

# Neurociencia Básica

## Práctico computacional 1: Simulador NEURON

Daniel Herrera  
dherrera@fcien.edu.uy

Leonel Gómez

*All models are wrong, but some  
are useful*

---

George Box

## 1 Objetivos

1. Introducir a los estudiantes a la simulación computacional como herramienta de estudio del sistema nervioso
2. Utilizar las simulaciones computacionales para reforzar los conceptos de excitabilidad vistos en el teórico (los estudiantes pueden usar los ejercicios para identificar qué conceptos no tienen claros)
3. Estudiar el comportamiento de modelos más complejos que los vistos en el teórico

## 2 Introducción

### 2.1 Generalidades de la simulación computacional

En las clases teóricas vimos algunos de los fenómenos biofísicos básicos que rigen el comportamiento de las células excitables y las ecuaciones que los describen. Pero aunque podemos encontrar las soluciones a estas ecuaciones para algunos sistemas muy simples (ej. célula esférica isopotencial, célula cilíndrica isopotencial), para la mayoría de los sistemas interesantes las ecuaciones no pueden resolverse de forma analítica. Sin embargo, en esos casos más complejos podemos usar esas fórmulas para simular a la neurona mediante el uso de computadoras. La simulación computacional consiste en definir las ecuaciones que describen al sistema (que estarán determinadas por su geometría y los mecanismos biofísicos presentes), y luego resolverlas en la computadora.

En este práctico usaremos el simulador NEURON, que está especializado para hacer estas simulaciones de forma masomenos sencilla. NEURON nos permite definir neuronas complejas combinando muchas partes simples, cada una guiada por su propio sistema de ecuaciones como las vistas en el teórico. Estas partes simples que son la base del NEURON consisten en pequeños cilindros (llamados *secciones* en el NEURON) que se conectan entre sí para generar geometrías complejas. Cada sección tiene su propia geometría (ancho, largo) y sus propios mecanismos biofísicos (representando diferentes tipos de canales, diferentes membranas, etc). Luego de definir todas las partes que hacen a la neurona y cómo se conectan, el NEURON simula a la neurona resolviendo numéricamente las ecuaciones que describen la evolución en el tiempo de cada cilindro.

La simulación computacional de un sistema biológico puede servir varios objetivos, como testear hipótesis sobre la utilidad y el funcionamiento de las diferentes partes de las neuronas, y suele usarse en conjunto con estudios experimentales. Los estudios experimentales permiten generar hipótesis para poner a prueba en el simulador, y este último permite generar nuevas hipótesis para volver a poner a prueba en el laboratorio experimental. Así, en las simulaciones al igual que con cualquier experimento, comenzamos con una pregunta (más o menos específica) que queremos responder. Luego debemos definir qué simulación llevar a cabo para responder esta pregunta, implementarla en la computadora, correr la simulación y analizar los datos. En este práctico veremos cómo definir e implementar un sistema a simular en el programa NEURON.

## 2.2 Lógica de uso del NEURON

### 2.2.1 Definir el modelo

NEURON nos permite definir por separado los diferentes aspectos de la neurona. A continuación explicamos los componentes que hay que definir, poniendo como ejemplo una neurona con un soma, una dendrita y un axón, a la que le inyectaremos una corriente en la dendrita para ver cuanto tarda en transmitirse el PA hasta el final del axón:

- **Propiedades de la célula:**

- **Topología/Geometría.** Primero debemos definir las secciones (es decir, los cilindros) que tendrá el sistema. En nuestro ejemplo usaremos 3 secciones: el soma, la dendrita, y el axón (si quisieramos definir más dendritas, o dendritas con ramas, cada una de estas sería una nueva sección). Para cada sección hay que definir la geometría (largo y diámetro). Luego, hay que definir la topología de la célula, que dice nos dice cómo se conectarán esas secciones. En nuestro ejemplo definiremos que tanto la dendrita como el axón se unen por uno de sus extremos al soma.
- **Mecanismos biofísicos.** Luego, debemos definir los mecanismos biofísicos que rigen en en cada sección. Los mecanismos biofísicos

son, aproximadamente, los canales que tiene la sección. Por ejemplo, podemos poner a cada sección los canales pasivos, no dependientes de voltaje que mantienen el potencial de reposo. Para nuestro ejemplo, debemos poner en el axón y en el soma canales dependientes de voltaje (ej. Hodgking y Huxley) que les permitan generar PA. Cuando agregamos estos mecanismos a cada sección de la célula, el simulador agrega nuevas ecuaciones para las mismas, simulando así el efecto de los mecanismos.

