Flint 2.1: The User Guide

Flint project

October 15, 2018

Abstract

This document describes how to use Flint 2.1. Readers also find some OS-specific notes and trouble-shooting techniques which users would like to know when using Flint 2.1.

Contents

Chapter 1

Introduction

1.1 Brief summary about Flint

Flint 2.1 is new reimplementation of Flint 1.x, a simulator for biological and physiological models. Flint can run simulations of multi-level physiological models written in PHML. It means that Flint parses given models, performs numerical analysis for their simulation, and renders simulation outcome into a line graph via an external program such as gnuplot [1]. Likewise, Flint can handle CellML and SBML as well as SBML-PHML hybrid models.

1.1.1 Markup languages

Flint 2.1 supports the following standard languages of models:

- PHML [2] (including its precursor ISML [3])
- SBML [4]
- CellML [5] (supported as an experimental feature)

1.1.2 Solver methods for ordinary differential equations

Flint 2.1 supports the following algorithms to solve ODEs numerically:

- Euler method
- Runge-Kutta 4th-order method
- Adaptive stepsize Runge-Kutta method, based on the ARKode solver of SUNDIALS [6]

1.2 Notation

In this document, a sentence starting with `$` describes a command line in a command shell on your system, such as

```
$ echo this is a command line.
```

Chapter 2

Getting started

2.1 Install Flint 2.1

Flint project makes both Windows and macOS version of Flint 2.1's installer freely available at https://flintproject.github.io/. For Windows, double-clicking the .msi package will start the installation process. The .dmg archive for macOS contains Flint 2.1's .pkg file; extracting and double-clicking it to start an application bundle called "Flint.app".

2.2 Try your first simulation with Flint 2.1

This section describes a simple procedure with Flint 2.1 to run a simulation of an example model "HodgkinHuxley_1952_neuron_model.phz", which is distributed as part of the PhysioDesigner [7] installation.

Launch Flint

To launch Flint, double-click flint.exe on Windows, or Flint.app on macOS. It shows a window like Fig. 2.1.

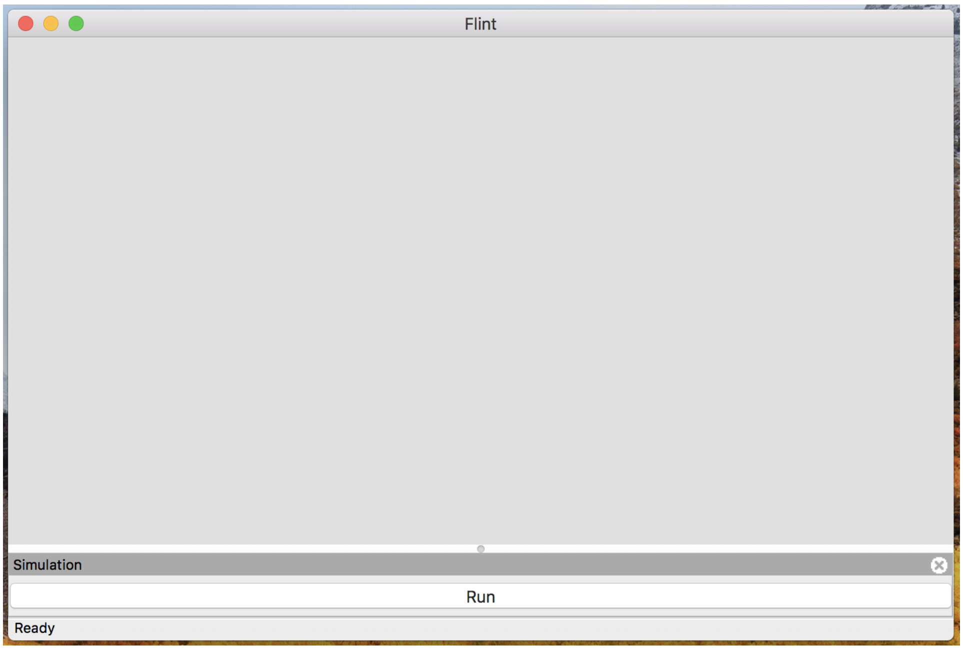


Figure 2.1: The initial window of Flint.

Open a model

In the "File" menu, select "Open" to choose a model file. Then you will see a file dialog like Fig. 2.2. Select "HodgkinHuxley_1952_neuron_model.phz" in the file dialog, and click "Open" button. Then the model window will appear as in Fig. 2.3.

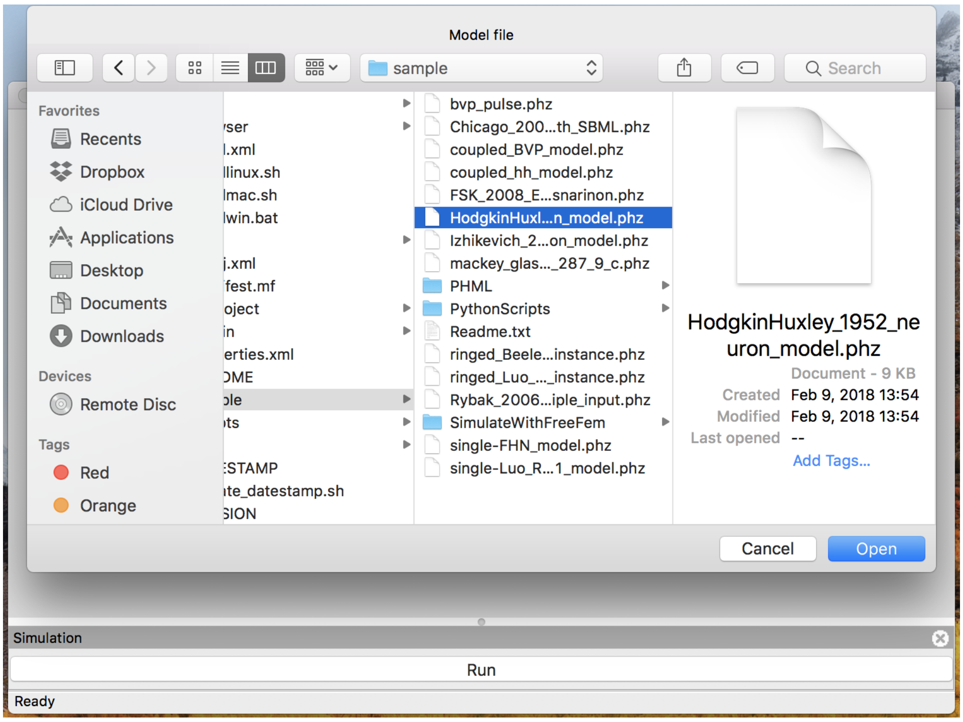


Figure 2.2: The file dialog to open a model.

