VOICE COMMANDS OF A 2D GRAPH

Dorian NEDELCU[1]

[1]"Eftimie Murgu" University, Resita, Romania, email: d.nedelcu@uem.ro

The paper focuses on the voice commands of a 2D graph, using the following software tools: Microsoft Speech SDK as a speech recognition tool (http://www.microsoft.com/speech/), Python as a programming language (http://www.python.org) and wx.Python as a graphical user interface toolkit (http://www.wxpython.org). The "PythonGraph" software resulted from this paper is an instrument for analyses and visual view of numerical data, created only through free and Open Source resources. The main functions of the graph are activated through voice commands, but also in Windows style: menus and toolbar. The graph data is imported from Excel or CSV file with a template structure. The software is provided with zooming instruments (fit, pan, zoom in, zoom out), spline curves interpolation, graph intersection with constant X or Y values and Excel export of the results, visual follow of the graph points to view coordinates, export of data in HTML, CSV or TXT format, save the graph as image file and the modification of the graph general setting (the colour, the size/the font, the visibility of the curves, markers, parameters, graph title, labels, grid, axis).

Keywords: Voice, recognition, commands, graph, Python.

1. INTRODUCTION

There are some software packages on the market for 2D curves representation. Why create a new software ? The answer can be synthesized by the following considerations:

- a challenge to create a 2D graph software using only free and Open Source resources;
- to obtain an engineering and practical tool for exploring 2D curves;
- to intersect 2D curves with X or Y values and obtain numerical intersection coordinates;
- to command the main operation by voice and in Windows style (menus & toolbar);
- to set the main properties of the graph with user preferences;
- to have a software which can be extended with the future necessities.

2. THE SOFTWARE PACKAGES

The "PythonGraph" software is created with the following free and Open Source resources, figure 1:

- **Python** – a high-level programming language (www.python.org). Python runs on Windows, Linux/Unix, Mac OS X, and has been ported to the Java and .NET virtual machines. Python is free to use, even for commercial products, because of its OSI-approved open source license. Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear syntax. It has interfaces to many system calls and libraries, as well as to various window systems, and is extensible in C or C++. It is also usable as an extension language for applications that need a programmable interface [1].
- **SQLite** – is a library (http://www.sqlite.org) that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world, with the source code launched in the public domain [2].

- **wxPython** – is a graphical user interface toolkit (G.U.I.) for the Python programming language (http://www.wxpython.org). It allows Python programmers to create programs with a robust, highly functional graphical user interface, simply and easily. It is implemented as a Python extension module (native code) that wraps the popular wxWidgets cross platform G.U.I. library, which is written in C++. Like Python and wxWidgets, wxPython is Open Source which means that it is free for anyone to use and the source code is available for anyone to look at and modify. Or anyone can contribute fixes or enhancements to the project. wxPython is a cross-platform toolkit. This means that the same program will run on multiple platforms without modification. Currently supported platforms are 32-bit Microsoft Windows, most Unix or unix-like systems, and Macintosh OS X [3].
- **Microsoft Speech SDK 5.1** – this kit is used to develop speech applications with Visual Basic, ECMAScript and other Automation languages. The SDK also includes freely distributable text-to-speech (TTS) engines (in U.S. English and Simplified Chinese) and speech recognition (SR) engines (in U.S. English, Simplified Chinese, and Japanese). (http://microsoft-speech-sdk.software.informer.com/).
- **PySpeech** – is a Python module (http://code.google.com/p/pyspeech) that provides a clean interface to Windows's voice recognition and text-to-speech capabilities. It's can be used within a program that needs to listen for specific phrases or general speech, or that needs to speak. PySpeech works in any language that Microsoft Speech supports. PySpeech is a wrapper that tells Windows to start up Microsoft Speech, and which hooks into the callback structure provided by Microsoft Speech.