- **Estimulación.** Luego de definir la célula, podemos agregarle estímulos externos para que se active. En nuestro ejemplo, queremos ver cómo una corriente inyectada en la dendrita genera un PA. Para ello, definiremos una corriente determinada en el extremo de la dendrita, definiendo su duración y su intensidad. Otras opciones de estímulos pueden ser, por ejemplo, simulación de sinápsis en las dendritas.
- **Datos a analizar.** Finalmente, es necesario definir los datos que queremos extraer y graficar. Estos serán valores de la simulación que le pedimos a NEURON que guarde y grafique. Para nuestro ejemplo, puede servirnos una gráfica del voltaje a lo largo del tiempo para el extremo del axón, que nos permita ver la forma temporal del PA en esa región.

Cuando nos encontremos definiendo las células, también veremos un factor más que tenemos que definir: el número de segmentos de cada sección. Esto es de cierta forma la "definición" con que el programa estará modelando los movimientos de corriente dentro de la sección: cuanto más secciones, mayor definición.

### 2.2.2 Correr el modelo

Luego de definir los aspectos del sistema listados arriba, queda correr la simulación. NEURON cuenta con dos mecanismos para definir y correr las simulaciones. Uno es la GUI (por *Graphical User Interface*), que se basa en el uso de menús visibles que permiten al usuario seleccionar y modificar los elementos de la simulación de forma visual. La otra vía es la de la *línea de comando*, o de los *scripts*, que se basa en escribir las instrucciones para especificar el sistema en el lenguaje de programación del simulador. NEURON usa el lenguaje HOC, pero también se puede utilizar Python<sup>1</sup>. La primera vía puede ser más rápida para casos simples, y no requiere conocimiento de programación, mientras que la segunda vía ofrece una flexibilidad y un potencial mucho mayor. En este tutorial utilizaremos la GUI para definir nuestra simulación.

---

<sup>1</sup>Como nota de interés vale la pena comentar que, a pesar de que cuando NEURON se creó HOC era un lenguaje en uso, desde entonces cayó en desuso y ahora es un lenguaje muerto, que sólo se usa en NEURON. Por otro lado, hoy en día Python es uno de los lenguajes más populares, razón por la que se ha creado una interfaz a NEURON para el mismo.

### 3 Tutorial: Célula ‘esférica’ isopotencial

En esta sección mostramos cómo definir un modelo simple en NEURON utilizando la GUI: una célula de un único compartimiento, que es el equivalente a una célula isopotencial (sólo un soma).

#### 3.1 Instrucciones

Los parámetros que definiremos para el sistema son los siguientes:

- **Geometría:** Célula de  $30\ \mu\text{m}$  de diámetro y  $30\ \mu\text{m}$  de largo. Neuron sólo permite definir secciones cilíndricas, por lo que tendremos un soma cilíndrico
- **Biofísica:** Célula isopotencial. Mecanismos de membrana pasivos:  $V_{\text{reposo}} = -65\ \text{mV}$ ,  $g_{\text{pas}} = 0.001\ \text{S}/\text{cm}^2$ ,  $c_m = 1\ \mu\text{F}/\text{cm}^2$
- **Estimulación:** Inyección de un pulso cuadrado de corriente, de 10 ms de duración, y 1 nA de amplitud
- **Datos:** Graficar el  $V_m$  de la célula hasta 20 ms después de comenzar a inyectar el pulso, para observar su comportamiento

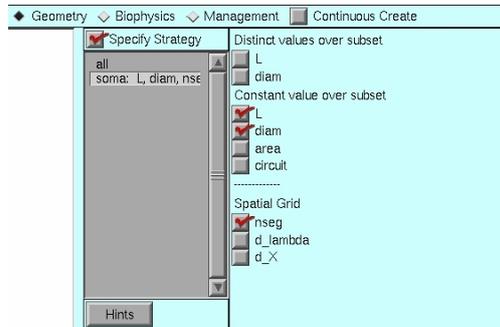
#### 3.2 Resumen de menús:

Primero iniciamos la interfaz gráfica del NEURON, yendo a la carpeta donde lo instalamos y haciendo doble click en el archivo *nrngui*. A continuación describimos cómo definir esta célula utilizando el GUI. Primero, introducimos los menús del NEURON donde se define cada aspecto de la simulación, y a los que accedemos desde la barra del *Main Menu* que se abre al iniciar NEURON:

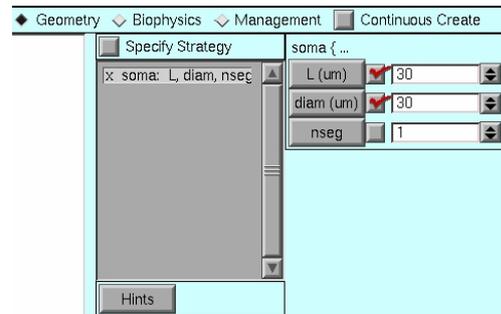
- **Build** → **Cell Builder:** Permite especificar las propiedades de la célula en sus diferentes sub-menús:
  - *Topology:* Crear las secciones y definir cómo se conectan
  - *Geometry:* Definir tamaño (diámetro y largo) de cada sección
  - *Biophysics:* Introducir y modificar mecanismos biofísicos en cada sección
- **Tools** → **Point Processes** → **Managers Point Manager:** Insertar estímulos en puntos específicos de la célula, como un clampeo de corriente, o una sinápsis
- **Graph:** Herramientas para graficar los resultados
- **Tools** → **RunControl:** Correr la simulación, y definir algunos parámetros como su duración

