PyDigitizer – A Python module for digitizing 2D curves

Dorian NEDELCU[1], Tihomir LATINOVIC[2]

[1]"Babes-Bolyai" University of Cluj-Napoca, Romania, Faculty of Engineering, Department of Engineering Sciences, Traian Vuia square, no.1-4, 320085, Resita.
[2]University of Banja Luka, Bosnia and Herzegovina, Faculty of Mechanical Engineering, Department of Mechatronic and Robotics, Stepe Stepanovica 71, 78000, Banja Luka.

E-mail: dorian.nedelcu@ubbcluj.ro

Abstract. The digitizing process consists of extracting a curve numerical coordinates from an explicit image. The paper describes the PyDigitizer module for digitizing 2D curves from images, created with the help of free and Open Source resources, using Python as a programming language and wx.Python as a graphical user interface toolkit. Digitizing is done manually, only for XY graphs type, by placing mouse points with left clicking over the curves in the image. After digitization, spline interpolation curves can be plotted over the image curves, their intersection points can be accurately calculated for specified X or Y values and the coordinates of the digitized points can be displayed in tabular form; moreover, regression digitized curves can be generated, the results can be exported to an Excel file or saved in a dataset to be restored later. It is also possible to open Excel files, with point coordinates of one or more curves, to operate intersections for the X or Y axis.

1. INTRODUCTION

There are several applications on the market for digitizing 2D curves [1], [2], [3] from images.
WebPlotDigitizer software [1] can be operated on-line or on desktop computer, works with a wide variety of charts (XY, bar, polar, ternary, map, etc.), is free to use, open source and cross-platform (web and desktop) and use manual or automatic extraction algorithms to extract data points.
Online PlotDigitizer software [2] can be operated on-line to extract numerical data from images, supports XY graphs, bar graphs (horizontal & vertical), polar diagrams, ternary plots, pie charts, maps, use auto-tracing algorithms like cluster, points, curves, bar, histogram.
Graphreader software [3] can be operated on-line to extract values from image based graphs and plots, and export those data into comma separated values (CSV).
Why create a new package? The answer can be synthesized by the following considerations:

- a challenge to create a 2D digitizer module using only Open Source and free resources;
- to obtain an engineering and practical tool to generate 2D curves from images;
- to intersect 2D curves with X / Y values and obtain numerical intersection coordinates;
- to calculate regression of digitized curves with polynomial functions;
- to have a module which can be extended with the future necessities or included in other future software.

2. THE PyDigitizer MAIN WINDOW

The PyDigitizer main window contains the following areas.

- the application title bar – contains the application name "*PyDigitizer*" and the image file name;
- the application toolbar – contains icons for the main functions of the application, Fig. 1;
- the "*Digitizer Window"* area – where the image will be loaded and points can be placed over the curves; PyDigitizer supports multiple image formats, including JPG/JPEG, PNG, BMP, TIF/TIFF.
- the application status bar – reserved for the application's messages.

Figure 1. The PyDigitizer toolbar

3. THE PyDigitizer INTERFACE

The PyDigitizer toolbar from Fig. 1 is detailed in Table 1.

Table 1 The PyDigitizer toolbar icons

No.	Toolbar icon	Icon name	Description
1		**Open**	Open the source image file
2		**Load**	Load and restore chart data saved in a dataset
3		**Calibrate**	Axis calibration
4		**Curve**	Manage 2D curves (name, sort type, colours)
5	1-Xdir-Red		List control with names of all curves
6		**View**	View the graph of the current curve selected from the list
7		**Regression**	Polynomial regression of a digitized curve
8		**Trace**	Trace all curves marked with points
9		**Erase**	Erase all traced curves
10		**Intersect**	Chart intersection with X/Y constant values
11		**Table**	Tabulates the coordinates of the points marked on the curves and copies them to the clipboard
12		**Excel**	Export in Excel the source image and the coordinates of the digitized points
13		**Save**	Save chart data in a dataset
14		**Help**	Access the PyDigitizer help
15		**About**	Show information about the PyDigitizer software
16		**Exit**	Exit the application

