On the fragmentation in the open-source world

and on the challenges to create a healthy and sustainable project maintenance

Alexander Semke

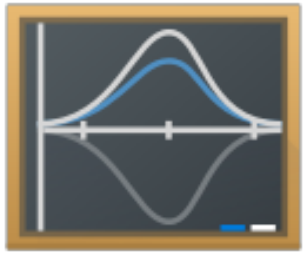
Challenges

- Open source movement - millions of volunteers/contributors
- Not true for every single open-source project

- Human factor is more critical for small projects
  - Maintainers and core developers are humans, too
  - Limited amount of time
  - Maintaining such a project becomes “work”, the amount of stress increases
  - Personal interests and motivations change over time
  - Very hard to safe-guard the reliability (maintenance, etc.) of the project in the long run
- People tend to underestimate the importance of the invest in the long-term maintenance

Fragmentation of resources, or lack of collaboration, is an issue!

LabPlot – KDE application for interactive graphing and analysis of scientific data



- GUI based approach
- Import of data in different formats (ASCII, binary, HDF5, FITS, JSON, etc.)
- Support for live-data (reading from local and network sockets, MQTT, serial port)
- 2D visualization
- Analysis functions (fitting, smoothing, FFT, interpolation, etc.)
- Basic statistical capabilities
- Support for different open-source computer algebra systems like Maxima, Octave, etc.
- Export of the results to different formats (SVG, PNG, etc.)

Fit tests [Changed] — labplot2

File Edit Spreadsheet Matrix Worksheet CAS Worksheet Analysis Datapicker Windows Tools Settings Help

Project Explorer

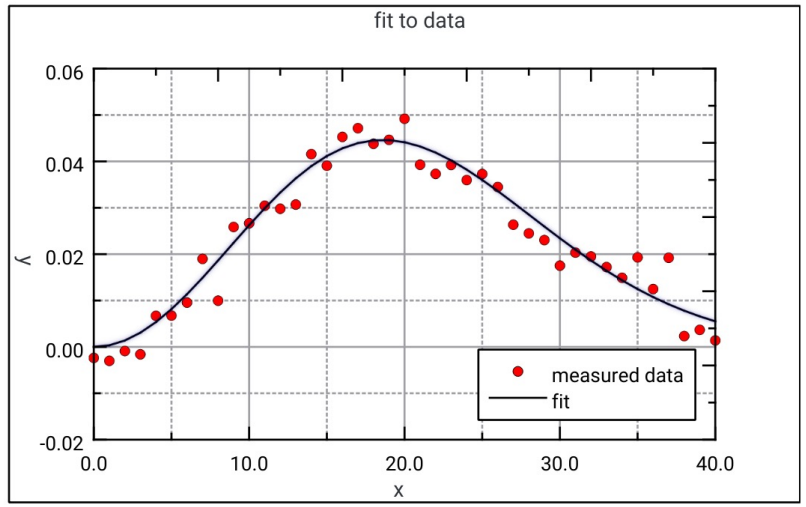
Search/Filter: Search/Filter text

Name	Type
Lorentz_data	Spreadsheet
Lorentz_plot	Worksheet
Maxwell-Boltzmann_data	Spreadsheet
x	Column
y_true	Column
y_meas	Column
Maxwell-Boltzmann_plot	Worksheet
plot with data and fit	CartesianPlot
x axis 1	Axis
x axis 2	Axis
y axis 1	Axis
y axis 2	Axis
measured data	XYCurve
fit	XYFitCurve
residuals	Column
legend	CartesianPlotL
plot with residuals	CartesianPlot
Sigmoid_data	Spreadsheet
Sigmoid_plot	Worksheet
Gompertz_data	Spreadsheet
Gompert_plot	Worksheet
Weibull_data	Spreadsheet
Weibull_plot	Worksheet
Gumbel_data	Spreadsheet

Maxwell-Boltzmann_data

	x {Numeric}	y_true {Numeric}	y_meas {Numeric}
1	0,000000e+00	0,000000e+00	-2,373721e-03
2	1,000000e+00	3,459171e-04	-3,011469e-03
3	2,000000e+00	1,371808e-03	-8,963710e-04

Maxwell-Boltzmann_plot



Fit

General Line Symbol Values Filling

Name: fit

Comment:

Data:

x-Data: x

y-Data: y_meas

Weights:

x-Range: Auto

Min 0,000000

Max 99,000000

Fit:

Model: Maxwell-Boltzmann

f(x): **sqrt**(2/**pi**)*x^2***exp**(-x^2/(2*a^2))/a^3

Parameters Options

Results:

Parameters:
a = 13.1565±2.0508

Goodness of fit:
sum of squared errors (χ²): 0.00214907
mean squared error: 2.14907e-05
root-mean squared error (reduced χ²): 0.0046358
mean absolute error: 0.363451

Recalculate

visible

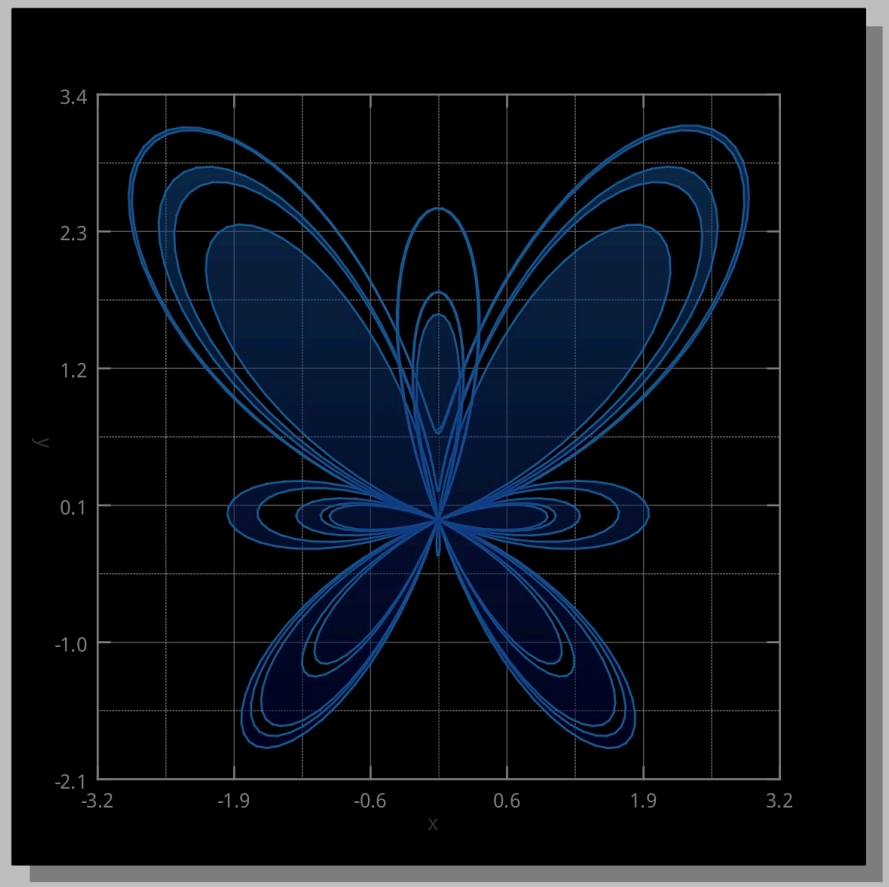
Mathematical functions - [Butterfly curve] — labplot2

File Edit Spreadsheet Matrix Worksheet CAS Worksheet Analysis Datapicker Windows Tools Settings Help

Project Explorer

Search/Filter: Search/Filter text

Name	Type
Mathematical functions	Project
Hilbert curve	Worksheet
data	Folder
Butterfly curve	Worksheet
xy-plot	CartesianPlot
x axis 1	Axis
x axis 2	Axis
y axis 1	Axis
y axis 2	Axis
f(x)	XYEquationCurve
Piriform curve	Worksheet
Eight curve	Worksheet
Lemniscate	Worksheet
Teardrop Curve	Worksheet
Rose Curve	Worksheet
Devil's Curve	Worksheet



xy-Equation

General Line Symbol Values Filling

Name	f(x)
Comment	http://mathworld.wolfram.com/ButterflyCurve.html
Equation type	parametric
x=f(t)	sin(t)*(exp(cos(t))-2*cos(4*t)-pow(sin(t/12), 5))
y=f(t)	cos(t)*(exp(cos(t))-2*cos(4*t)-pow(sin(t/12),5))
t, min	0
t, max	10*pi
Number of points	1000