### 3.3 Implementación paso a paso

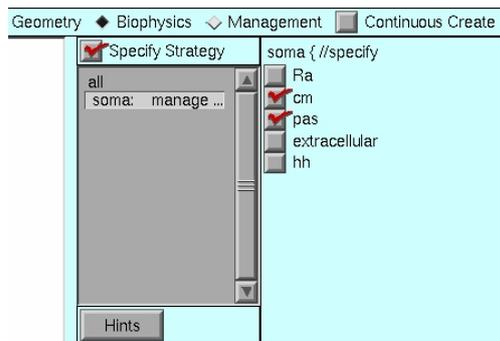
A continuación se explica paso a paso cómo usar estas herramientas para definir nuestra simulación:



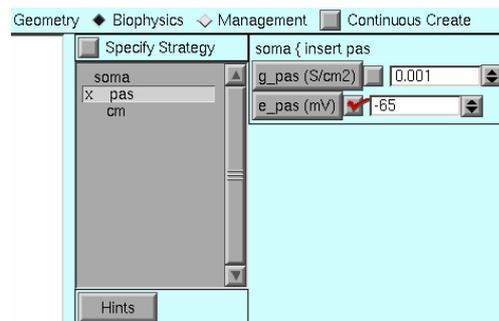
(a) Submenú *Geometry*: Seleccionar parámetros a modificar



(b) Submenú *Geometry*: Apretar *Specify Strategy* e ingresar valores



(c) Submenú *Biophysics*: Seleccionar mecanismos biofísicos a utilizar



(d) Submenú *Biophysics*: Apretar *Specify Strategy* e ingresar valores

Figure 1: Pasos para definir la geometría y biofísica de la célula desde la GUI

**1) La célula:** Primero definimos la célula. Apretamos: *Main Menu* → *Build* → *Cell Builder* y se abre el menú del *CellBuilder*, que se muestra en la Figura 1.

- **Topología.** Si vamos al menú *topology* vemos que ya se creó por default una sección llamada *soma*<sup>2</sup>. Esa sección que ya se crea cuando abrimos el "cell builder" es el único que necesitamos, así que luego pasamos a definir las características de la misma.
- **Geometría.** Para definir la geometría entramos al submenú *Geometry* (Figura 1a) El panel gris del medio muestra las secciones disponibles. De ahí seleccionamos el *soma*. A la derecha, vemos los parámetros geométricos

<sup>2</sup>Es importante saber que el nombre de la sección no le importa al programa. Aunque esta sección se llama *soma*, no tiene nada que lo caracterice como tal. También si creamos secciones nuevas, automáticamente se llaman "dend", pero bien podemos agregarles mecanismo de Hodking y Huxley y convertirlo en un axón

de la célula. Seleccionamos los que queremos modificar ( $L$ ,  $diam$  y  $nseg$ ), como se ve en la Figura 1a. Apretando el botón *Specify Strategy*, alternamos entre el menú para elegir los parámetros, y el menú para modificar valores. Apretamos el botón *Specify Strategy* y fijamos el diámetro y la longitud de la célula en  $30\ \mu\text{m}$ , y el número de segmentos en 1 (Figura 1b)<sup>3</sup>.

- **Biofísica.** Ahora vamos al submenú *Biophysics*. Nuevamente seleccionamos *soma* en la barra gris, y a la derecha vemos los valores que podemos definir. La capacitancia específica de membrana  $cm$  y resistividad axial  $Ra$  ya se setean por default en la célula, por ahora no los tocamos. El mecanismo *pas* contiene los mecanismos pasivos de membrana, y el mecanismo *hh* los de Hodgking-Huxley (Figura 1c). Seleccionamos sólo *pas* por ahora, y clickeando en *Specify Strategy* podemos ver y modificar los valores del mecanismo pasivo. Fijamos  $e_{pas}$  en  $-65\ \text{mV}$  (el  $g_{pas}$  el valor de la conductancia, o la "densidad de canales").
- **Activar los cambios** Con esto terminamos de definir las propiedades de nuestra célula. Antes de abandonar este menú, hacemos click en *Continuous Create* en el borde superior derecho. Esto hace que todos los cambios que hicimos, y que vayamos haciendo más adelante en este menú, se añadan automáticamente a la célula que se simulará.