4. THE PyDigitizer OPERATING PROCEDURE

The automatic extraction of a curve points from an image is always affected by small errors, which must be corrected manually. That's why PyDigitizer uses only manual digitizing of curves by placing multiple points over each curve with a left mouse click. The number of points is not limited, but these points must follow the shape of the curve. For each curve a sort order can be defined (in Xdir or Ydir direction respectively without sorting). Sorting in X or Y direction is done automatically by the application, so points can be specified in any order or new points can be added later. For the unsorted option, the points must be specified in order without the possibility of adding additional points. For example, in Fig. 2 for curve 1 Xdir sorting must be selected, for curve 2 sorting on Ydir and for curve 3 no sorting must be specified. Operating the PyDigitizer application involves the following steps:

- Open the PyDigitizer application.
- Click on **Open** icon to load an image into "*Digitizer Window*" or click on **Load** icon to load a dataset previous saved by **Save icon.**
    - The **Open** option will display "*Select image file*" window to open an image file with curves to be digitized. The selected image file will be display into "*Digitizer Window*" and "*Axis calibration*" window, Fig. 3, will be activated to specify the coordinates (X1,Y1) and (X2,Y2) for P1 and P2 axis calibration points. The calibration points will be saved by "*Save calibration points*" button. Next, the P1 and P2 points must be specified in "*Digitizer Window*" by two left click mouse, Fig. 4. The P1 and P2 points can always be changed through through **Calibrate** toolbar icon. The "*Manage 2D curve names*" window will be activated to define the names of future curves to be digitized, Fig. 5; the **New** button create a new curve name with the following fields: **Curve name**, **Sort type** and **Color**; with **Save** button the name will be loaded into "*List of curve names*" list. The **Exit** button will close the "*Manage 2D curve names*" window and the curve names will be loaded in the list from the "*Digitizer Window*" toolbar. New curve names can be added through **Curves** toolbar icon.
    - The **Load** option will display "*Select dataset file*" window to open the data file containing the name and the image itself, the calibration points, the name list of the curves and the points digitized in a previous session. All this data will be restored in "*Digitizer Window*" and the digitizing process can continue.
- Select a name curve from the "*Digitizer Window*" toolbar list (see row 5 from Table 1). The points that will be extracted will belong to the selected curve from this list.
- Place points along the curve over the image in the "*Digitizer Window*" window, Fig. 6. The points will be created with left mouse click; move points by right mouse click; the first click triggers the operation followed by moving the mouse and the second click finalizes the position of selected point; a click with central mouse button will delete the closest point to the mouse cursor. This information's will be displayed on the left region of the main windows status bar. As you move the cursor over the image, the right side of the status bar will display the X, Y real coordinates of the current point. The selected current curve can be displayed graphically in a separate window through the **View** toolbar icon if at least 3 points have been digitized, Fig. 7.
- All digitized curves can be displayed over those in the source image through the **Trace** toolbar icon if at least three points have been specified, Fig. 8. These curves can be removed through the **Erase** toolbar icon, without deleting the associated digitized points.
- The table of digitized point coordinates can be view through the **Table** toolbar icon, Fig. 9. This table is also copied to the Windows clipboard from where it can be used in other applications.
- The **Excel** toolbar icon will export table of digitized point coordinates to Excel application. Also the original image, the digitized points, the intersected points generated through **Intersect** toolbar icon and the curves generated through **Trace** toolbar icon will be exported to Excel file.

