

# USER'S MANUAL

---

*ASTRO 1.0.*

*Astrocyte *in silico**

© University College London, MIT licence

2017

## TABLE OF CONTENTS

INTRODUCTION.....	4
Key system and software requirements .....	4
The strategy of building the model.....	4
Experimental data or estimates desirable to build a realistic astrocyte model.....	6
GETTING STARTED.....	7
Installing ASTRO.....	7
Setting up and running ASTRO: Main regimes of modelling.....	8
Important. <i>Each of these regimes can operate independently of each other. The user work can be limited to one of these regimes only.</i> .....	9
Setting up and running ASTRO: Worker computer preparation .....	9
ADOPTING EMPIRICAL DATA ON ASTROCYTE MORPHOLOGY .....	10
Main cell processes and branches.....	10
Generating nanoscopic processes of astroglia.....	10
Uploading a 3D-reconstructed fragment of astroglia.....	11
Storing a representative sample of 3D-reconstructed nanoscopic processes .....	11
Transforming 3D-reconstructed processes into sets of disk-shaped compartments .....	14
Biophysical comparison of original and transformed structures: a Monte Carlo test.....	15
Statistics of NEURON-compatible cylindrical compartments .....	18
GENERATING COMPLETE ASTROCYTE MORPHOLOGY .....	19
Preparing Host computer (under Windows) .....	19
Generating nanoscopic processes .....	21
Setting limiting parameters for nanoscopic process morphology.....	22
Adjusting tissue-filling (volumetric) properties of nanoscopic process morphology.....	23
SIMULATING ASTROGLIAL FUNCTION .....	24
FRAP experiments: probing and adjusting inter-cellular connectivity of astroglia .....	24
FRAP with round-spot bleaching.....	25
FRAP with linear bleaching.....	25
Probing membrane mechanisms of astroglia .....	26
Steady-state membrane voltage distribution.....	27
Electric stimulation settings.....	27
Frequency stimulation .....	28

Simulating membrane voltage in response to local current hotspots.....	29
Modelling intracellular calcium dynamics.....	29
Main Ca <sup>2+</sup> simulation menu.....	30
Ca <sup>2+</sup> wave simulations.....	32
Simulating astroglial glutamate transporters.....	33
Simulating astroglial intracellular potassium concentration.....	34
<b>SYSTEM PREPARATIONS FOR HIGH-END CALCIUM SIMULATIONS .....</b>	<b>36</b>
Preparing Worker computer / cluster (HPC, OS Linux) for Ca <sup>2+</sup> simulations.....	36
All files from directory HPC should be downloaded on the cluster, keeping the structure of directories unchanged. ....	36
All files from directory HOST should be downloaded on the local computer (OS Windows), keeping the structure of directories unchanged.....	36
Preparing Host computer (client, OS Windows) for Ca <sup>2+</sup> simulation .....	36
Notes .....	38

## INTRODUCTION

Electrically non-excitatory astrocytes are able to transduce, integrate and propagate physiological signals by engaging multi-modal fluctuations and non-dissipative diffusion waves of intracellular  $\text{Ca}^{2+}$ .

Deciphering this type of signalling, however, poses a conceptual challenge because it requires an understanding of molecular interactions in the complex system of ultrathin processes which constitute the bulk of astrocyte morphology. How a particular mode of intracellular  $\text{Ca}^{2+}$  signalling could generate a particular physiological message remains poorly understood. Gaining mechanistic insights into astrocytic physiology will be therefore difficult without a detailed biophysical model that recapitulates realistic astrocyte morphology and known cellular mechanisms.

### Key system and software requirements

#### Full version:

1. Two computers: the Host computer operating under Windows, and Worker (remote) computer cluster operating under Linux.
2. Basic preinstalled software: MPIC++ (Worker) (<https://www.open-mpi.org/software/ompi/v3.0/>), MATLAB not older than 2013 (Worker and Host) and NEURON 7.0 (Worker and Host) <https://neuron.yale.edu/neuron/download>
3. Platform: Linux and Windows. Optional modes of operation: sequential and parallel (MPI) computing.

**Important:** If a user has not MatLab, the user can install MATLAB Runtime. It is a free package that can be downloaded from the webpage:

<https://www.mathworks.com/products/compiler/mcr/>.

#### Basic version

### The strategy of building the model

The morphology of brain astroglia differs significantly from that of nerve cells, notably in two main aspects. Firstly, it features a complex system of nanoscopic processes that originate from several primary (and perhaps secondary and some tertiary) processes filling the tissue volume in between all process branches. Because the size of these nanoscopic processes is beyond the diffraction limit of conventional optical microscopy (including two-photon excitation imaging), they are normally detected as a cloudy structure surrounding thicker branches. Secondly, tissue domains occupied by individual astroglia do not overlap thus leaving individual cells to fill in all the tissue space that corresponds to the cell territory. Finally, astrocytes from different brain regions could have different basic biophysical characteristics, such as membrane unit conductance and axial cytoplasm resistance, let alone the plethora of known and unknown ion channel and receptor mechanisms. Thus, building an astrocyte model that recapitulates real astroglia morphology and known astroglial function consists of three main stages (Fig. 1).

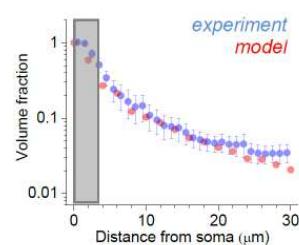
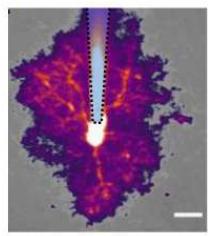
**The first stage** is to reproduce *in silico* the geometry of all main cell processes that could be identified and traced in an optical microscope. Here the process is somewhat similar to well-established routines of

## Experimental measurements

## Replication *in silico*

### Stage 1. Gross morphology: Principal processes

3D tracing, 2PE microscopy *in situ*



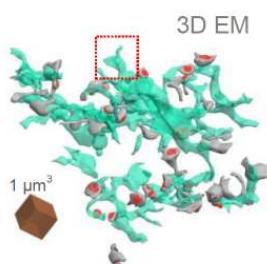
Realistic cell arbour 'skeleton' imported



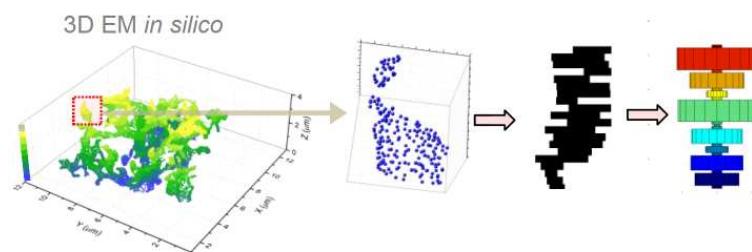
### Stage 2a. Nanoscopic morphology: generation of ultrathin processes

3D EM of astroglial structures

Biophysically equivalent nano-processes



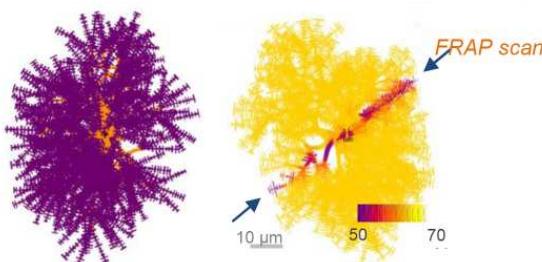
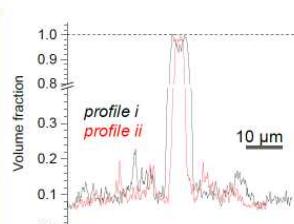
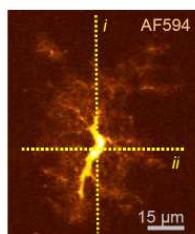
3D EM



### Stage 2b. Nano-to-macro morphology: tissue volume filling

3D-EM + 2PE: volume/surface landscape

Generating and testing nanoscopic architecture



### Stage 3. Incorporating known functions, on the scale from 0.01-100 μm

Glutamate uncaging probing

Exploring cell physiology

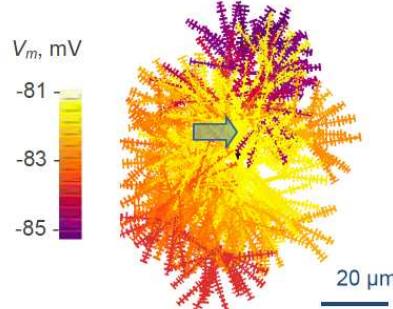
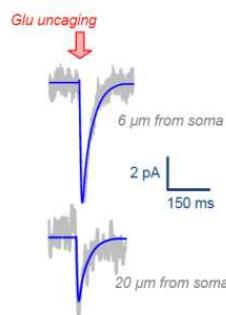
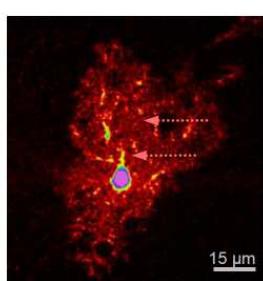


Figure 1. Astroglia *in silico*: summary of the modelling approach

neuronal reconstruction including: (i) 3D optical reconstruction of the cell filled in with a fluorescent indicator, in live or fixed tissue; (ii) tracing the soma and all identifiable (thick) branches while cutting off the 'cloudy' area representing nanoscopic processes; and (iii) importing the main-branch 3D geometry (including variable thickness of individual branches) into NEURON. This procedure could be performed using various available software types that provide digitised output morphology compatible with NEURON, such as NEUROLUCIDA or Vaa3D (see sections below for detail).

**The second stage** is to generate *in silico* a complex system of nanoscopic processes that would originate from the thicker branches (reconstructed during the **first stage**) filling the space between them. Because the exact reconstruction, *in situ* or *in silico*, of many hundreds or thousands of ultrathin cell branches is not technically feasible, the method adopted here is to generate such branches automatically based on the statistical morphometric data obtained using 3D electron microscopy and the volumetric data obtained with two-photon excitation imaging. Furthermore, complex and irregular 3D geometry of nano-processes reconstructed *in situ* is transformed into NEURON-compatible shapes involving disk-shaped and cylindrical compartments: importantly, the latter shapes are tested and validated for biophysical compatibility (key electrodynamic and diffusion properties) with the original irregular 3D shapes.

**The third stage** is to adjust basic biophysical properties of the astroglial cytoplasm and the plasma membrane using the data of electrophysiological or optical/ ligand probing tests *in situ*.

Thus, a realistic cell model that represents the astroglia (population) under study will require a set of experimental data, either published separately or obtained *de novo*, that are sufficient to carry out the above stages of model construction.

## Experimental data or estimates desirable to build a realistic astrocyte model

1. A 3D reconstructed tree of main identifiable astroglial processes importable into NEURON. Alternatively, this could be an artificially generated cell arbour with the branching pattern and branch diameters representing the average (typical) astrocyte from the population of interest.
2. A sample (20-50) of nanoscopic astroglial processes reconstructed using 3D (serial-section) EM, with rendered surface co-ordinates. This sample will be used to obtain statistical properties of the ultrathin processes to be generated in the model.
3. Average tissue volume fraction occupied by astroglia, as distributed radially from the soma to the cell edges. This data set is obtained from two-photon excitation measurements *in situ* (or from published data).
4. The mean membrane surface density and the surface-to-volume fraction values obtained from 3D reconstructions of nanoscopic astroglial processes.
5. The characteristic I-V curve (somatic patch-clamp, square-pulse current injections) for the astroglia of interest, other (optional) available functional data such as electrical responses to glutamate uncaging or changes in extracellular potassium, intracellular calcium wave speed, etc.

## GETTING STARTED

### Installing ASTRO

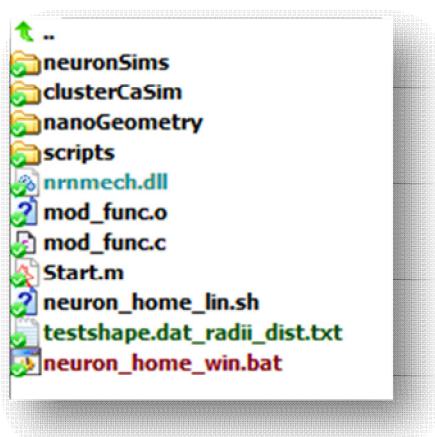
The latest installation version can be downloaded from (<https://github.com/LeonidSavtchenko/Astro>).

The web page with the software product looks like this.

The screenshot shows the GitHub repository page for 'Astro' by LeonidSavtchenko. The repository name is 'Astro'. Key statistics displayed are 41 commits, 1 branch, 0 releases, and 2 contributors. The 'Code' tab is selected. A list of recent commits is shown:

Commit	Description	Date
clusterCaSim	Update model	3 days ago
nanoGeometry	Update model	3 days ago
neuronSims	Added windows for leaves	2 days ago
scripts	Add files via upload	4 months ago
.gitignore	Deleted build artifacts and other files created during the simulation.	4 months ago
Manual ASTRO.v1.pdf	New Version of Manual	3 days ago
README.md	Update Readme	17 hours ago
Start.m	Update model	3 days ago
neuron_home_lin.sh	Add files via upload	4 months ago
neuron_home_win.bat	Add files via upload	4 months ago
testshape.dat_radii_dist.txt	Update model	3 days ago

In order to download Astro, the user must press the green key “Clone or download” and save Download.Zip at any place of the personal computer operating under Windows. Then the archive must be open and save on the Host computer (Windows) keeping the following folder structure (Fig. 2).