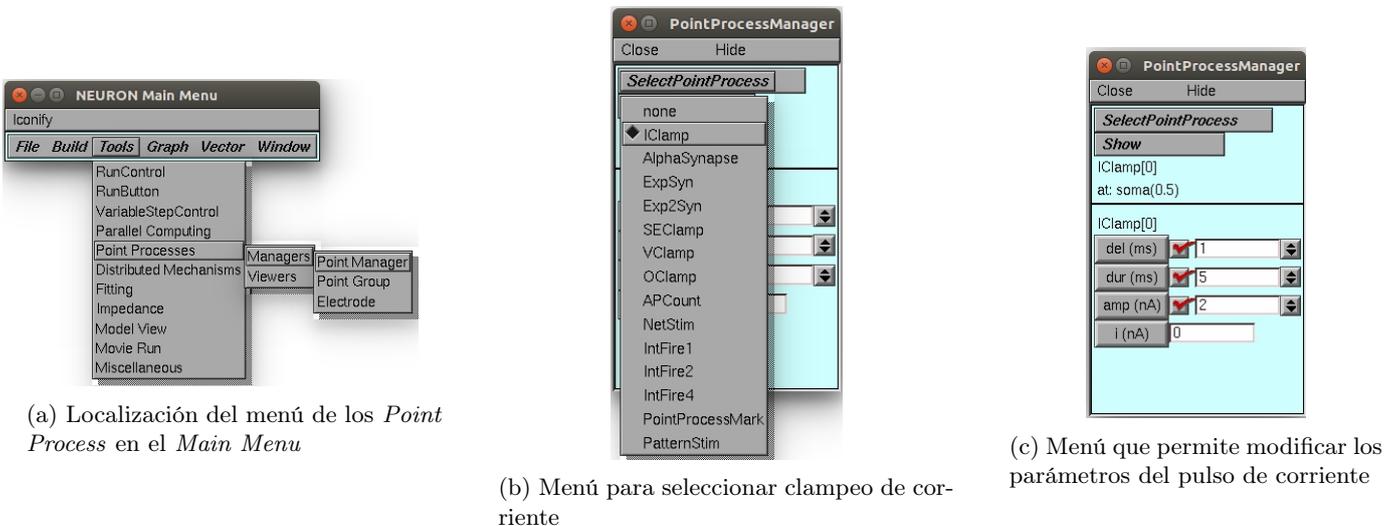


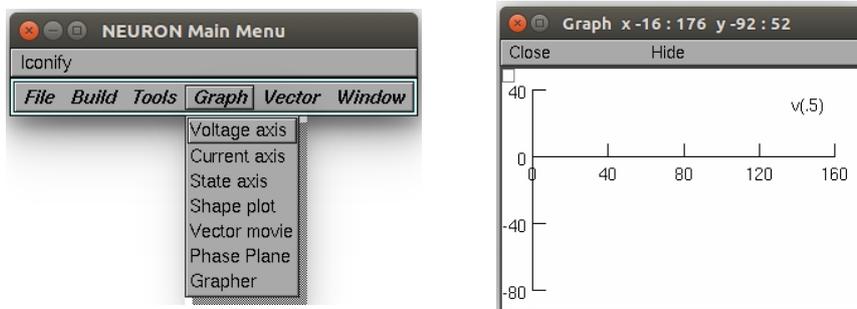
Figure 2: Pasos para incluir un pulso de corriente desde el GUI

**2) Estimulación.** A continuación especificamos el pulso de corriente que inyectaremos a la célula, en el menú *Point Process Manager*, que permite crear inyecciones de corrientes, sinápsis, etc. Accedemos seleccionando *Main Menu*  $\rightarrow$  *Tools*  $\rightarrow$  *Point Processes*  $\rightarrow$  *Managers*  $\rightarrow$  *Point Manager* (Figura 2a). Seleccionamos en la barra el tipo *SelectPointProcess* la opción *IClamp*, que es una

<sup>3</sup>Seteamos el número de segmentos en 1 porque queremos una célula isopotencial, entonces no queremos resolución espacial entre las diferentes regiones del soma. Esto es poco realista en términos biofísicos pero simplifica el modelo a estudiar

inyección de pulso cuadrado de corriente (Figura 2b). Abajo aparecen en el menú los parámetros *IClamp* que podemos definir (Figura 2c). Introducimos los valores deseados:  $dur = 10$  ms (duración del pulso),  $amp = 1$  nA (amplitud del pulso), y  $del = 3$  ms (el tiempo entre que comienza la simulación y que se inyecta la corriente). Si quisieramos usar más de una corriente (por ejemplo, una corriente inyectada en  $t=3$  ms y otra en  $t=15$  ms), podemos abrir un nuevo menú de *Point Manager* por cada corriente que queramos inyectar. Al cerrar la ventana de un Point Process se borra el mismo.

