Flint 1.8: The User Guide

Flint project

June 9, 2017

Abstract

This document describes how to use Flint 1.8. Readers also find some OS-specific notes and trouble-shooting techniques which users would like to know when using Flint 1.8.

Contents

Chapter 1

Introduction

Welcome to Flint 1.8.

1.1 Brief summary about Flint

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 on the plot window's canvas. Likewise, Flint can handle SBML and SBML-PHML hybrid models. Furthermore, Flint offers a plug-in feature with which different types of plotting applications such as gnuplot are available.

1.1.1 Markup languages

Flint 1.8 supports the following three languages for models:

- PHML/ISML
- SBML

1.1.2 Solver methods for ordinary differential equations

Flint 1.8 supports the following two algorithms to solve ODEs numerically:

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

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 1.8

Flint project makes both Windows and macOS version of Flint 1.8 freely available at https://flintproject.github.io/. The .dmg archive for macOS contains Flint's application bundle, named "Flint.app"; extracting it and copying it into your favorite path is all you have to do for installation. For Windows, double-clicking the .msi package will start the installation process.

2.2 Try your first simulation with Flint 1.8

This section describes a simple procedure with Flint 1.8 to run a simulation of a sample model "HodgkinHuxley_1952_neuron_model.phml", which is distributed as part of the PhysioDesigner installation.

Launch Flint

To launch Flint, double-click flint.exe on Windows, or Flint.app on macOS.

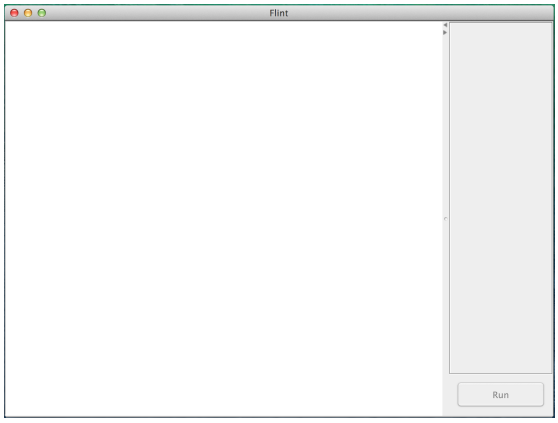


Figure 2.1: The initial window of Flint.

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.phml"

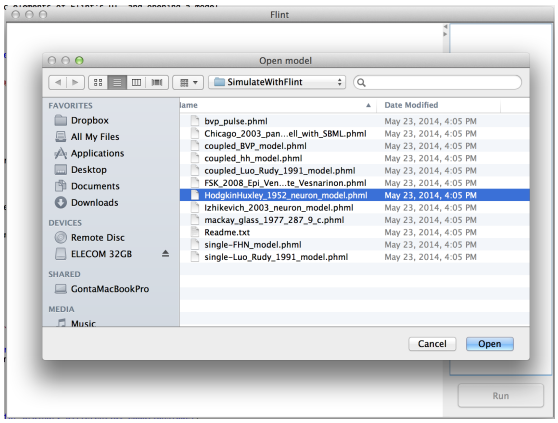


Figure 2.2: The file dialog to open a model.

in the file dialog, and click "Open" button. Then the model window will appear like Fig. 2.3.

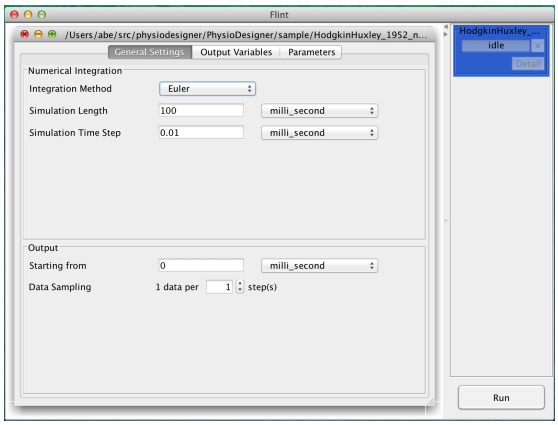


Figure 2.3: The model window.

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 "Cancel" (with the cross mark) and "Detail" buttons will be enabled.

Wait until a message dialog shows up to tell that the simulation completed.