Recalculate

visible

Project — labplot2

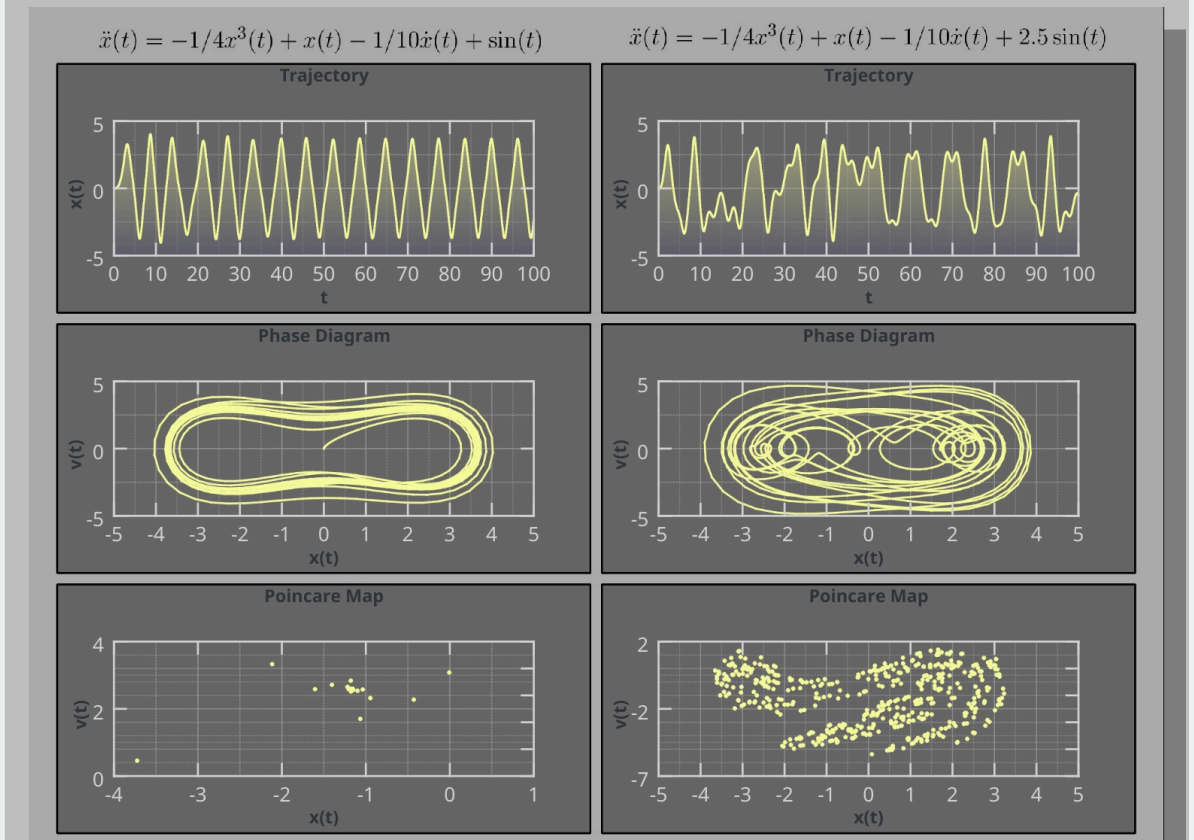
File Edit Spreadsheet Matrix Worksheet CAS Worksheet Analysis Datapicker Windows Tools Settings Help

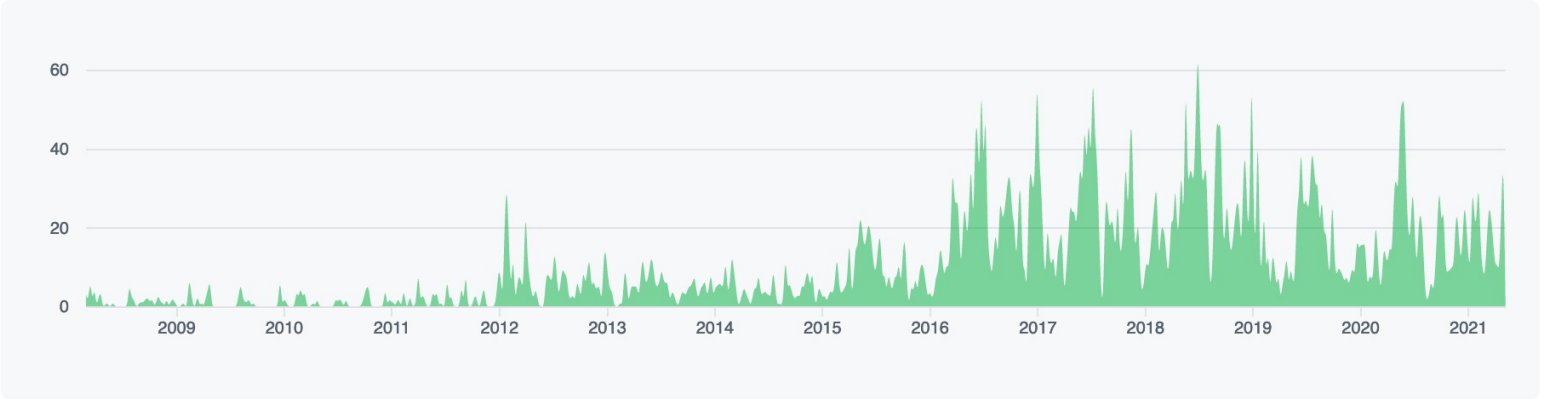
Restart Backend Evaluate Worksheet

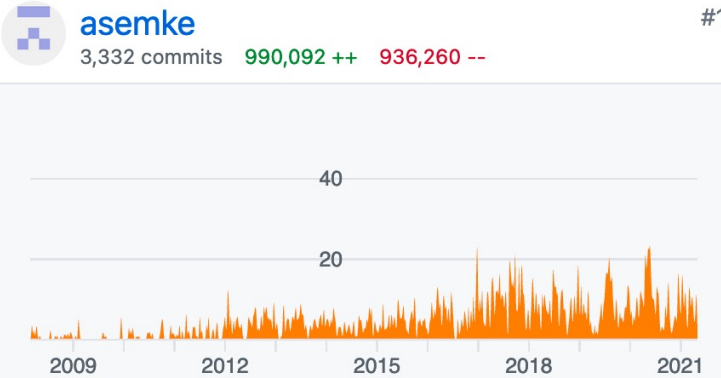
Maxima(Duffing)

```
>>> iter:500$
>>> %tau:bfloat(%pi)$
>>> /* first example with the driving force sin(t) */
>>> /* solve the differential equation with the Runge-Kuta method */
>>> duffing:[v,-v/10+x-x^3/4+sin(t)]$
>>> solution:rk(duffing,[x,v],[0,0],[t,0,iter/5,0.1])$
>>> /* extract data */
>>> t_data:map(lambda([x],x[1]),solution)$
>>> x_data:map(lambda([x],x[2]),solution)$
>>> v_data:map(lambda([x],x[3]),solution)$
>>> /* calculate poincare map */
>>> solution2:rk(duffing,[x,v],[0,0],[t,0,iter*2*%tau,%tau/30])$
>>> poincare_list:create_list(solution2[i], i, makelist(i*60,i,1,iter))$
>>> poincare_data:makelist(poincare_list[i],i,1,iter)$
>>> /* extract data */
>>> poincare_x_data:map(lambda([x],x[2]),poincare_data)$
>>> poincare_v_data:map(lambda([x],x[3]),poincare_data)$
>>>
>>> /* ########## second example with the driving force 2.5*sin(2*t) ##########
   */
>>>
>>> duffing2:[v,-v/10+x-x^3/4+2.5*sin(2*t)]$
>>> solution3:rk(duffing2,[x,v],[0,0],[t,0,iter/10,0.1])$
>>> /* extract data */
>>> t_data_2:map(lambda([x],x[1]),solution3)$
>>> x_data_2:map(lambda([x],x[2]),solution3)$
>>> v_data_2:map(lambda([x],x[3]),solution3)$
>>> /* calculate the Poincare map */
>>> solution4:rk(duffing2,[x,v],[0,0],[t,0,iter*%tau,%tau/30])$
>>> poincare_list_2:create_list(solution4[i], i, makelist(i*30,i,1,iter))$
>>> poincare_data_2:makelist(poincare_list_2[i],i,1,iter)$
>>> /* extract data */
>>> poincare_x_data_2:map(lambda([x],x[2]),poincare_data_2)$
```