**3) Recolección de datos.** Queremos ver la respuesta de voltaje a la corriente, así que haremos una gráfica de Voltaje en en suma en función de tiempo. Para eso vamos a *Main Menu* → *Graph* → *Voltage Axis* (Figura 3a), con lo que se abre una gráfica de voltaje (Figura 3b). Esta gráfica ya está configurada para lo que queremos por lo que no debemos modificar nada más, pero más adelante veremos cómo modificarla para graficar otros parámetros.



(a) Localización de la gráfica para voltaje en el *Main Menu*

(b) Gráfica de voltaje abierta previa a la simulación

Figure 3: Localiación en el menú y aspecto de la gráfica del voltaje

**4) Guardar la sesión.** Ahora que terminamos de definir el sistema, podemos guardar la sesión en un archivo que podemos cargar en el futuro. Seleccionando *Main Menu* → *File* → *save session* se abre una ventana que nos permite guardar la sesión. Es una buena práctica de experimentación computacional guardar los modelos usados para poder replicar los resultados obtenidos.

**5) Iniciar simulación.** Con el modelo a simular especificado, queda correr la simulación. Abrimos el menú de control de la simulación abriendo *Main Menu* → *Tools* → *RunControl* (Figura 4a). Ahí hay varios parámetros que controlan las características de la simulación. *Tstop* marca hasta qué momento se hará la simulación. Lo ponemos en 20 ms. Con la opción *Init & Run*, nuestro modelo se correrá hasta el tiempo *Tstop*, y se graficarán los datos solicitados. También se la puede correr la simulación de a pasos, utilizando la opción *Continue for*, que toma el estado actual de la simulación da un paso más con la duración indicada en su caja blanca (más adelante usaremos esta opción). Ahora apriete *Init & Run* y vea el resultado de la simulación.

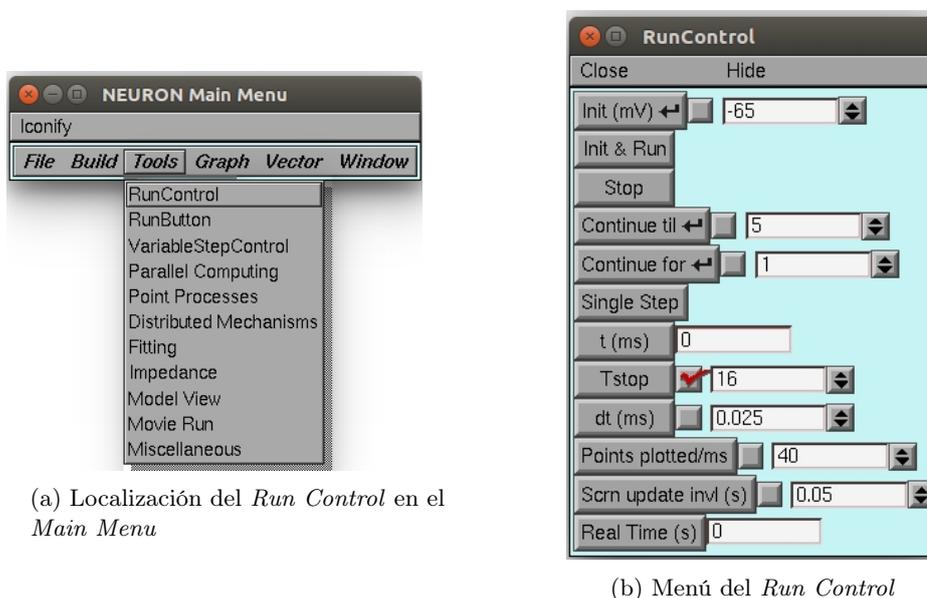


Figure 4: Ilustración de cómo acceder al control de la simulación

## Herramientas gráficas

Para resolver algunos ejercicios será necesario cambiar los parámetros de las gráficas. En este recuadro se explican algunas herramientas para hacerlo:

**Cuantificaciones rápidas en las gráficas.** Si mantenemos el click apretado sobre una gráfica obtenida de la simulación, aparece una cruz que sigue el trazo de la gráfica dibujado, y muestra entre paréntesis los valores de (X,Y) (Figura 5a). Con esto podemos estimar los valores de los parámetros estimados en los puntos de interés.

**Graficar otros parámetros en función del tiempo.** En la gráfica *Voltage axis* o *State axis*, vemos un cuadrado blanco en el eje superior izquierdo de la gráfica. Si mantenemos apretado el click sobre el cuadrado blanco aparece un menú con opciones (Figura 5b). Si soltamos el click sobre la opción *Plot what?*, podemos elegir graficar otros parámetros (y parámetros de otras secciones de la célula) (Figura 5c). Primero hacemos doble click en el primer panel sobre la sección de la célula que queremos graficar. En el panel del medio aparecen los parámetros que podemos graficar. Por ejemplo, si queremos graficar la corriente capacitiva seleccionamos *i\_cap*, y si queremos graficar la conductancia de potasio de HH, seleccionamos *gna\_hh*.