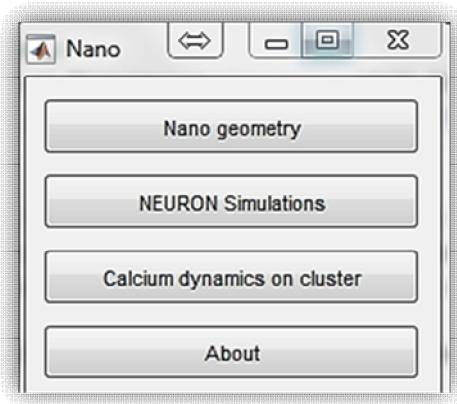


## **Figure 2. Folder structure of Astro 1.0 on the Host computer under Windows.**

The full version of Astro requires preliminary preparation of a computer cluster that runs under Linux. See details in the section “**Preparing Worker computer / cluster (HPC, OS Linux) for Ca<sup>2+</sup> simulations**” and “**Setting up and running ASTRO: Worker computer preparation**”

To get started with the 'Astro' in the local version, the Host has to have a MATLAB (2013 or later), NEURON (7.0 or later).

Executing *Start.m* in Matlab environment preinstalled on the host computer, should generate an introductory menu (Fig. 3).



**Figure 3. Introductory menu**

The menu provides the choice of the three main modelling regimes, 'Nano Geometry', 'NEURON Simulations', and 'Calcium Dynamics on Cluster'. The two former regimes can be run routinely on a stand-alone Host computer whereas the 'Calcium Dynamics on Cluster' regime, which deals with resource-consuming simulations of intracellular astroglia Ca<sup>2+</sup> requires the *Worker computer / cluster*, as explained below.

### **Setting up and running ASTRO: Main regimes of modelling**

1. **Constructing astrogial morphology** ('*Nano-geometry*', Host computer only required). Importing the 3D main-branch morphology of astroglia into NEURON; generating, within the NEURON environment, the nanoscopic astroglial protrusions that reflect experimental data. This regime can be run separately from other parts of ASTRO, it requires a Host computer with MATLAB (2012 or later) and NEURON (7.2 or later, <https://neuron.yale.edu/neuron/download>) installed under Windows 7 or 10.
2. **NEURON-based simulations of membrane mechanisms** ('*NEURON simulations*', Host computer only required). Further adjustment of the astrocyte morphology (in accord with volumetric data); populating the cell with membrane mechanisms; setting up simulation configurations and protocols. This regime can be run separately from other parts of ASTRO, it requires a Host computer with

MATLAB (2012 or later) and NEURON (7.2 or later, <https://neuron.yale.edu/neuron/download>) installed under Windows 7 or 10.

3. **Simulating full-scale  $\text{Ca}^{2+}$  dynamics ('Calcium Dynamics on Cluster')**, Host and Worker computers normally required). Design and simulations of longer-term (seconds to minutes) intracellular calcium dynamics within realistic geometry using the cluster / cloud-based parallel computing. This regime can be run separately from other parts of ASTRO, it requires a Host computer with MATLAB (2012 or later) and NEURON (7.2 or later, <https://neuron.yale.edu/neuron/download>) installed under Windows 7 or 10, and Worker computer / cluster operating under Linux and with preinstalled NEURON ([https://neuron.yale.edu/neuron/download/compile\\_linux](https://neuron.yale.edu/neuron/download/compile_linux)) and MPI.

Briefly, in this regime, the user working on the Host computer with MATLAB creates a MAT-file containing instructions for computation; uploads this file to the Worker cluster and launches there the simulations of astroglial  $\text{Ca}^{2+}$  dynamics (independently of the Host computer). The Host computer connects intermittently to the Worker time (a) to monitor computation progress, and (b) to download intermediate simulation results that are displayed and saved in MATLAB. Once simulations have been completed, the MATLAB module running on the Host computer downloads the output MAT-file and visualises the computation results.

**Important.** *Each of these regimes can operate independently of each other. The user work can be limited to one of these regimes only.*

#### Setting up and running ASTRO: Worker computer preparation

The files from GITHUB directory [Astro/clusterCaSim/hpc/](#) are to be uploaded to the cluster, with the folder structure kept unchanged (Fig. 4). Importantly, the Worker computer should have NEURON (7.2 or later) installed beforehand.

Name	Size
..	
model	
results	
scripts	
user_geometry	
x86_64	
hostfile_BusyMaster	1 KB
hostfile_IdleMaster	1 KB
init.hoc	3 KB
mcomplex.dat	1 KB
multipsplit.hoc	5 KB
params.hoc	2 KB
perf.dat	1 KB
perf.hoc	3 KB
pltsplit.hoc	2 KB
printtopology.hoc	1 KB
trajec.hoc	3 KB

Figure 4. Folder structure for the Worker computer/ cluster0

To connect the Host and the Worker, the user should run an executable file ***params.bat*** located on the host computer:...[\*\*\*/host/Core/scripts/win-lin/params.bat\*\*\*](#)

The program path is defined inside the file “params.bat” has to be set to the cluster IP address (HEADNODEIP), the password of the cluster login (PASSWORD), the cluster login (LOGIN) and location of **HPC** file on the cluster (HEADNODEWORKERDIR ).

A piece of the file “params.bat” that describes the program path between the host and file.

```
*****  
set HEADNODEIP=144.82.46.83  
set LOGIN=my_login  
set PASSWORD=my_password  
set HEADNODEWORKERDIR=/home/********/hpc  
*****
```

## ADOPTING EMPIRICAL DATA ON ASTROCYTE MORPHOLOGY

### Main cell processes and branches

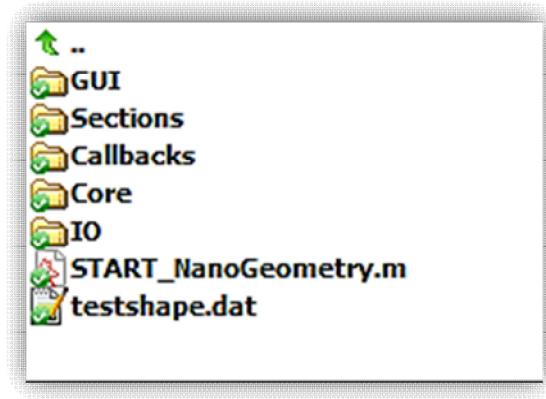
The first stage of morphological reconstruction deals with the main processes of the astrocyte under study. These are thought of as the processes that can be identified and traced in an optical (two-photon excitation imaging) microscope. The procedure to 3D-reconstruct such processes and to import the resulting data file into NEURON is, in large part, similar to the well-established 3D reconstruction routines for nerve cells. The software (and detailed instructions) providing an output compatible with NEURON include, for instance, Vaa3D (Allen Institute) <http://www.alleninstitute.org/what-we-do/brain-science/research/products-tools/vaa3d/> or NEUROLUCIDA (MBF Bioscience) <http://mbfbioscience.com/neurolucida>.

Alternatively, for exploration purposes the branch structure can be downloaded from the database of nerve or neurogliaform cells NEUROMORPHO (<http://neuromorpho.org/>), or built independently using the function in NEURON function 'Create a cell'.

### Generating nanoscopic processes of astroglia

The MATLAB code '**Nano**' was designed (a) to analyse the astrocyte nanostructure obtained via ultrathin serial-section 3D reconstruction, (b) to transform it into NEURON-compatible shapes consisting of coaxial disks / cylinders, and (c) to validate biophysical compatibility of the latter shapes with the original 3D shapes.

The Nano code can either be activated from the introductory menu (Fig. 3) or launched separately from the folder **nanoGeometry** (Fig. 5):



**Figure 5. Folder structure for the 'nanoGeometry' containing MATLAB code of Nano simulation for ASTRO.1.0**

#### Uploading a 3D-reconstructed fragment of astroglia

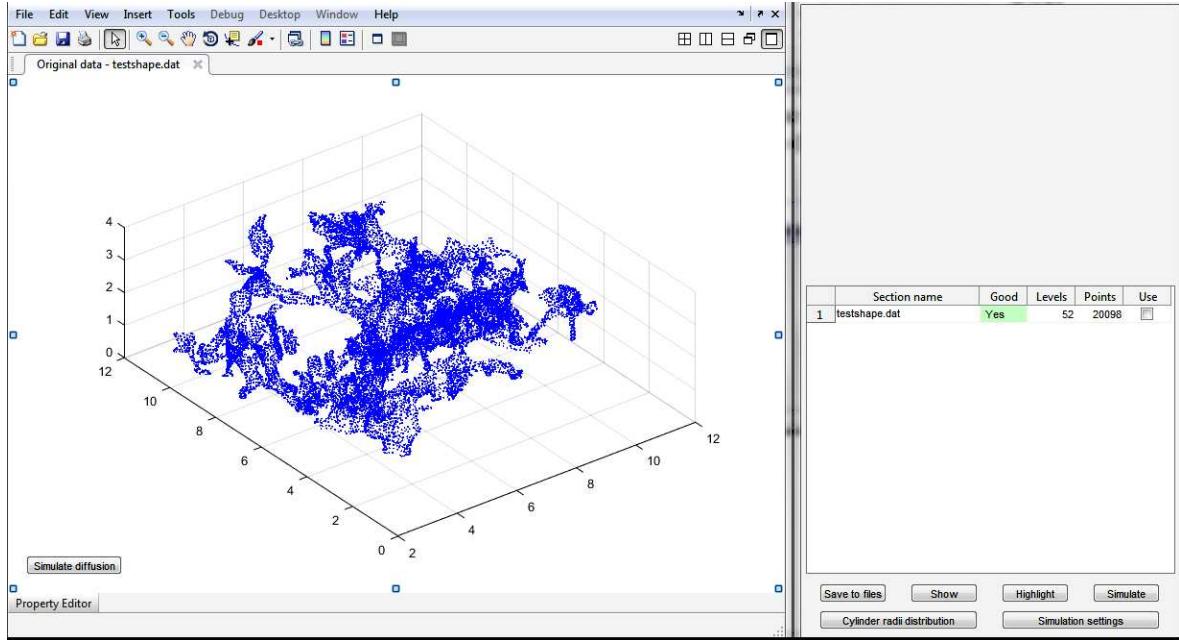
By default, the **nanoGeometry** folder (Fig. 5) hosts a file titled '*testshape.dat*'. This ASCII-format file contains XYZ surface coordinates of a 3D-reconstructed (serial-section EM) fragment of an astrocyte representing the typical occurrence of nanoscopic astroglial processes in the CA1 neuropil of the hippocampus. The co-ordinates of the cell process surfaces are represented by a (pseudo-random) scatter of reference points on the surface, at a density of  $100\text{-}300 \mu\text{m}^{-2}$ .

These data provide the basis for obtaining key statistical properties of the nanoscopic shapes to be transformed into NEURON-compatible, cylinder-compartment-based nanoscopic astroglial processes.

Whilst the nano-architecture of astrocytes in CA1 neuropil could be characteristic of various brain regions, the user is encouraged to generate their own file '*testshape.dat*' containing the 3D data representing the astroglia of interest. The method of obtaining the surface point scatter in the 3D reconstruction data could vary depending on the software involved.

#### Storing a representative sample of 3D-reconstructed nanoscopic processes

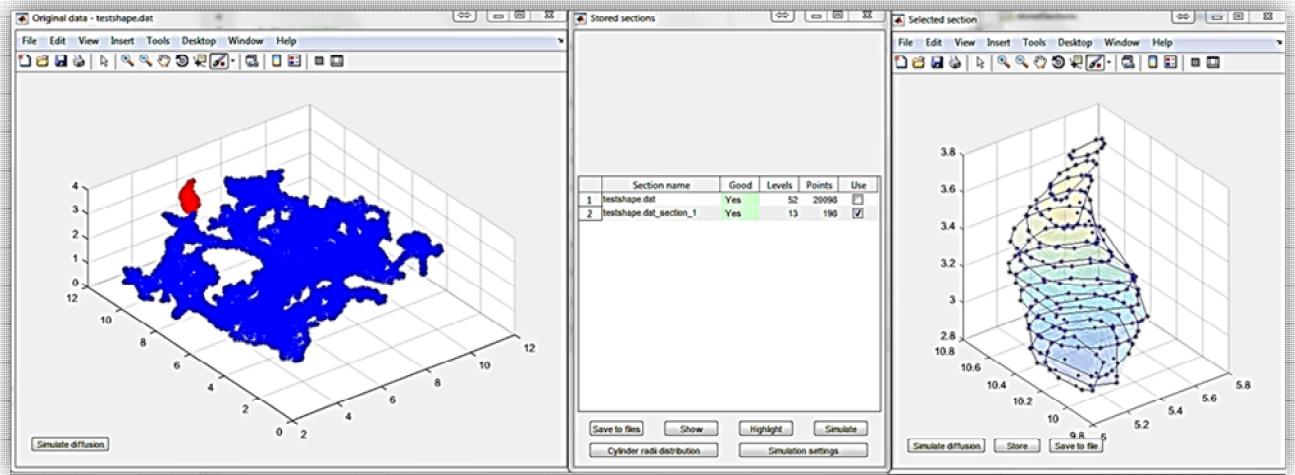
Once the '*testshape.dat*' file has been in place, the **nanoGeometry** regime can be launched by launching the file '*START\_NanoGeometry.m*'. This prompts two adjacent windows (Fig. 6).



**Figure 6.** Windows panels: selection and analyses of nanoscopic processes extracted from the 3D data file 'testshape.dat' (left), with a list containing the selected structures (right).

The left window (Fig. 6) displays the digital data from the 'testshape.dat' file.. The right window (Fig. 6) indicates stored selections. Here the aim is to select as many as possible representative nanoscopic astroglial processes, based on which the statistics of cylindrical compartments will be generated (to be used in the NEURON script).

The 3D structure could be rotated to view and select individual processes using the 'Brush' selection menu: the process is selected once it has been 'painted', i.e. highlighted by the Brush (Fig. 7, left panel). The purpose of this stage is to select as many as possible 3D-reconstructed nano-processes, to accumulate the morphometric statistics suitable for generating NEURON-compatible shapes.