Worksheet 1



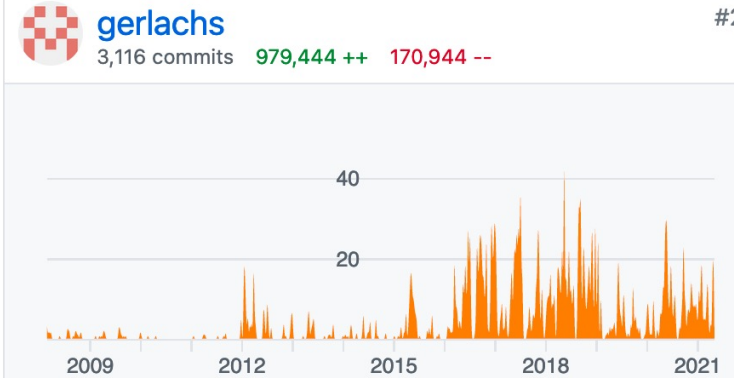




asemke

#1

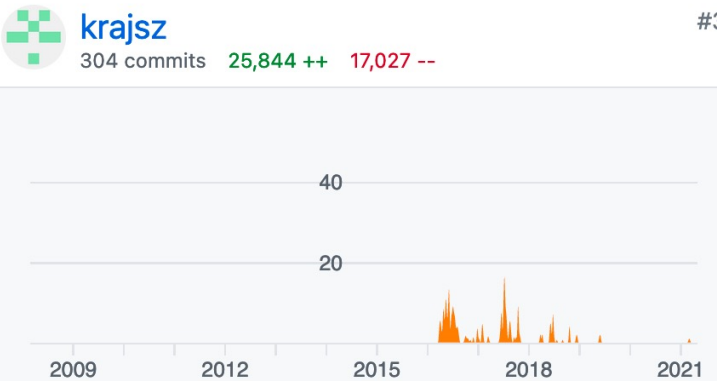
3,332 commits 990,092 ++ 936,260 --



gerlachs

#2

3,116 commits 979,444 ++ 170,944 --



krajsz

#3

304 commits 25,844 ++ 17,027 --

kovacsferencz98

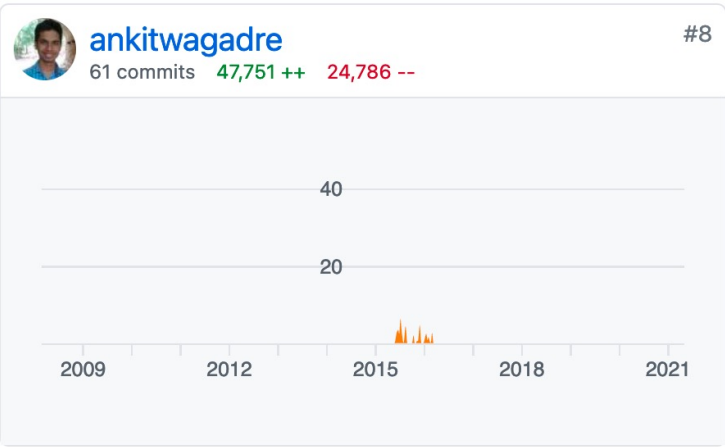
#4

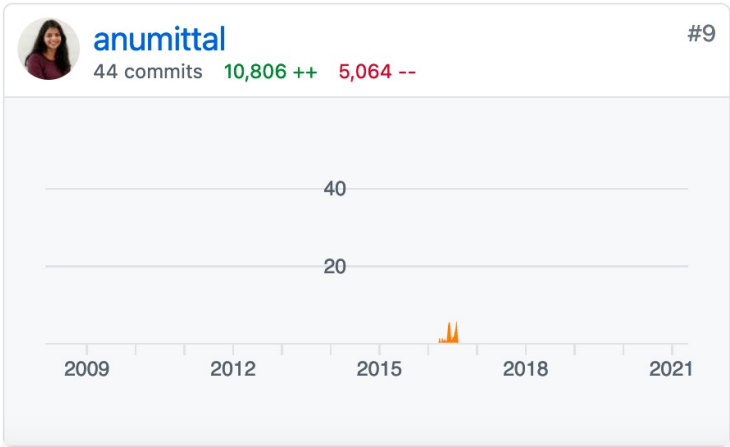
215 commits 79,710 ++ 31,477 --


Murmele
92 commits 11,558 ++ 5,625 --
#5
40
20
2009
2012
2015
2018
2021

garvitdelhi
86 commits 30,022 ++ 25,178 --
#6
40
20
2009
2012
2015
2018
2021


yurchor
83 commits 2,606 ++ 855 --
#7
40
20
2009
2012
2015
2018
2021

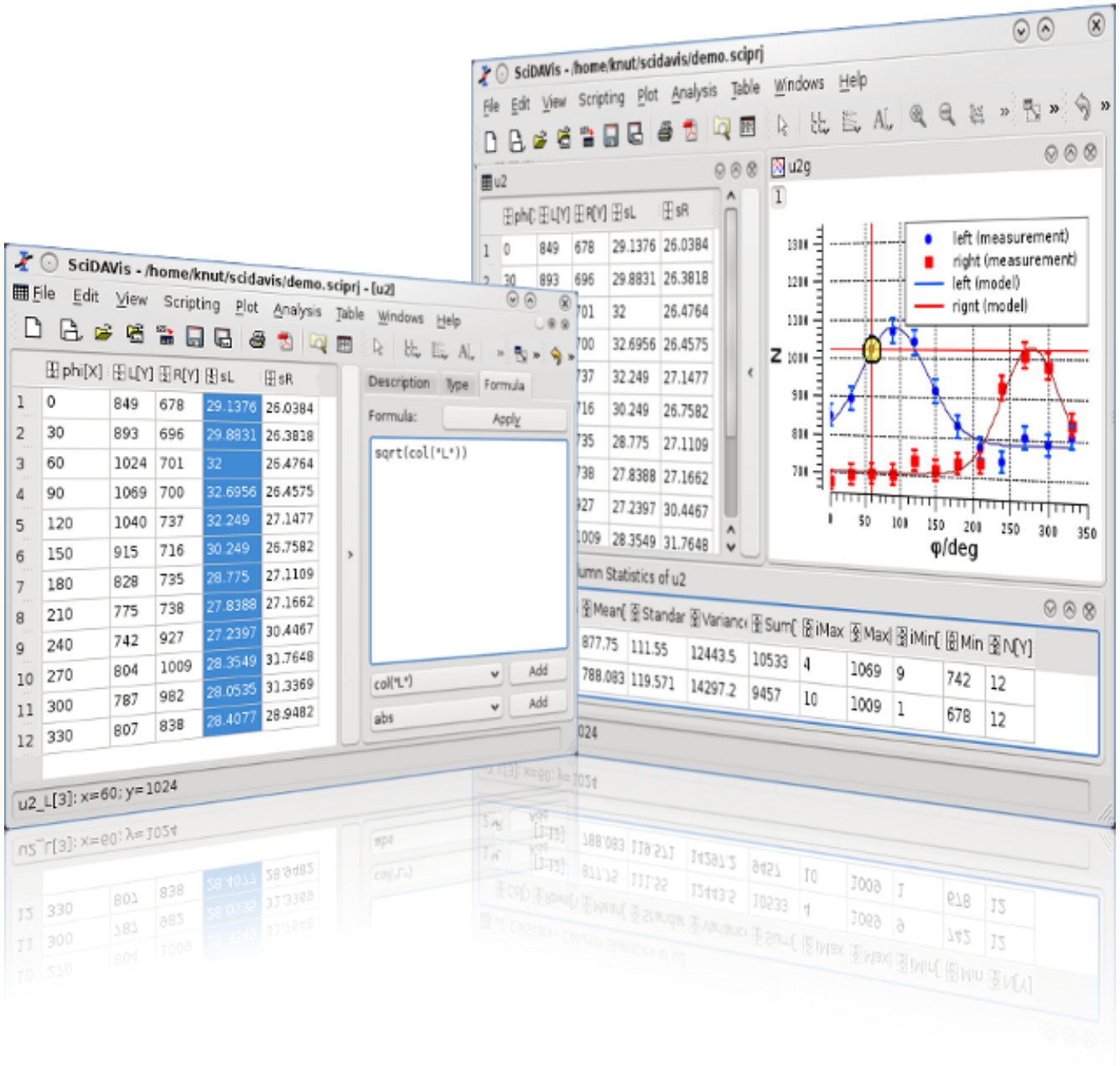

ankitwagadre
61 commits 47,751 ++ 24,786 --
#8
40
20
2009
2012
2015
2018
2021


anumittal
44 commits 10,806 ++ 5,064 --
#9
40
20
2009
2012
2015
2018
2021

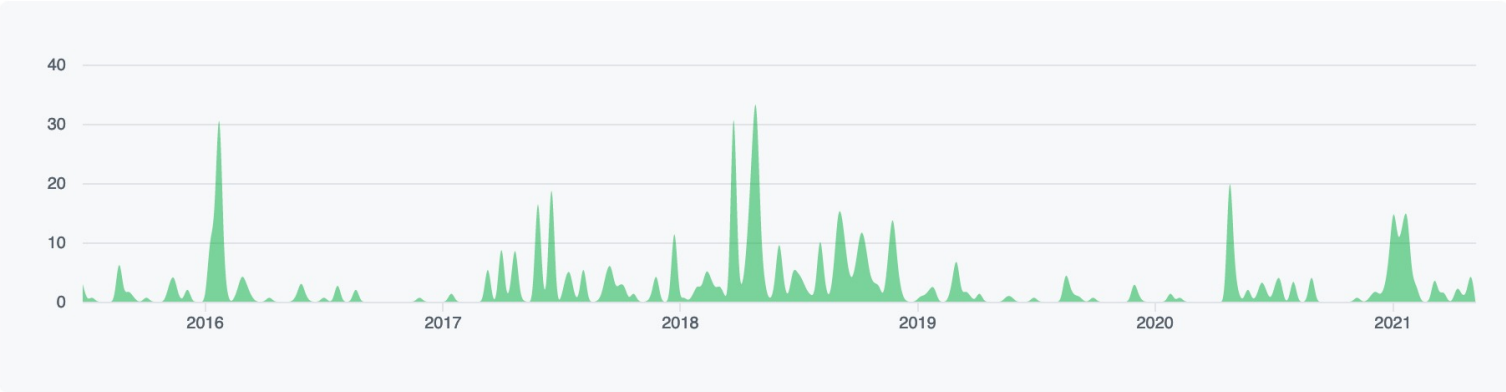

croick
39 commits 5,688 ++ 3,072 --
#10
40
20
2009
2012
2015
2018
2021

Similar Projects

SciDAVis is a free application for *Sci*entific *D*ata *A*nalysis and *Vis*ualization.

