % resistencia intracelular por unidade de comprimento
% (ohm/cm)
re=Re/(3*pi*a^2);    % resistencia extracelular por unidade de comprimento
% (ohm/cm)
```

```
I.memb=zeros(n_iter,N); % vector para a corrente transmembranar (mA/cm^2)
I.Na=zeros(n_iter,N); % vector para a corrente de sodio (mA/cm^2)
I.K=zeros(n_iter,N); % vector para a corrente de potassio (mA/cm^2)
I.L=zeros(n_iter,N); % vector para a corrente de leakage (mA/cm^2)
I.ion=zeros(n_iter,N); % vector para a corrente ionica (mA/cm^2)
I.C=zeros(n_iter,N); % vector para a corrente capacitiva (mA/cm^2)
I.i=zeros(n_iter,N); % vector para a corrente axial intracelular (mA/cm^2)

for i=1:n_iter

    for j=1:N

        g.Na(i,j)=(nmh.m(i,j)^3)*nmh.h(i,j)*g_max.Na; % condutancia
        % especifica do sodio (mS/cm^2)
        g.K(i,j)=(nmh.n(i,j)^4)*g_max.K; % condutancia especifica do
        % potassio (mS/cm^2)

        % calcula I.memb e I.i aplicando sealed boundary conditions
        if(j==1)
            I.memb(i,j)=( ( Vm(i,j+1) - Vm(i,j) )/(dx^2) - re*ip(i,j) )/...
                (2*pi*a*(ri+re));
            I.i(i,j)=-( (Vm(i,j+1)-Vm(i,j))/dx - re*ip(i,j)/(2*pi*a) )/...
                (ri+re);
        elseif(j==N)
            I.memb(i,j)=( ( Vm(i,j-1) - Vm(i,j) )/(dx^2) - re*ip(i,j) )/...
                (2*pi*a*(ri+re));
            I.i(i,j)=( re*ip(i,j)/(2*pi*a) )/(ri+re);
        else
            I.memb(i,j)=( ( Vm(i,j+1) - 2*Vm(i,j) + Vm(i,j-1) )/...
                (dx^2) - re*ip(i,j) )/(2*pi*a*(ri+re));
            I.i(i,j)=-( (Vm(i,j+1)-Vm(i,j))/dx - re*ip(i,j)/(2*pi*a) )/...
                (ri+re);
        end

        I.Na(i,j)=g.Na(i,j)*(Vm(i,j)-E.Na(1))/1e3;  % corrente do sodio
                                                     % (mA/cm^2)
        I.K(i,j)=g.K(i,j)*(Vm(i,j)-E.K(1))/1e3;      % corrente do potassio
                                                     % (mA/cm^2)
        I.L(i,j)=g.L*(Vm(i,j)-E.L(1))/1e3;           % corrente leakage
                                                     % (mA/cm^2)
        I.ion(i,j)=I.K(i,j)+I.Na(i,j)+I.L(i,j);      % corrente ionica
                                                     %(mA/cm^2)
        I.C(i,j)=I.memb(i,j)-I.ion(i,j);             % corrente capacitiva
                                                     % (mA/cm^2)

        % VALORES FUTUROS
        % incrementos
        if(i~=n_iter)
            dVm=I.C(i,j)*(1e3*dt)/Cm; % (mV)
            [alpha,beta]=HHrates(Vm(i,j)-Vr); % output:(ms^-1); intput:(mV)
            dn=dt*(alpha(1)*(1-nmh.n(i,j))-beta(1)*nmh.n(i,j));
            dm=dt*(alpha(2)*(1-nmh.m(i,j))-beta(2)*nmh.m(i,j));
            dh=dt*(alpha(3)*(1-nmh.h(i,j))-beta(3)*nmh.h(i,j));

            % actualizacao dos valores futuros
            Vm(i+1,j)=Vm(i,j)+dVm; % (mV)
            nmh.n(i+1,j)=nmh.n(i,j)+dn;
            nmh.m(i+1,j)=nmh.m(i,j)+dm;
            nmh.h(i+1,j)=nmh.h(i,j)+dh;
        end
    end
end
```

```
end
```