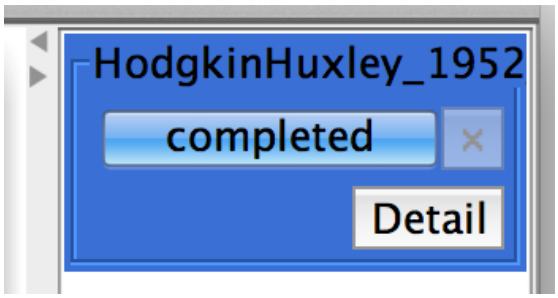


Figure 2.4: The progress bar for the model.

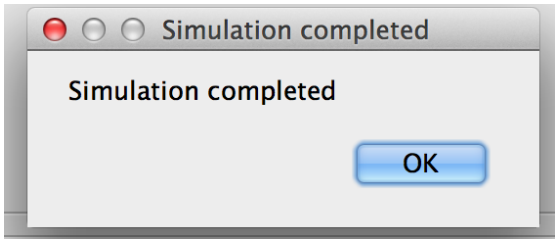


Figure 2.5: The message dialog says the simulation completed.

See detail of the simulation

Click the "Detail" button to get the simultion result. Then a detail window will appear like Fig. 2.6.

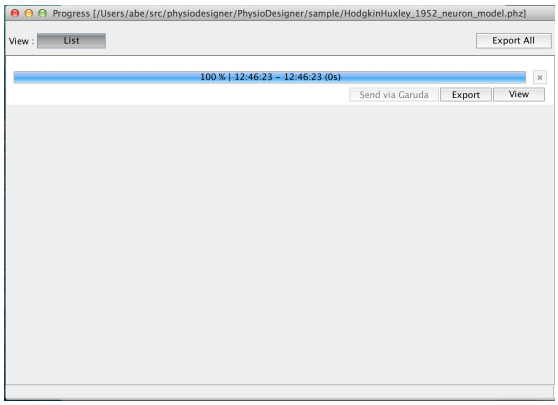


Figure 2.6: The detail window.

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. Drag a name card "V" from the variable list, and drop it into the right axis field labelled Y1. Soon the corresponding line graph will appear on the canvas, like Fig. 2.8. At the same time you can also pick up another name "I_Na" from the list, moving it into the left axis field labelled Y2, like Fig. 2.9.

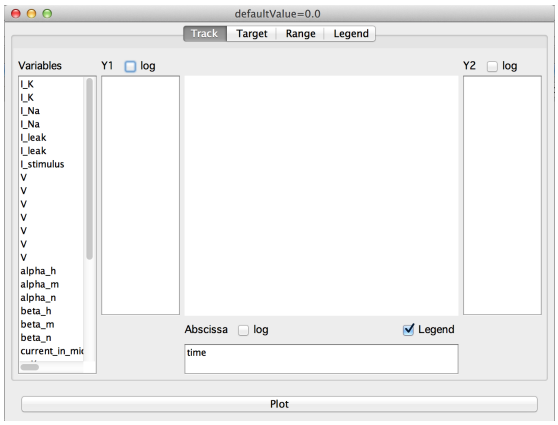


Figure 2.7: The plot window.

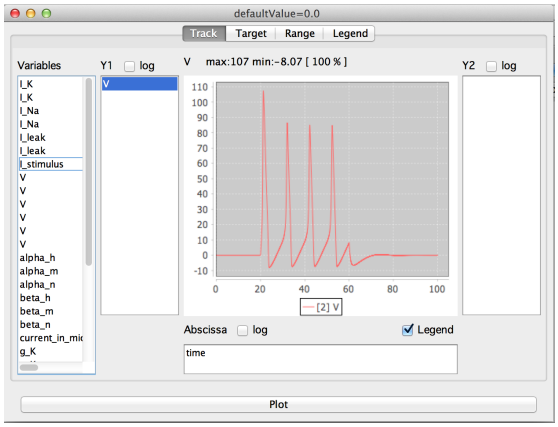


Figure 2.8: The plot window with “V” on Y1.

Plot a line graph with gnuplot

Push button “Plot” to draw the line graph about “V” and “I_Na” with gnuplot. Soon a gnuplot window will pop up like Fig. 2.10.

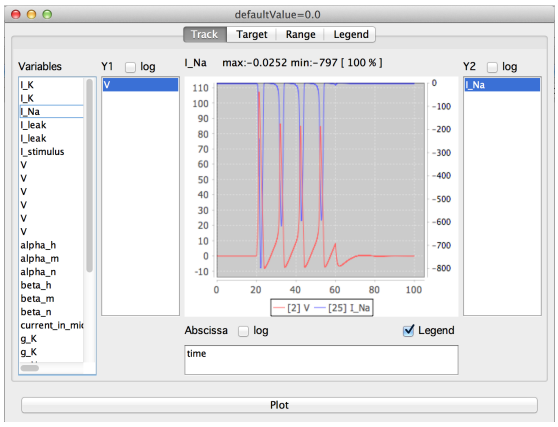


Figure 2.9: The plot window with “V” on Y1 and “I_Na” on Y2.