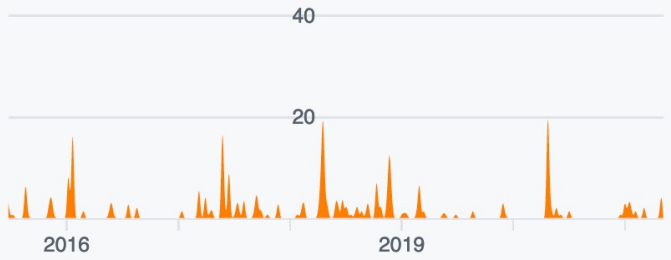


http://scidavis.sourceforge.net/



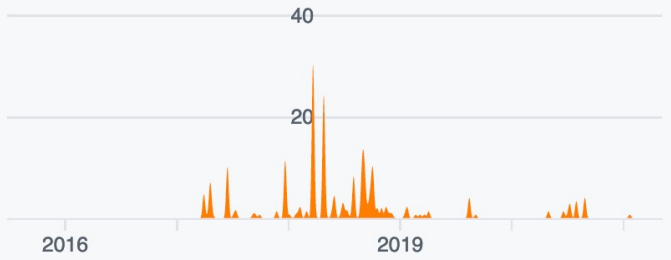
highperformancecoder #1

366 commits 710,736 ++ 647,200 --



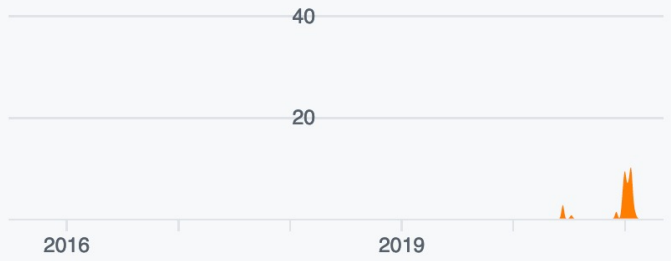
gbm19 #2

282 commits 10,807 ++ 14,037 --



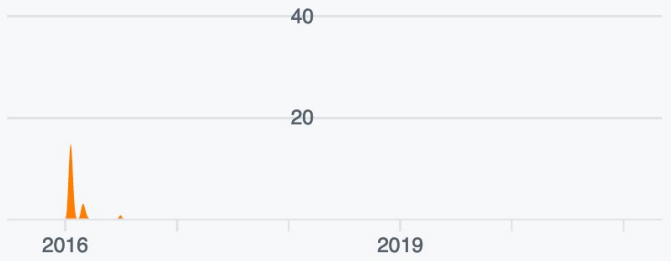
MehdiChinoune #3

54 commits 66,290 ++ 75,047 --

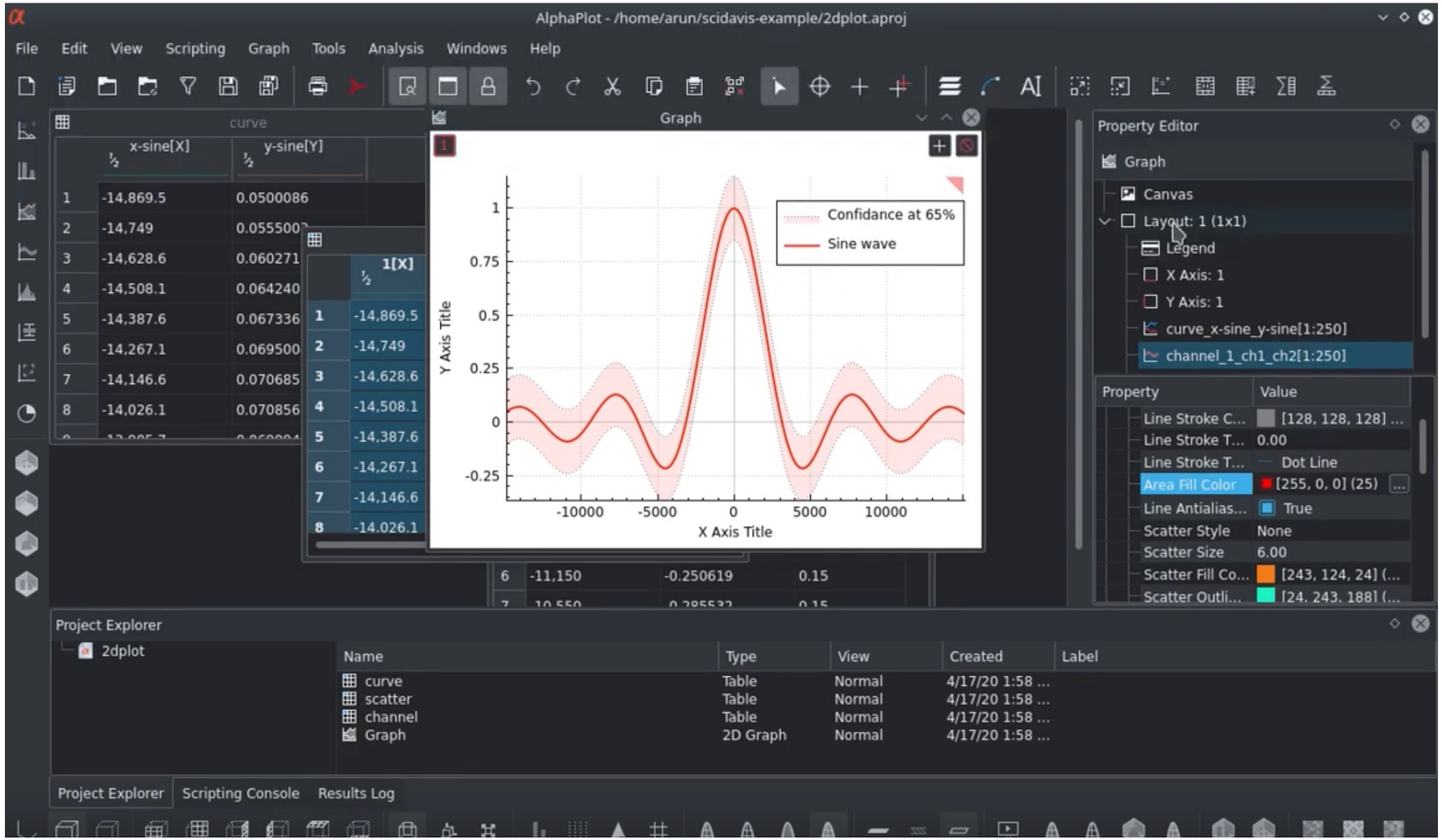


narunlifescience #4

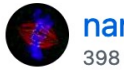
43 commits 3,883 ++ 427 --



AlphaPlot is an open-source computer program for interactive scientific graphing and data analysis. It can generate different types of 2D and 3D plots (such as line, scatter, bar, pie, and surface plots) from data that is either imported from ASCII files, entered by hand, or using formulas.