Choose duration and time step

Specify the duration of simulation in “Simulation Length” and the time step length of the simulation in “Simulation Time Step” optionally.

Run a simulation

Click the “Run” button to start a simulation.

Once simulation started running, the progress bar will appear in the control panel in the right side like Fig. 2.4, and both the cross mark (as “Cancel”) and “Detail” buttons will be enabled.

Wait until the status bar tells that the simulation completed (see Fig. **??**).

See detail of the simulation

Click the “Detail” button to get the simultion result. Then a detail window will appear as in Fig. 2.6.

Select ordinates

Click the “View” button on the detail window, then a plot window to render line graphs about the simulation result, like Fig. 2.7. Check the Y1 column of “V” in the variable list, which calls gnuplot. Soon the corresponding line graph will appear on a separate window, like Fig. 2.8. Moreover you can also check the Y2 column of another variable “I_Na” in the list to arrange two line graphs in the same canvas, as in Fig. 2.9.

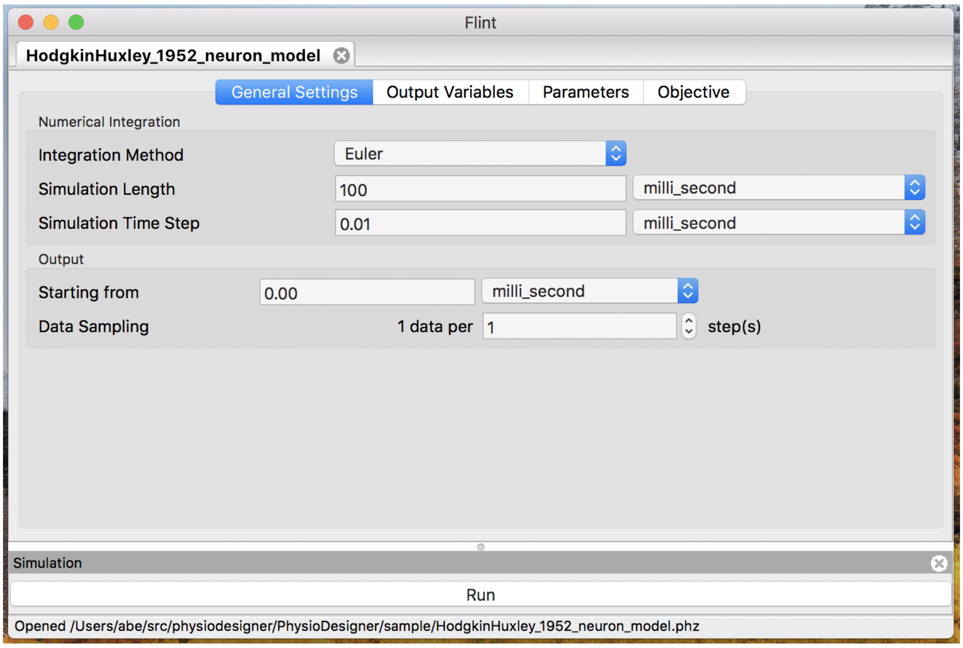


Figure 2.3: The model window.

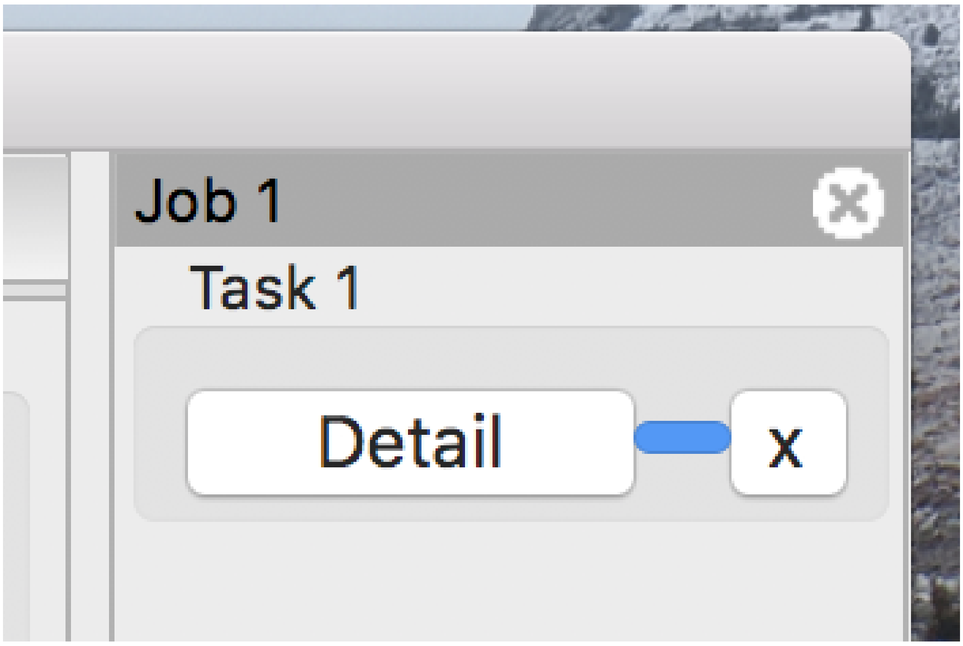


Figure 2.4: The progress bar for the model.