**Figure 7.** Window panels: storing and analysing 3D morphology of nanoscopic processes. *Left and middle* (as in Fig. 6), one process is selected with the 'Brush' (highlighted red). *Right*, the

**selected process (red in *Left* window) magnified and shown as a set of serial sections, including scattered data points (original data) representing the rendered surface of the structure.**

The selected process is automatically highlighted in red, with the corresponding surface points (original data), shown organised into the original serial-section layers (Fig. 7, right). In this window (Fig. 7, right), the user can examine the process shape using the functions shown on the menu (window top). The unwanted (erroneously sampled) data points could be deleted using 'delete/remove' features of the Brush menu. In some versions of MATLAB, it is more convenient to use the '**'NaN'**' function for point removal.

Once the user has been satisfied with the selected process, its structure can be added to the statistics accumulation list using 'Store' menu option. The selected structure should be suitable for statistics accumulation, in which case it is displayed (Fig. 7, middle) as '**'Yes'**' and shown in green. Some shapes could appear mathematically unusable (due to data outliers, insufficient data point numbers, etc.), in which case they will be indicated as '**'No'**' and shown in red. The panel in Fig. 8 shows '**Stored selections**' window (as in Fig. 7, middle) containing five processes. To accumulate sufficient shape statistics, it is recommended to select and store at least 15-20 individual processes for further analyses.

	Section name	Good	Levels	Points	Use
1	testshape.dat	Yes	52	20098	<input type="checkbox"/>
2	testshape.dat_section_1	Yes	14	220	<input checked="" type="checkbox"/>
3	testshape.dat_section_2	No	9	66	<input checked="" type="checkbox"/>
4	testshape.dat_section_3	Yes	8	64	<input checked="" type="checkbox"/>
5	testshape.dat_section_4	Yes	6	48	<input checked="" type="checkbox"/>

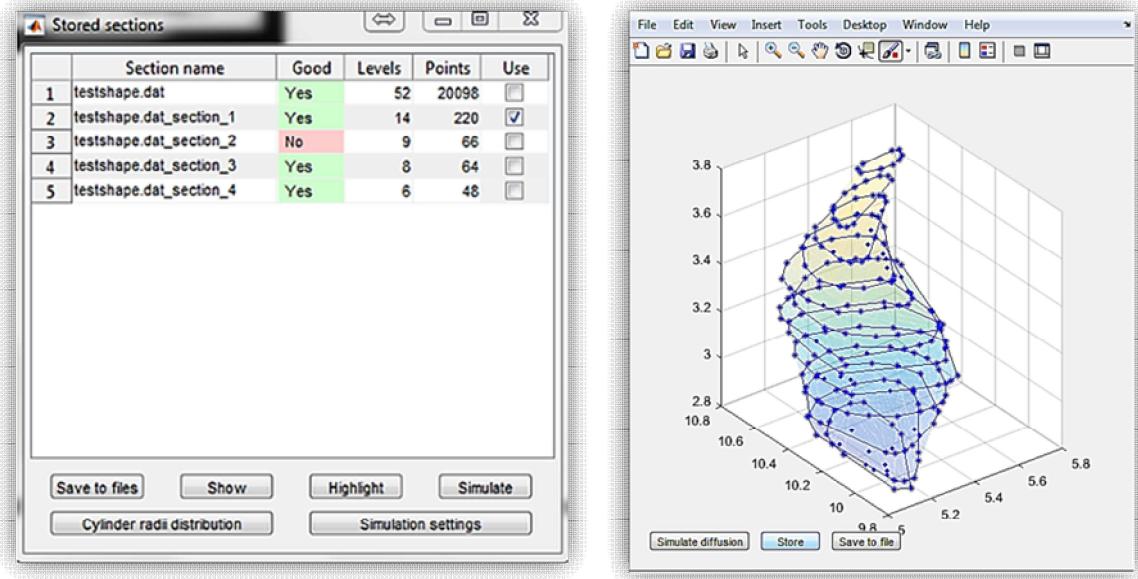
**Figure 8.** Window panel displaying file names for the entire sample (*testshape.dat*) and four individual added processes (of which one, No 3, is unsuitable for analyses and has to be excluded by unclicking). The menu has six executive options:

- 'Save to files': Data will be saved to fil;
- 'Show': The selected processes will be shown in detail (as in Fig. 7, right);
- 'Highlight': The selected processes will be highlighted on the sampled structure (as in Fig. 7, left);
- 'Simulate': This will start Mote Carlo simulations (Brownian motion of neutral molecules or charged ions through the selected process) testing the biophysical features of the selected process;
- 'Cylinder radii distribution': Calculating statistical data for the selected structures;
- 'Simulation settings': Setting parameters of Monte Carlo simulation.

The '**Stored section**' window panel has six different executive options (Fig. 8). The ultimate aim at this stage of astrocyte model building is to generate the cylinder diameter statistics that would represent real (3D-reconstructed) processes using sets of NEURON-compatible cylindrical (disk-shaped) cellular compartments, as explained in the sections below.

### Transforming 3D-reconstructed processes into sets of disk-shaped compartments

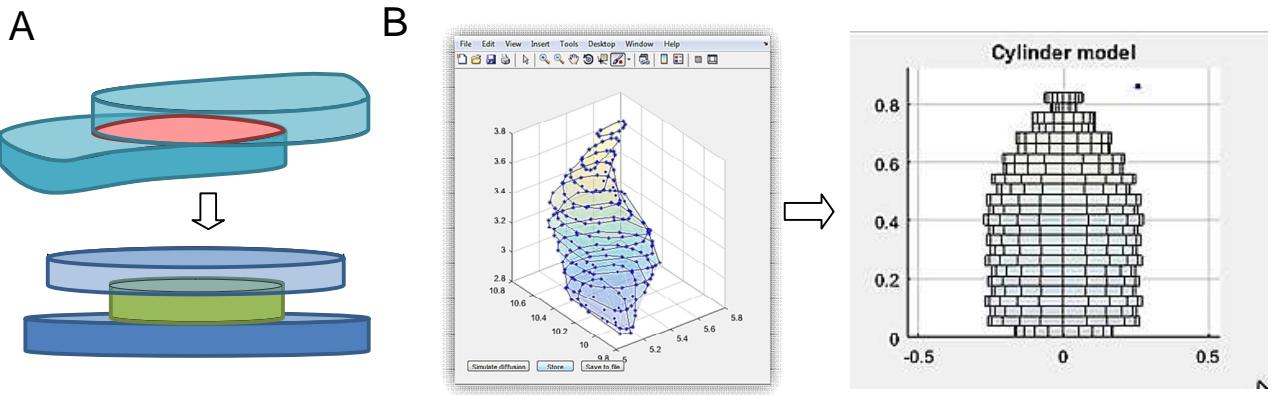
Once the user has selected one nanoscopic process for analyses (Fig. 9, left), the process will display as a 3D structure consisting of  $N$  serial sections formed by  $N$  polygon-based slabs (Fig. 9, right). The number of slabs  $N$  normally matches the number of ultrathin (EM) serial sections in the original structure.



**Figure 9.** Window panels displaying a set of stored processes, with one process selected for analyses (left, ticked) and shown in detail as a set of layers represented by polygonal slabs (right).

Here, the task of **Nano** is to transform these irregular polygon-based structures into coaxial circular cylinder-based structures with similar biophysical properties. To achieve this, each pair of the adjacent irregular polygonal slabs, which are expected to be joined eccentrically, is transformed into three disk-shaped slabs (round cylinders), two 'regular' and one 'transitional' (Fig. 10, left): the base area of the regular cylindrical slab corresponds to that of the polygonal slab whereas the base area of the transitional slab is the joint area for the two polygon-based slabs (red area in Fig. 10, left). Thus, the cross-section and side area properties of the original structures are in large part preserved in the cylinder-based one.

By repeating this routine, **Nano** generates a cylindrical structure consisting of  $2N$  cylindrical slabs ( $N$  regular and  $N$  transitional; Fig. 10, right) and calculates the distribution of slab diameters.



**Figure 10. Transformation of irregular polygon-based structures into coaxial cylinder-based structures.**

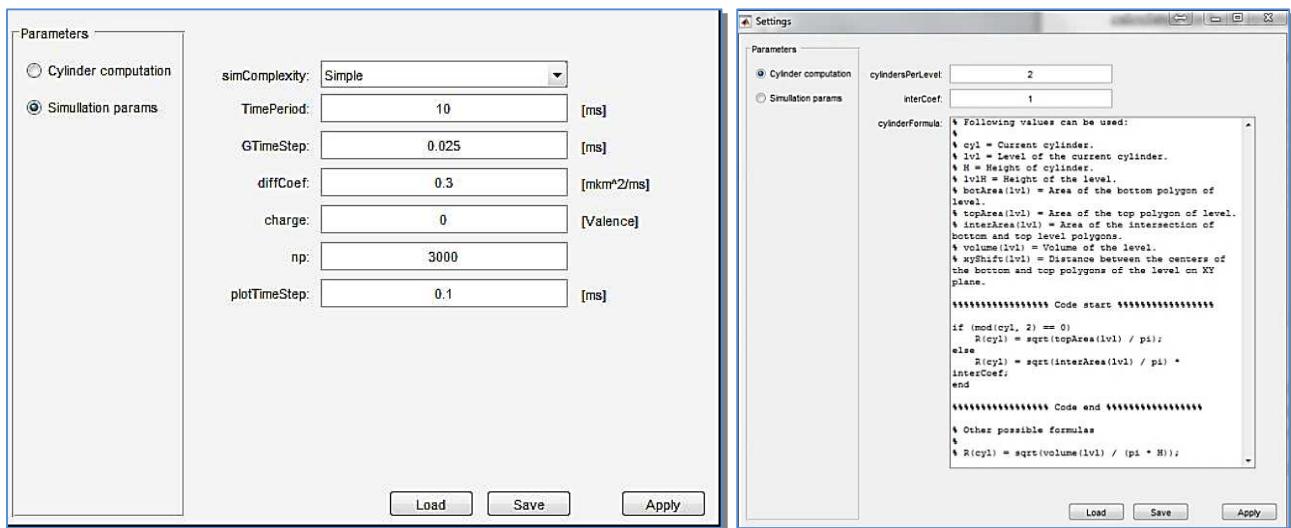
(A) Diagram illustrating the transition from pairs of adjacent polygonal slabs (top; 'smoothed' polygonal slabs shown in blue; red indicates the contact surface) to three cylindrical slabs (bottom; regular slabs shown in navy, transitional slab in green). The base area of the transitional cylinder (green) equals to the joint area (red).

(B) The original structure (left) consisting of  $N$  slabs is thus systematically transformed into a cylinder-based structure consisting of  $2N$  slabs (right).

#### Biophysical comparison of original and transformed structures: a Monte Carlo test

The key test for physical compatibility of the original (polygon-based) and transformed (cylinder-based) structure is the dynamics of particle flux through the structure. 'Nano' is equipped with a Monte Carlo experiment generating Brownian diffusion and electrodiffusion (i.e., electric current) of free particles injected instantaneously at one end of the structure and monitored for their escape at the other end. Simulations (displayed in 'real-time') produce two plots of the particle flux dynamics for both structure types.

Before running the test, the user has to set up its parameters using '**Simulation settings**' (Fig. 9, left) and then '**Simulation params**' options (Fig. 10 left).



**Figure 11: Window panels to set Monte Carlo simulations of Brownian motion.**

*Left*, parameters to be set (default values shown):

'simComplexity', ...; is the parameter which defines two type of simulations: simple and complex. Simple modelling employs a straightforward algorithm for the interaction of particles with the wall of the astrocyte. In this algorithm, the particle stops merely before the wall and does not interact if it can go beyond the wall of the astrocyte at this integration step. The simple algorithm can provide a large integration step and improve the computation speed. A complex algorithm takes into account the detailed interaction of particles with the wall of the astrocyte by an elastic collision. In this case, the algorithm has a small integration step and requires a significant amount of computational time. However, this algorithm gives results that are significantly better than a simple algorithm.

'TimePeriod', simulation run time;

'GTimeStep', diffusion simulation time step

'DiffCoef', diffusion coefficient;

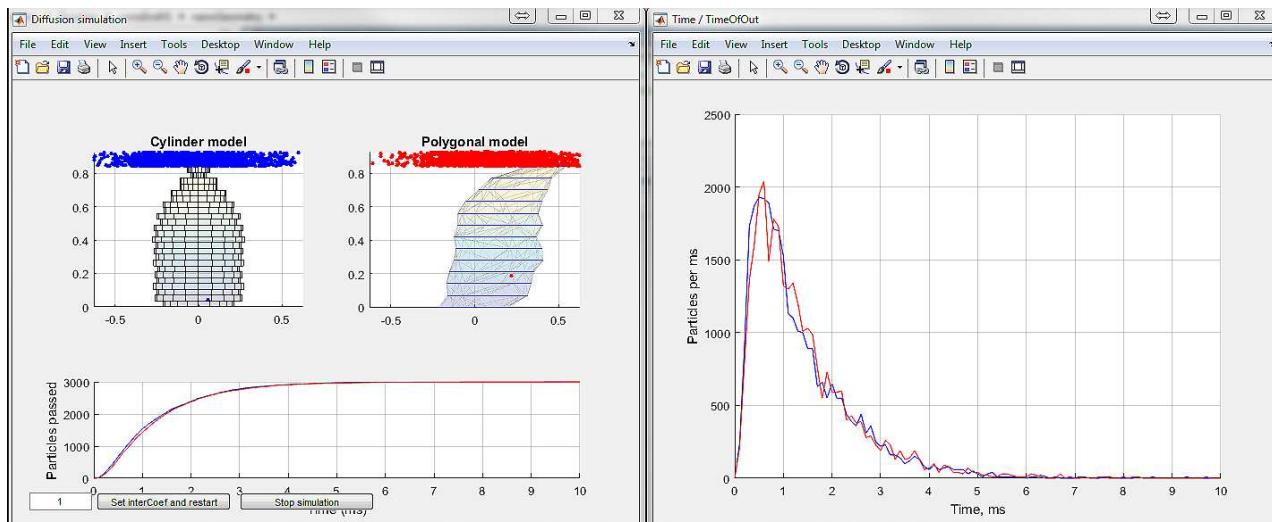
'Charge', particle charge (0 for free diffusion, 1 for electric current);

'np', total number of simulated particles;

'plotTimeStep', time step in the outcome dynamics plot.