**Ajustar la escala de las gráficas** Cuando hacemos una gráfica en neuron, hay una sola escala en el eje de Y para todos los parámetros graficados (aunque los mismos tengan unidades distintas). Por ello, es conveniente graficar sólo parámetros de escala similar en una misma gráfica. Entonces si seleccionamos un nuevo parámetro, a veces hay que borrar los viejos. Esto se logra seleccionando la opción *Delete* en el menú del cuadrado blanco, y luego presionando

en la gráfica sobre el nombre del parámetro que queremos borrar (los que aparecen a la derecha en el gráfico). También, a veces ocurre que al graficar un parámetro no se ve nada. Esto suele ser porque la escala de la gráfica no está acorde a la del parámetro, lo que se solucionamos cambiando la escala de la gráfica seleccionando *Cuadrado blanco*  $\rightarrow$  *View*  $\rightarrow$  *View = plot*.

**Cambiar el color de los parámetros graficados.** En cualquiera de las gráficas, en el menú del cuadrado blanco si seleccionamos la opción *Color/Brush*, aparece un panel de colores. Seleccionando un color y luego el nombre del parámetro en la gráfica, cambiamos el color con que se grafica el mismo (también se cambia el color del nombre).

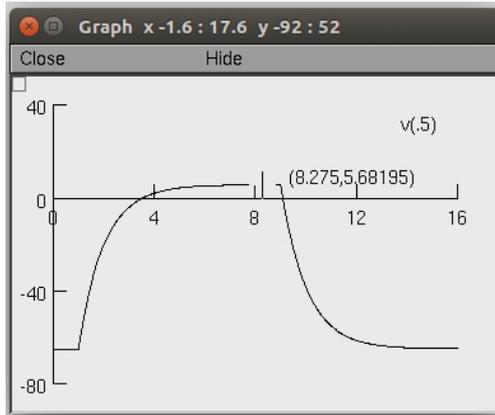
### 3.4 Ejercicios: Célula esférica

Utilizando las herramientas de la sección anterior, haga lo siguiente:

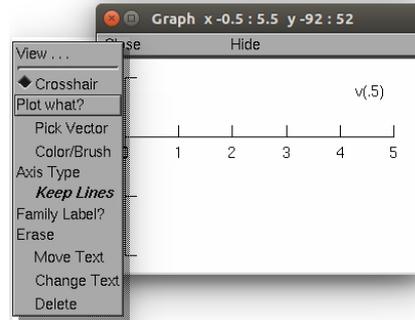
1. Para la célula generada en el tutorial, estime la constante de tiempo  $\tau$  (cuanto tarda la membrana en cargarse hasta el 63% de su valor final) para 3 amplitudes de corriente distintas. ¿Depende la constante de tiempo de la amplitud de la corriente? (consultar el recuadro ‘Herramientas gráficas’, el título ‘Cuantificaciones rápidas en las gráficas’)
2. Estime la resistencia de entrada de la célula (definida como  $R_{input} = \Delta V_m(\infty)/I_{inyectada}$ ). ¿Qué pasa con la misma si cambia el tamaño de la célula?
3. La vaina de mielina alrededor del axón disminuye la capacitancia del mismo (la cantidad de carga que necesita para generar un cambio de voltaje específico). La capacitancia específica ( $cm$ ) default de NEURON es  $1 \mu\text{F cm}^{-2}$ , cambíela a  $0.1 \mu\text{F cm}^{-2}$ . ¿Cómo cambia el comportamiento de la célula? ¿Piensa que  $\tau$  se vió afectada? Intente explicar el comportamiento de comportamiento en términos de la corriente capacitiva ( $I_C$ ) y la corriente a través de los canales pasivos ( $I_{Pas}$ ). Cuando termine devuelva la capacitancia de la célula a  $1 \mu\text{F cm}^{-2}$ .

Agregue el mecanismo biofísico de HH a la célula (menú de *Biophysics* en el *Cell Builder*), con sus parámetros en los valores de default. Reduzca la duración del pulso de corriente a 1 ms

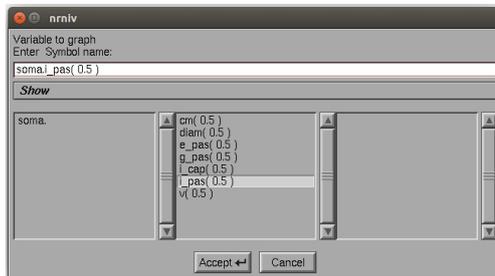
3. Estimar el voltaje umbral para la generación del potencial de acción, modificando la amplitud del pulso de corriente (use hasta 4 cifras decimales en la amplitud del pulso). Repita para un pulso de corriente de 12 ms de duración. ¿Cambia el umbral? Intente explicar la observación en términos de los canales de  $Na^+$  de HH.
4. Abra una nueva gráfica (*Main Menu*  $\rightarrow$  *Graph*  $\rightarrow$  *State axis*). Grafique en una misma ventana los parámetros compuerta en el soma ( $m\_hh$ ,  $h\_hh$ ,