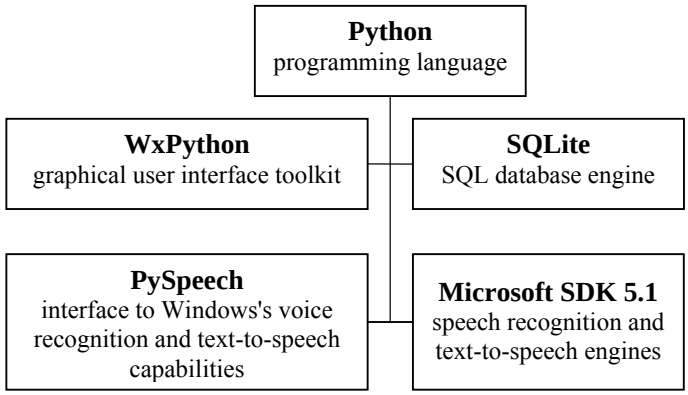


Figure 1. The software packages

3. THE DATA INPUT FORMAT FILE

The graph data is imported from an Excel or CSV file with the following template structure:

- the first line contains the number of curves followed by the graph title, in two cells for the Excel file or separated by a comma in the CSV file;
- the second line contains the X and Y labels of the graph, in two cells for the Excel file or separated by a comma in the CSV file;
- a number of lines equal to the number of curves, containing the curve parameter label followed by the number of points of the curve, in two cells for the Excel file or separated by a comma in the CSV file; for every curve, the file format continues with:
  - one empty line;
  - one line containing the curve parameter label;
  - a number of lines equal to the number of points of the curve, containing an point index, an X and Y value for every point, in three cells for the Excel file or separated by a comma in the CSV file.

4. THE "PythonGraph" MAIN WINDOW

The "PythonGraph" main window is presented in figure 2.

- the application title bar (**1**) – contains the application name, version and year number, followed by the author's name and location;
- the application main menu (**2**) – contains the menu options: **File**, **Graph**, **View**, **Help**;
- the application toolbar (**3**) – contains icons for the main functions of the application;
- the application graph area (**4**) – reserved for the drawing of graph components;
- the application status bar (**5**) – reserved for application's messages.