*Right*, the panel depicting MatLab code for cylinder shape parameters (for error checking , in-depth analyses and development). Parameter 'CylinderPerLevel' sets the number of cylinder slabs per polygonal slab.

Pressing 'Simulate' key (Fig. 11, left) opens the window panels displaying simulation progress (Fig. 12, left) and the outcome (Fig. 12, right).

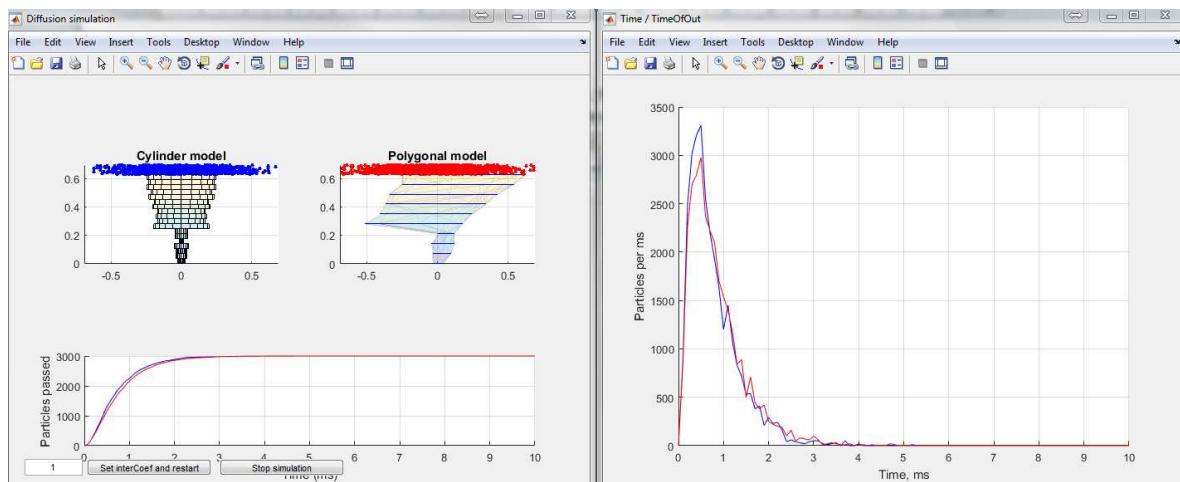


**Figure 12. Monte Carlo simulations of particle flux dynamics in the original and the cylinder-based nanoscopic processes.** *Left*, two structures shown, cylinder-based transformed process

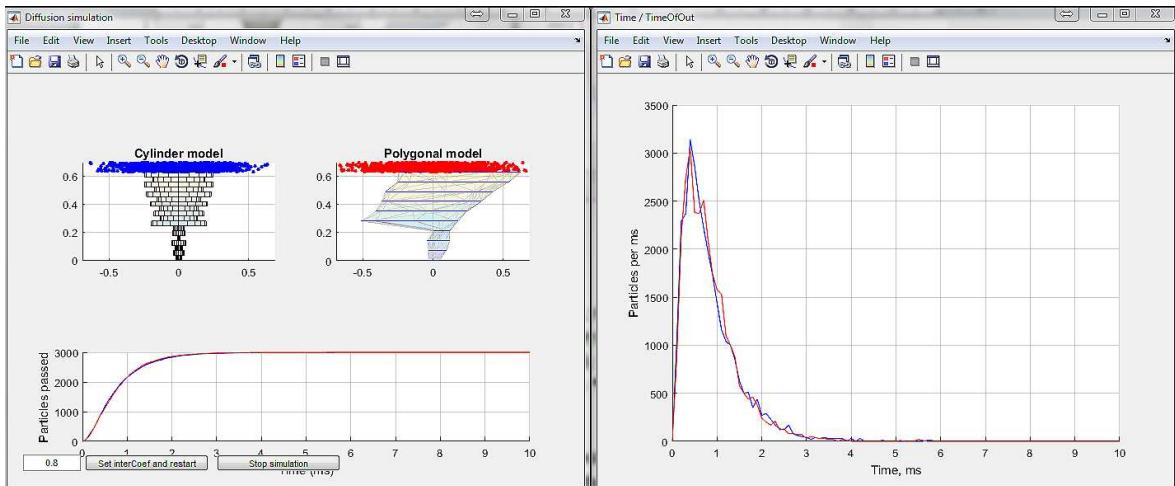
(top left) and the original polygon-based process (top right), with the particle escape (passing through the structure) rate dynamically monitored (bottom). *Right*, summary of the particle flux (number of particles per milliseconds) escaping the cylinder-based process (blue) and the polygon-based process (red).

When the particle flow dynamics are indistinguishable between the original and cylinder-based nanoscopic processes (Fig. 12, right), this indicates that the diffusion and electrodynamic properties of the two structures are similar. Thus, the transformed (cylinder-based) structure can be added to the statistics of cylinder diameters used in the NEURON script to generate astroglial processes: this is done using 'Save to file' option (Fig. 10B; left panel).

When the flux dynamics are dissimilar (example in Fig. 13), the user can adjust effective diameters of transitional cylinders using the 'Set interCoef' option (Fig. 13, bottom left): reducing the average transitional diameter will slow down the particle fluxes, and vice versa. Following the adjustments, the simulation run should be repeated until the readout of particle dynamics are matched between the structures (Fig. 14). The adjusted structure is therefore added to the data pool generating the statistics of cylinder diameters.



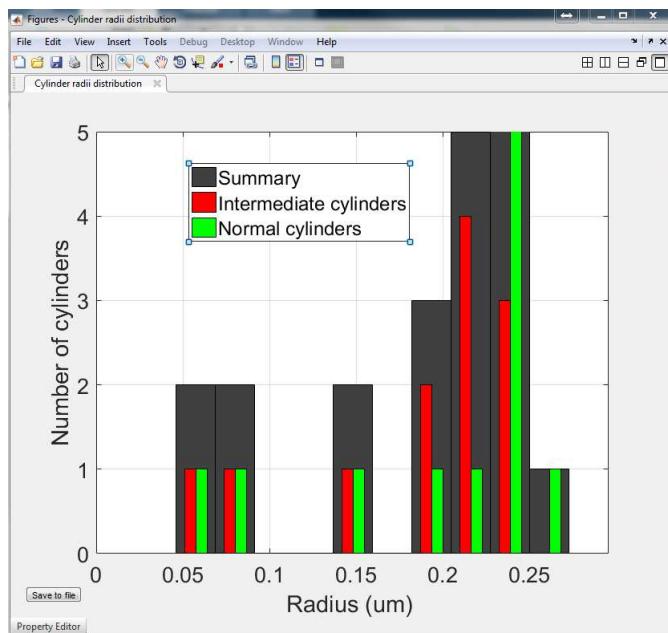
**Figure 13. Example of Monte Carlo simulations, as in Fig. 12, but indicating that the cylinder-based structure lets through Brownian particles at a higher rate than does the original polygon-based structure (see Fig. 13 for notation detail).** This requires an adjustment of transitional cylinder diameters using 'Set interCoef' key (bottom left).



**Figure 14.** Monte Carlo simulations, as in Fig. 13, but with the adjusted transitional cylinders (top left panel), thus producing the matching outcomes for the particle dynamics (right).

### Statistics of NEURON-compatible cylindrical compartments

Once the accumulation of data on individual cylinder-based structures is complete, the program (see. Fig. 9 , the bouton “Cylinder radii distribution” on the Panel “Store selection”) generates the distribution of cylinder-compartment diameters representing the astroglial processes (Fig. 15).



**Figure 15.** Distribution of cylinder-compartment diameters (example based on Fig. 10B data). Green and red, regular and transitional cylinders, respectively (see Fig. 10A, bottom); black, cumulative data.

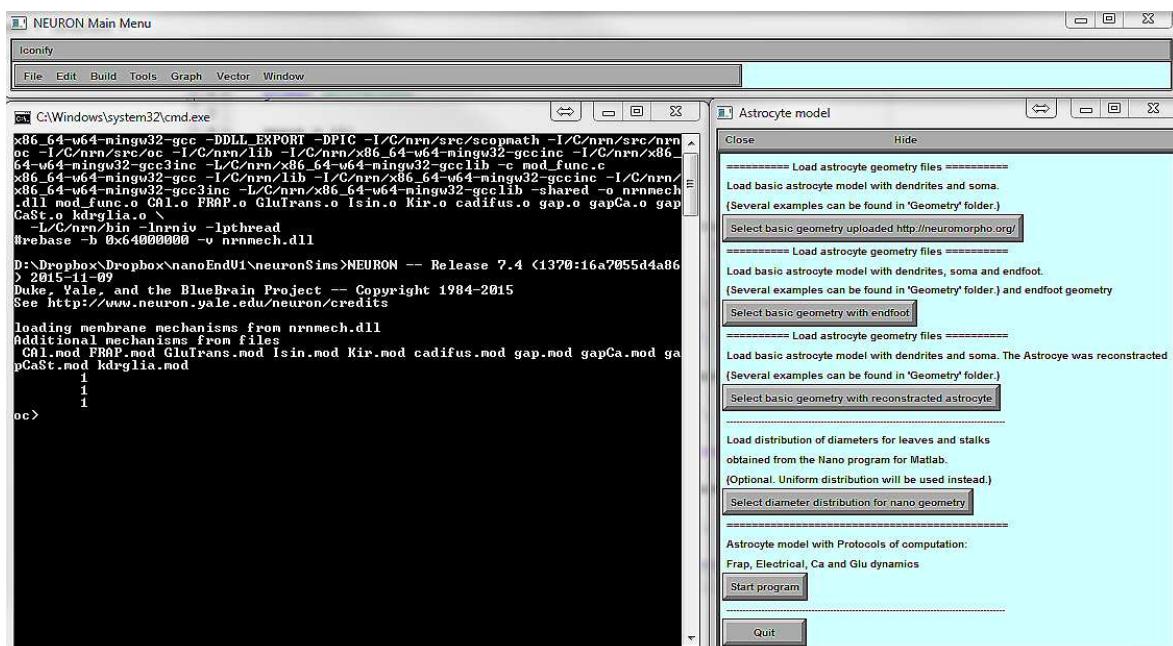
This resulting distribution can be saved into a data file (see. Fig. 9 , the bouton “Cylinder radii distribution” on the Panel “Save to files”)which will be used to generate nanoscopic astrocytic processes with the NEURON script, as explained in sections below.

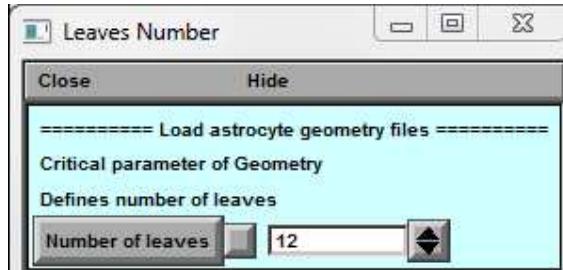
## GENERATING COMPLETE ASTROCYTE MORPHOLOGY

### Preparing Host computer (under Windows)

This section explains initial steps to launch and run NEURON environment adapted for astroglial modelling, on the Host computer under Windows.

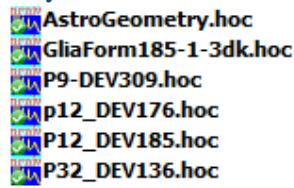
1. Execute the 'init.hoc' file located in the host computer directory ...***neuronSims\init.hoc*** or use button “NEURON simulations” from the panel of the start menu (Figs. 2-3).
2. The path to NEURON on the user computer should be set in the batch file : **neuron\_home\_win.bat**, by definition, the following path is set : set **NEURON\_HOME\_WIN='C:\nrn'**
3. If the user works with the main panel (Figs. 2-3), Astro generates the executable NEURON \*.mod files themselves. Therefore, the user does not need to do anything extra. Otherwise, the user must create the NEURON executive mod-files applying the file **mknrndll.exe** placed in NEURON installation directory and after activation **mknrndll.exe** user needs to point the working directory of Astro ...|***neuronSims\***
4. The file **init.hoc** opens three windows (Fig. 16):
  - a. System window (**cmd.exe**)
  - b. Window for the definition of leaves number.
  - c. Window for definition of basic astrocyte geometry.



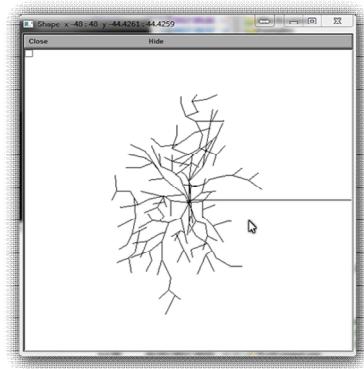


**Figure 16.** System windows (incorporating Astro) initiated by launching NEURON.

5. To design a new astrocyte model, choose and press one of the three following buttons
  - a. 'Select Basic Geometry uploaded from <http://neuromorpho.com>'. Pre-prepared files are in the directory /Geometry.