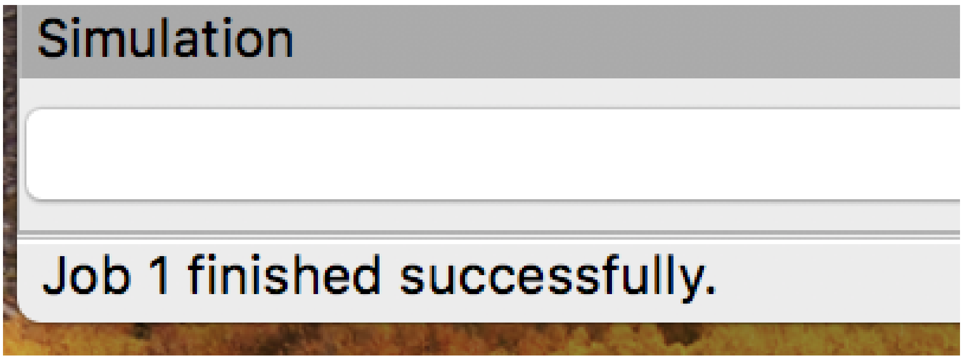


Figure 2.5: The status bar indicates the simulation completed.

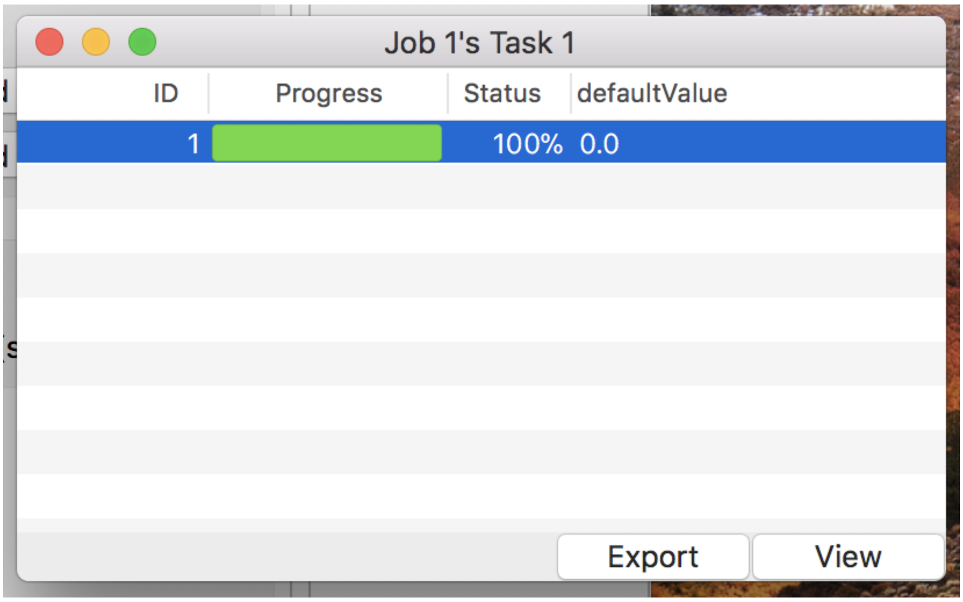


Figure 2.6: The detail window.

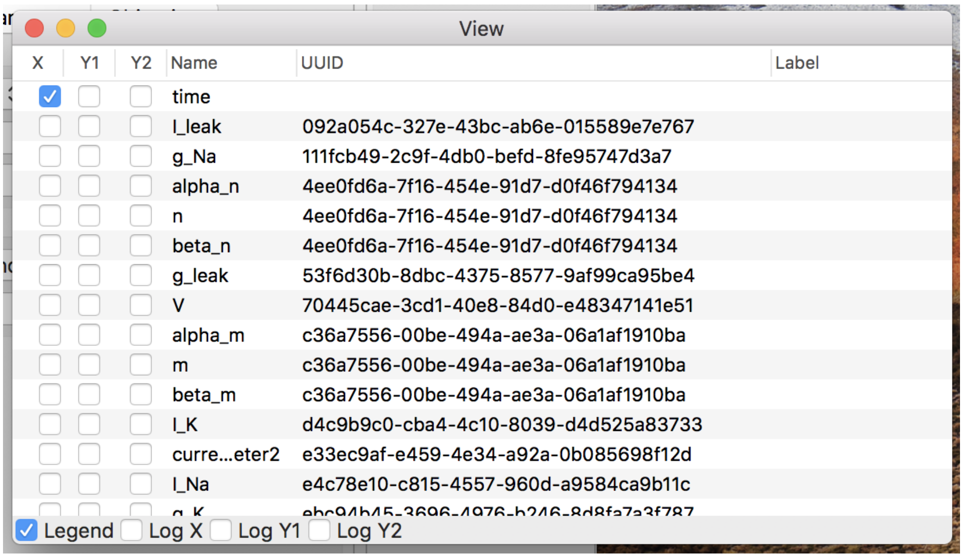


Figure 2.7: The plot window.

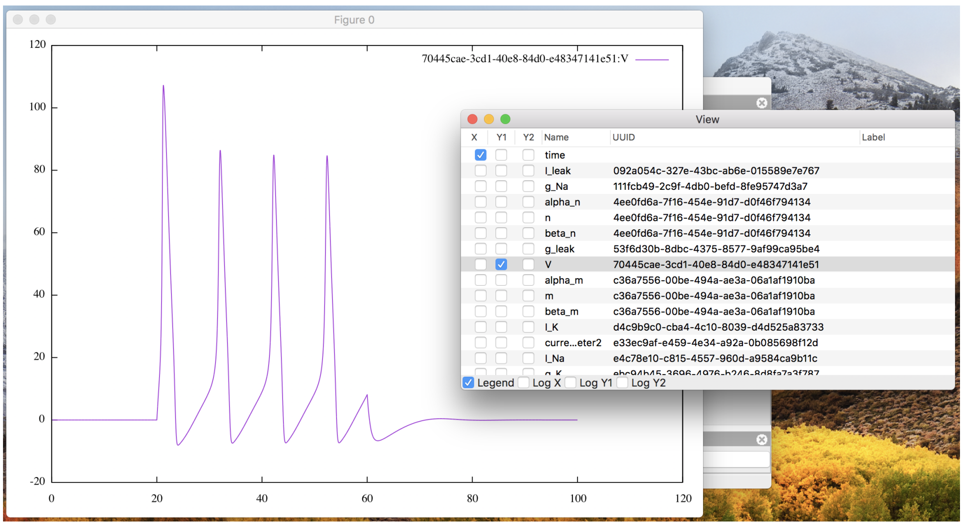


Figure 2.8: The plot window with "V" on Y1.

