User’s Guide

Spline Calculus, version 2.12a-20080419

Gábor Peintler[1]

Home page: http://www.staff.u-szeged.hu/~peintler/

April 19, 2008

[1] University of Szeged, Department of Physical Chemistry

This documentation and the program Spline Calculus[1] described in it may be used and distributed freely complying with the following regulations:

- The programs and included data files can be distributed only in their original compressed archives, without any modification. The distribution of the documentation is possible only in PDF form without any change. The programs, data files and documentations are free, only the expense of the media and shipping/handling cost can be charged on the user.
- Anyone can use the program freely even for commercial purposes. Charging the use of the program is strongly prohibited in any way (e.g. rental fee, service charge, fee of distribution together with an instrument, etc.).
- If the use of Spline Calculus conduced the user to reach any result, this fact should clearly be indicated when the result is published.

All names and abbreviations (e.g. DOS, Windows, Linux, PostScript, Acrobat Reader, Zip, Unzip, GRX, DISLIN, etc.) together with the definitions of image formats (e.g., png) used in this document are either trademarks (^{TM}) or copyrighted (©) properties of the appropriate companies, organizations or persons.

The program are provided on an "as is" basis. It may contain inaccuracies and bugs. The person using this software bears all risk as to the quality and performance of this software.

Contents

List of Figures

Chapter 1

Introduction

Spline Calculus fits a smoothing spline on experimental data included in a file. The program draws the data and the fitted curve on the screen. While the program is running, the results can be checked visually and the fitted curve can be changed by modifying the error values. Most of the parameters concerning the calculation or the figure can also be controlled interactively.

1.1 Theoretical Background

To explain the splines is not the purpose of this manual. Essentially, it is a method for interpolation even in case of erroneous data (i.e., they may have experimental uncertainty). The appendix includes a brief mathematical deduction of the natural cubic and smoothing splines. Numerous articles, books, text books, electronic documentation can be found about splines and the writer of this program found the following three ones to be the most useful ones:

[1] de Boor, C., *A Practical Guide to Splines.* New York. Springer, 1978.

[2] Reinsch, C., H.: *Smoothing by Spline Functions.* Numerische Mathematik, 10, 177–183, (1967).

[3] Reinsch, C., H.: *Smoothing by Spline Functions II.* Numerische Mathematik, 16, 451–454, (1971).

The feature what makes the splines to be extremely useful is that complete mathematical analysis can be carried out numerically in the range of input data. This means that the function values, the first three derivatives and the definite integral values can easily be calculated.

1.2 Installation

Minimum requirements:

- 386DX 40 MHz processor,
- 0.5 Mbyte (DOS, Windows) or 2 Mbyte (Linux) free disk space,
- 4 Mbyte memory,
- VGA-compatible graphics card and
- MS-DOS 5.0, Windows 95 or Linux with kernel 2.2.

Acceptable configuration:

- 486DX 100 MHz processor or better,
- 1 Mbyte (DOS, Windows) or 3 Mbyte (Linux) free disk space (+1 Mbyte for every variant of the manual in PDF format),
- 8 Mbyte memory,
- VESA-compatible graphics card with 800×600 resolution and 8bit color depth, and
- MS-DOS 6.22, Windows 98SE or Linux with kernel 2.4.

The files installed on the user's computer:

Executables:
`spcalcd.exe` (DOS and Windows)
`spcalcw.exe` (Windows)
`spcalcx`, `spcalc.shared` (Linux)

Documentation:
`00readme` (English, text file)
`change.log` (english, text file)

Documentation downloadable independently from InterNet:
`SpCalceo.pdf` (english, on-line)
`SpCalcep.pdf` (english, printable)

Files for graphics:
`spcalc.png` (Linux, it can be used as icon)

Configuration file with default values:
`spcalc.cfg`

Sample files:
`sine.dat`, `smooth.dat`, `pH-Vtitr.dat`

1.2.1 DOS, Windows 95, 98, Millenium, 2000, XP and 2003 Operating Systems

The freely downloadable `spcalcdw.zip` contains the executable program, a part of the documentation and the example files.[1] The installation itself is very simple. The user has to create a directory (or folder, in windows nomenclature) and the contents of `spcalcdw.zip` has to be extracted into that directory. In this directory, the program (`spcalcd.exe` or `spcalcw.exe`) is ready to use. If the user wants to access the program from other directories, then either *(1)* the installation directory should be inserted into the `PATH` environment variable or *(2)* the program name has to be called together with its path (e.g. `C:\SPCALC\SPCALCW.EXE`). In the latter case the configuration file (*see* Section 2.2) must be copied into the specific directory if the user wants to use it there.