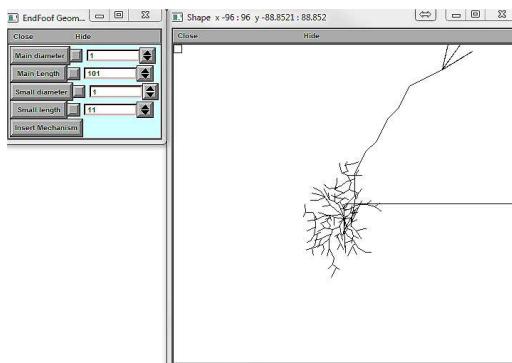


After selecting a file, a window appears with the selected structure of the dendritic tree:



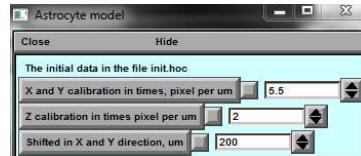
**Figure 17.** Window panel showing the main processes (skeleton) of astroglia.

5. To design a new astrocyte model, choose and press one of the three following buttons
  - b. 'Select Basic Geometry with endfoot'. The option the same as in (a), but with self-generating endfoot structure. In this option, an additional window appears that allows defining geometric parameters of the main branch of and biophysical mechanisms.



- c. “Select basic geometry with reconstructed astrocyte” loads prepared in advance by the user ‘cell skeleton’ file from the directory [/Geometry](#) (see page 9). Now the user can upload a file **GeometryReal.hoc** with a fully prepared cytoskeleton of CA1 astrocyte. The structure was reconstructed from CA1 astrocyte using Vaa3D software. The user can freely download the software from the web site <http://www.alleninstitute.org/what-we-do/brain-science/research/products-tools/vaa3d/>

An additional window appears that allows defining geometric parameters of the main branch and shifted to the zero coordinate.



6. Nevertheless, the user must download the primary cell 'skeleton' geometry before initiating any further design of the model.

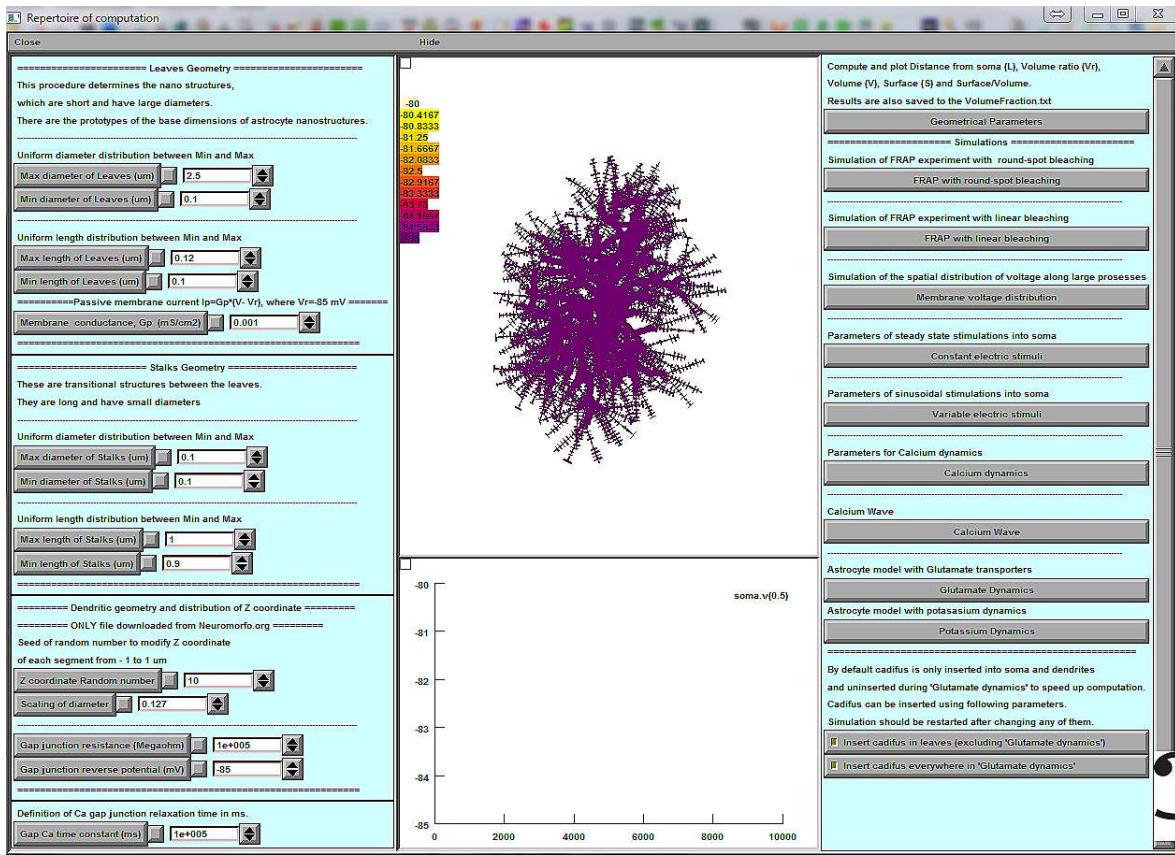
After the user has downloaded the base cytoskeleton, there are two options. First one is to load the nanostructure that the user has prepared in advance (using the Nanostructure application). The option ‘**Select diameter distribution for nano geometry**’ allows the user either to load a file with the statistics of diameters produced by the **Nano** module (from sampled 3D-reconstructed astroglial processes, see above). Alternatively, press Start button generate nanostructures computationally using the built-in tools. In both cases, the user can repeatedly adjust key morphometric features of the generated nanostructures.

The “**Select diameter distribution for nano geometry**” is providing population-average structure of nanoscopic processes and downloading the file “testshape.dat\_radii\_dist.txt” providing a 3D-reconstructed structure of main processes and population-average morphology of nano-processes.

In case of full morphology go to the chapter **SIMULATING ASTROCYTE PHYSIOLOGY**.

## Generating nanoscopic processes

The main window of **Astro** (Fig. 18) provides a path to the further steps in modelling complete astrocyte morphology, as described in sections below.



**Figure 18.** The main window of Astro: left panel, control of geometry of astrocyte; right panel, computational scenarios; centre, simulated function output mapped onto astrocyte morphology (top; membrane voltage shown), with selected digital output plot (bottom).

### Setting limiting parameters for nanoscopic process morphology

**Important.** Before beginning the physiological simulations check the geometric parameters of the astrocyte by clicking on the button 'Geometrical parameters' (see the next chapter) and the main diameters. If main diameters are constant just click the button scaling of the diameters on the main panel and check 'Geometrical parameters' again.

Sections **Leaf Geometry** (Fig. 18, top left) sets limiting values on the shapes of nanoscopic processes generated based on the uploaded statistics of 3D-reconstructed astroglial structures (see previous chapter).

Similarly, section **Stalk Geometry** (middle left) sets limiting values on the length and diameters of thin astroglial branches connecting nanoscopic processes ('leaves'). These parameters determine how densely astroglial branches are to be populated with nanoscopic processes.

Section "Passive membrane current" sets the density of membrane conductance membrane conductivity is installed taking into account the lateral surface of the leaves.

Section '**Dendritic Geometry and distribution of Z coordinate**' (bottom left) sets four parameters:

**'Z coordinate. Random number'** generates the seed of random number to add Z coordinate of each segment between - 1 to 1 um. This function is introduced for 2D cytoskeleton in plane of Y-X, in order to create a third dimension in the direction of the Z-axis.

- **'Scaling of Diameter'** defines the scaling coefficient for the main branch diameters as a function of distance from the soma. The parameter “`scalingDiam`” scales the diameters of main branch according to the experimental recording. The following formula was used in the model:

$$d(r) = \frac{d_{\text{const}}}{\sqrt{S(r+1)}}$$

Where S= “`scalingDiam`” and r is a distance from the soma. The default value of this parameter is 0.127, which agrees well with the experimental data.

This option has to be ignored if you download real cell skeleton.

Astro complex has two types of gap junctions, one electrical for modulation of the membrane potential, and the second one is a calcium junction for changes the intracellular concentration of calcium.

The electrical gap junction sets the local transmembrane current according to the formula:

$$I_{\text{gap}} = (V_g - V)/r_g$$

**'Gap Junction Resistance,  $r_g$ , MOhm'**: the default value,  $10^{10}$  MOhm, is equivalent to having no gap junction current.

**'Gap Junction Reverse potential,  $V_g$  (mV)'**: the default value, -85 mV, is to be set when relevant experimental data are available.

The Gap junction for Ca concentration sets the local calcium flux according to the formula of local calcium current:

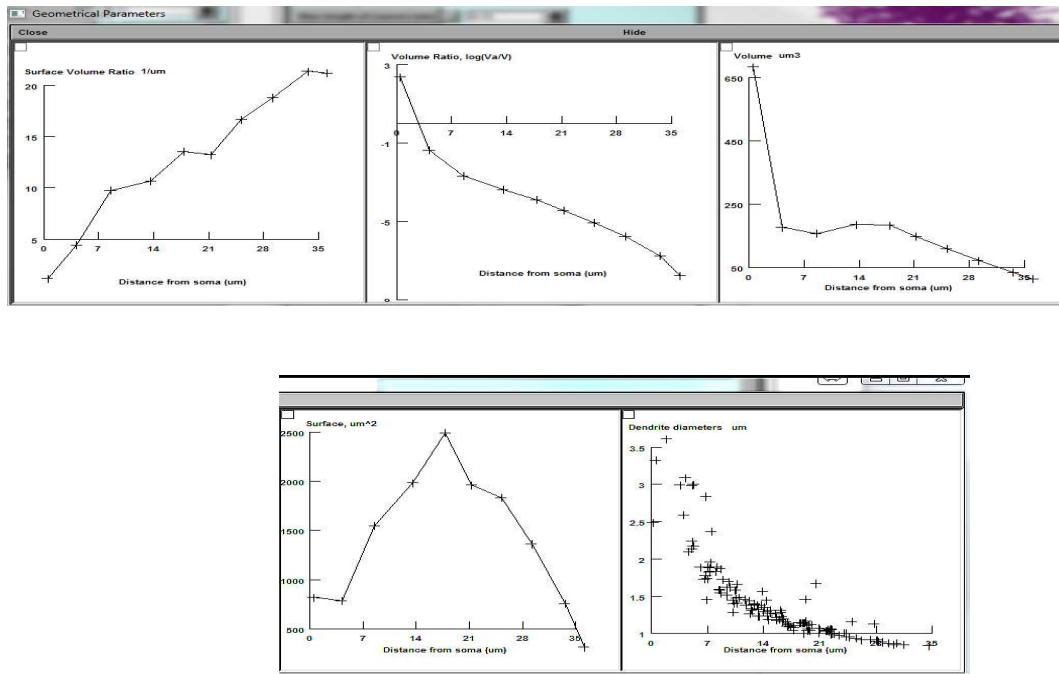
$$I_{\text{ca}} = ([Ca]_i - [Ca]_b)/(2 [Ca]_b T_R F)$$

Where  $[Ca]_i$  is intracellular calcium concentration,  $[Ca]_b$  is resting calcium concentration, the default value is 50 nM,  $T_R$  is the time constant of Ca relaxation between real and basic value of calcium, F is Faraday constant.

**'Gap Ca time constant, ms'**,  $T_R$ , sets the time constant of calcium relaxation. The default value is 100000 ms is so huge that no Ca flux through Ca gap junction. The value can be set by user when relevant experimental data are available.

### Adjusting tissue-filling (volumetric) properties of nanoscopic process morphology

The volume-filling properties and surface-to-volume ratios of the nanoscopic processes (generated as described above) can be monitored using the '**Geometrical parameters**' key (Fig. 18, top right). Pressing it opens the following windows (Fig. 19):



**Figure 19.** Window panels providing the volumetric characteristics of modelled astroglia (launched by 'Geometrical parameters' key).

The plotted data (Fig. 19) also are saved to the file '[...\\neuronSims\\Text results\\VolumFraction.txt](#)'. These values are to be compared with the corresponding empirical data obtained using 3D EM reconstructions and two-photon excitation imaging data of the astroglia of interest. The user is free to evaluate the mismatch and adjust the density of nanoscopic processes (using **'Stalk Geometry'** and **'Dendritic Geometry'** options where relevant; Fig. 18) correspondingly, until an acceptable match is produced.

At that stage, the modelled astroglial morphology is complete (see 'FRAP experiments' below for further subtle morphological adjustments), and the user is can begin to simulate various functions of astroglia while also implementing a variety of membrane and intracellular biophysical mechanisms, as briefly explained in the sections below. The windows depicting key geometrical parameters (Fig. 19) can be closed and open whatever at any time.