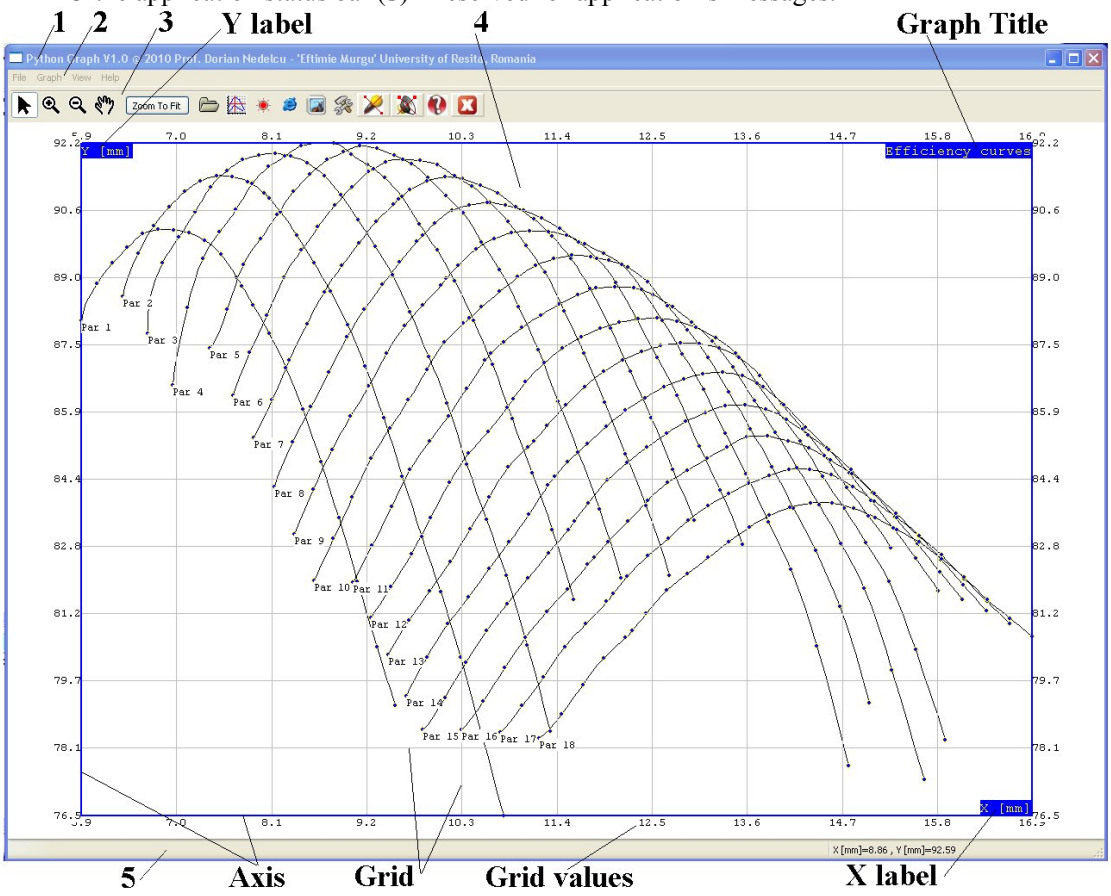


Figure 2. The "PythonGraph" main window

5. THE "PythonGraph" INTERFACE

The "PythonGraph" interface instruments (menu, toolbar, voice command recognition) are presented in table 1.

Table 1 - The "PythonGraph" functions

No.	Menu option	Toolbar icon	Recognition voice command	Description
1	File → Open		OPEN	Open data file in Excel or CSV format
2	File → Save TXT	-	-	Export data graph in TXT format
3	File → Save CSV	-	-	Export data graph in CSV format
4	File → Save HTML		DATA	Export data graph in HTML format

Table 1 - The "PythonGraph" functions

No.	Menu option	Toolbar icon	Recognition voice command	Description
5	File → Save PNG		IMAGE	Save graph as PNG image file
6	File → Settings		SETTINGS	Modify PythonGraph settings through "Setting" window
7	File → Exit		EXIT	Quit application
8	Graph → Intersect		INTERSECT	Graph intersection with X/Y constant values through "Graph Intersect" window
9	Graph → InfoPoints		POINTS	Identify and visualize graph points
10	View → Fit		FIT	Zoom to fit window
11	View → Redraw	-	-	Redraw the graph with default settings
12	Help → Help		Help	Accesing Python Graph help
13	Help → Author	-	-	Information about the author of the GraphPython software
14	Help → Licenses	-	-	Information about the licenses used for GraphPython software
15	Help → About	-	-	About the Python Graph
16	-		-	Select
17	-		-	Zoom In
18	-		-	Zoom Out
19	-		-	Pan
20	-		-	Activate / Dezactivate Voice Recognition commands
21	-		-	Activate / Dezactivate Voice Speech (Audio version of application messages)

6. THE "PythonGraph" SETTINGS WINDOW

The "Settings" windows is activated through the **File → Settings** menu, through the toolbar icon  or by "**SETTINGS**" voice command recognition. The Settings window is divided in the following sections: **General**, **Curves**, **Markers**, **Limits/Axis/Grid**, **Voices** and is provided with the **Default**, **Apply**, **Save** and **Quit** buttons. The general setting of the graph can be modified through this window: colour, size/ font, visibility of the curves, markers, parameters, graph title, labels, grid, axis.

The four buttons can be accessed through left mouse click or by voice commands recognition: **DEFAULT**, **APPLY**, **SAVE**, **QUIT**. The **DEFAULT** command  loads and applies the default graph settings. The **APPLY** command  regenerates the graph with the modified settings. The **SAVE** command  saves the graph settings in a configuration file, automatically loaded when the application restarts. The **QUIT** command  hides the **SETTINGS** window.

The five sections can be accessed through left mouse click or by voice commands recognition: **GENERAL**, **CURVES**, **MARKERS**, **LIMITS**, **VOICES**.

The **General** section, figure 3, shows the following graph information: graph title, curves number, maximum number of points per curves, total number of points per graph, the following numerical values Xmin, Xmax, Ymin, Ymax, X label, Y label, DX=Xmax-Xmin and DY=Ymax-Ymin differences and three option controls for **Show title**, **Show X label**, **Show Y label**.

