
Loader Plotter Documentation

Release 1.0

A. L. Almeida

Dec 20, 2017

This is a auto-generated documentation of the Loader-Plotter produced. The main objective of this loader is to grab the Data from ArrayFire format files and a designated folder. With that and the class requirements, it loads the data to *Numpy* arrays so that it can Plot Surface, Plot 3D and Construct Movies.

Requirements:

To run this Loader one need to have installed various modules: *** Numpy ** ArrayFire ** Matplotlib ** ImageIO ** DateTime*

class `Loader.meshPlot` (*InPath, Plot1D=True, PlotSlices=False, PlotTemporal=False, Mov=False*)

Class that contains information about the mesh of the data loaded.

Only works with ArrayFire file format in the following protocol: SF_XXXX.af :: For different suffix and prefix the *load_parameters()* has to be changed

At the moment it only has the hability to process 1D input data and turn them in a 2D (or semi-3D) output figure/movie.

The format of movie/figures can be altered in the specific function, namely *plot_sequence(InPath)*

The Initializer does Initiate the object to proceed with the data loading and dump them in a specific required format as output.

Parameters

- **InPath** – The relative path where the group of ArrayFire data is located.
- **Plot1D** – A Boolean parameter to inform the class if the user wants the set of input data to be converted to a 2D plot.
- **PlotSlices** – A Boolean parameter to inform the class if the user wants a Semi-3D plot of the group data available in the InPath folder.
- **PlotTemporal** – A Boolean parameter to inform the class if the user wants a output in the 2D color-contrast mode graph (pcolor/imshow format)
- **Mov** – A simple Boolean to inform the class if the user wants a final video of the movement to be rendered.

dissSech (*xx, A, B, O, x0, d*)

This function has the same behavior as the one called *dissSechAbs* but has no absolute conversion to be handled for statistical analysis. Be careful when using it.

dissSechAbs (*xx, A, B, O, x0, d*)

Defined the function to analyze Bright Solitons. These are one of the many types of solitons observed in these kind of equations. Bright solitons are the ones studied so far

This version, before returning the final array converts the data to its absolute value so that Real Data analysis can be performed.

Parameters

- **xx** – The collection of x-axis data where the soliton is based
- **A** – This real parameter represents the amplitude of the soliton.
- **B** – This is the squeezing factor. It is also inversely proportional to the FWHM parameter
- **O** – Related to the propagation chirp of the soliton wave-form
- **x0** – Displacement of the maximum in the x-axis base.
- **d** – Exponent factor related to the imaginary part of the power.

Returns This function return an *Numpy* array describing the Bright Soliton that fulfills the given parameters.

load_envelope (*filename*)

Function to load the file in ArrayFire format (*.af*) and convert it to a 1D Array in the NumPy format so it can be handled and plotted.

Parameters filename – The file name of the ArrayFire format file to load and convert to *Numpy* Array format.

load_parameters ()

Reads the file with parameters of the simulation and returns a set of data in the following order: :: Dimensions, Time Step, Spatial Step, Number of Points, Vector Limits

No arguments must be passed and a file *parameters.dat* in a defined protocol format has to exist with accurate mesh information

plot_sequence (*InPath, DoMovie*)

Function to plot the set of saved data with the matplotlib plot functions. This makes the output beautiful!!!

Statistical informations are saved to the folder named “Statistical”. Those informations are fundamental to perform analysis of the propagation.

Parameters

- **InPath** – String value indicating the relative path of all *ArrayFire* (*.af*) files containing the data computed by the GPU
- **DoMovie** – Simple Boolean variable indicating if the user pretends a final Movie to be rendered. (typical fps can be changed in the source code)