hurwitzKondoClass ReadMe

Robert Drost (robert.drost@aalto.fi)

Overview

The hurwitzKondoClass is a MatLAB object designed to fit supplied x and y data using the Hurwitz-Kondo model in https://arxiv.org/abs/2310.09326. The latest version was written and tested in MatLAB 2023b. To use the hurwitzKondoClass, simply save `hurwitzKondoClass.m` file on your computer and run your MatLAB client. The file must either be in the active MatLAB folder or on the MatLAB search.

Required Inputs

To initialise an instance of the hurwitzKondoClass, call the function as:

```
obj = hurwitzKondoClass(xData, yData, temperature)
```

The hurwitzKondoClass requires three input arguments in the following formats:

`xData` The x-axis data from your data set, supplied as a vector (1xN or Nx1 array). In a typical STM-experiment, this will be the bias voltage. Please note that xData and yData must be of the same size.
The object will guess the physical units of xData (V/mV/uV) from the exponent.

`yData` The y-axis data from your data set, supplied as a vector (1xN or Nx1 array). In a typical STM-experiment, this will be the lock-in amplifier signal.
Please note that xData and yData must be of the same size.

`temperature` The experimental temperature in K, supplied as a scalar.

To process larger data sets, use a loop to generate arrays of objects such as

```
for i=1:k
   obj(i) = hurwitzKondoClass(xData(i, :), yData(i, :), temperature(i));
end
```

Object properties

In addition to `xData`,`yData`, and `temperature`, the hurwitzKondoClass object contains the following parameters which can be read and modified after initialisation to control the object's behaviour:

`li` The lock-in oscillation amplitude in the same units as `xData`

`a` Scale factor used in fitting.

`Gamma` Encodes Kondo peak width in the fitting model.

`phi` Encodes Kondo peak asymmerty in the fitting model.

`bgSlope` Slope of the linear background included in the fitting mode.

`bgIntercept` Axis intercept of the linear background included in the fitting mode.

`biasOffset` X-axis offset to included in the fitting model (the Kondo peak is expected at zero bias).

`fit` Fit curve after the object fit function has been called.

`useLi` Include lock-in broadening in fitting model (true/false).

`model` Fitting model; specify 'hurwitz' or 'frota'.

`precision` Numerical precision for computing the Hurwitz Zeta function. Increasing this value will speed up the fitting procedure at the expense of accuracy.

`options` Optimisation options controlling the behaviuor of `lsqnonlin`. Refer to the MatLAB documentation for further details.

`speed` If true, the fitting function will attempt to guess initial parameters by downsampling data before doing a fit of the full data set. 'True' will generally be faster,
but can fail if the data varies strongly over the bias window.

Fitting and Plotting

The hurwitzKondoClass contains in-built functions to fit the $x-$ and $y-$data with the chosen model and display the result. To perform a fit, call the object fit function as:

```
obj = dofit(obj);
```

The object will fit the model specified by the `model` property to its data using the model properties `a`,`Gamma`, `phi`, `bgSlope`, `bgIntercept`, and `biasOffset`. Final parameters will be saved in the object properties `a`,`Gamma`, `phi`, `bgSlope`, `bgIntercept`, and `biasOffset`. The object plot function will plot the object `yData` over the object `xData` along with the latest fit stored in `obj.fit`.

Example

Using the data set provided in `example.txt`:

```
data = readmatrix('example.txt');
h = hurwitzKondoClass(data(:, 1), data(:, 3), 2);
h = dofit(h);
```

will return a hurwitzKondoClass object with the following properties:

```
        xData:  [400×1 double]
        yData:  [400×1 double]
            T:  2
           li:  []
            a:  0.7113
        Gamma:  2.8680
          phi:  0.0735
      bgSlope:  -0.0012
  bgIntercept:  0.3189
   biasOffset:  0.1716
          fit:  [400×1 double]
        useLi:  1
        model:  'hurwitz'
    precision:  1.0000e-06
      options:  [1×1
                optim.options.Lsqnonlin]
        units:  'mV'
        speed:  1
```

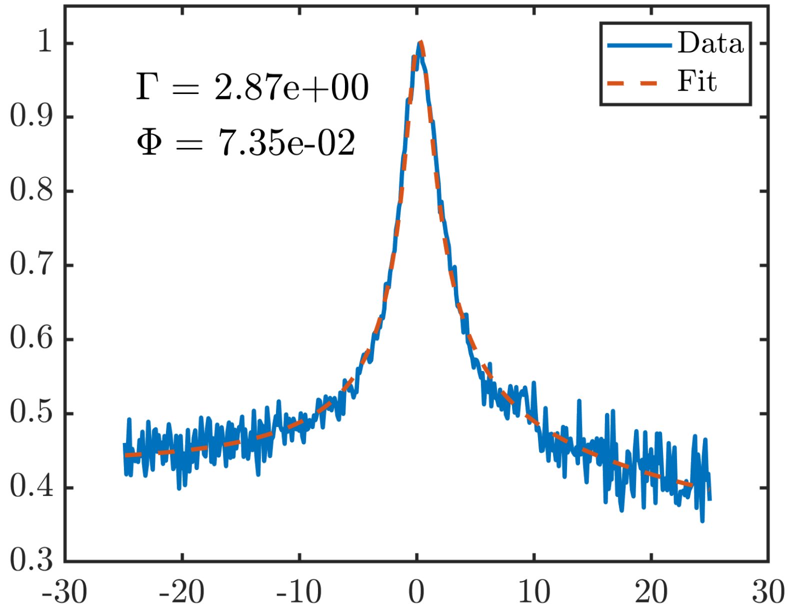

Figure 1: Plot of the fit result.