The **Curves** section, figure 4, contains the following controls:

- **Show curves** – imposes the visibility of the curves;
- **Curves visibility list** – imposes the visibility of the individual curves, if **Show curves** control is activated;
- **Select/Deselect all curves** – for quick selection or deselection of all individual curves from the previous list control;
- **Uniform color** – imposes the same color of all curves; if this control is deactivated, the curves have different colors;
- **Uniform curves color** – activates the color matrix to select the uniform color of curves;
- **Show parameters** - imposes the visibility of the curve's parameters;
- **Parameters color** - activates the color matrix to select the color of parameters;
- **Parameters font size** – text control to specify the font's size for the parameters;
- **Alternate width** – imposes the alternate width of the curves; if this control is deactivated all the curves have the same width;
- **Alternate width value** – a list of values between 0 to 5; the selected value represents the width of all curves, if **Alternate width** is unchecked; if **Alternate width** is checked the curves width alternate between 0 and the selected width value;
- **SPLINE INTERPOLATION** section – contains the following controls:
  - **No. of spline interpolation points** – which imposes the number of interpolation points between every two initial data point of the curve; set 0 value to avoid spline interpolation or >=2 to activate the spline interpolation process;
  - **Show spline curves** – imposes the visibility of the spline curves;
  - **Spline curves color** – imposes the color of the spline curves.

The **Markers** section, figure 5, contains the following controls:

- **Show markers** – imposes the markers visibility of the curves;
- **Markers visibility list** – imposes the markers visibility of the individual curves, if **Show markers** control is activated;
- **Select/Deselect all curves** – for quick selection or deselection of all individual curves from the previous list control;
- **Uniform color** – imposes the same color of all curve's markers; if this control is deactivated the markers have different colors;
- **Diameter** – imposes the diameter size of the markers;
- **Show coordinates** - imposes the visibility of the coordinate's points of the curves;
- **Coordinates color** - activates the color matrix to select the color of coordinates;
- **Coordinates font size** – text control to specify the font's size of the coordinates;
- **SPLINE INTERPOLATION** section – contains the following controls:
  - **Show spline markers** – imposes the visibility of the spline curves markers;
  - **Spline markers color** – imposes the color of the spline markers.

The **Limits/Axis/Grid** section, figure 6, contains the following controls:

- **Show axis** – imposes the graph axis visibility;
- **Titles, Axis & Label color** – activates the color matrix to select the title, axis and labels color;
- **Titles Labels font size** – text control to specify the font's size of the title and labels;
- **Show grid** – imposes the visibility of the grid's graph;
- **Grid color** – activates the color matrix to select the color of the grid;
- **Interval numbers for X/Y grid** – the number of intervals for X/Y grid;
- **Decimal number for X/Y axis** – the number of decimals for X/Y grid values;
- **Show grid values** – imposes the visibility of the grid values;
- **Grid values color** – activates the color matrix to select the color of the grid values;
- **Grid values font size** – text control to specify the font's size of the grid values.

The **Voices** section, contains the following controls:

- **Voice Recognition** – activate / deactivate the voice recognition process of the commands;
- **Voice Speech** – activate / deactivate the voice announcements of the application messages.