It is important to note that the graphics of DOS version is untrustworthy if it is started from the command line of Windows 2000, XP or 2003 operating system due to incompatibility issues. Nevertheless, `spcalcw.exe` runs flawless under any Windows versions.

Spline Calculus does not write anything into the registry of Windows so the programs, data files or even the whole directory can be moved freely.

In case the user wants to start the program by clicking on an icon, then the icon should be created in one of the folders belonging to the Start menu or on the Desktop according to the followings:

- Right-click on an empty area of the opened folder or the Desktop. (A folder can be opened in a window by *(1)* highlighting the name of the folder through the «Start|Programs» or «start|All Programs» menu, *(2)* right-clicking the name, and *(3)* choosing the «Open» menu.)
- The icon can be defined through the «Shortcut» option of the «New» menu.

1.2.2 Linux Operating System

It is important to note before all, that the Linux variants of the program have been used successfully on Debian, SuSE and Slackware distributions with KDE, XCFE, Window Maker,

[1] Self-extracting archive is not distributed for security reasons. The `zip` file can be unpacked using e.g. `Unzip.exe` which can be downloaded from the `http://www.info-zip.org/pub/infozip/UnZip.html` web page.

IceWM and fvwm2 window managers. The successful running is likely but not guaranteed under other systems (e.g. GNOME).

The executables, some parts of the documentation and the example files are to be found in the freely downloadable `spcalcxf.tgz`, `spcalcxo.tgz` or `spcalcxo64.tgz` files. These installation packages are identical but the first one was created under XFree with kernel version 2.4 while the latter ones were created under XOrg with kernel version 2.6. The first two ones are for `i386` compatible machines while the last one is for `amd64` architecture. The first one can practically be used under any Linux distribution with kernel 2.2 or higher but it does not understand the UTF characters.The installation is suggested to be done the following way:

- A directory should be created within the `/home/`*username* directory of the user in which Spline Calculus will be installed.
- The file `spcalcxf.tgz` (or `spcalcxo.tgz` or `spcalcxo64.tgz`) has be copied in the previously created directory and extracted with the

  `tar -xzf spcalcxf.tgz` (or `spcalcxo.tgz` or `spcalcxo64.tgz`)

  command.
- The file `spcalcxf.tgz` (or `spcalcxo.tgz` or `spcalcxo64.tgz`)can be deleted after extraction.

Certainly there are other means of the installation. If more than one users want to use the program and they do not want to have multiple copies of it on the fixed drives then the administrator (`root`) may copy the executables to the appropriate directories. The above described method, however, does not require the help of the system administrator, any user is able to carry out the installation alone.

Two identical executables can be found in each package. The statically linked executable (`spcalcx`) surely works with any Linux distribution (since it contains everything which is necessary during runtime) and this form can be installed and run without the help of the root user, but its size is huge. It may removed if the much smaller dynamically linked one (`spcalcx.shared`) works. The latter executable, however, depends on many libraries. Essentially, the DISLIN, GRX, tiff, png, jpeg, z and Motif related libraries should be installed to work the dynamically linked executable. The exact list of the required shared libraries are the followings:

Under XFree with kernel 2.4 for `i386` architecture, the `libdislin.so.9`, `libpng12.so.0`, `libjpeg.so.62`, `libtiff.so.4`, `libgrx20X.so.2`, `libX11.so.6`, `libm.so.6`, `libc.so.6`, `libpthread.so.0`, `libXm.so.3`, `libXt.so.6`, `libz.so.1`, `libdl.so.2`, `/lib/ld-linux.so.2`, `libXmu.so.6`, `libSM.so.6`, `libICE.so.6`, `libXext.so.6` and `libXp.so.6` libraries must be present.