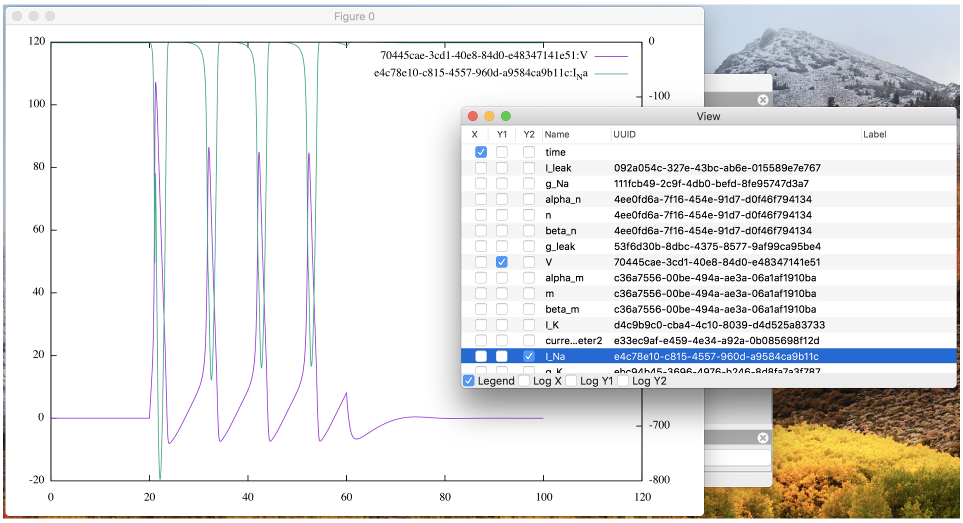


Figure 2.9: The plot window with “V” on Y1 and “I_Na” on Y2.

Chapter 3

Graphical User Interface

Flint 2.1 comes with a graphical user interface out of the box. This chapter explains features of the GUI and how to use them.

3.1 Launching Flint

On Windows, double-clicking "flint.exe" in the start menu starts Flint. On macOS, double-clicking "Flint.app" works similarly.

3.2 Quitting Flint

To quit Flint, use the menu "File"→"Exit".

3.3 Loading models

Flint must load models before running simulations for them. Users tell Flint which model should be loaded by opening the model file. Loading a model can fail due to some reasons; for example, it may fail if the model file contains an error or unsupported elements. An error dialog will display a diagnosis message when loading a model fails. Once loading a model successfully, the model window shows up and stays in the main window until closed, like Fig. 3.1.

3.4 Configuring simulation tasks

Before starting simulations for a loaded model, users may want to configure them in terms of numerical integration, simulation time, output data, and parameters.

3.4.1 Integration method

Users have to choose a solver method for ordinary differential equations at the "Integration method" combobox.

3.4.2 Simulation Length

Users must specify the total length of simulation time at the "Simulation Length" field; the given number is interpreted in terms of the selected time unit.

3.4.3 Simulation Time Step

Similarly to "Simulation Length", users can specify the time step at the "Simulation Time Step" field.

3.4.4 Starting from

Users can specify when (in the sense of simulation time) output starts from at this field. By default, simulation process produces output from time 0.

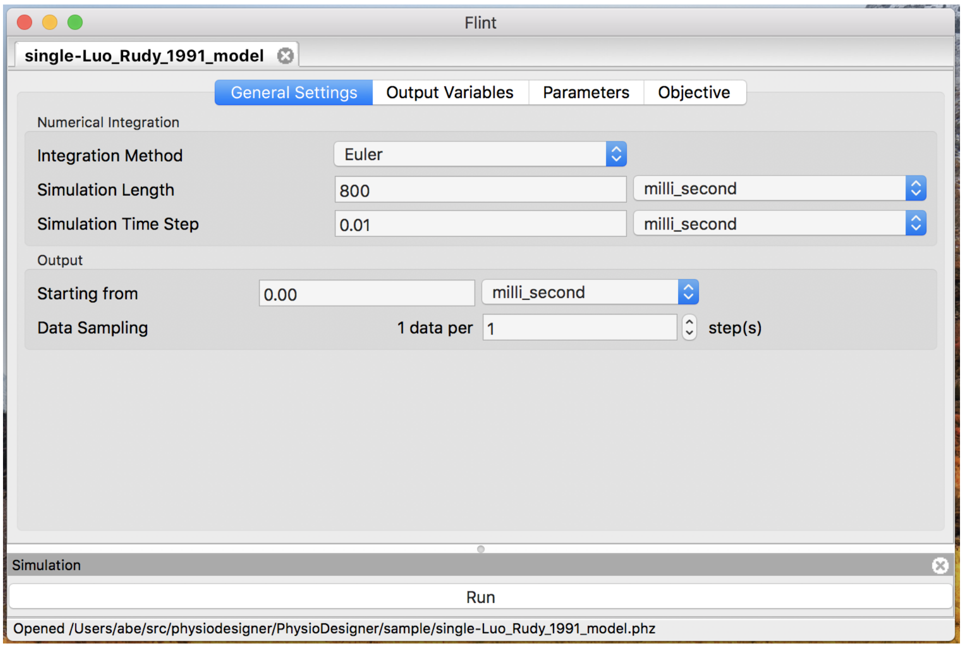


Figure 3.1: The model window.

3.4.5 Data Sampling

This setting is for determining how often the result data are written in. Note that the sampling rate does not affect the calculation for simulation.

3.4.6 Select output variables

Before starting simulations for a loaded model, users may want to choose a limited number of variables for output among available variables. Filtering output variables will reduce the burden of writing output, and thus may improve the simulation performance. The "Output Variables" panel (Fig. 3.2) allows user to select output variables by matching its properties, such as name, with a given string.