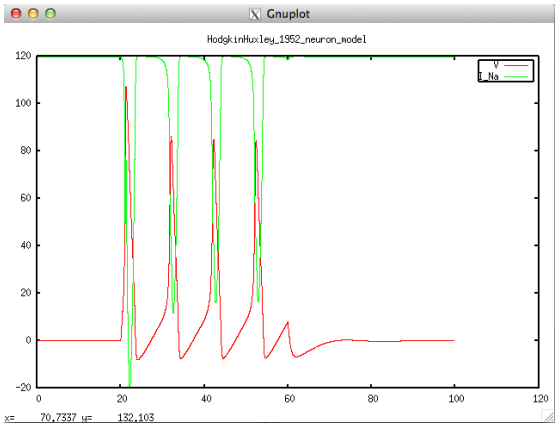


Figure 2.10: Line graph by gnuplot about “V” and “I_Na”.

Chapter 3

Graphical User Interface

Flint 1.8 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.

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.

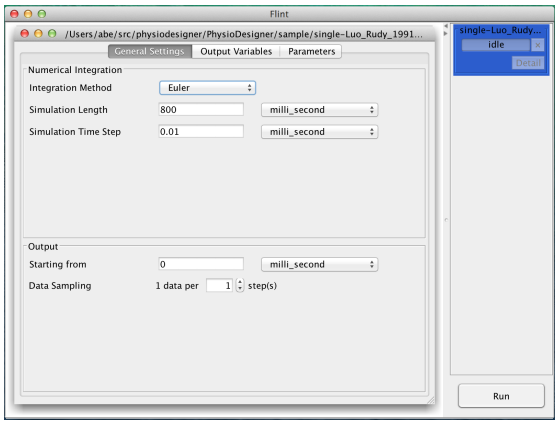


Figure 3.1: The model window.

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.

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 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.

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

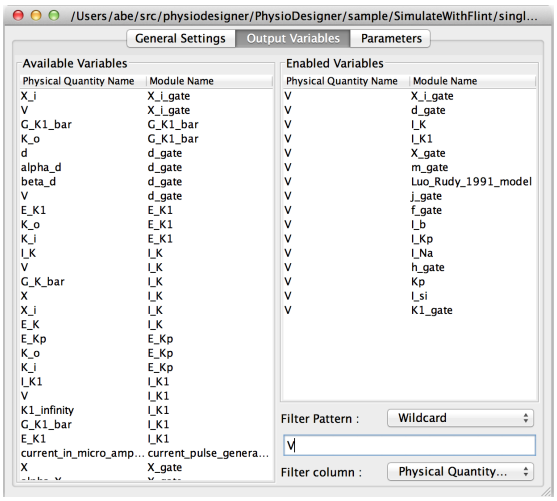
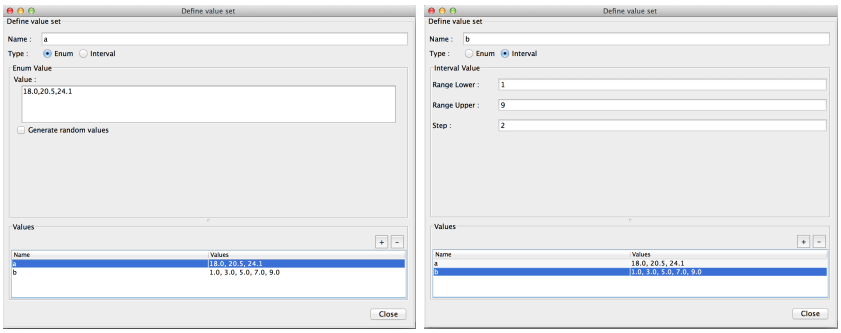


Figure 3.2: The “Output Variables” panel.

of multiple value-sets. If users define value-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` of type enum (b) `b` of type interval

Figure 3.3: Editing value-sets in the “Define value 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.

There are two types of value-sets: enum and interval. 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.

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`.

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. 3.4.