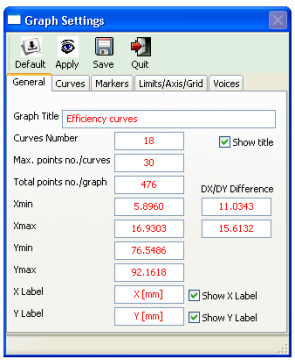


Figure 3. The "General" section

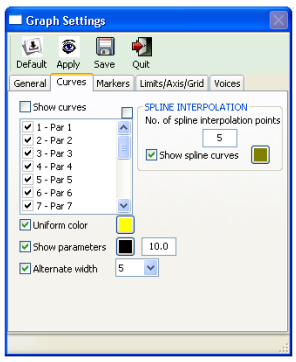


Figure 4. The "Curves" section

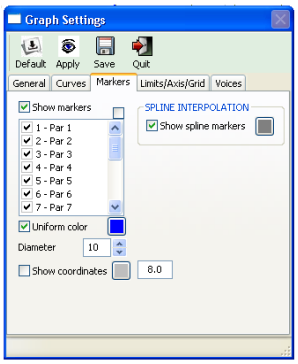


Figure 5. The "Markers" section

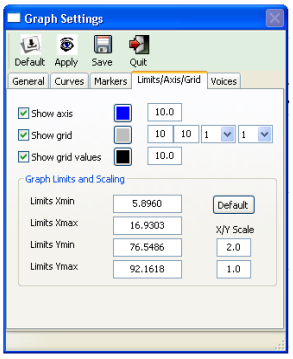


Figure 6. The "Limits" section

7. THE "PythonGraph" INTERSECT WINDOW

The "**Intersect**" windows is activated through the **Graph → Intersect** menu, through the toolbar icon or by the "**INTERSECT**" voice command recognition, figure 7. This window is designated to generate the intersection points of the graph with X/Y constant values. Through mouse click in the graph window, the application will generate the graph intersection with both X/Y directions passing through the clicked point. To intersect the graph with only one direction, X or Y direction control must be activated and an input numerical value is required in the intersection value text control; finally, the **Intersect** button will calculate the intersection points of the graph with the previous selected value and direction. The **Excel** button will send the intersection points coordinates to Excel. The **Erase** button will erase all the previous intersection points from the graph area and coordinate's table. The **Quit** button hides the "**Intersect**" window. The four buttons can be accessed through left mouse click or by voice commands: **INTERSECT**, **EXCEL**, **ERASE** and **QUIT**. The coordinate's table shows the ID, the intersected curve's parameter and the X/Y coordinates of the intersections points. The points resulted from the intersections are marked with distinct colour in the graph area. The values of the intersection points are calculated by cubic spline interpolations functions [4]; the number of found intersections points is placed on the title bar of the "**Intersect**" window.

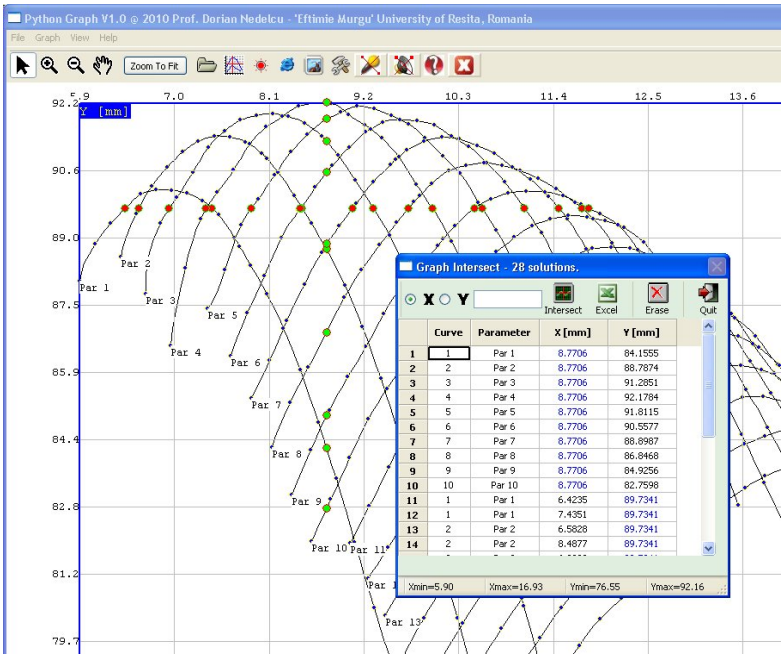
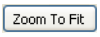


Figure 7. The "**INTERSECT**" window

8. THE "PythonGraph" INFOPOINTS COMMAND

The "**InfoPoints**" command is activated through the **Graph → InfoPoints** menu, through the toolbar icon  or by "**POINTS**" voice command recognition. This command will generate a special marker which follows the mouse position by placing the marker to the nearest graph point; also, the status bar will show the ID curve and ID point, the parameter of the curve and X/Y coordinates of the marked point. While **Select** icon  is active, a right click mouse will exit the "**InfoPoints**" command.

9. THE "PythonGraph" ZOOMING TOOLS

The "**Fit**" command is activated through the **View → Fit** menu, through the toolbar icon 

or by "**FIT**" voice command recognition. This command places the entire graph area in the current application window size.

The "**ZoomIn**" command is activated through the toolbar icon  . This command will magnify the graph area. If one point is clicked by the left mouse button in the graph area, the magnified process will be centered around this point. The same effect can be obtained by scrolling up the central mouse wheel. If two points are clicked, only the rectangular area between this points will be magnified. While "**ZoomIn**" command is active, a right mouse button will launch the "**ZoomOut**" command.

The "**ZoomOut**" command is activated through the toolbar icon  . This command will reduce the graph area around a point that was clicked by the left mouse button. The same effect can be obtained by scrolling down the central mouse wheel. While "**ZoomOut**" command is active, a right mouse button will launch the "**ZoomIn**" command.