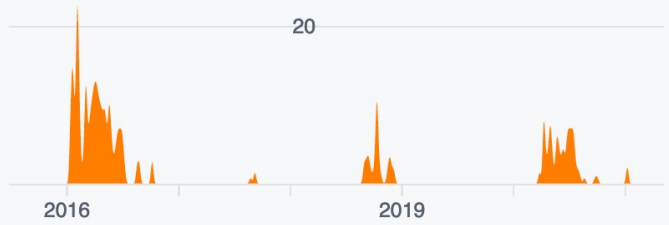


alphaplot.sourceforge.net



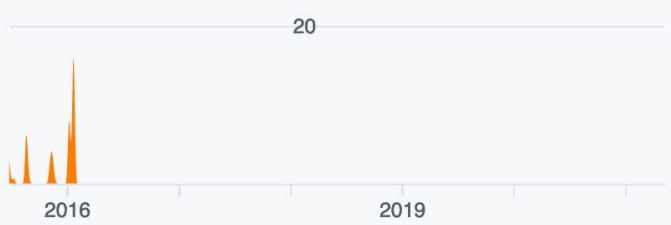
narunlifescience #1

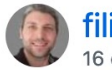
398 commits 1,252,495 ++ 1,212,317 --



highperformancecoder #2

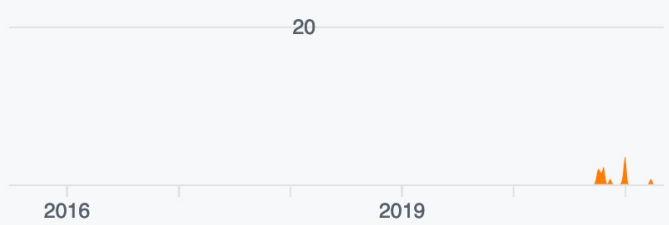
59 commits 32,522 ++ 34,059 --





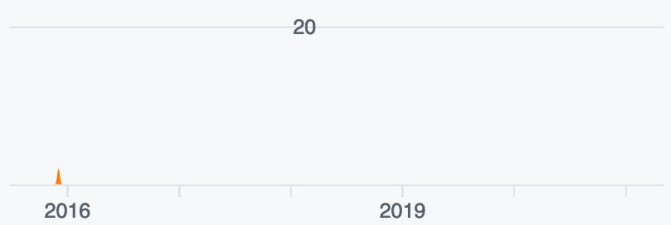
filipestevao #3

16 commits 305 ++ 84 --

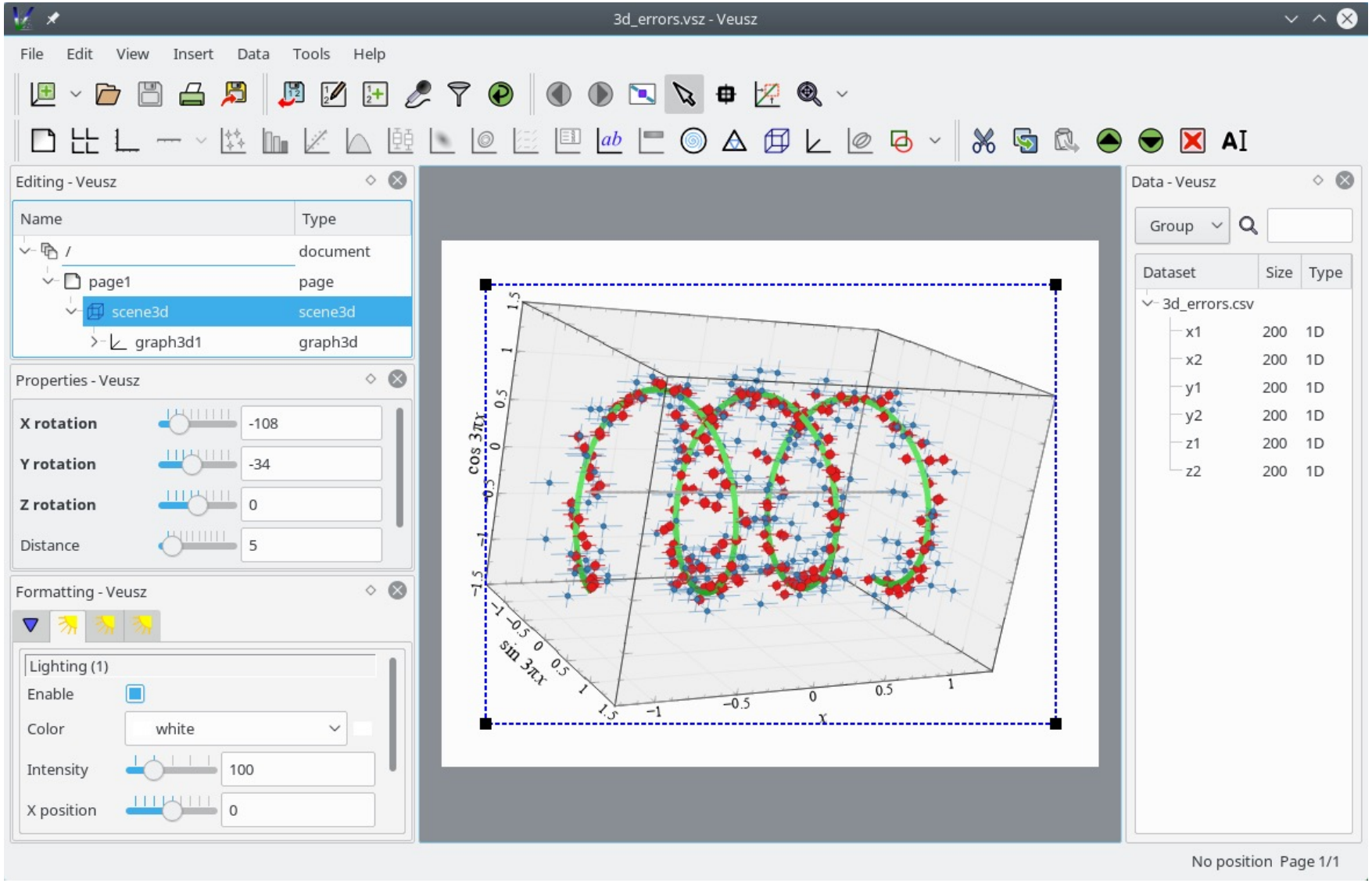


Fellypao #4

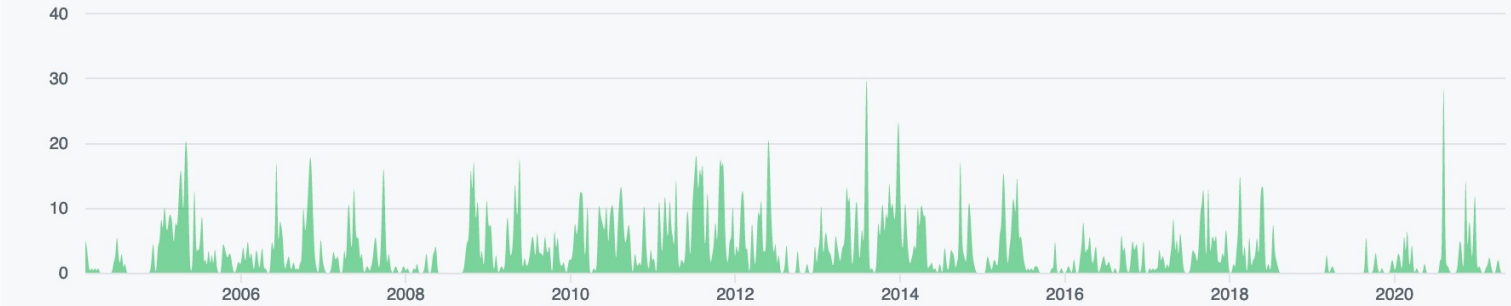
3 commits 2,170 ++ 2,168 --



Veusz is a scientific plotting and graphing program with a graphical user interface, designed to produce publication-ready 2D and 3D plots. In addition it can be used as a module in Python for plotting