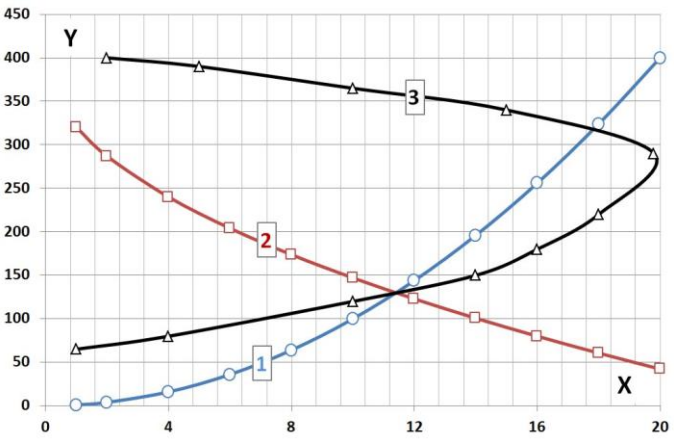

Figure 2. The demo image

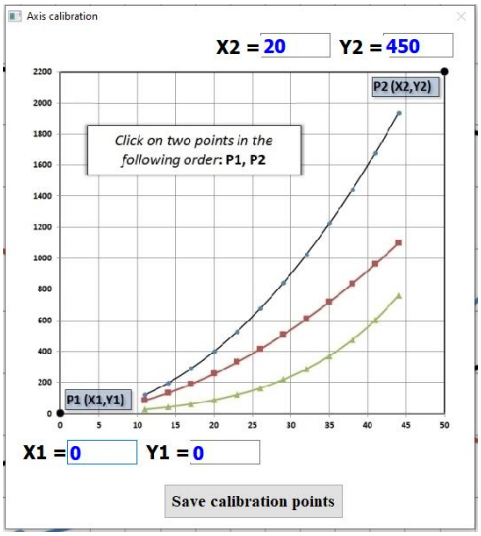

Figure 3. The **"Axis calibration"** window

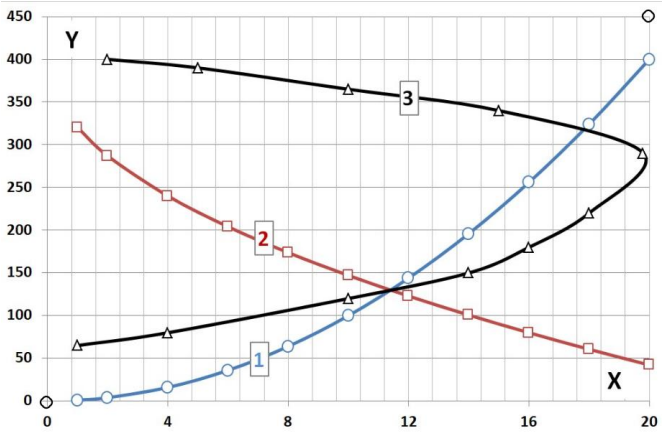

Figure 4. The P1 & P2 calibration points placed over image

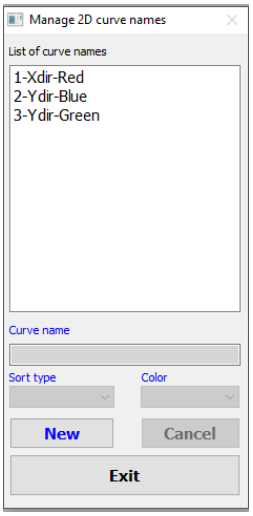

Figure 5. The "*Manage 2D curve names*" window

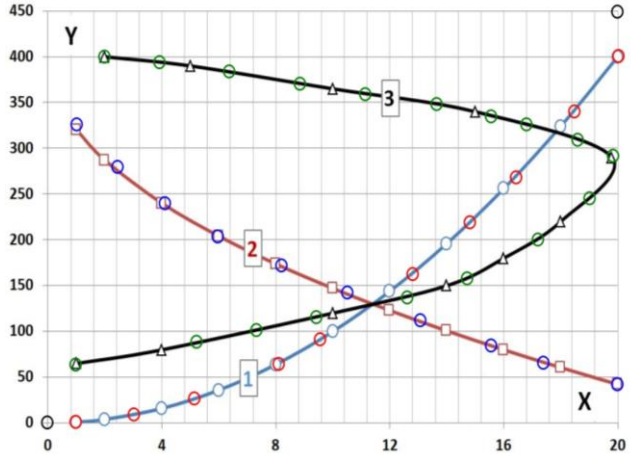

Figure 6. The digitized points placed over image

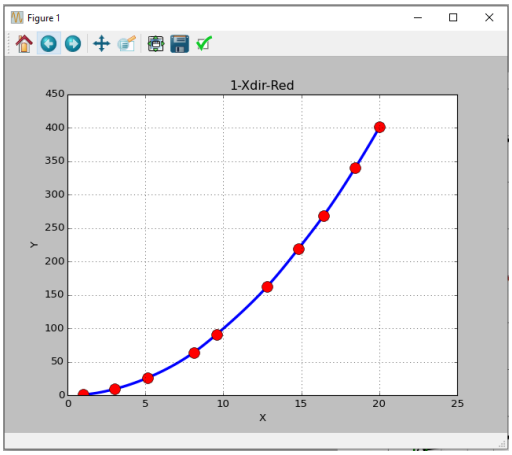

Figure 7. The chart of curve no. 1

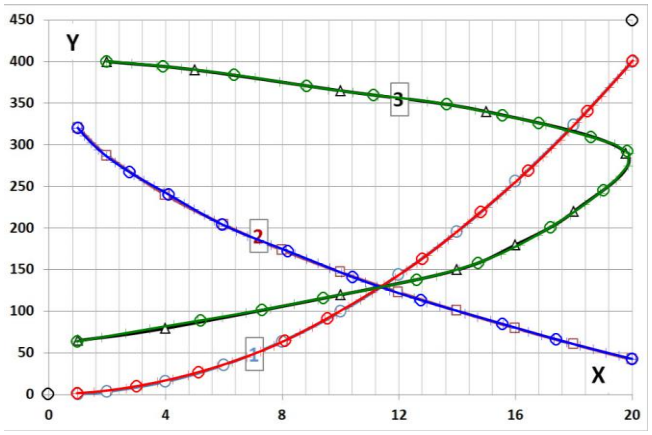

Figure 8. The digitized curves traced over image

Values copied to Windows clipboard

ID curve	X	Y
1-Xdir-Red	0.993524837449	0.884012776915
1-Xdir-Red	3.02311075305	9.3102039068
1-Xdir-Red	5.14820659409	26.1625861666
1-Xdir-Red	8.10901428273	64.0804462511
1-Xdir-Red	9.56554064569	91.0442578667
1-Xdir-Red	12.8128781106	162.666882471
1-Xdir-Red	14.8185865449	219.122363041
1-Xdir-Red	16.4422552774	268.836890707
1-Xdir-Red	18.471841193	340.459515311
1-Xdir-Red	20.0238774814	401.128091447
2-Ydir-Blue	1.01740231881	320.2366566
2-Ydir-Blue	2.78433593945	267.151652481
2-Ydir-Blue	4.12147489561	240.187840866
2-Ydir-Blue	5.96004096033	203.955219007
2-Ydir-Blue	8.20452420817	171.935692714
2-Ydir-Blue	10.4251299746	140.758785533
2-Ydir-Blue	12.7651231479	112.952354804
2-Ydir-Blue	15.558788467	84.3033049628
2-Ydir-Blue	17.3973545318	65.7656844771
2-Ydir-Blue	20.0	42.1723493134

Figure 9. The table of digitized point coordinates

5. THE INTERSECT COMMAND

The **Intersect** window is activated through the **Intersect** toolbar icon, Fig. 10. That window is designated to generate the intersection points of the digitized curves with X/Y constant values. To intersect the chart with X or Y values, the option must be selected from the left control list and an input numerical value is required in the associated text control; finally, the **Intersect** button will calculate the intersection points of the chart with the imposed values. The **Copy** button will send the intersection points coordinates to Windows clipboard. The points resulted from the intersections are marked with black colour and hexagon shape in the "*Digitizer Window*" window. The **Excel** button will send the intersection points coordinates to Microsoft Excel. The **Erase** button will erase all the previous intersection points from the "*Digitizer Window*" window and coordinate's table. The coordinate's table shows the intersected curve's name and the X/Y coordinates of the intersections points. The values of the intersection points are calculated by cubic spline interpolations functions; the number of founded intersections points is placed on the title bar of the **Intersect** window. Interpolation will be applied to all digitized curves that are covered by at least 3 points.