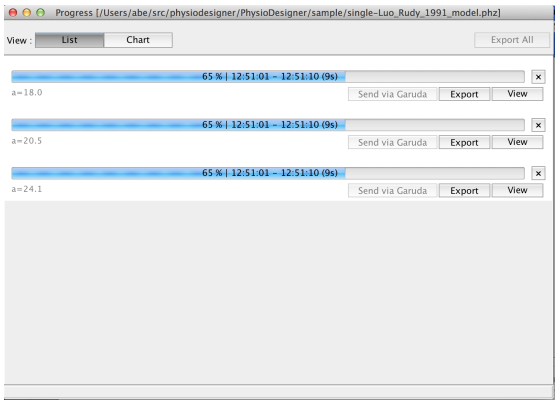


Figure 3.4: The detail window during simulation.

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 are two ways to cancel running jobs; one is pushing the cross mark on the control panel (see Fig. 3.5), which cancels all of related jobs. The other is pushing the cross mark at the detail window, which cancels the corresponding job only.

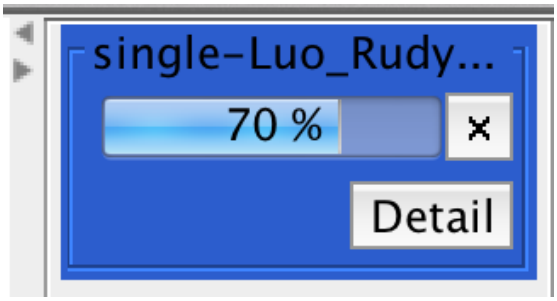


Figure 3.5: The progress bar / cross mark / "Detail" button on the control panel.

3.6.2 Pause and resume jobs

As in Fig. 3.6, 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.

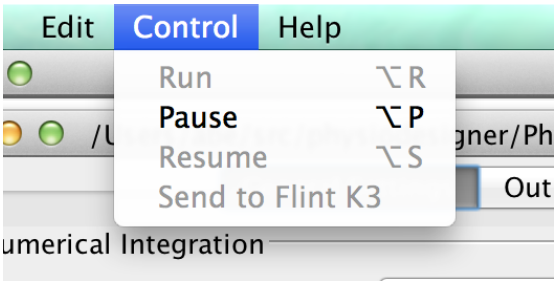


Figure 3.6: The Control Menu.

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 drag-and-dropping variables from the left side frame.

If there are so many variables that some of them scroll out on a frame, a scroll bar appears on the left edge of the frame. Besides scrolling them by wheeling with mouse, especially on the left side frame, the focus can jump to a variable by typing an alphabet key (i.e., `[A-z]`) which specifies the first character of the variable's name.

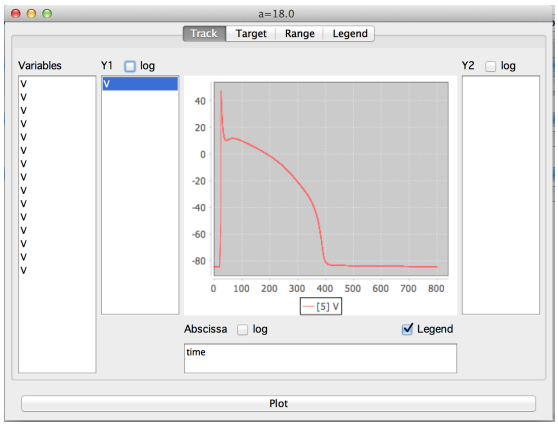


Figure 3.7: Selecting `V` as a Y1 ordinate.

3.7.2 Configuration for gnuplot

Pushing the "Plot" button calls gnuplot to draw a line graph with the specified abscissa and ordinates. Users have to install gnuplot in advance, and to specify the location of the gnuplot executable (see section 3.11).

Options for gnuplot

Users can control the following options for gnuplot in Tab "Range" or "Legend":

- Minimum and/or maximum value(s) for the abscissa (X) and/or ordinates ($Y1/Y2$)
- Labels for the abscissa (X) and/or ordinates ($Y1/Y2$)
- With or without the title

These will be enabled on next time plotting.

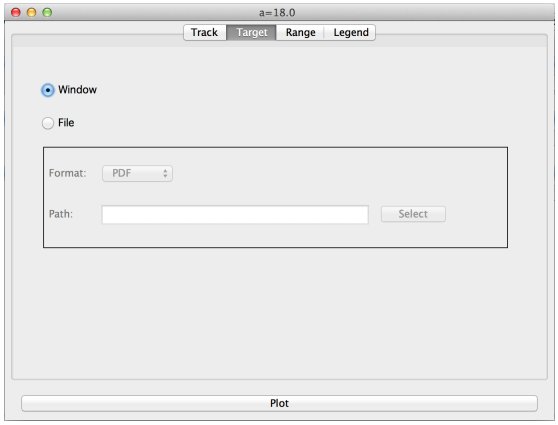


Figure 3.8: The "Target" tab.