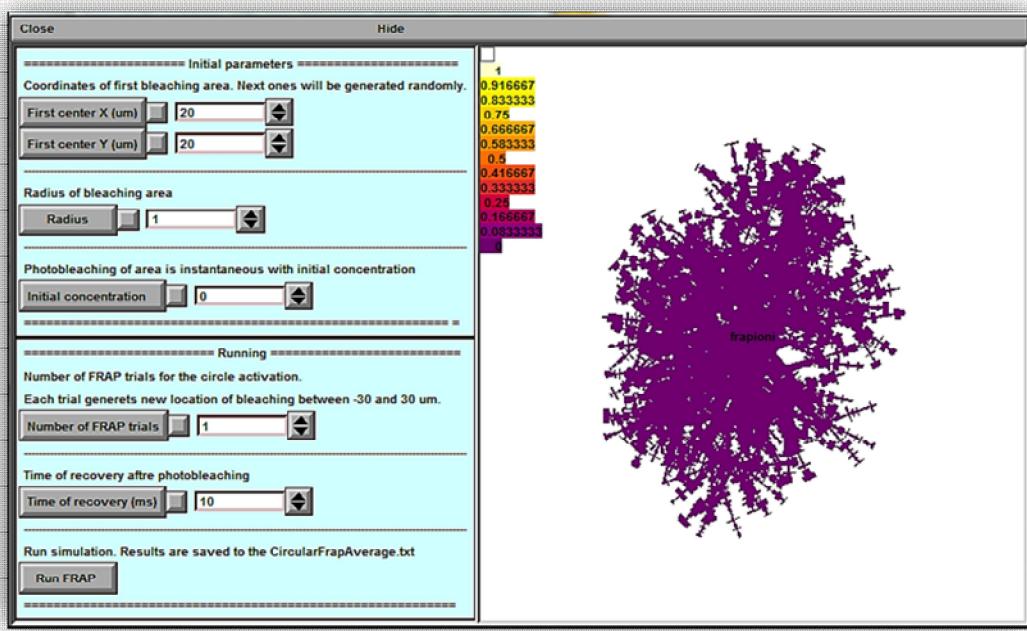
## SIMULATING ASTROGLIAL FUNCTION

### FRAP experiments: probing and adjusting inter-cellular connectivity of astroglia

Fluorescence recovery from photobleaching (FRAP) applied with respect to water-soluble intracellular indicators assesses effective diffusivity across the cellular compartments adjacent to the bleached area. FRAP experiments and the corresponding simulations (launched from main **Astro** panel; Fig. 18 right) could be used therefore to test whether the modelled astrocyte morphology reproduces intracellular diffusivity (connectivity) properties documented empirically. The default values correspond to the data obtained in CA1 astroglia using linear (line-scan) photobleaching tests with Alexa Fluor 594.

## FRAP with round-spot bleaching

This option starts simulations of FRAP with a round bleaching area. Pressing '**FRAP with round-spot-bleaching**' opens the window as follows:



**Figure 20. Simulation panel for FRAP with round-spot bleaching.** Left, parameter settings; right, concentration dynamics of non-bleached molecules, colour-coded and mapped on cell morphology.

The initiation panel (Fig. 20, left) provides the following settings:

**'First Center X'**, X coordinate of the bleaching area centroid

**'First Center Y'**, Y coordinate of the bleaching area centroid

**'First Center Z'**, Z coordinate of the bleaching area centroid

**'Radius'**, the bleaching area radius.

**'Number of trials'** sets the number of trials generated quasi-randomly across the astrocyte morphology (X and Y are generated with a uniform distribution, between -30 and 30  $\mu\text{m}$ ).

**'Time of recovery'** is a time of recovery after photobleaching in ms. The recovery starts from '**the initial concentration**' of bleached molecules. In this option, the photobleaching was assumed to be instantaneous (duration of time was 0 ms).

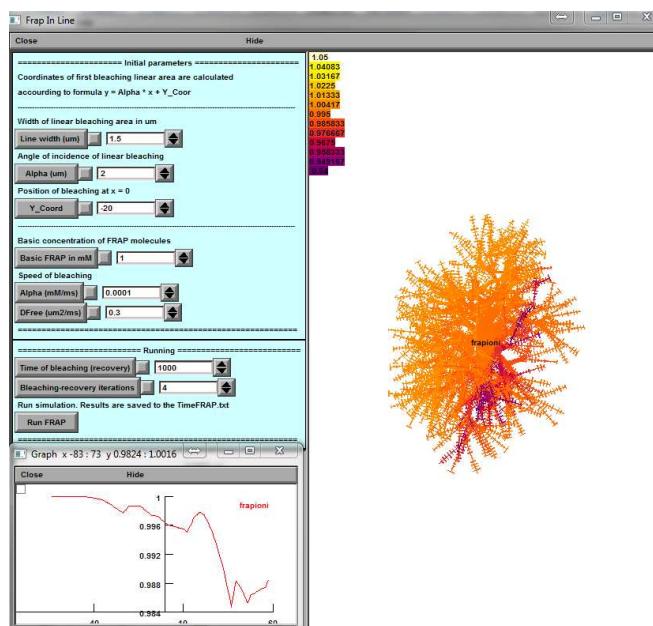
**'Initial concentration'** sets the initial concentration of bleached molecules.

**'Run'** launches FRAP simulations.

The results of computations are saved to the file '*CircularFrapAverage.txt*'

## FRAP with linear bleaching

This option starts simulations of FRAP with a line-scan bleaching area. Pressing '**FRAP with linear bleaching**' opens the window as follows:



**Figure 21. Simulation panel for FRAP with linear (line-scan) bleaching.** *Left*, parameter settings; *right*, concentration dynamics of non-bleached molecules, colour-coded and mapped on cell morphology.

The initiation panel (Fig. 21, left) provides the following settings:

**'LineWidth'**, **'Angle'** and **'Y at x=0r'** set, respectively, the width and the inclination angle of the bleaching line, and where the line intersects the central line ( $x = 0$ ).

**'Basic FRAP in mM'** sets the basic concentration of photobleaching molecules. The value is important to compare with experimental results.

**'Bleaching rate'**, the rate of photobleaching.

**'Dfree'**, the intracellular diffusivity.

**'Bleaching recovery interactions'**, the number of bleaching-recovery periods per trial.

The default duration of FRAP trials is 8000 ms, with photobleaching occurring every 1000 ms. The results of computation are saved to the file '*TimeFRAP.txt*'

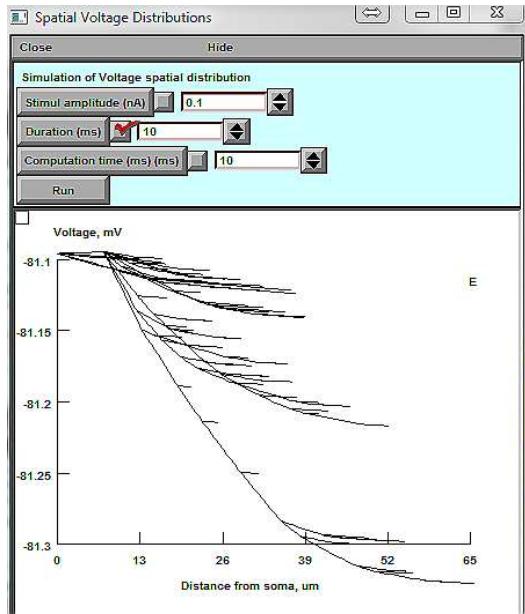
## Probing membrane mechanisms of astroglia

This part of **Astro** deals with simulations of membrane mechanisms (passive electrical properties, voltage-dependent channels, ion exchange, receptor current, etc.) and intracellular processes ( $\text{Ca}^{2+}$  entry, buffering, diffusion, and removal mechanisms) acting in morphologically realistic astrocytes. Most of the programming routines involved are an inherent part of the NEURON environment, with the full guidance available in the NEURON documentation

(<https://www.neuron.yale.edu/neuron/docs>). The sections below provide therefore brief information relevant to astroglial probing.

### Steady-state membrane voltage distribution

The 'Compute the spatial voltage distribution' key (Panel 'Repertoire of computation') opens a window (Fig. 22) which reports membrane voltage across selected cell processes upon constant current injection at the soma (amplitude in nA). The default resting potential is -85 mV.



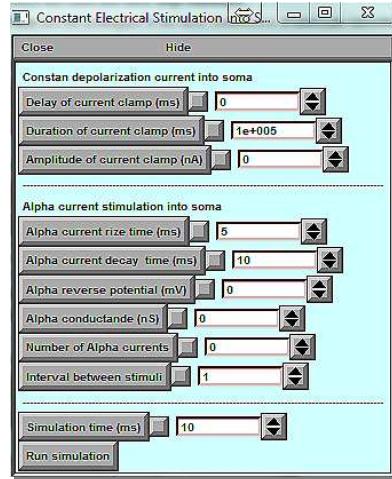
**Figure 22. Window panel to monitor membrane voltage distribution.**

The initiation panel (Fig. 21, left) provides the following settings:

- 'Stimulus amplitude nA', amplitude of constant depolarising current injected into the soma.
- 'Duration ms', current duration.
- 'Computation time', experiment run time.
- 'Run', to start simulation.

### Electric stimulation settings

The electric stimulation panel was design to generate different patterns on electrical stimulations into a soma. The button 'Electrical stimulation' defines two types of electrical stimulation into the soma. The following panel offers the parameters of these stimuli:



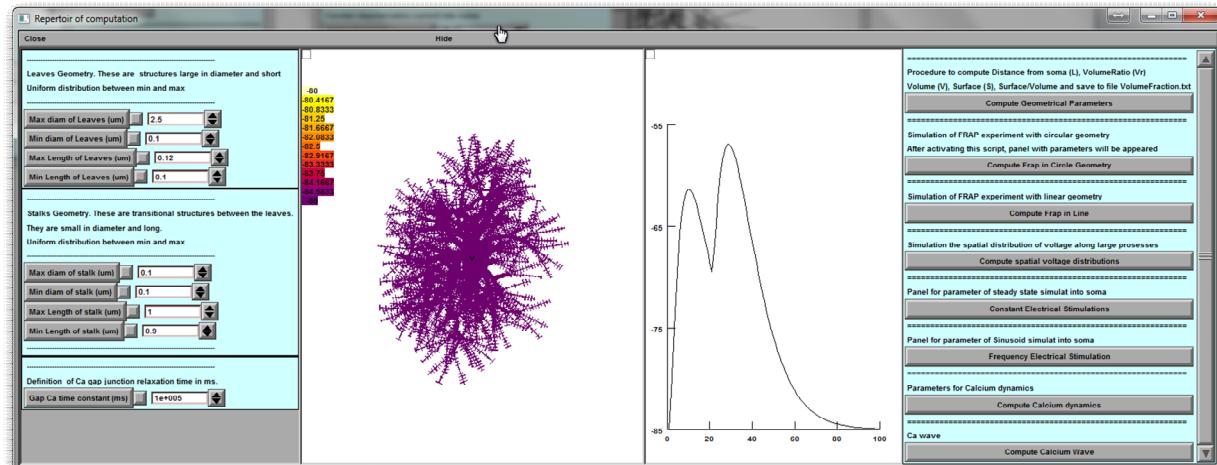
**Figure 23. Constant electric stimulation.** Set of parameters for the simulation of voltage and current dynamics in astrocytes in response to the constant and alpha current.

Top block defines three parameters of the constant current injected into the soma: delay, duration and amplitude.

Middle block. Five parameters define alpha current into the soma: rise time, decay time, reverse potential, conductance, the number of inputs and time between stimuli.

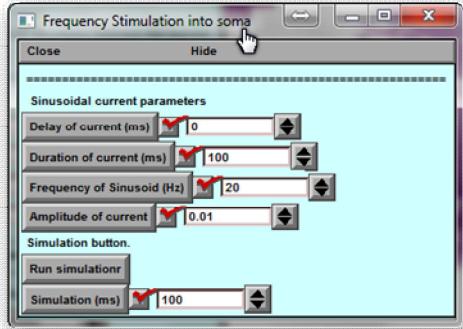
The last bloc is to start computation during the specific time.

The result of the computation is on the main plots with spatial and time scale.



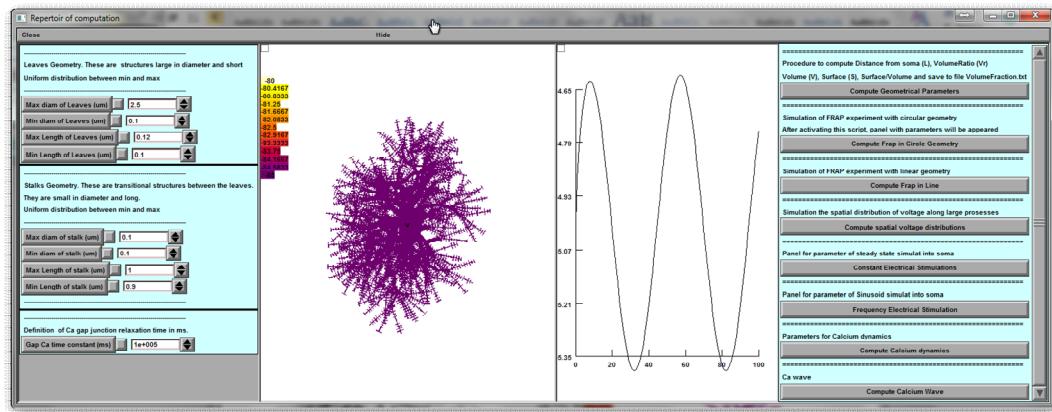
## Frequency stimulation

The button 'Frequency Electrical stimulations' is the definition of sinusoidal current stimulation into the soma via the following panel:



**Figure 24.** Set of parameters for the simulation of voltage and current dynamics in astrocytes in response to the Sinusoidal current.

The set of parameters describes the sinusoidal current injected into Soma: delay of current, duration of current, a frequency of the current in Hz, the amplitude of current in nA and length of simulations. The main plot shows the results of simulations.



#### Gap junction for Calcium concentration

$$ica = ((([Ca]i - [Ca]b)/(2[Ca]btrF))$$

#### Simulating membrane voltage in response to local current hotspots

This **Astro** function can be performed by modifying settings in the **Astro** menu section titled 'Simulating astroglial glutamate transporters' (see p. 30 below). The latter provides full monitoring of the dynamic membrane voltage landscape mapped onto the cell morphology.