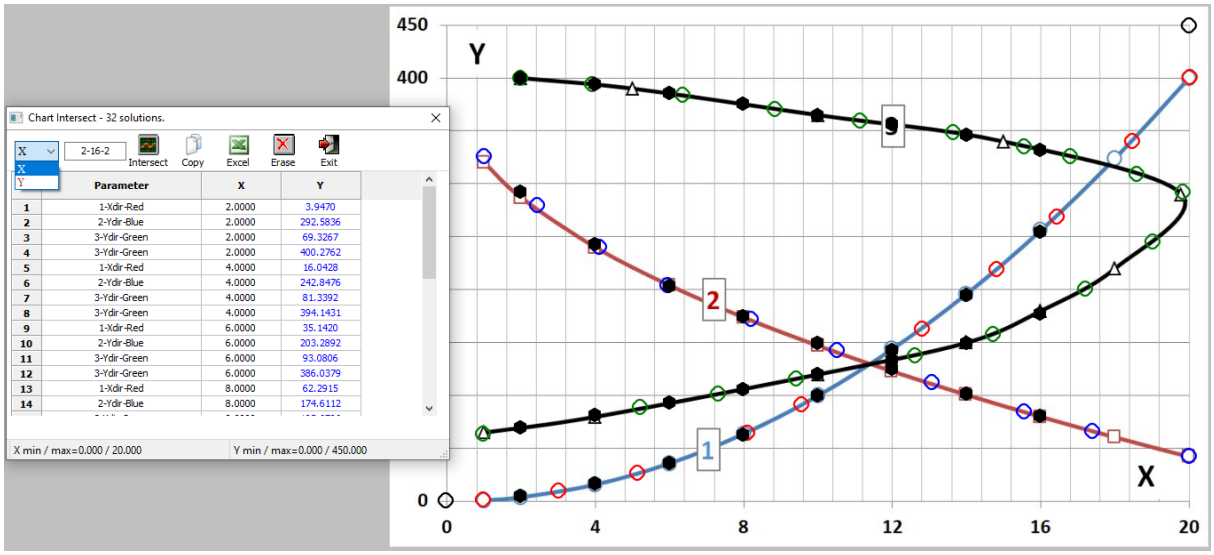

Figure 10. The chart intersect with multiple X values

6. THE REGRESSION COMMAND

The polynomial regression is a form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modelled as an n^{th} degree polynomial in x argument. Polynomial regression models are usually fit using the method of least squares. The PyDigitizer software uses polynomial regression for the degree of the polynomial between 1 and 6. Polynomial regressions can only be calculated for digitized curves with at least 3 points. The PyDigitizer software calculates all regression polynomials with degree between 1 and 6, for each displaying the corresponding chart and R squared (R2), which is an indicator of how well data fits the model of regression. The value of R2 ranges between [0,1], higher is the R2 value, better is the model and the results. The polynomial regression is exemplified for curve no. 2, Fig. 11, digitized by 15 points. The six regression polynomials are plotted against the digitized points (marked with black circles) in Fig. 12, respectively for each regression polynomial in Fig. 13. From these figures the R2 coefficient is calculated between 0.96 and 0.97 for degree 1 and degree 6 of the polynomial regressions. It can be seen that, as the degree of the polynomial increases, the regression curve has more inflections; so, the user can select the optimal polynomial approximation not only based on the value of the R2 coefficient but also on the shape of the regression curve. The coefficients of the 6 polynomials, the corresponding R2 coefficients, regression curve plots and 40 points calculated by polynomials are automatically exported to an Excel file.