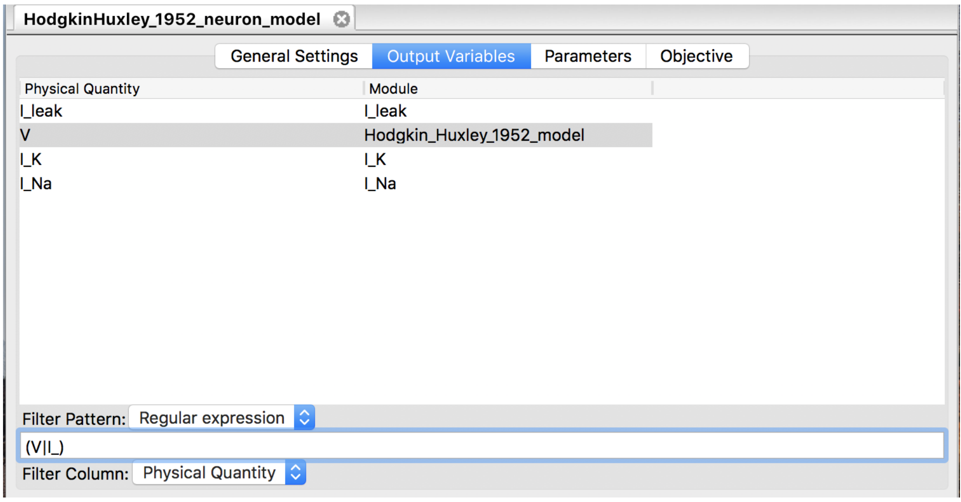


Figure 3.2: The "Output Variables" panel.

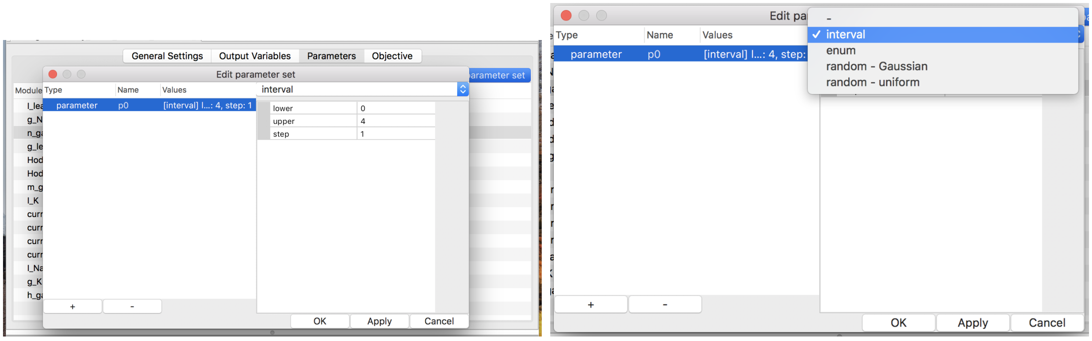
3.4.7 Parametrize constant values

By default, Flint run a simulation job for a loaded model. At the same time it is possible to run a bunch of simulations at once for a single model with different values of parameters.

A parameter is bound to a set of possible values, called value-set. The whole set of possible tuples of multiple parameters is defined as a cartesian product of multiple value-sets. If users edit parameter sets and use them, then Flint will run as many jobs for the model as cardinality of the cartesian product. In other words, each simulation job corresponds to a value tuple for multiple parameters.

Users can see and modify parametrization of constant elements in a loaded model, such as the initial values of ordinary differential equations and values of static-parameters of PHML, at the "Parameters" panel.

The table at the "Parameters" consists of each row corresponding to a constant element in the model; the "Expression" field of the row accepts a formula (in an infix notation) defining the parametrized value of the constant element.



(a) a edit a parameter set calld "p0"

(b) b choosing value type

Figure 3.3: Editing parameter sets in the "Define parameter set" window.

Define a value-set

In order to define or modify value-sets, push button "Define value set" at first. Then a window will pop up. It allows users to define new value-set, see existing value-sets, and modify them (see Fig. 3.3a).

There are four types of value-sets: enum, interval, Gaussian, and uniform (see Fig. 3.3b). For a value-set of type enum, each of possible values must be specified. On the other hand, only the lower and upper (both inclusive) of a range of values with a step are required to define a value-set of type interval. Note that possible values of an enum should be separated by a comma. The latter two types of value-sets are for generating (pseudo)random values according to specified probability distribution in simulation time.

Use value-sets

Once users have defined a value-set, it is available in the "Expression" field of any row in the "Parameters" table. For example, users can specify `a+2*b` in the field if there exists a couple of value-set named `a` and `b` (see Fig. 3.4).

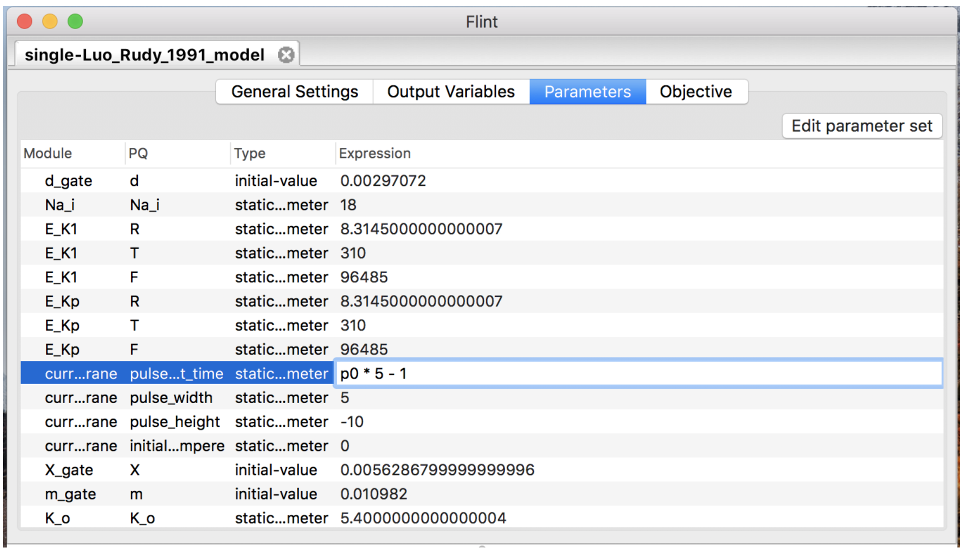


Figure 3.4: Parameterize with defined value-set "p0".