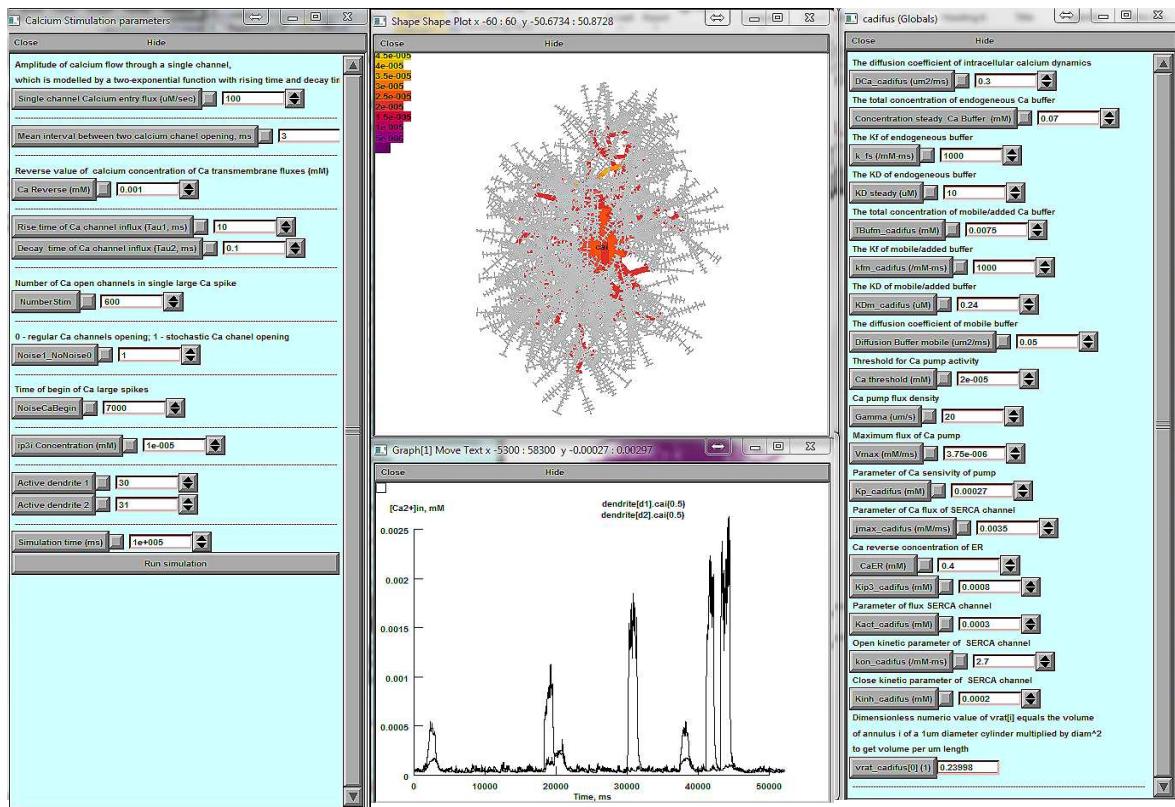
#### Modelling intracellular calcium dynamics

**Astro** provides simulation and analyses of the intracellular  $\text{Ca}^{2+}$  dynamics - including entry, buffering, diffusion, regenerative waves, and removal - across the entire astrocyte morphology. This type of

simulation may require significant computational resources: it might be feasibly to run short-duration (1–10 ms) trials on the Host computer but simulating longer events over the large cell areas will probably require the Worker computer / cluster (see below for detail). For the sake of simplicity, the current version enables simulation of either a uniform  $\text{Ca}^{2+}$  change across large cell areas or active (oscillatory, single-channel type)  $\text{Ca}^{2+}$  entry within two selected processes only.

## Main $\text{Ca}^{2+}$ simulation menu

'Calcium dynamics' key (Fig. 18 right) opens four new windows (Fig. 22):



**Figure 22. Window panels pertinent to intracellular  $\text{Ca}^{2+}$  dynamics and monitoring.** *Left*, parameter settings concerning single-channel  $\text{Ca}^{2+}$  dynamics located on “Active dendrites” (bottom; dendrites d1 and d2 are shown). *Right*, parameter settings concerning  $\text{Ca}^{2+}$  homeostasis mechanisms (buffering, diffusion, extrusion, etc.). *Middle*, visualization of  $[\text{Ca}^{2+}]$  mapped on cell morphology (top), and  $[\text{Ca}^{2+}]$  kinetics in two selected dendrites (bottom; dendrites d1 and d2 are shown).

Window panel 'Calcium stimulation parameters' (Fig. 22, left) provides parameter settings for active  $\text{Ca}^{2+}$  entry kinetics. The latter is assumed to consist of a series of stochastic  $\text{Ca}^{2+}$  channel openings (individual events) including higher-frequency bursts. Note that many of the parameters involved are currently unknown and therefore represent the subject of investigative exploration and probing.

'Single channel calcium entry flux' sets the rate of calcium entry during an individual event.

'Mean interval between two calcium events' is the average time interval between events.

**Ca Reverse** is  $\text{Ca}^{2+}$  reverse concentration of linear Ca fluxes from an inside to an outside the astrocyte.

'Rise' and 'Decay' set the rise and decay time of  $\text{Ca}^{2+}$  flux during a single channels opening.

'Events per burst' sets the number of calcium channels opening per burst.

'Randomness' sets 0 for random channels opening with uniform distribution and 1 for deterministic channels opening.

'Burst onset' sets the time of the event burst.

'IP3 concentration' sets the basal an initial concentration of ip3 ions.

'Active process 1' and 'Active process 2' are astroglial processes where active  $\text{Ca}^{2+}$  entry is enabled.

'Stimulation time' sets the run time of simulations. In astroglia,  $\text{Ca}^{2+}$  dynamics is relatively slow long time process and the simulation needs at least 100 seconds.

Window panel setting  $\text{Ca}^{2+}$  homeostasis parameters (Fig. 22, right) displays self-explanatory keys and the corresponding explanations.

Panel of parameters of Ca dynamics

'The diffusion coefficient of intracellular calcium dynamics', DCa\_cadifus, sets the value of Ca diffusivity.

'The total concentration of endogenous Ca buffer', TBufs\_cadifus sets the initial concentration of endogenous Ca buffer.

'The Kf of endogenous buffer', kfs\_cadifus, sets the forward kinetics of endogenous buffer.

'The KD of endogenous buffer', KDs\_cadifus, sets the dissociation constant of endogenous buffer.

'The total concentration of mobile/added Ca buffer', TBufm\_cadifus, sets the total concentration of mobile buffer.

'The Kf of mobile/added buffer', kfm\_cadifus, sets the forward kinetics of mobile buffer

'The KD of mobile/added buffer', KDM\_cadifus, sets the dissociation constant of mobile buffer.

'The diffusion coefficient of mobile buffer', DBufm\_cadifus, sets the diffusion coefficient of mobile buffer.

'Threshold for Ca pump activity', cath\_cadifus, sets the influx only when a critical threshold of intracellular calcium is reached

'Ca pump flux density', gamma\_cadifus, sets the density of Ca pump flux.

'Maximum flux of Ca pump', vmax\_cadifus, sets the maximum Ca flux

'Parameter of Ca sensivity of pump', Kp\_cadifus, sets the sensitivity of Ca pump according to the formula  $j_{\text{pump}} = \alpha v_{\max} [Ca]_i^2 / ([Ca]_i^2 + K_p^2)$ .

'Parameter of Ca flux of SERCA channel', jmax\_cadifus, sets the maximum value of SERCA channels.

'Ca resting concentration of ER', CaER, sets the initial concentration of Ca inside the ER.

'IP3 degradation rate', Kip3\_cadifus, sets the degradation rate of IP3

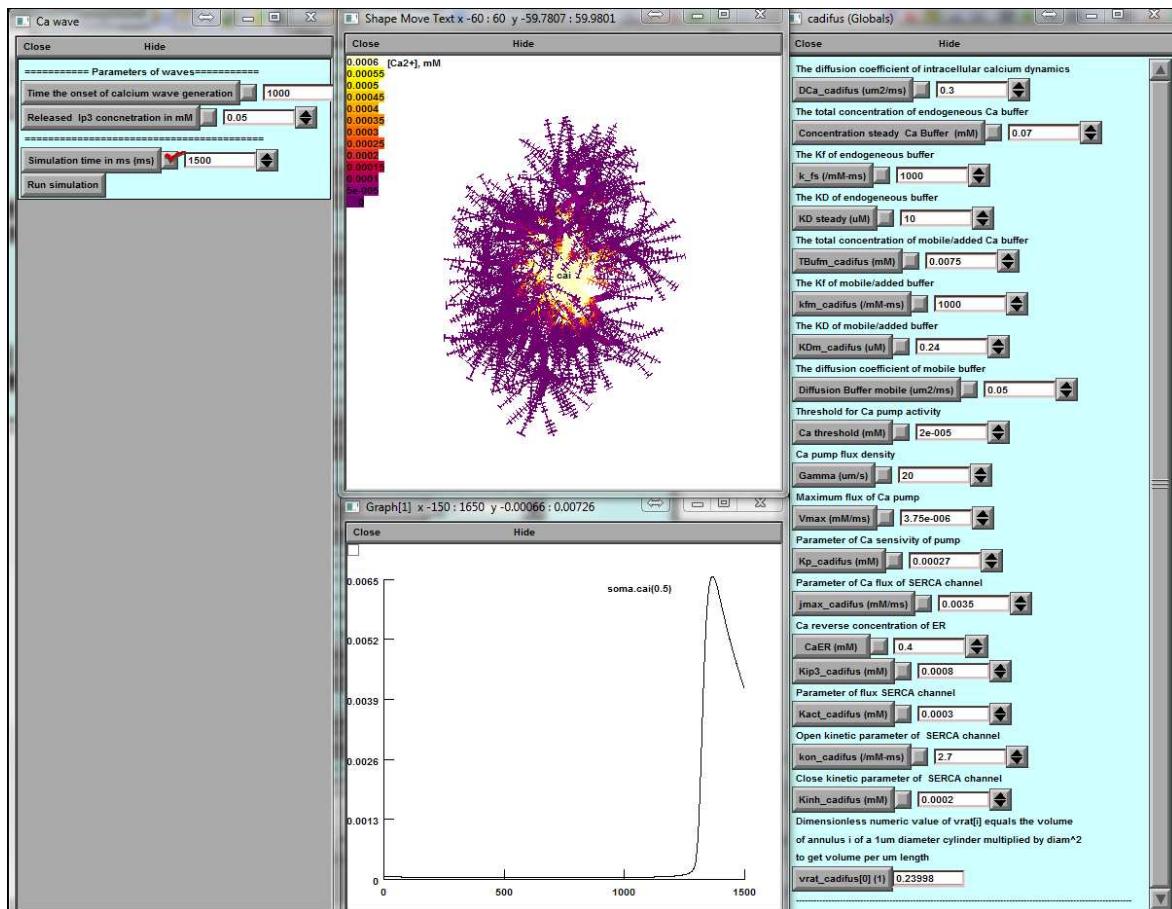
'Parameter of flux SERCA channel', Kact\_cadifus, sets the dynamics of SERCA channels according to the formula

$$j_{\text{chnl}} = \alpha j_{\max} (1 - ([Ca]_i^2 / caer) (ip3 / (ip3 + Kip3)) * (([Ca]_i^2) / ([Ca]_i^2 + Kact)))$$

- ‘Open kinetic parameter of SERCA channel’, kon\_cadifus, sets the forward rate of SERCA channels kinetics
- ‘Close kinetic parameter of SERCA channel’, Kinh\_cadifus, sets the back forward rate of SERCA channels kinetics
- ‘Dimensionless numeric value of vrat[i] equals the volume of annulus i of a 1um diameter cylinder multiplied by diam<sup>2</sup> to get volume per um length’ vrat\_cadifus[0]

## Ca<sup>2+</sup> wave simulations

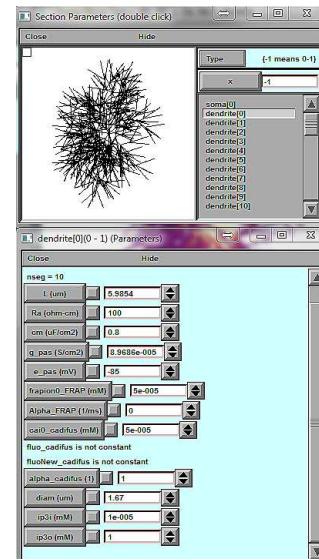
‘Ca wave’ key (Fig. 18, right) opens the new window panel as follows



**Figure 23. Window panels enabling simulating of intracellular Ca<sup>2+</sup> waves dynamic.** *Left*, colour-coded dynamic [Ca<sup>2+</sup>] landscape mapped onto cell morphology. *Right*, parameter setting panel (top), and the readout plot of [Ca<sup>2+</sup>] dynamics, by default in the soma (bottom).

'Ca wave' panel (Fig. 23, top right) gives the onset of a step increase in the IP<sub>3</sub> concentration ('Onset', time point 1000 ms shown) and the step amplitude ('IP<sub>3</sub> Max', amplitude 0.05 μM shown). Default time is 10 s. A new simulation run can start once the current run has ended. 'Graph' panel (Fig. 23, bottom right) provides [Ca<sup>2+</sup>] time course in the soma.