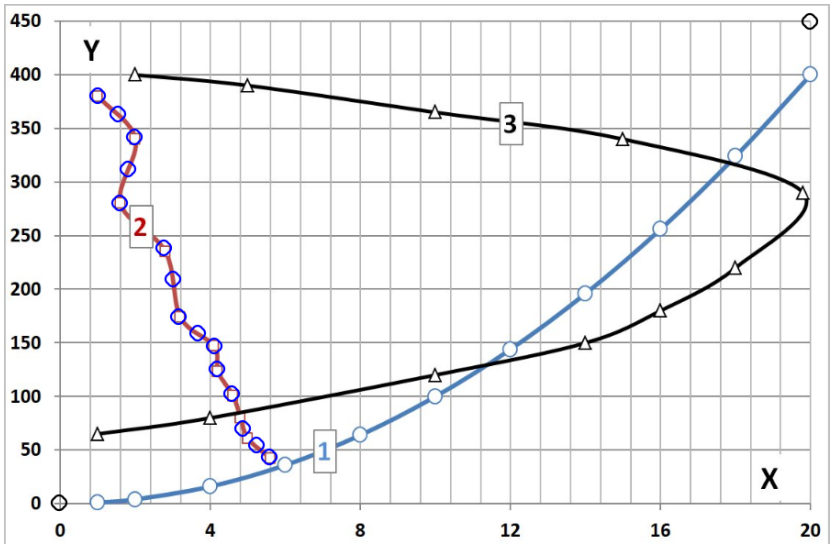

Figure 11. Curve no. 2 digitized by 15 points

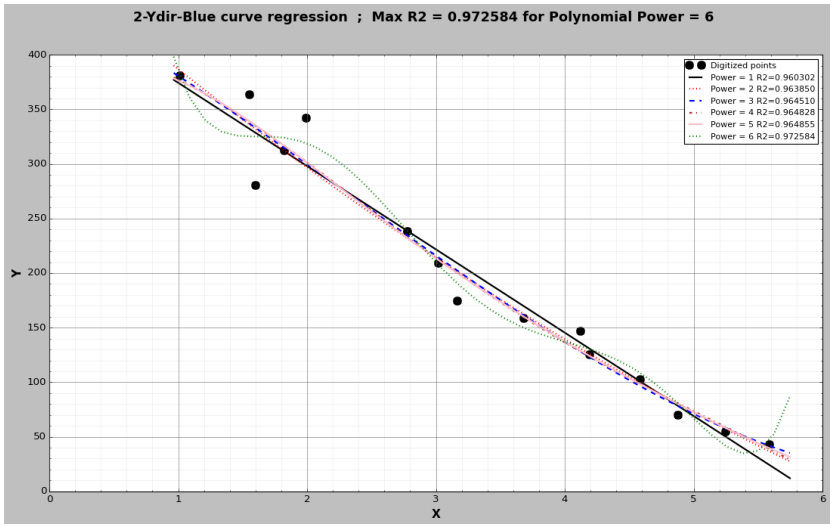

Figure 12. Digitized points and regression polynomials for curve no. 2

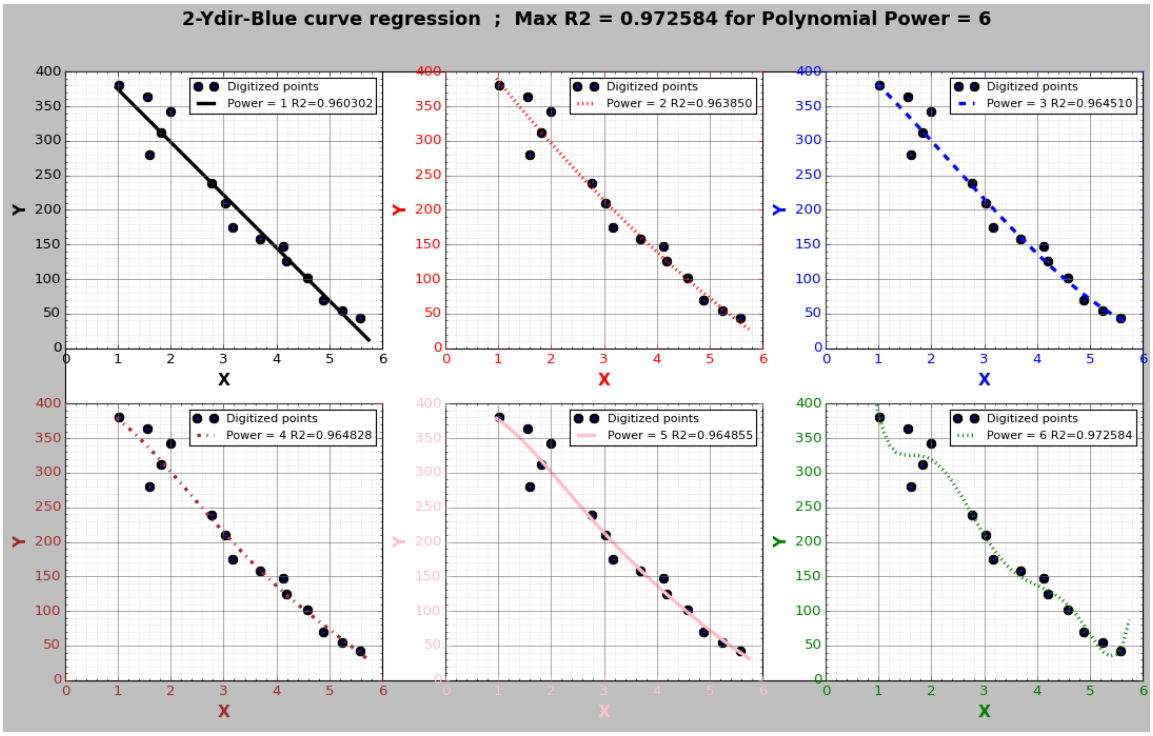

Figure 13. Digitized points and regression polynomials of curve no. 2 for the degree of polynomials between 1 and 6

