Double pendulum simulator
User manual

Bence Göblyös

November 22, 2023

1 Installation

1.1 Linux and BSD

1.1.1 Dependencies

A C compiler is required to build the application. It has been tested with GCC (and briefly with clang), but any other compiler should work as well. It is also advised to have access to the `make` utility to run the build scripts.

The application relies on `gnuplot` to draw the phase space plots, so it is highly recommended to install it.

ImageMagick may be used to convert the PPM output into more common formats, but it is only required if the image viewer of choice doesn't support PPM images (KDE's `gwenview` can open them by default, for instance).

Platform-specific install commands:

- Debian:
  `apt install build-essential gnuplot-nox imagemagick`
- Fedora:
  `dnf install @c-development gnuplot-minimal ImageMagick`
- FreeBSD:
  `pkg install gnuplot-lite ImageMagick7-nox11`

1.1.2 Building

To build the application, run `make` as an unprivileged user. In case `make` is not available, the following command will compile the application:

```
cc -s -O2 src/main.c src/input.c src/sim.c -o bin/dpsim -lm
```

1.1.3 Installing

The build process will produce a binary in the bin directory. It is possible to install it to `/usr/local/bin/dpsim` by running `make install` as root.

1.2 Windows

1.2.1 Dependencies

The application uses `gnuplot` to genreate phase space plots and ImageMagick to convert PPM output images into more common formats. The latter is optional but highly recommended, as the default photo viewer cannot open PPM images.

1.2.2 Building with GCC (MinGW-w64)

For this method, a Windows build of GCC is required, obtained either as a standalone installation of MinGW-w64, or through Cygwin or MSYS2.

Once inside the build environment, the application may be compiled by running `win/mingw.bat`. The output binary will be `bin/dpsim.exe`.

1.2.3 Building with MSVC

After the MSVC development environment is set up (a guide can be found on this page), running `win/msvc.bat` from a developer terminal will build the application. The output binary will be `bin/dpsim.exe`.

1.3 MacOS

1.3.1 Dependencies

The application requires `gnuplot` to generate phase space plots (downloads can be found on this page) and ImageMagick to convert PPM output images into more common formats. The latter is optional, as the default image viewer can open PPM files.

Additionally, the Xcode command line tools are required to compile the application. They may be installed by running `xcode-select --install` in the terminal.

1.3.2 Building

The application may be compiled by running `make` in the project directory.

2 Usage

After starting the program, a text menu will appear with 3 submenus:

- **General options**: This menu contains the common simulation parameters:
  - m for mass of the pendulum
  - l for length of the pendulum
  - g for gravitational acceleration

  - t for simulation time (cutoff time for flipover map)
  - f for frequency, the number of steps to take per second

- **Full-trajectory simulation**: This menu contains the options for simulating the entire trajectory of a double pendulum and saving the phase space:

  - **theta1** and **theta2** are the starting angles of the upper and lower pendulums respectively, in radians.
  - **Plotting frequency** specifies the number of data points per simulated second that should be used to generate the phase space plots. Higher simulation frequencies result in more accurate simulations, but plotting them isn't always necessary, so the application can skip data points to make the plotting faster. Setting it at or above the simulation frequency will cause every data point to be plotted.
  - **Run simulation** will start the simulation and keep results in memory.
  - **Save data to csv** will export the simulation data into the specified CSV file. If no simulation has been done yet or if the parameters have changed, it will start the simulation as well.
  - **Plot phase space** uses `gnuplot` to generate an SVG plot of the phase space. Similarly to the previous option, it will start a simulation if no up-to-date results are found.

- **Flipover time simulation**: Run multiple simulations and plot the time it takes for the lower pendulum to flip over as a function of the starting angles:

  - **Pixels per side** defines the side length of the resulting matrix.
  - **Run simulation** will start the simulations and keep the results in memory.
  - **Save output to PPM** saves the result matrix into a PPM image. If no up-to-date results are found, a new simulation will be started.
  - **Convert PPM to another image format** calls ImageMagick to convert an existing PPM file into a more common format. It does not check for an up-to-date simulation.