3.5 Starting simulation

To start simulation, use the menu "Control"→"Run" or button "Run" on the control panel. It kicks simulation jobs for all loaded models. Users can monitor the progress in total on the control panel, as well as the one for a single job on the detail windows like Fig. 3.5. Note that a context menu allows users to cancel simulation assigned to a specific parameter value in a task.

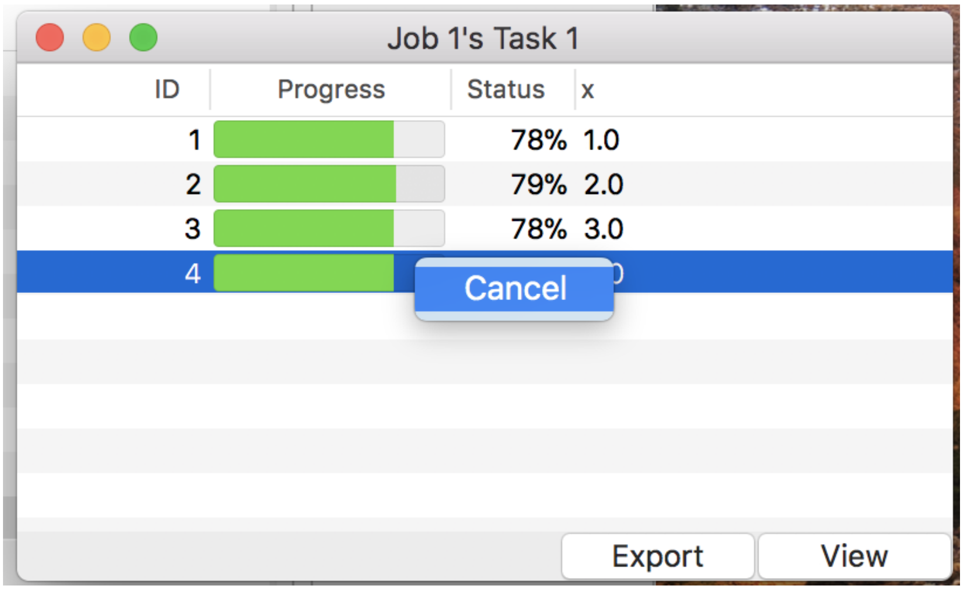


Figure 3.5: The detail window during simulation.

3.6 Controlling simulation jobs

After starting simulation jobs, users can control them instead of just waiting for them finishing.

3.6.1 Cancel jobs

There is another way to cancel running jobs; pushing the cross mark on the control panel (see Fig. 3.6), which cancels a job i.e. all of its tasks together.

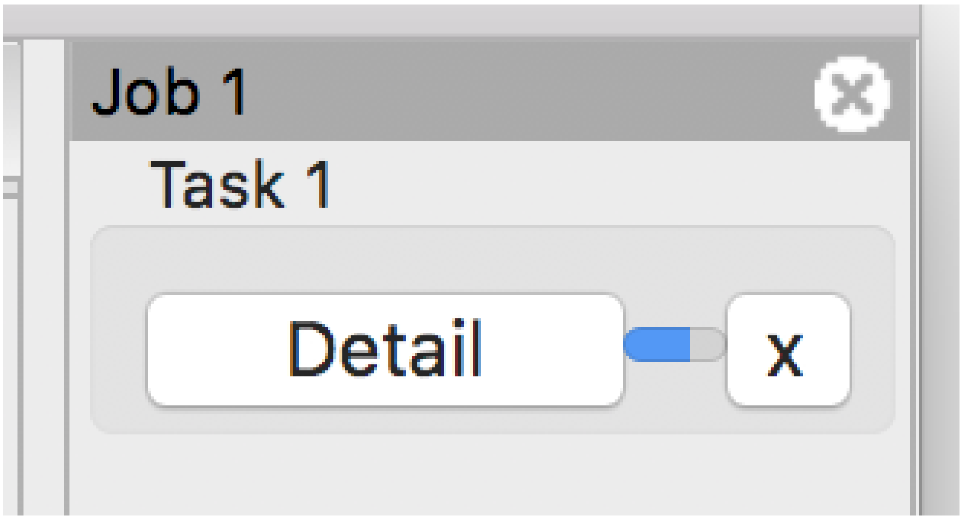


Figure 3.6: The progress bar / cross mark / "Detail" button on the control panel.

3.6.2 Pause and resume jobs

As in Fig. 3.7, users can pause jobs at any time during simulation by using the menu "Control"→"Pause". Resumeing paused jobs can be done with the menu "Control"→"Resume". Note that these operation affects all of alive jobs simultaneously.

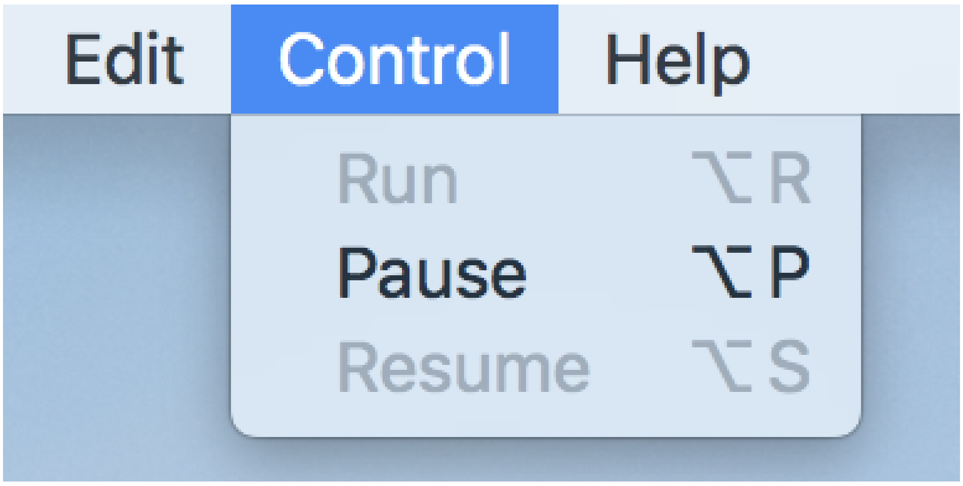


Figure 3.7: The Control Menu.