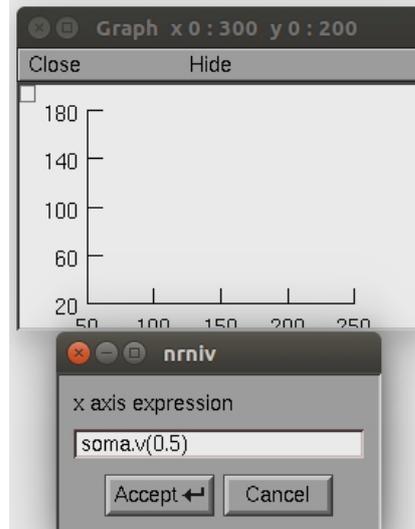
(a) Puntero que indica valor en la gráfica en formato (X,Y), que se muestra al mantener el click apretado



(b) Ubicación del menú para seleccionar nuevos parámetros a graficar



(c) Menú que permite seleccionar parámetros a graficar. En el primer panel se selecciona la sección, en el segundo el parámetro



(d) Gráfica de *Phase Plane* que se abre al seleccionarlo en el *Main Menu*, con la ventana en la que ingresar el parámetro del eje de X

Figure 5: Herramientas extra

- n\_hh), utilizando un color para cada una (vea el recuadro de herramientas gráficas).
5. En otra ventana grafique las conductancias dependientes del voltaje gk\_hh y gna\_hh, un color para cada una.
  6. Observando las gráficas con las conductancias y los parámetros de compuerta, simule 3 inyecciones de pulsos de corriente de  $-1$  nA con 3 duraciones distintas: 2 ms, 4 ms y 8 ms. Si siguió los pasos descritos, debería observar que se genera un potencial de acción en la última simulación, lo que es contraintuitivo dado que la corriente es hiperpolarizante. Intente explicar esta observación en términos de los canales de HH y de las corrientes activas de  $Na^+$  y  $K^+$ .<sup>4</sup>
  7. Manteniendo las gráficas abiertas, cambie la duración del pulso a 1 ms. Busque, utilizando sólo una cifra decimal, la mínima amplitud de corriente (usando corrientes positivas) que genera un PA. Introduzca un nuevo pulso de corriente, comenzando 7 ms después que el primer pulso, y con una duración de 1 ms. Utilice la misma amplitud que en el primer pulso. ¿Genera el segundo pulso un PA? Intente explicar esta observación. Ahora reduzca la amplitud del primer pulso en 0.1 nA, lo que debería eliminar el primer PA. ¿El segundo estímulo sigue por debajo del umbral? Discuta brevemente las implicancias de estos resultados para la activación de las neuronas.
  8. Como último ejercicio, planteese alguna pregunta o dato que le interese averiguar, que pueda realizar con lo visto del NEURON. Plantee la pregunta, las simulaciones que hizo para responderla, y una breve reflexión sobre los resultados. Como ayuda creativa, se listan abajo algunos datos aleatorios sobre el comportamiento de las neuronas, y la indicación de algunos otros parámetros de la simulación que puede modificar. Puede, si así quiere, basar su pregunta/simulaciones en esa información.

Datos interesantes:

- Una clasificación funcional que se hace en algunas neuronas de mamíferos es si son de tipo I o de tipo II. Se diferencian en que cuando les inyectamos una corriente, las de tipo I aumentan su frecuencia de disparo (i.e. de generación de PA) de forma continua a medida que aumentamos la amplitud de corriente, mientras que las de tipo II muestran un salto, pasando de frecuencia 0 a una determinada amplitud de corriente a frecuencias elevadas a una amplitud ligeramente mayor.
- La constante de tiempo  $\tau$  determina cuanto tiempo tarda en perder una neurona una despolarización que se le haya generado con un estímulo. Esto determina, en parte, cuál es la ventana temporal sobre la que integra sus entradas, o de cierta forma, su memoria reciente.

---

<sup>4</sup>Es importante que el parámetro e\_pas esté en  $-65$  mV, y que esté usando el *nrnogui*, y no el *nrndemo*

- Existen neuronas llamadas neuronas resonadoras, que luego de que son activadas pueden continuar generando PA periódicos en ausencia de estimulación.
- La presencia de canales de  $K^+$  de HH no es esencial para la generación de un PA, pero sí determina algunas de sus características.
- La frecuencia con que las neuronas generan PA es el parámetro más estudiado en codificación neuronal (es decir, el parámetro que se suele medir para ver a qué responde una neurona). Una característica importante de una neurona es el rango dinámico al que responde, o dicho de otra forma, el rango de valores del estímulo en el que su frecuencia de disparo sigue cambiando con los cambios del estímulo. En neuronas visuales, por ejemplo, el rango dinámico puede ser el valor de intensidades de luz en el que funciona la neurona (ej. con muy poca, o mucha luz, no responde o se satura respectivamente). En neuronas aisladas, puede medirse el rango dinámico en términos de una corriente inyectada.
- Tanto el voltaje umbral del PA como el período refractario que tarda una célula en volver a poder generar un PA dependen de la historia reciente de la célula.