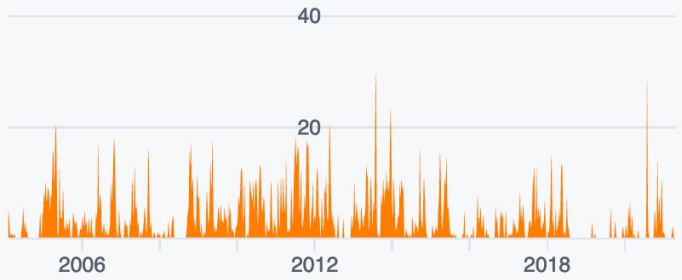


https://veusz.github.io/



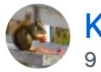
jeremysanders #1

3,190 commits 516,066 ++ 348,525 --



bks #2

16 commits 267 ++ 91 --

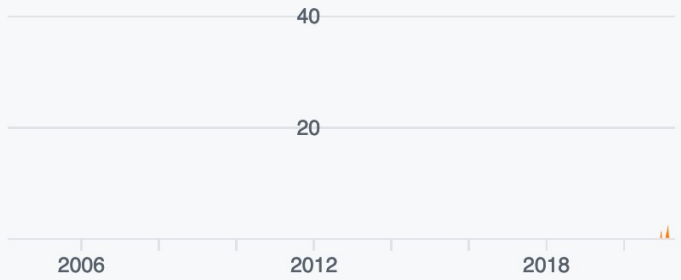


KDB2 #3

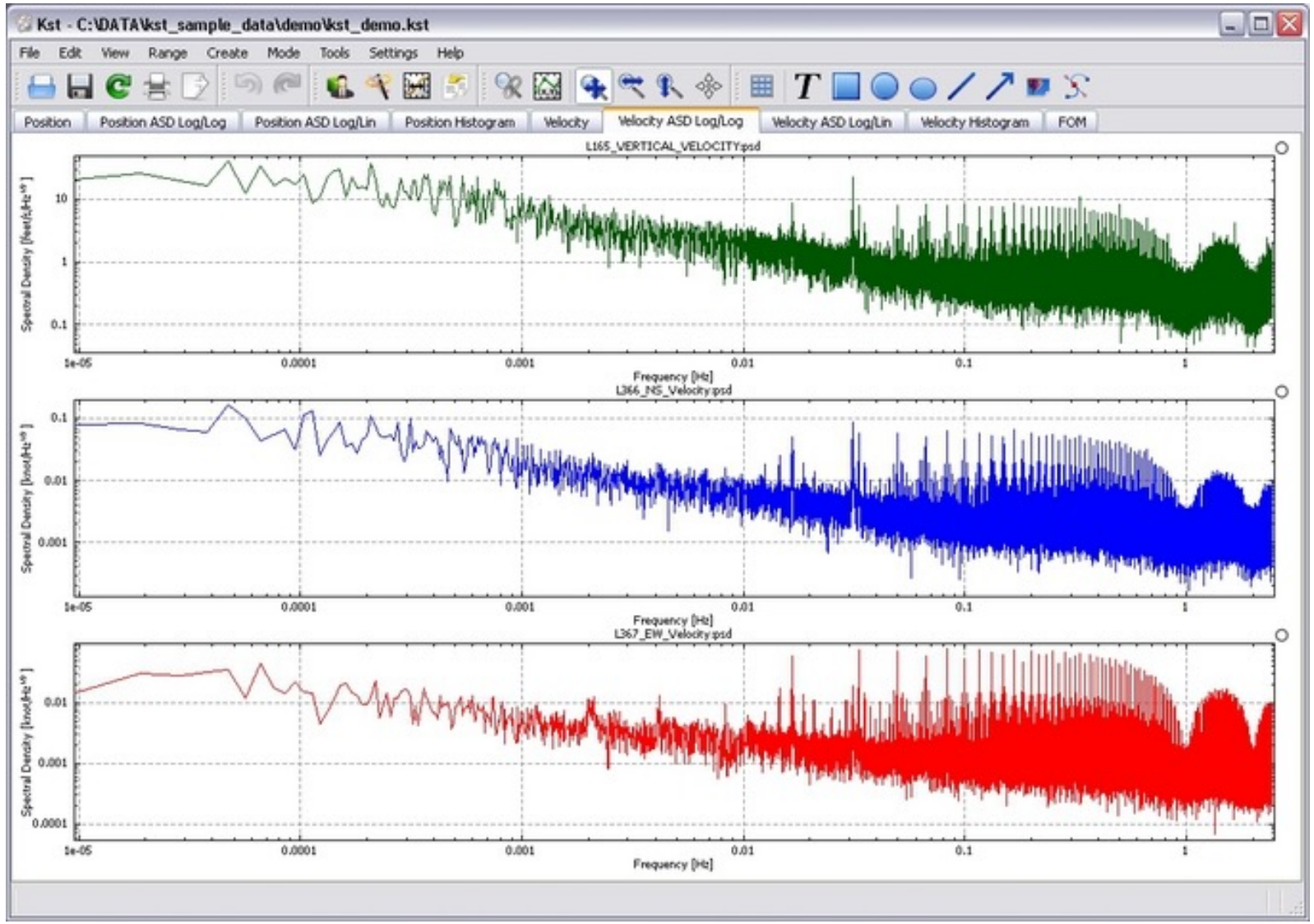
9 commits 20 ++ 8 --

korintje #4

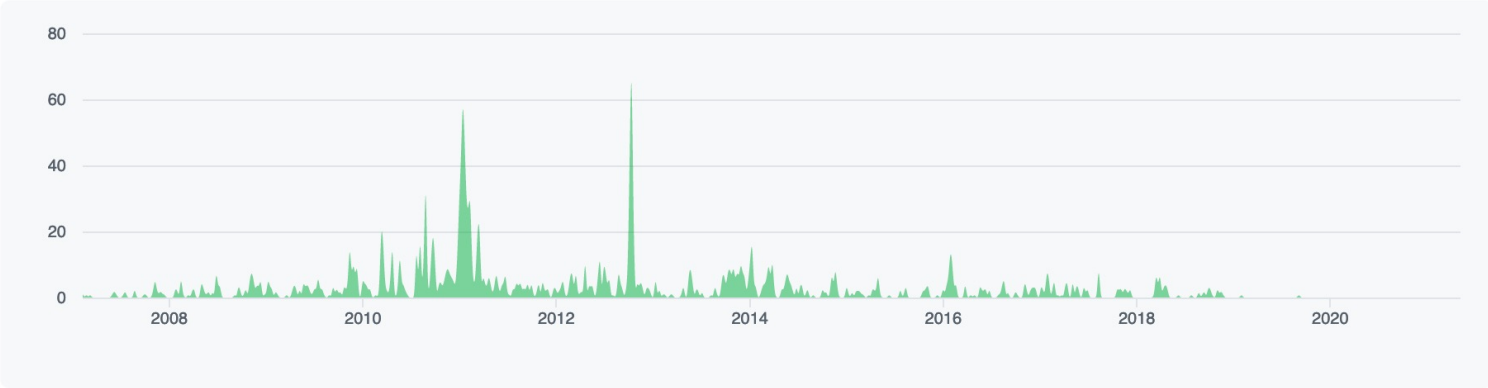
8 commits 181 ++ 49 --

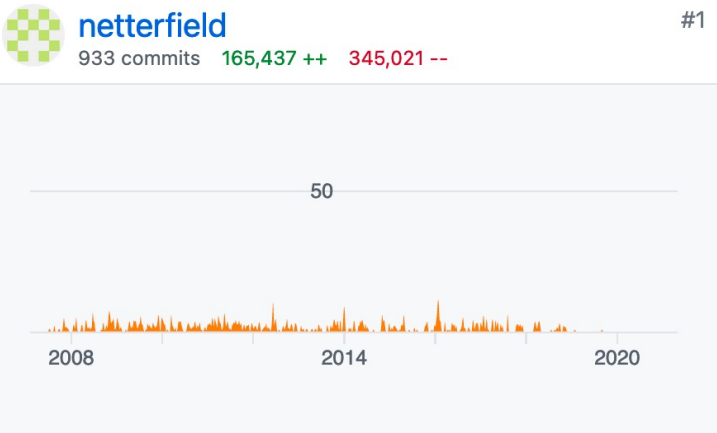


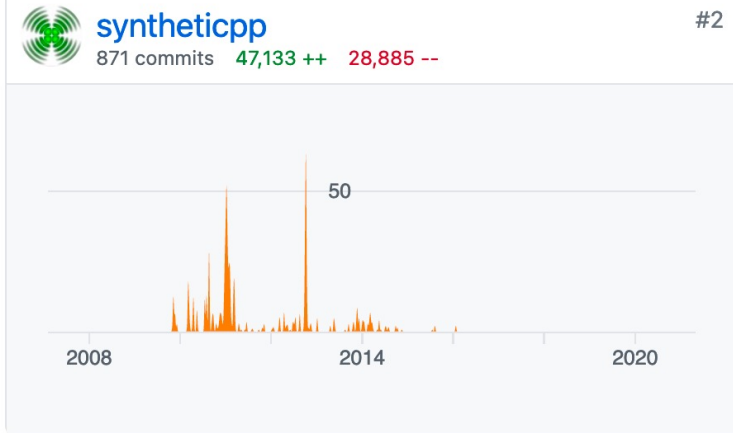