Under XOrg with kernel 2.6 for `i386` architecture the `linux-gate.so.1`, `libdislin.so.9`, `libpng12.so.0`, `libjpeg.so.62`, `libtiff.so.4`, `libgrx20X.so.2`, `libX11.so.6`, `libm.so.6`, `libc.so.6`, `libpthread.so.0`, `libXm.so.4`, `libXt.so.6`, `libz.so.1`, `libXau.so.6`, `libXdmcp.so.6`, `libdl.so.2`, `/lib/ld-linux.so.2`, `libXmu.so.6`, `libXext.so.6`, `libXp.so.6`, `libXft.so.2`, `libXrender.so.1`, `libfontconfig.so.1`, `libfreetype.so.6`, `libSM.so.6`, `libICE.so.6` and `libexpat.so.1` libraries must be present.

Under XOrg with kernel 2.6 for `amd64` architecture the `libdislin.so.9`, `libpng12.so.0`, `libjpeg.so.62`, `libtiff.so.4`, `libgrx20X.so.2`, `libX11.so.6`, `libm.so.6`, `libc.so.6`, `libpthread.so.0`, `libXm.so.4`, `libXt.so.6`, `libz.so.1`, `libXau.so.6`, `libXdmcp.so.6`, `libdl.so.2`, `/lib64/ld-linux-x86-64.so.2`, `libXmu.so.6`, `libXext.so.6`, `libXp.so.6`, `libXft.so.2`, `libXrender.so.1`, `libfontconfig.so.1`, libfreetype.so.6, libSM.so.6, libICE.so.6 and libexpat.so.1 libraries must be present.

After the installation, the programs are ready to use. If the directory of the Spline Calculus is listed in the PATH environment variable then the program can be started from an X terminal window with the spcalcx or spcalcx.shared commands. Otherwise ./spcalcx or ./spcalcx.shared commands should be used. The programs can also be started by clicking on an icon. The creation procedure of an icon depends very much on the window manager or graphical desktop used. The file spcalc.png can be used as the image file of the icon.

There is a third option to use Spline Calculus under Linux. The Windows executable (spcalcw.exe) runs smoothly through WINE.

Chapter 2

Using Spline Calculus

The program uses simple text files both for input and output. Additionally, the graphical representation of the result can also be saved as `PNG`[1] binary image files.

The program reads two files for input: *(1)* the *data file* includes the experimental data and optionally their estimated uncertainties and *(2)* the *configuration file*. Four kinds of output file can be written by the program: *(1)* the *result file* includes the coefficients of the interpolating cubic polynomials and the curves of the graphics display, *(2)* the *user output file* contains the data selected by the user through the graphics display, *(3)* the *message file* includes the messages of Spline Calculus during calculations and *(4)* the *image files* contain the user selected graphics displays.

The configurable options of the program can be set in the following order:

1. Options are set through the command line parameters have the higher precedence.
2. Options are set in the configuration file have the next precedence.
3. Under Windows and Linux, if the name of the *data file* is not set either through a command line parameter or in the configuration file then a file open dialog asks the user to choose the *data file*.
4. Options are not set use their default values.

2.1 Data File

The structure of this file is simple as the sample files illustrates. There can be redundant rows at the beginning of the file including titles, descriptions, remark, etc. The rest of the file contains the input data arranged in columns. At least two columns must exist, the values of the independent (x) and dependent (y) data. The column of the deviation values (*see* Subsection 2.2.3 in detail) is optional. The sequence of the data must be the same in each row.

If a string in the x- or y-column does not represent a real number, the whole row is omitted. If a string in the deviation column cannot be a real number, the default value will be employed for δ. The x-values must be in increasing sequence downward in the file. If it is not so, the program sorts the rows before fitting the spline. If two rows has the same x-value, the program is stopped.

2.2 Configuration File

The name of the configuration file is supposed to be "`spcalc.cfg`" initially. The program looks for this file in the actual directory at first. If it is not found then the name of the executable is supposed for the configuration file. For example, if the name of the program is `spcalcw.exe` then the `spcalcw.cfg` file is searched on the actual directory.[2] If the actual

[1] `PNG` is the abbreviation of Portable Network Graphics. Homepage: `http://www.libpng.org/pub/png/`.