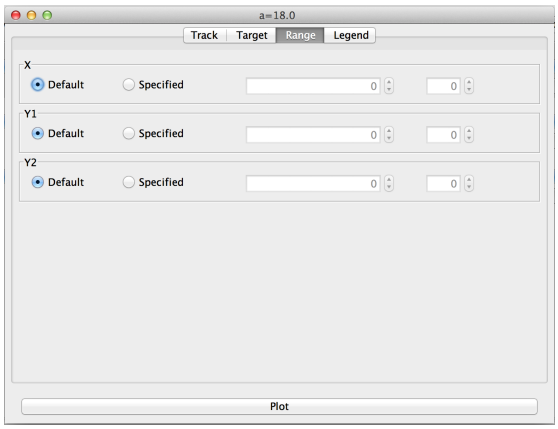


Figure 3.9: The “Range” tab.

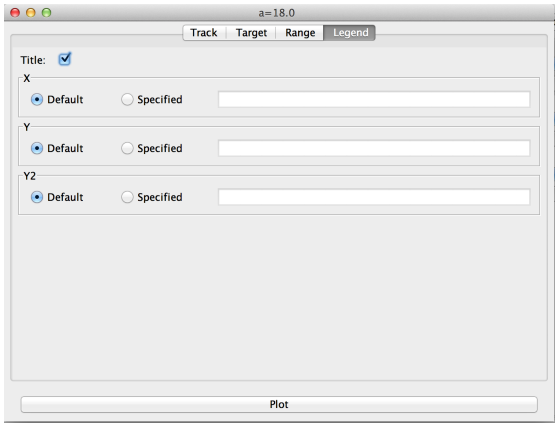


Figure 3.10: The “Legend” tab.

Writing the line graph to a file

It is possible to make gnuplot write the line graph to some file, as follows:

- choose Tab “Target”.
- check the “file” button.
- select a file format and path to be written.
- push button “Plot”.

Then a dialog will inform users of the resulting status.

Trouble shooting

- Pushing button “Plot” with no response, check the gnuplot initialization file to load. It is called `.gnuplot` on Unix and macOS, and `GNUPLOT.INI` on other systems.

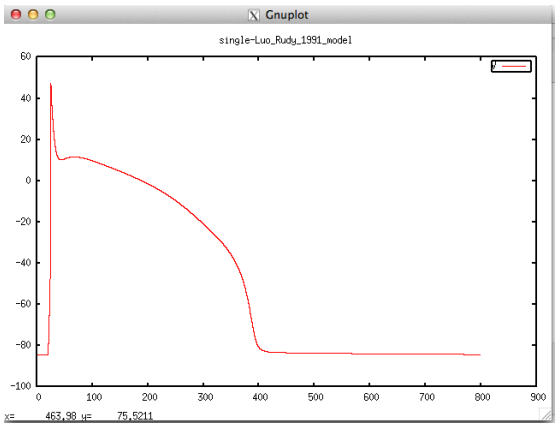


Figure 3.11: The line graph generated by gnuplot.

3.8 Saving output data

Users may save the resulting simulation data for later investigation.

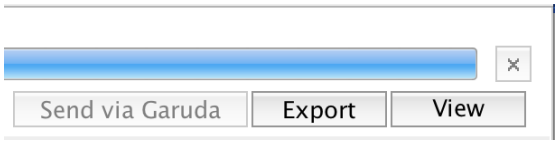


Figure 3.12: Buttons to export data in the detail window.

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. Push button “Export” (see Fig. 3.12)
3. Choose a target filename *ending with .csv or .txt* in the file dialog.

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. Push button “Export” (see Fig. 3.12)
3. Choose a target filename *which extension is neither .csv nor .txt* in the file dialog.

3.8.3 Exporting all data into a directory

It is sometimes useful to export all data for simulation jobs into a directory. The procedure is as follows:

1. Open the "Detail" window
2. Push button "Export all" (see Fig. 3.4)
3. Choose a target directory in the file dialog.

3.8.4 Sending data to a Garuda gadget

If Flint is enabled for the Garuda protocol (see subsection 3.11.2 about how to turn it on), and an Garuda gadget is able to receive data in CSV or ISD, it is easy to send the result from Flint through the Garuda protocol as follows:

1. Open the "Detail" window
2. Push button "Send via Garuda" (see Fig. 3.4)
3. Choose a target gadget from the list in the next dialog.