7. THE SOFTWARE PACKAGES

The PyDigitizer software is created with the following free and Open Source resources:

- **Python** – a high-level programming language [4]. Python is free to use, even for commercial products, is an interpreted, interactive, object-oriented programming language.
- **wxPython** – is a graphical user interface toolkit for the Python programming language [5]. It allows Python programmers to create programs with a robust, highly functional graphical user interface, simply and easily [6].
- **Matplotlib** – is a comprehensive library for creating static, animated and interactive visualizations in Python. Its variety of output formats, several chart types, and capability to run interactively makes Matplotlib suitable for use in many different situations [7].
- **NumPy** – is the fundamental package for scientific computing with Python. [8].
- **SciPy** – is a library [9] that provides algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic & differential equations and many other classes of problems.
- **SQLite** – is a library [10] that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.

8. THE PROGRAMME DESCRIPTION AND PROGRAMMING TOOLS

The PyDigitizer application does not share data over the Internet, the code runs entirely on the user's computer, does not contain any viruses and not store data to any external server. The programme consists of more than 2500 instructions, grouped in the following main classes:

- **Vectorize** – the main class for digitizing 2D type XY curves;
- **AxisCalibration** – a class for XY axis calibration;
- **Add_Curves** – a class to define curves by name, sort type and color;
- **Intersect** – a class for 2D chart intersections with X or Y values, with functions for: spline calculations, points export to clipboard or Excel, deletion of the previous calculated points;

- **Regression** – a class to compute regression polynomial functions, for curves digitized with minimum 3 poins;
- **Table_Grid** – a class that shows the digitized points values and copies to Windows clipboard;
- other specific classes and functions, like: SplineInterpolation, Put_Clipboard, Sgn(x), etc.

Python was the chosen programming language to create the application. The wxPython module was used to create the frames and all Windows controls: toolbar, individuals controls (text, buttons, lists, etc.). The application was generated and tested through Python 2.7 version. The application was written in UliPad, a free wxPython powered, programmer oriented and flexible editor, with features such as class browser, code auto-complete, HTML viewer, directory browser and many others [11].
The SQLite module was used to create a database to store additional elements of the software: icon images, help file contents, text files, additional image files. To manage the application database the SQLite Expert was used [12], a powerful visual tool that enables to easily administer the SQLite databases. SQLite Expert integrates database management and maintenance into a single, seamless environment, with a clear and intuitive user interface. The cx_Freeze application was used to produces a folder containing an executable file for the program, along with the shared libraries (DLLs or others files) needed to run it [13]. The installation kit is created through the Inno Setup application and delivered as a single file "PyDigitizer_Setup.exe" [14].

9. CONCLUSIONS

The developed application PyDigitizer is designated to extract points from 2D curves contained in an image file, including constant X or Y curve intersection calculated through spline interpolation functions and generates polynomial regression functions. One of the application's goals: to use only free and Open Source resources, was achieved through the Python language and its associated modules wxPython, matplotlib, NumPy, SciPy, SQLite. The installation kit is available as dataset at Mendeley [15] for free install and use of it. PyDigitizer is released under CC0 1.0 license: "*You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission*". The operating procedure is detailed in a video on YouTube channel: https://youtu.be/WifxfTgQKcY.

REFERENCES

[1] Rohatgi A, Web Plot Digitizer, 2015, https://automeris.io/WebPlotDigitizer/citation.html.
[2] https://plotdigitizer.com/app
[3] http://www.graphreader.com/
[4] http://www.python.org/about/apps/.
[5] http://www.wxpython.org
[6] Rappin N, Dunn R, *wxPython in Action*, Manning Publications Co., 2006
[7] Hunter J D, *Matplotlib: A 2D Graphics Environment*, Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007
[8] https://numpy.org/
[9] https://scipy.org/
[10] Owens M, *The Definitive guide to SQLite*, Apress, 2006
[11] http://code.google.com/p/ulipad/
[12] http://www.sqliteexpert.com/
[13] https://pypi.org/project/cx-Freeze/
[14] https://jrsoftware.org/isinfo.php
[15] Nedelcu, Dorian; Latinovic, Tihomir (2022), "PyDigitizer – A Python module for digitizing 2D curves", Mendeley Data, V1, doi: 10.17632/jnwyr7jzrd.1, https://data.mendeley.com/