[2] Another example for Linux: if the name of the executable is `spcalcx.shared` then the supposed name for the configuration file is `spcalcx.cfg`.

directory does not contain any configuration file then the file is searched in the directory where the program is placed. If no configuration file is found, the default values are used.

In the configuration file, each line sets one option. The only exception is the first one: it must include the "`[SpCalc]`" string. If a row includes a hash mark (#) then everything behind this character (including the hash mark itself) is considered as remark. The file can include empty lines and remark lines,[3] too.

An option can be set by the help of the "`keyword=value`" syntax. There can be space(s) and/or tab key(s) before and after the "=" sign. Each option must occur after the row including "`[SpCalc]`" since everything is omitted before this row. The following subsections detail the keywords[4] that can be used

2.2.1 Keywords for the File Names

`InputFileName` *default value*: `spcalc.dat`

The value of this keyword is the name of the *data file*. The name can also include the full path with maximum 255 characters.

`OutputFileName` *default value*: `spcalc.out`

The value of this keyword is the name of the *result file*. The name can also include the full path with maximum 255 characters.

`CalcDataFileName` *default value*: `spcalc.cal`

The value of this keyword is the name of the *user output file*. The name can also include the full path with maximum 255 characters.

`LogFileName` *default value*: `spcalc.log`

The value of this keyword is the name of the *message file*. The name can also include the full path with maximum 255 characters.

`ImageBaseFileName` *default value*: `spcal`

The value of this keyword is the first part of the names of the *image files*. For example, if this value is "`plot`" and an image is to be saved, the program searches for the `plot000.png`, `plot001.png`, etc. files until the file does not exist in the actual directory. This non-existent filename is used for the image file to be saved next.

2.2.2 Keywords for the Data to be Used

`ColumnX` *default value*: `1`

The value of this keyword is the number of the column of the independent x-values in the *data file*.

`ColumnY` *default value*: `2`

The value of this keyword is the number of the column of the dependent y-values in the *data file*.

`OmittedRows` *default value*: `0`

The value of this keyword is the number of the rows to be skipped in the beginning of the *data file*.

`ColumnError` *default value*: `0`

The value of this keyword is the number of the column of the relative deviation values (*see* Subsection 2.2.3 in detail) in the *data file*. If the value is zero then default value is used for the relative deviations of the dependent y-values. It should be emphasized that if `ColumnError` is set (i.e., its value is larger then zero) then the *data file* must contain at least three columns.

[3] In remark lines, the first character is a hash mark.

[4] The keywords are case insensitive.

2.2.3 Keywords for the Deviations

Initially, all deviations of y-values (δ_i) are set to $(y_{max} - y_{min})/10^6$ (i.e., one millionth of the y-range). This value is called *initial base error*. After the initialization, the δ_i values are multiplied by the relative deviations. The latter values are taken from the column of the *data file* defined by the keyword "`ColumnError`". If no column defined for the relative deviations or the defined column contains missing values[5] then the default value (`1.0`) is used for the relative deviations. This default value can be changed by the keyword "`ErrorValue`". Beside the `ColumnError`, the following keywords influences the δ_i values:

`ErrorValue` *default value*: `-1.0`

If this value is set to a positive real number then this number becomes the default relative deviation value. If the value of this keyword is negative then the default relative deviation is `1.0`.

`ErrorRatio` *default value*: `no`

This keyword should equal to the "`no`" or should start with the "`yes`" string (case insensitively). In the latter case, a small positive integer number can follow the "`yes`" word. If the number is not indicated it is supposed to be `1`.

In case of "`yes`", the deviation values are weighted according to the slope of the curve defined by the x- and y-values. The slope is approximated by linear interpolation. The integer number determines how many neighboring points are used for the linear interpolation both from and from right. For example, if the number is `2` then the actual point itself, the two closest points from left and the two closest points from right are used, so the linear interpolation uses five successive points all together.