Otros parámetros a cambiar, además de los ya vistos:

- Tamaño del soma
- Conductancia y valor de equilibrio de la corriente pasiva
- Conductancias máximas de las corrientes de sodio y de potasio del mecanismo de HH ( $gnabar\_hh$  y  $gkbar\_hh$  respectivamente)
- El tipo de estímulo. Seleccionando *AlphaSynapse* se simula una sinápsis en lugar de una corriente inyectada. Ahí el parámetro onset es cuando se activa la neurona, tau es la constante de tiempo con que se activa y desactiva la sinápsis, gmax es la conductancia máxima de la sinápsis, y e es el voltaje de reversión de la sinápsis (el voltaje al que tiende)
- La cantidad, duración, magnitud de los estímulos

## 4 Material extra

### Discretización en la simulación

Supongamos que queremos simular un axón de forma realista en condiciones complejas. El comportamiento del axón está regido por varias ecuaciones diferenciales, por ejemplo:

*Ecuación del cable*

$$\frac{1}{r_i} \frac{\partial^2 V_m}{\partial x^2} = c_m \frac{\partial V_m}{\partial t} + \frac{V_m}{r_m} i$$

### Modelo de Hodking y Huxley

$$I_K = n^4 \bar{g}_K (V_m - V_K)$$
$$\frac{dn}{dt} = \alpha_n (1 - n) - \beta_n n$$

con  $\alpha_n$  y  $\beta_n$  dependientes de  $E$ , y con ecuaciones similares para  $I_{Na}$ , con coeficientes  $m$  y  $h$  dependientes del voltaje y del tiempo, multiplicando la conductancia máxima.

Entonces dividimos el axón en muchos pequeños cilindros<sup>a</sup> de largo  $\Delta x$ , y simulamos a cada uno de estos como un fragmento isopotencial. Cada cilindro tiene en cada momento sus valores de  $V_m$ ,  $I_k$ ,  $n$ ,  $m$ ,  $h$ , etc. Haciendo esto discretizamos el espacio. También discretizamos el tiempo, en lugar de resolver lo que ocurre en todo instante, nos limitamos a resolver el estado del sistema en los tiempos múltiplos del intervalo  $\Delta t$  (ej. si  $\Delta t = 0.01ms$ , hallamos el estado del sistema en los tiempos 0ms, 0.01ms, 0.02ms, 0.03ms, etc).

Dada esta discretización, ahora tenemos aproximaciones a las ecuaciones diferenciales, por ejemplo:

$$\frac{\partial V_m}{\partial t}(x) \approx \frac{V_m(x, t + \Delta t) - V_m(x, t)}{\Delta t}$$
$$\frac{\partial^2 V_m}{\partial x^2}(x, t) \approx \frac{V(x + \Delta x, t) - 2V(x, t) + V(x - \Delta x, t)}{\Delta x^2}$$

donde  $V_m(x, t)$  es el valor del voltaje del fragmento del cilindro en posición  $x$ , en el tiempo  $t$ ,  $V_m(x, t + \Delta t)$  es el voltaje del mismo fragmento en el próximo tiempo (luego del intervalo  $\Delta t$ ), y  $V(x + \Delta x, t)$  y  $V(x - \Delta x, t)$  son los potenciales de los cilindros adyacentes al cilindro en posición  $x$ , en el tiempo  $t$ .

Entonces la simulación parte de una condición inicial, y en cada instante, a partir de los parámetros de cada segmento, resuelve el sistema de ecuaciones relevantes, y así determina qué parámetros tendrá cada segmento en el siguiente paso  $t + \Delta t$ . De esta forma no hay que encontrar la solución a las complejas ecuaciones diferenciales del sistema, con derivadas e integrales, sino simplemente resolver muchas ecuaciones algebraicas que nos dicen cómo cambia el sistema paso a paso. Si agregamos o quitamos corrientes u otros factores a lo largo de la simulación, estas entran simplemente como nuevos parámetros en estos sistemas de ecuaciones.

Mientras menores sean  $\Delta x$  y  $\Delta t$ , más fino es el modelo, y más se acerca al caso continuo ideal, pero también mayor es el costo computacional, ya que la computadora tiene que resolver más ecuaciones.

<sup>a</sup>Pero siempre una cantidad finita de pequeños cilindros