The "**Pan**" command is activated through the toolbar icon  . This command translates the graph area simultaneously with the move of the pressed left mouse button.

10. THE "PythonGraph" VOICE RECOGNITION AND SPEECH SYSTEM

The voice command is a voice recognition system which launches one command at a time. The application has a dictionary of command keywords. When the user speaks in the microphone, the application identifies the command keyword and launches the appropriate command. The voice recognition system listens for the following keywords: OPEN, IMAGE, FIT, DATA, POINTS, SETTINGS, INTERSECT, ABOUT, HELP, EXIT for the main window, DEFAULT, APPLY, SAVE, QUIT, GENERAL, CURVES, MARKERS, LIMITS, VOICES for the **Settings** window and EXCEL, ERASE, QUIT for the **Intersect** window. The voice recognition system can be activated or deactivated through the toolbar icons  or  .

The user can be audio informed about the application's messages using voice speech system, which can be activated or deactivated through the application toolbar icons  or  .

These capabilities require Microsoft Speech SDK 5.1 kit and PySpeech module on the computer which run the "PythonGraph" application in Windows XP. For Windows Vista, only PySpeech module is required, because Window Speech Recognition is already included in Microsoft Windows Vista.

11. THE PROGRAMME DESCRIPTION AND PROGRAMMING TOOLS

The programme consists of ~2500 instructions, grouped in the following main classes:

- **DrawFrame** – a class for graph drawing: drawing normal and spline curves, normal and spline markers, point's coordinates values, grid lines, grid values, axis, X and Y labels, title graph;
- **Intersect** – a class for graph intersections with X and/or Y values, which includes functions for: spline calculations, points export to Excel, deletion of the previous calculated points;

- **Settings** – a class to modify the graph settings, which includes functions for: load default settings, save current settings in a configuration file named "GraphPython.cfg", apply the current settings to graph window;
- **Matrix** – a class for matrix manipulation;
- global functions for spline system calculation and spline intersection with a constant value;
- other specific classes and functions.

Python was the chosen programming language to create the application. The wxPython module was used to create the frames and all Windows controls: menu, toolbar, individuals controls (text, buttons, lists, etc.). Python is a remarkably powerful dynamic programming language that is used in a wide variety of application domains (http://www.python.org/about/apps/).

The speech and recognition process is recreated through the PySpeech module and Microsoft Speech SDK 5.1 engine. To create communication between Python and PySpeech & Microsoft engine the free Python Win32 Extensions [5] must be used.

The application was written in UliPad, a free wxPython powered, programmer oriented and flexible editor, with many features such as class browser, code auto-complete, html viewer, directory browser, wizard and many others (http://code.google.com/p/ulipad/).

The SQLite module was used to create a database to store additional elements of the software: icon's images, file help contents, text file of the licenses, additional image files. To manage the application database the SQLite Expert was used, a powerful visual tool that enables to easily administer SQLite databases. SQLite Expert integrates database management and maintenance into a single, seamless environment, with a clear and intuitive graphical user interface.

12. CONCLUSIONS

The developed application is designated to visualize and explore 2D curves, including constant X and/or Y graph intersection through spline interpolation functions. One of the application's goals: to use only free and Open Source resources, was achieved through Python language and associates modules (wxPython, PySpeech), SQLite, a database module, and text-to-speech and speech recognition engines, Microsoft Speech SDK 5.1 kit. The speech and recognition process is based on the Microsoft engine, which is free, but unfortunately does not have support for Romanian language at this time. The quality of the recognition process can be improved through a training process (http://support.microsoft.com/kb/306901).

REFERENCES

1. GUIDO, V.R.., DRAKE, F. L., *Tutorial Python,* Python Software Foundation, http://python.net/~gherman/propaganda/tut-ro/tut-ro.pdf, http://docs.python.org/tutorial/.
2. OWENS, M, *The Definitive guide to SQLite*, Apress, 2006.
3. RAPPIN, N., DUNN, R., *wxPython in Action*, Manning Publications Co., 2006.
4. BEU, T., *Analiza numerica in TurboPascal*, Editura MicroInformatica, Cluj-Napoca, 159-165, 1992.
5.HAMMOND, M., ROBINSON, A., *Python Programming On Win32. Help for Windows Programmers*, O'Reilly Media, January, 2000.