3.9 Flint K3

Instead of runnsing simulation with Flint, it is also possible to submit a loaded model to Flint K3 (`https://flintk3.unit.oist.jp/`) for simulation. A K3 account is necessary to use this advanced feature.

3.9.1 From menu

To submit a model to Flint K3,

1. Load a model
2. Select the menu "Control"→"Send to Flint K3"
3. Enter the User ID and Password of a K3 account and the title of new job (see Fig. 3.13)
4. Push button "Submit"

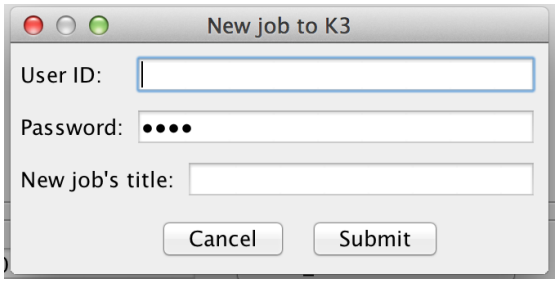


Figure 3.13: The "New job to K3" dialog.

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

3.10 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.

3.10.1 From menu

To export C code from a model,

1. Load a model
2. Select the menu "File"→"Export"→"C"
3. Choose a target filename via the file dialog that follows.

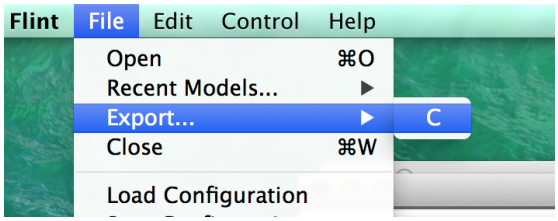


Figure 3.14: The menu "File"→"Export"→"C".

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.

3.10.2 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

```
$ 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`.

3.11 Preference

Users can customize Flint's preference.

3.11.1 Plotter

This is an advanced option. For gnuplot, select the path of `gnuplot` (or `pgnuplot.exe` on Windows), e.g., "`/usr/bin/gnuplot`". On macOS, say, `/Applications/gnuplot.app` as an application bundle of gnuplot, its value should be

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

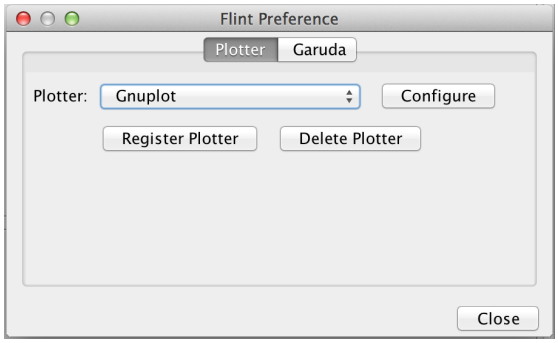


Figure 3.15: The "Plotter" panel on the preference dialog.

3.11.2 Garuda

Turning it on makes Flint enabled for the Garuda protocol on the desktop, like Fig. 3.16.

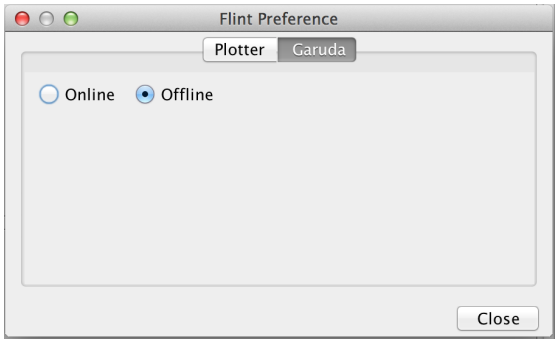


Figure 3.16: The "Garuda" panel on the preference dialog.

3.12 Shortcut keys

There are useful shortcut keys as follows:

3.12.1 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
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

3.12.2 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 1.8 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 a sampleing 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 On Windows

Failure at loading a model

If the error message shows an error message like

```
'flint-open' is not recognized as an internal or external command,
operable program or batch file.
```

then make sure that the PATH environment variable contains

```
C:\Program Files (x86)\Flint\
```

and try Flint again after rebooting Windows.