3.7 Visualizing simulation results

Flint has a feature to show a line graph for the result of a simulation on the fly, not only after its job finished, but also int the middle of ongoing simulation.

From the detail window, users can display the plot window by clicking button "View" for each simulation job.

3.7.1 Choose abscissa and ordinates

In order to draw a line graph, users have to specify the abscissa and ordinates by checking an X column as well as either Y1 or Y2 column. Immediately after choosing abscissa and ordinates, Flint calls gnuplot in the background to draw a line graph. Thus users have to install gnuplot in advance, and to specify the location of the gnuplot executable (see section 3.10).

Trouble shooting

- Choosing abscissa and ordinates results in no response, make sure if the gnuplot initialization file is valid and correct. It is called `.gnuplot` on Unix and macOS, and `GNUPLOT.INI` on other systems.

3.8 Saving output data

Users may save the resulting simulation data for later investigation.

3.8.1 Exporing data as CSV

Flint can export the result data into a CSV file. The header column contains the variable names as well as their unit name if any.

The procedure is as follows:

1. Open the "Detail" window
2. Select as many tasks as you would like to save.
3. Push button "Export"
4. Choose a target directory in the file dialog (see Fig. 3.8)

The names of files saved in the target directory are of form "(ID).csv."

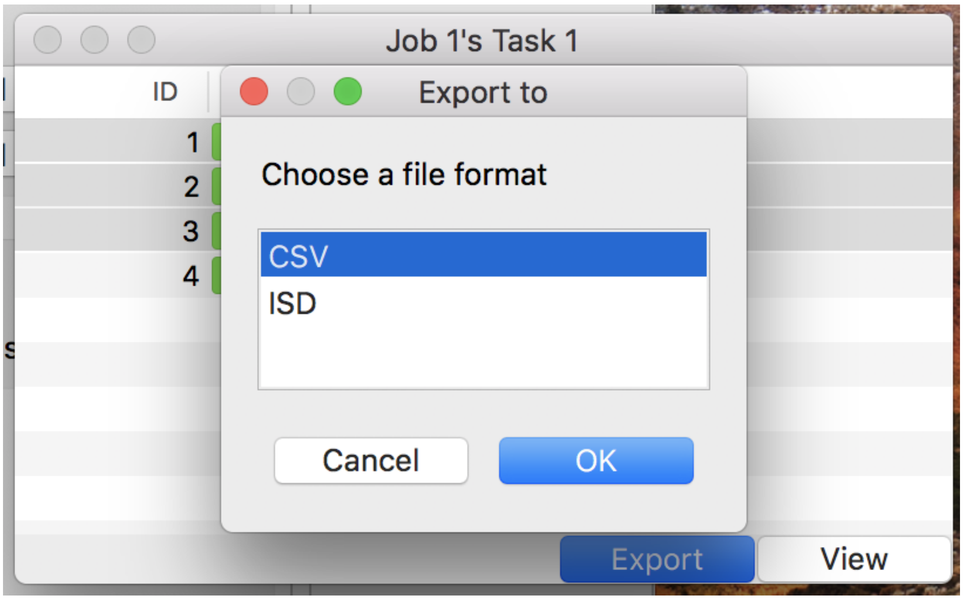


Figure 3.8: The dialog to save data.

3.8.2 Exporting data as ISD

Flint can also export the result data into a ISD file. The ISD file format is a binary file format for preserving multi-variate data.

The procedure is as follows:

1. Open the "Detail" window
2. Select as many tasks as you would like to save.
3. Push button "Export"
4. Choose a target directory in the file dialog (see Fig. 3.8)

The names of files saved in the target directory are of form "(ID).isd."

3.9 Exporting C source code from model

Not only running online simulation, but also Flint can export simulation code as a C99 source file from a loaded model. So far it works only for pure ODE models.

3.9.1 From menu

To export C code from a model,

1. Load a model
2. Select the menu "File"→"Export to C" (see Fig. 3.9)
3. Choose a target filename via the file dialog that follows.

Then a dialog will appear to tell whether it is done successfully or not.

Please note that the numerical method used in the exported code is the one specified in the original model, e.g., Euler or Runge-Kutta 4th-order method; the ARKode solver of SUNDIALS has not been supported yet.

3.9.2 How to build a program from exported code

Once a C source file exported, what to do next is building the program by a C compiler conforming C99 standard.

If, for example, gcc is available, then invoking the following code

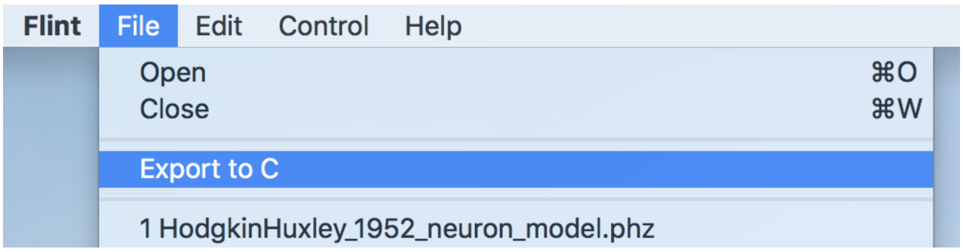


Figure 3.9: The menu "File"→"Export to C".

```
$ gcc -O3 -std=c99 -o simulate exported.c
```

will produce an executable named `simulate` from the C source file `exported.c`.

Finally,

```
$ ./simulate output.isd
```

will run a simulation, writing the whole output into `output.isd`.

3.10 Preference

Users can customize Flint's behavior via preference, which UI looks like Fig. 3.10.

3.10.1 Concurrency hint

The concurrency hint helps Flint run multithread simulation with an optimized number of concurrent threads. By default Flint automatically detects the number of cores and preset it for the hint.

3.10.2 Plotter

This is a necessary option to render line graphs. Select the path of `gnuplot` (or `gnuplot.exe` on Windows), e.g., "`/usr/bin/gnuplot`". If macOS users have, say, `/Applications/gnuplot.app` as an application bundle of gnuplot, its value should be

`/Applications/gnuplot.app/bin/gnuplot`.