2.2.4 Keywords for Modifying the Outputs

`ShowCalculus` *default value*: `yes`

If this value is set to "`yes`" the points minimum, the points of maximum and the points of inflection are indicated on the screen and they are written into the *result file*. Otherwise, this keyword should set to "`no`" and the result of the mathematical analysis are indicated neither on the screen nor in the *result file*.

`WriteCurve` *default value*: `yes`

If this value is set to "`yes`" the points of the displayed calculated curve are also written into the *result file*. These data have fine resolution and they can be used for quick estimation of the function values, the first three derivatives and definite integral values. Otherwise, this keyword should set to "`no`" and the *result file* will not contain the data of the displayed calculated curve.

`NumberDigits` *default value*: `7`

This value determines the precision, i.e. it gives the number of significant digits of the floating point numbers in the *result file*.

`DisplayDigits` *default value*: `5`

This value determines the precision, i.e., it gives the number of significant digits of the floating point numbers on the screen.

2.2.5 Keywords for Choosing an Appropriate Graphics Driver

`GraphColorBits` *default value*: `4`

This value determines the color depth, i.e., how many bits gives the color of one pixel on the screen. The possible values are `4`, `8`, `15`, `16`, `24` and `32`. In principle, values `1` and `2` (i.e., black & white mode and CGA mode) also work. In these cases, however, there is no

[5] Such string is considered to be a missing value which is not able to represent a real number.

default values for the color codes, so they have to be set in the configuration file. The color codes are detailed in Subsection 2.2.7.

GraphGRXDriver *default value*: VESA under DOS, win32 under Windows and xwin under Linux

This keyword determines the graphics driver to be used. The possible values are herc, stdvga, stdega, et4000, cl5426, mach64, ati28800, s3, VESA and memory under DOS; win32 and memory under Windows; xwin and memory under Linux. If the value of this keyword is a question mark (?) then the program tries to detect the graphics driver automatically. A detailed description can be found at http://grx.gnu.de/grx246um.htm for the drivers of the GRX graphics library but it is not required for the use of Spline Calculus.

GraphResolution *default value*: 640x480

This keyword determines the width and height of the screen in pixels, i.e., they are the horizontal and vertical resolutions. The possible standard values are 320x200, 320x240, 320x400, 360x480, 400x300, 512x384, 640x200, 640x350, 640x400, 640x480, 800x600, 1024x768 and 1280x1024. Besides them, some drivers allow arbitrary resolution within the physical limit of the memory and the graphics card. In the latter case, the value of this keyword must be two integer values (width and height) joined with an "x" without spaces.

These three keywords are not independent of each other, only certain triplets work well. The program is able to detect all correct triplets automatically. The program carries out this detection and writes the possible triplets into the *message file*[6] if *any* of the following conditions is fulfilled:

- The keyword GraphGRXDriver has an invalid value.
- The value of keyword GraphResolution does not correspond to the above rules syntactically.
- The given triplet does not correspond to a valid mode of the used graphics card.
- The values of the keywords GraphGRXDriver and GraphResolution are question marks.

For example, the programmer's computer gives the following list of the valid drivers and graphics modes:

```
SplineCalculus v2.00a-20060727 has started ...

The required graphic mode cannot be initialized. Supported graphic modes for
the available drivers are (BPP: color depth in bits, Width and Height:
horizontal and vertical sizes of the graph given in pixels):

BPP Width Height   BPP Width Height   BPP Width Height   BPP Width Height
                          *** Driver: stdvga ***
  1   320x200        1   640x200        1   640x350        1   640x480
  4   320x200        4   640x200        4   640x350        4   640x480
  8   320x200        8   320x240        8   320x400        8   360x480
                          *** Driver: stdega ***
  1   320x200        1   640x200        1   640x350        4   320x200
  4   640x200        4   640x350
                          *** Driver: mach64 ***
  1   320x200        1   640x200        1   640x350        1   640x480
  4   800x600        4  1024x768        4   320x200        4   640x200
  4   640x350        4   640x480        8   640x480        8   800x600
  8  1024x768        8  1280x1024       8   320x200        8   320x240
```