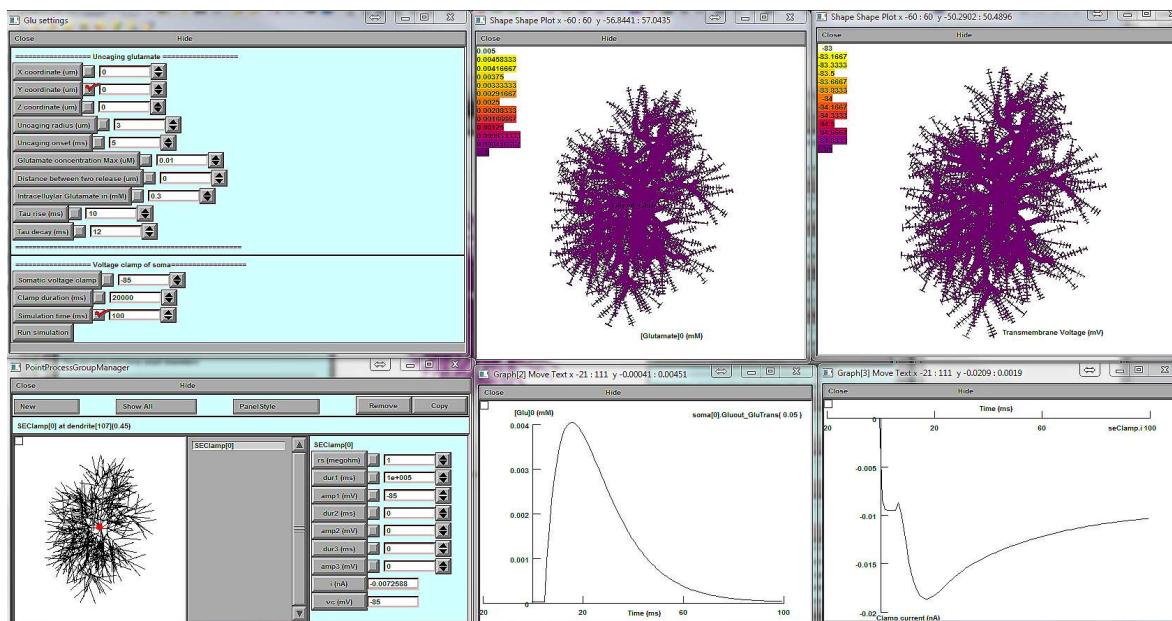
The user can explore the dynamics of calcium waves, locally releasing IP<sub>3</sub> in any place of the astrocyte (Section parameters panel), using the following built-in panels of NEURON:



run

## Simulating astroglial glutamate transporters

Astroglial plasma membranes are enriched in high-affinity glutamate transporters which generate rapid inward current upon glutamate binding. In **Astro** main panel, 'Glutamate transporters' key (Fig. 18, right bottom) opens the **Astro** menu (Fig. 24), which enables simulations of the dynamic membrane voltage landscape mapped onto cell morphology, in response to volume-limited application ('uncaging') of glutamate at a selected area of the cell.



**Figure 24.** Window panels with parameter settings (top left) and readout plots, to simulate the dynamic membrane voltage landscape mapped onto cell morphology, in response to volume-limited application ('uncaging') of glutamate. The left-bottom panel is for voltage clamp electrode position.

'Uncaging glutamate' panel (Fig. 24, top left) provides parameter settings for volume-limited glutamate application (uncaging) within a round area of the cell, including uncaging disk with coordinate X, Y, and the area radius; the uncaging onset (must be >3 ms to stabilise membrane kinetics); the rise and decay time and the maximum glutamate concentration.

The user can add another, simultaneously occurring uncaging spot specifying the distance between the two spots at the X from the first one.

Transporters currents are computed in somatic voltage clamp configuration: the clamp parameters are indicated on the panel '**Somatic voltage clamp**' (Fig. 24, bottom left) including electrode resistance, clamp voltage and duration.

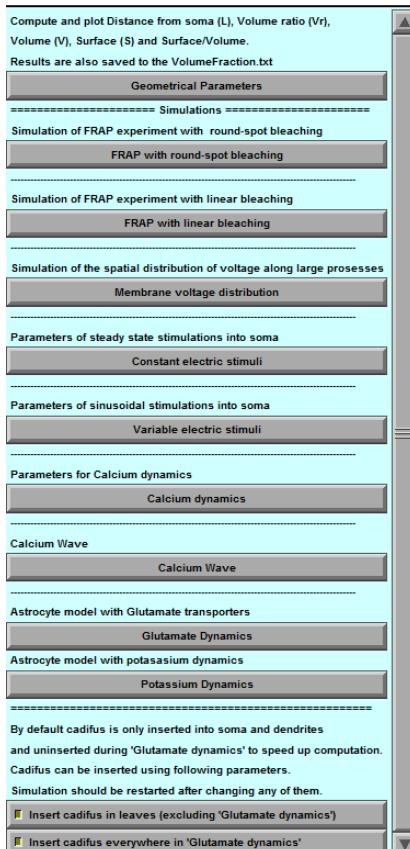
The voltage clamp current on the soma after uncaging of glutamate on the soma is shown on the left top panel.

The monitoring panels (Fig. 24, middle and right) display colour-coded membrane current landscapes (middle top) and extracellular glutamate concentration (right top) mapped onto cell morphology; the two corresponding plots (lower panels) display the variable time course of extracellular glutamate and clamp current at selected cell processes .

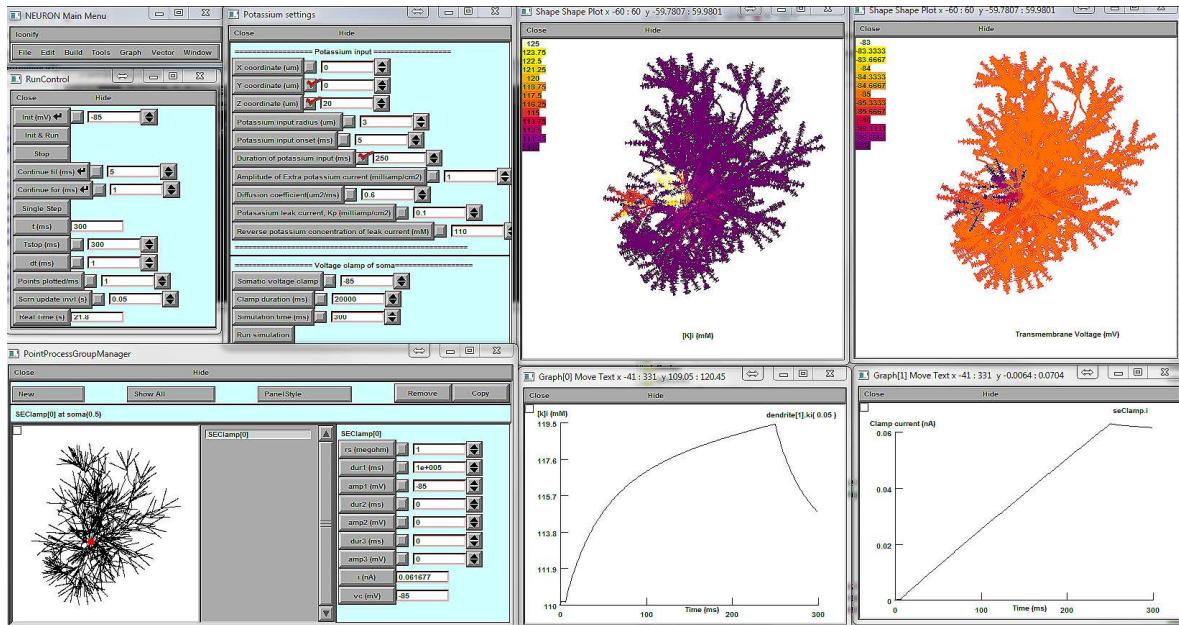
## Simulating astroglial intracellular potassium concentration

With this option, the user can simulate the dynamics of intracellular potassium as a result of local input potassium input and intracellular buffering (See the legend of Fig.25 for clarification).

From the main panel of Astro the user can active the potassium dynamics by clicking the button "Potassium Dynamics"



In result the user can find the following window:



**Figure 25.** Window panels with parameter settings (top left) and readout plots, to simulate the dynamic intracellular potassium landscape mapped onto cell morphology, in response to volume-limited application of potassium current. The simulation of potassium dynamics is done at fixed voltage with voltage clamp electrode indicated on the left-bottom panel.

'Potassium dynamics' panel (Fig. 25, top left) provides parameter settings for volume-limited potassium application (local current) within a round area of the cell, including sphere with coordinate X, Y, Z and the area radius; the onset; the rise and decay time and the maximum current amplitude concentration.

Potassium dynamics are computed in somatic voltage clamp configuration: the clamp parameters are indicated on the panel '**Somatic voltage clamp**' (Fig. 25, bottom left) including electrode resistance, clamp voltage and duration.

Intracellular potassium buffering is modelling with the linear approximation current

$$I_b = K_p \left( \frac{[K]_{in}}{[K]_b} - 1 \right), \text{ where } K_p \text{ is a density of current, } [K]_{in} \text{ is potassium intracellular concentration and}$$

$[K]_b$  is concentration of buffering level.

The voltage clamp current on the soma after uncaging of glutamate on the soma is shown on the left top panel.

The monitoring panels (Fig. 25, middle and right) display colour-coded potassium concentration landscapes (middle top) and membrane voltage (right top) mapped onto cell morphology; the two corresponding plots (lower panels) display the variable time course of intracellular potassium and clamp current at selected cell processes .

## SYSTEM PREPARATIONS FOR HIGH-END CALCIUM SIMULATIONS

### Preparing Worker computer / cluster (HPC, OS Linux) for Ca<sup>2+</sup> simulations

1. Install MPI (Message Passing Interface) on the cluster (most of the clusters have it already preinstalled). Free MPI software can be downloaded from  
<https://www.open-mpi.org/software/ompi/v3.0>
2. Install NEURON on the cluster. The latest version can be downloaded from the official site. We recommend using the installation from source code taking the sources from here  
<https://www.neuron.yale.edu/neuron/download/getstd> and following steps 1-5 of the next instruction [https://www.neuron.yale.edu/neuron/download/compile\\_linux](https://www.neuron.yale.edu/neuron/download/compile_linux).  
**--with-paranrn option should be added** to the configure command for NEURON installation to enable distributed computations.  
NEURON GUI is not required by cluster simulation. If the user wants to remove it from installation, don't download *iv-mm.tar.gz* archive and replace --with-iv=\$HOME/neuron/iv with –without-iv when calling configure for NEURON installation.
3. Put the *clusterCaSim/hpc* folder to the place shared between cluster nodes. For example, it can be saved in the directory */home/<username>*. The cluster setup is done.

The user can use PuTTY and WinSCP programs to work with console and file system of the remote cluster. Both programs are free.

---

..	
host	Deleted build artifacts and other files created during the simulation.
hpc	Deleted build artifacts and other files created during the simulation.

---

The structure of the directors, in GitHub, necessary for working with the cluster has the form:

All files from directory HPC should be downloaded on the cluster, keeping the structure of directories unchanged.

All files from directory HOST should be downloaded on the local computer (OS Windows), keeping the structure of directories unchanged.

### Preparing Host computer (client, OS Windows) for Ca<sup>2+</sup> simulation

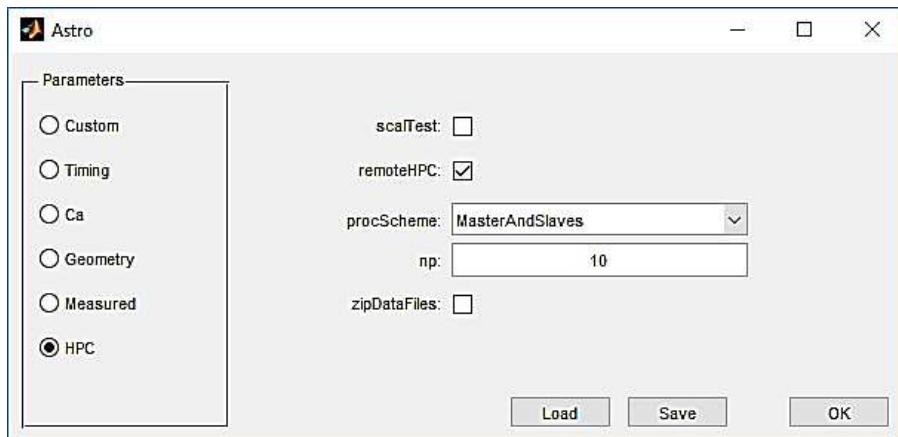
1. Open *clusterCaSim/host/scripts/win-lin/params.bat* and set your cluster connection parameters including the path to the *hpc* folder located in the cluster.

- Open *clusterCaSim/host/core/BasicParams.m* and fill *availableNodes* variable with names of your cluster nodes.

Also, fill *clusterCaSim/hpc/hostfile\_BusyMaster* and *hostfile\_IdleMaster* files with node names in the following manner: each line should contain the name of the node followed by ' max\_slots=1' without quotes.

Contents of both files should be the same except that *hostfile\_IdleMaster* should not include the master node (only slaves).

- After launching the program, the user can modify following parameters on the *HPC* panel:



*scalTest* – check if the scalability test is useful here. This test shows how well execution time scales relative to the number of processes

*remoteHpc* – uncheck if the software will be run on the client PC. The user should also modify *NRNDIR* and *HPCDIR* parameters that point to the NEURON and *hpc* folder locations on client PC in the *clusterCaSim/host/scripts/win-win/params.bat* and *params.sh*.

*procScheme* – processor distribution scheme

*np* – number of processors

If the user had compiled the executables to run the simulation, then the user should recompile them after changing any parameters in Matlab files using *clusterCaSim/host/BUILD\_AllHostExecutables.m*.

## Notes

### Nano geometry:

In data sets representing 3D-reconstructed nanoscopic processes there should be no serial-section layers without points, for diffusion simulations to work properly.

Because of a significant amount of computations in complex diffusion simulation Matlab cannot handle stopping or restarting it by pressing a button.

### Calcium dynamics on the cluster:

There are situations when selected geometry cannot be split into the specified number of processors. In this case, the user will see an MPI error before the computations begin. To solve the problem, the user can simply increase or decrease the number of processors.

Dendrites (astroglial processes will be called dendrites in the original NEURON environment) should be connected to the soma only in the 1 position. Otherwise cell splitting fails. Examples:

Good: soma[0] connect dendrite[125](0), 1

Wrong: soma[0] connect dendrite[125](0), 0.5

Wrong: soma[0] connect dendrite[125](0), 0.1

Wrong: soma[0] connect dendrite[125](0), 0