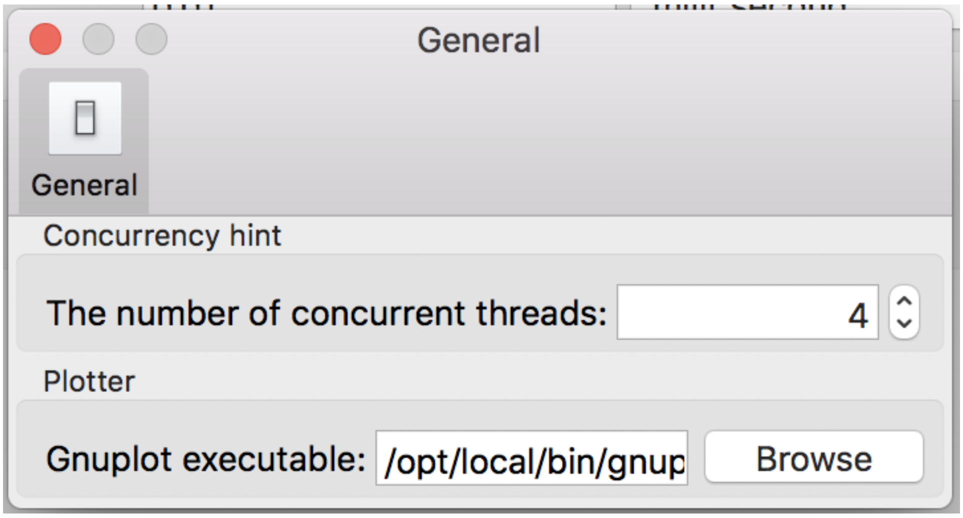


Figure 3.10: The "Plotter" panel on the preference dialog.

3.11 Shortcut keys

There are useful shortcut keys as follows:

3.11.1 Keys for main menu

Command	Shortcut keys on macOS	Shortcut keys on Linux or Windows
File → Open	`Cmd`+O	`Ctrl`+O
File → Exit	`Cmd`+Q	`Ctrl`+Q
File → Save configuration	`Cmd`+S	`Ctrl`+S
File → Save configuration as...	`Cmd`+`Shift`+S	`Ctrl`+`Shift`+S
Edit → Copy	`Cmd`+C	`Ctrl`+C
File → Cut	`Cmd`+X	`Ctrl`+X
Edit → Preference	`Cmd`+,	`Ctrl`+,
Control → Run	`Option`+R	`Alt`+R
Control → Pause	`Option`+P	`Alt`+P
Control → Resume	`Option`+S	`Alt`+S

3.11.2 Additional keys

Both `Esc` and `Ctrl`+`W` (or `Cmd`+`W` on Mac) can close an active subwindow in which there is no dedicated button to close it.

Chapter 4

Command Line Interface

Flint 2.1 allows users to run a simulation in a command shell.

4.1 Launching Flint

4.1.1 Invocation with no arguments

It is possible to launch Flint with the command `open(1)` of macOS as follows:

```
$ open Flint.app
```

Note that it does nothing but launches the graphical user interface of Flint. In a `cmd` session on Windows,

```
$ flint.exe
```

has the similar effect.

4.1.2 Invocation with filenames

If filenames of models are given in the command line on Windows:

```
$ flint.exe model1 model2 ...
```

then Flint tries to open them immediately after launching the GUI.

4.2 Showing help

Specifying `-help` in the command line shows the help message.

4.2.1 Synopsis

On Windows:

```
$ flint.exe -help
```

4.3 Running a simulation: the headless mode

Specifying `-headless` in the command line enable the headless mode, which runs a simulation of given model with the default configuration.

4.3.1 Synopsis

On Windows:

```
$ flint.exe -headless input output [-e file] [-g n] [-s file]
```

Load a model at `input`, simulation it with the default configuration, and leave the result at `output`. The following suboptions are available:

-e `file`
: save error messages during simulation as `file`.

-g `n`
: specify output sampling rate i.e. 1 output per `n` step.

-s `file`
: specify output variables with `file`.

Chapter 5

Frequently Asked Questions (FAQ)

Please read this chapter first when in doubt.

5.1 How to uninstall Flint

On windows, you can uninstall Flint through the system menu "Settings"→"Apps & features". On macOS, all you have to do for uninstallation is to remove Flint.app.

5.2 How to file a bug report or a feature request

Please visit Flint's GitHub Issues page at `https://github.com/flintproject/Flint/issues`, and find whether your issue has been reported or not. Feel free to file it if there is no similar issue.

Bibliography

[1] Thomas Williams, Colin Kelley, et al. gnuplot 5.0: An Interactive Plotting Program, August 2017. http://gnuplot.info/.

[2] PhysioDesigner project. Physiological Hierarchy Markup Language (PHML) Language Specification Version 1.0. http://physiodesigner.org/resources/specifications/specification_phml_ver1.0.pdf.

[3] Yoshiyuki Asai, Yasuyuki Suzuki, Yoshiyuki Kido, Hideki Oka, Eric Heien, Masao Nakanishi, Takahito Urai, Kenichi Hagihara, Yoshihisa Kurachi, and Taishin Nomura. Specifications of insilicoml 1.0: A multilevel biophysical model description language. *The Journal of Physiological Sciences*, 58(7):447–458, 2008.

[4] Michael Hucka et al. The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524531, 2003.

[5] Autumn A. Cuellar, Catherine M. Lloyd, Poul F. Nielsen, David P. Bullivant, David P. Nickerson, and Peter J. Hunter. An overview of cellml 1.1, a biological model description language. *SIMULATION*, 79(12):740–747, 2003.

[6] Alan C Hindmarsh, Peter N Brown, Keith E Grant, Steven L Lee, Radu Serban, Dan E Shumaker, and Carol S Woodward. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.

[7] Y. Asai, T. Abe, M. Okita, T. Okuyama, N. Yoshioka, S. Yokoyama, M. Nagaku, K. I. Hagihara, and H. Kitano. Multilevel modeling of physiological systems and simulation platform: Physiodesigner, flint and flint k3 service. In *2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet*, pages 215–219, July 2012.