Kst is the fastest real-time large-dataset viewing and plotting tool available and has built-in data analysis functionality. Kst contains many powerful built-in features and is expandable with plugins and extensions.



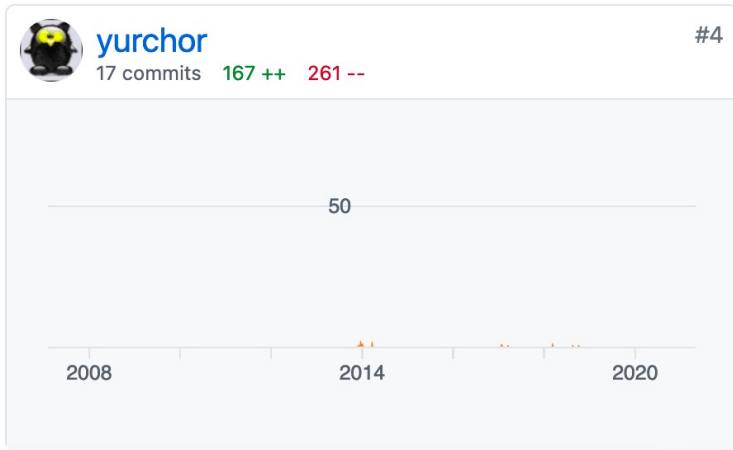
https://kst-plot.kde.org/


80
60
40
20
0
2008
2010
2012
2014
2016
2018
2020


netterfield
#1
933 commits 165,437 ++ 345,021 --
50
2008
2014
2020


syntheticpp
#2
871 commits 47,133 ++ 28,885 --
50
2008
2014
2020


brisset
#3
145 commits 18,056 ++ 7,079 --
50
2008
2014
2020


yurchor
#4
17 commits 167 ++ 261 --
50
2008
2014
2020

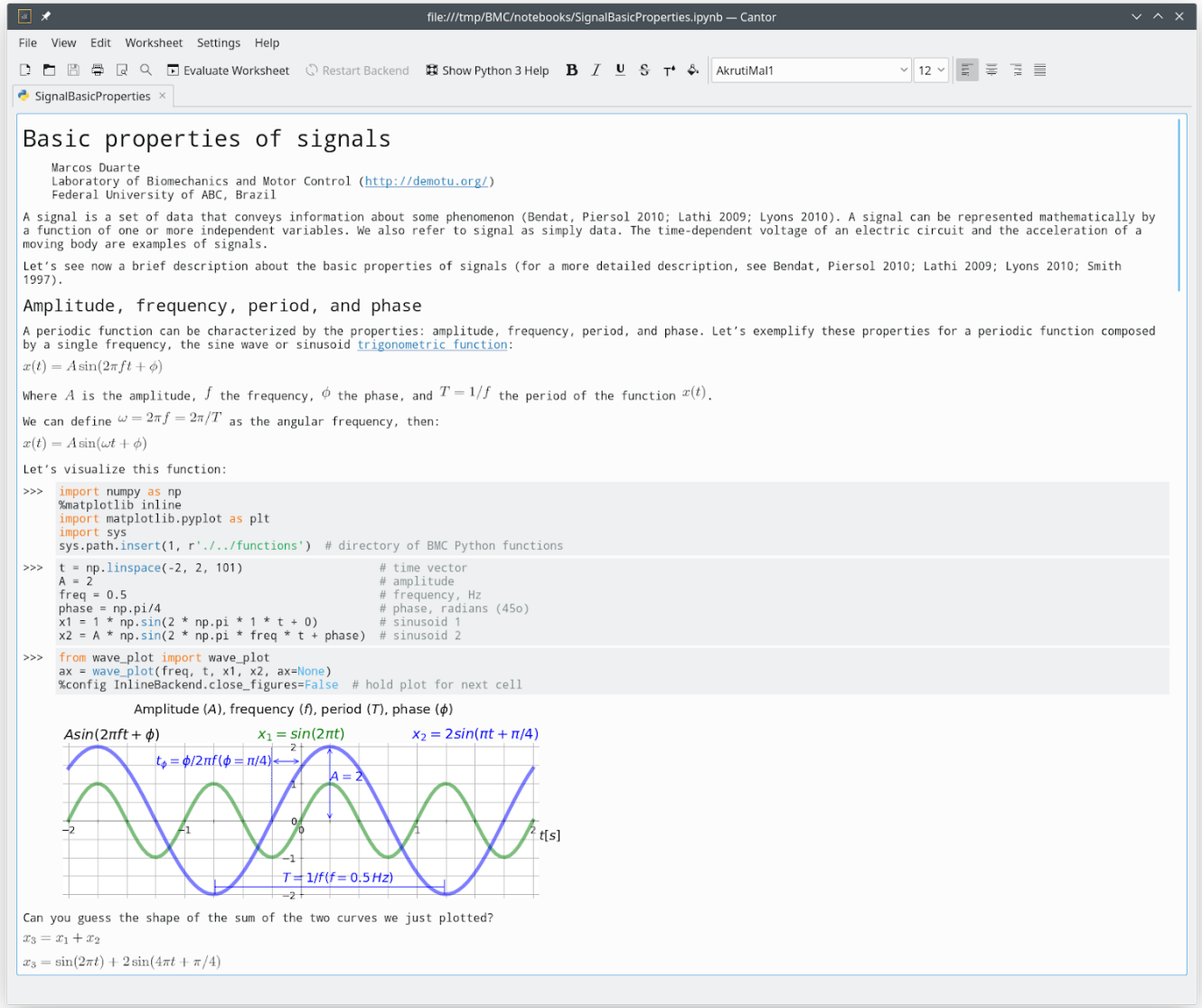
Summary:

- Development resources strongly fragmented, mostly one- to two-men projects
- Different development paces for different projects
- Active and healthy development right now for LabPlot, won't go on "forever"
- Still, the resources are limited which is blocking a bigger and faster development

How to change the current situation?

- Look for synergies and collaborations with other projects
- Intensify activities within the organization (packaging, stores, infrastructure for CI, documentation, etc.)
- Increase the frequency of releases
- Invest more into the promotional work to attract new contributors

Cantor - KDE Frontend to mathematical applications



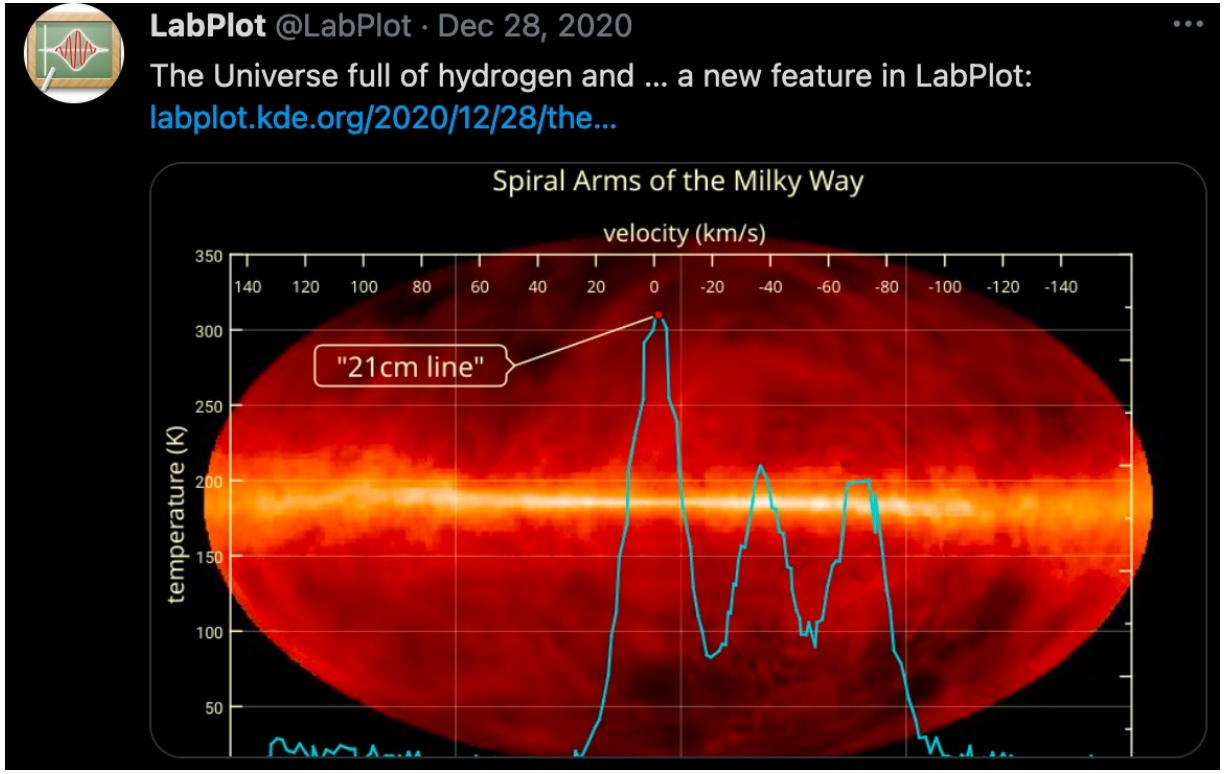
https://cantor.kde.org/

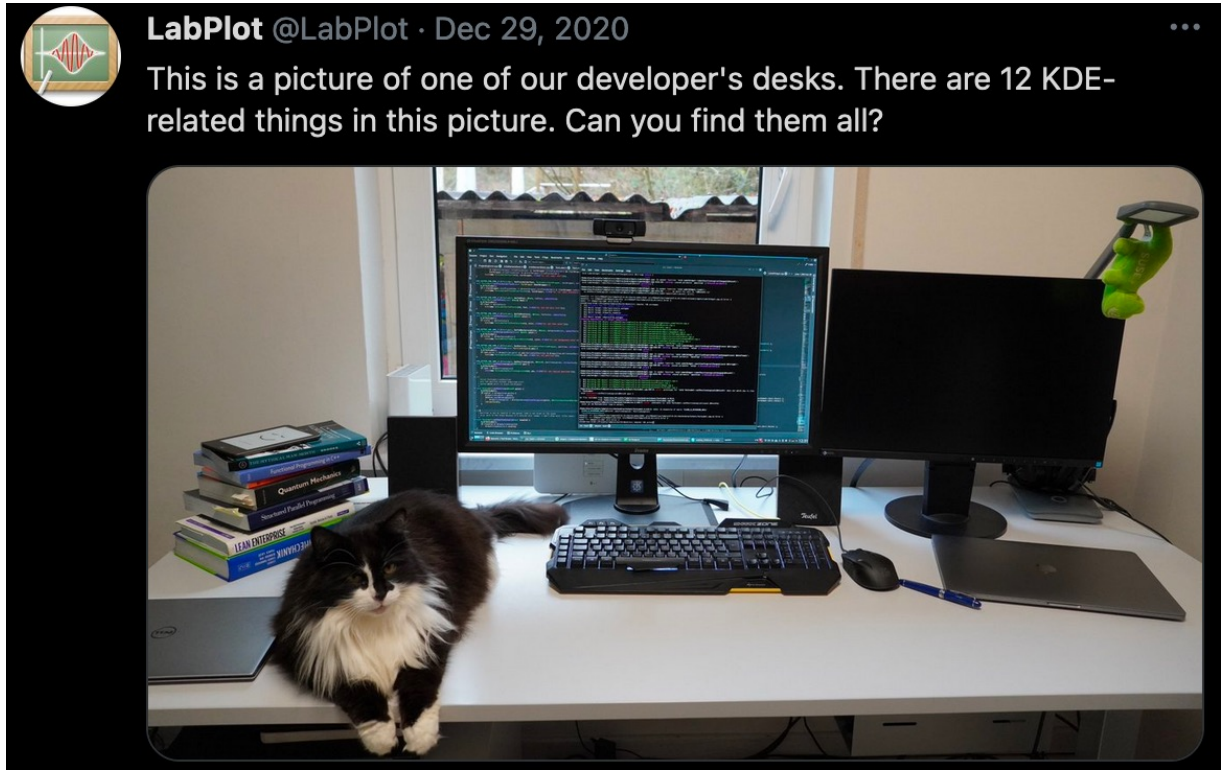
Change in the mindset
-> invest more into promotion

- Invest more into documentation, tutorials, example projects
- Make the usage of the application and the contribution to it easier
- Attract new contributors
- Convert users into contributors
- Make yourself redundant

All this requires even **more work and effort** for maintainers, at the least at the beginning

Do the promotion properly!





The cat brought 4 time more attention on Twitter than the distribution of hydrogen in our galaxy.

Thank You!

(https://labplot.kde.org/support/)