[6] In this case, the program is terminated after the detection, and it does not carry out any calculation, of course.

```
  8   320x400        8   360x480       15   640x480       15   800x600
 15  1024x768       16   640x480       16   800x600       16  1024x768
 24   640x480       24   800x600
                         *** Driver: ati28800 ***
  1   320x200        1   640x200        1   640x350        1   640x480
  4   800x600        4  1024x768        4   320x200        4   640x200
  4   640x350        4   640x480        8   640x480        8   800x600
  8   320x200        8   320x240        8   320x400        8   360x480
                           *** Driver: VESA ***
  1   320x200        1   640x200        1   640x350        1   640x480
  4   800x600        4  1024x768        4   320x200        4   640x200
  4   640x350        4   640x480        8   320x200        8   320x240
  8   320x400        8   360x480        8   320x200        8   320x240
  8   400x300        8   512x384        8   640x350        8   640x400
  8   640x480        8   800x600        8  1024x768        8  1280x1024
 15   320x200       16   320x200       15   320x240       16   320x240
 15   400x300       16   400x300       15   512x384       16   512x384
 15   640x350       16   640x350       15   640x400       16   640x400
 15   640x480       16   640x480       15   800x600       16   800x600
 15  1024x768       16  1024x768       15  1280x1024      16  1280x1024
 24   320x200       24   320x240       24   400x300       24   512x384
 24   640x350       24   640x400       24   640x480       24   800x600
 24  1024x768       24  1280x1024
                        *** Driver: memory (!) ***
  1   640x480        4   640x480        8   640x480       24   640x480
Behind the driver name, (!) denotes that (besides the standard sizes) the
driver allows to choose arbitrary width and height within the physical
limitation of the graphics adapter and the RAM.

Please, choose valid graphic driver and mode (i.e. correct BPP, width and
height values) in the "SpCalc.cfg" configuration file!

SplineCalculus v2.00a-20060727 has stopped.
```

Nevertheless, note that the automatic detection is not fool-proof! Sometimes, a triplet is detected to be valid but it does not work.

2.2.6 Keywords for the Appearance of the Graphics Screen

SizeSymbol *default value*: 4

The input data are plotted by solid circles on the screen. This value is the radius of the circles given in pixels.

WidthLine *default value*: 3

The calculated curve is drawn by a solid line on the screen. This value is the width of the line given in pixels.

WidthCalculus *default value*: 1

The results of the mathematical analysis of the calculated curve are nominated on the screen if keyword `ShowCalculus` is set to `yes`. The places of the minimums and maximums are indicated by vertical dashed lines and vertical dotted lines indicate the infection points. The value of this keyword is the width of these vertical lines given in pixels.

WidthXplace *default value*: 1

The *focus line* shows the actual x-value. The value of this keyword is the width of the *focus line* given in pixels.

2.2.7 Keywords for the Display Colors

The colors can be defined by integer numbers. The program uses either an indirect or a direct method for color definition depending on the color depth.

- The indirect method is used in cases of 1-, 2-, 4- and 8-bit color depth. The colors are stored in a palette, and each color has its own serial number. A specific color can be referenced by its serial number. The available colors can be displayed by the help of keyword `GetColorCodes`. In 4- and 8-bit modes, the most frequently used colors (also called VGA-colors) have the following serial numbers:

black:	0	red:	4	dark gray:	8	light red:	12
blue:	1	magenta:	5	light blue:	9	light magenta:	13
green:	2	brown:	6	light green:	10	yellow:	14
cyan:	3	light gray:	7	light cyan:	11	white:	15

- In cases of 15-, 16-, 24- and 32-bit modes, the direct method is used. It means that the integer number directly defines the color. An RGB-value[7] is used to define a color. The following binary representations of the integer number and five examples illustrate how the red, green and blue components of the color should be given:

	15-bit mode	16-bit mode	24- and 32-bit mode
	$\underbrace{\texttt{bbbbbb}}_{\text{red}}\underbrace{\texttt{bb\ bbb}}_{\text{green}}\underbrace{\texttt{bbbbb}}_{\text{blue}}$	$\underbrace{\texttt{bbbbb}}_{\text{red}}\underbrace{\texttt{bbb\ bbb}}_{\text{green}}\underbrace{\texttt{bbbbb}}_{\text{blue}}$	$(\texttt{bbbbbbbb})\ \underbrace{\texttt{bbbbbbbb}}_{\text{red}}\ \underbrace{\texttt{bbbbbbbb}}_{\text{green}}\ \underbrace{\texttt{bbbbbbbb}}_{\text{blue}}$
black:	`x0000000 00000000`	`00000000 00000000`	`(xxxxxxxx) 00000000 00000000 00000000`
red:	`x1111100 00000000`	`11111000 00000000`	`(xxxxxxxx) 11111111 00000000 00000000`
green:	`x0000011 11100000`	`00000111 11100000`	`(xxxxxxxx) 00000000 11111111 00000000`
blue:	`x0000000 00011111`	`00000000 00011111`	`(xxxxxxxx) 00000000 00000000 11111111`
white:	`x1111111 11111111`	`11111111 11111111`	`(xxxxxxxx) 11111111 11111111 11111111`

In 15-bit mode, all the three components are defined by a 5-bit number (0. . .31). In 16-bit mode, the green component is defined by a 6-bit number (0. . .63) while the other two components are still defined by a 5-bit number (0. . .31). In 24- and 32-bit modes, all teh three components are defined by a byte (0. . .255).

The following keywords can be used to adjust the colors:

`ColorBackGround` *default value*: 4-8 bit mode:`15`; 15 bit mode:`32767`; 16 bit mode:`65535`; 24-32 bit mode:`16579836`

This value determines the color of the background.

`ColorText` *default value*: 4-8 bit mode:`0`; 15 bit mode:`0`; 16 bit mode:`0`; 24-32 bit mode: `0`

The color determined by this keyword is used for the regular texts and for the frames on the graphics screen.

`ColorSymbol` *default value*: 4-8 bit mode:`4`; 15 bit mode:`21504`; 16 bit mode:`43008`; 24-32 bit mode: `11010048`

This color is used for the solid circles of the input data.

[7] The RGB-method for the color definition is supposed to be known in this manual. It is described more deeply at `http://grx.gnu.de/grx246um.htm`.

`ColorLine` *default value*: 4-8 bit mode:`1`; 15 bit mode:`21`; 16 bit mode:`21`; 24-32 bit mode: `168`

This color is used for the solid line of the calculated curve.

`ColorCalculus` *default value*: 4-8 bit mode:`8`; 15 bit mode:`10570`; 16 bit mode:`21130`; 24-32 bit mode: `5526612`

This color is used for the dashed and dotted vertical lines nominating the extrema and the points of inflection.

`ColorXplace` *default value*: 4-8 bit mode:`5`; 15 bit mode:`21525`; 16 bit mode:`43029`; 24-32 bit mode: `11010216`

This color is used for the *focus line* nominating the actual x-value.

`ColorKeys` *default value*: 4-8 bit mode:`12`; 15 bit mode:`32074`; 16 bit mode:`64170`; 24-32 bit mode: `16536656`

This color is used for the *shortcut keyns* operating on the graphics screen.

`GetColorCodes` *default value*: `no`

In 4- and 8-bit mode, if this keyword is set to "`yes`" then the colors of the actual palette are displayed and pressing any key terminates the program.

2.2.8 Other Keywords

`JumpSize` *default value*: `1`

This value determines the unit of the movement in pixels if the *focus line* (i.e., the actual x-value) is changed.

`SizingStep` *default value*: `0.04`

If the user changes the x- and/or y-ranges, this value determines the extent of this change. The value is interpreted as a relative ratio, i.e., `0.04` means that the range is increased (or decreased) by 4%.

`MaxIteration` *default value*: `1000`

The calculation of smoothing splines requires an iterative mathematical procedure. This value gives the maximum number of iterations. If the calculation fails, increasing this value (up to 100 000) sometimes helps. This situation occurs more frequently if the calculated curve is getting to be very similar to a straight line. In such case, the unique solution of the equation system detailed in the appendix is becoming more and more ill-conditioned.

2.3 Using Command Line Parameters

The more important options can be given through the command line parameters. If they are givin in this way, these values have precedence against the values given in the configuration file. The command line parameters are interpreted as follows:

Parameter 1 is the name of the input *data file* so this value corresponds to the value of keyword `InputFileName` (*see* page 9).

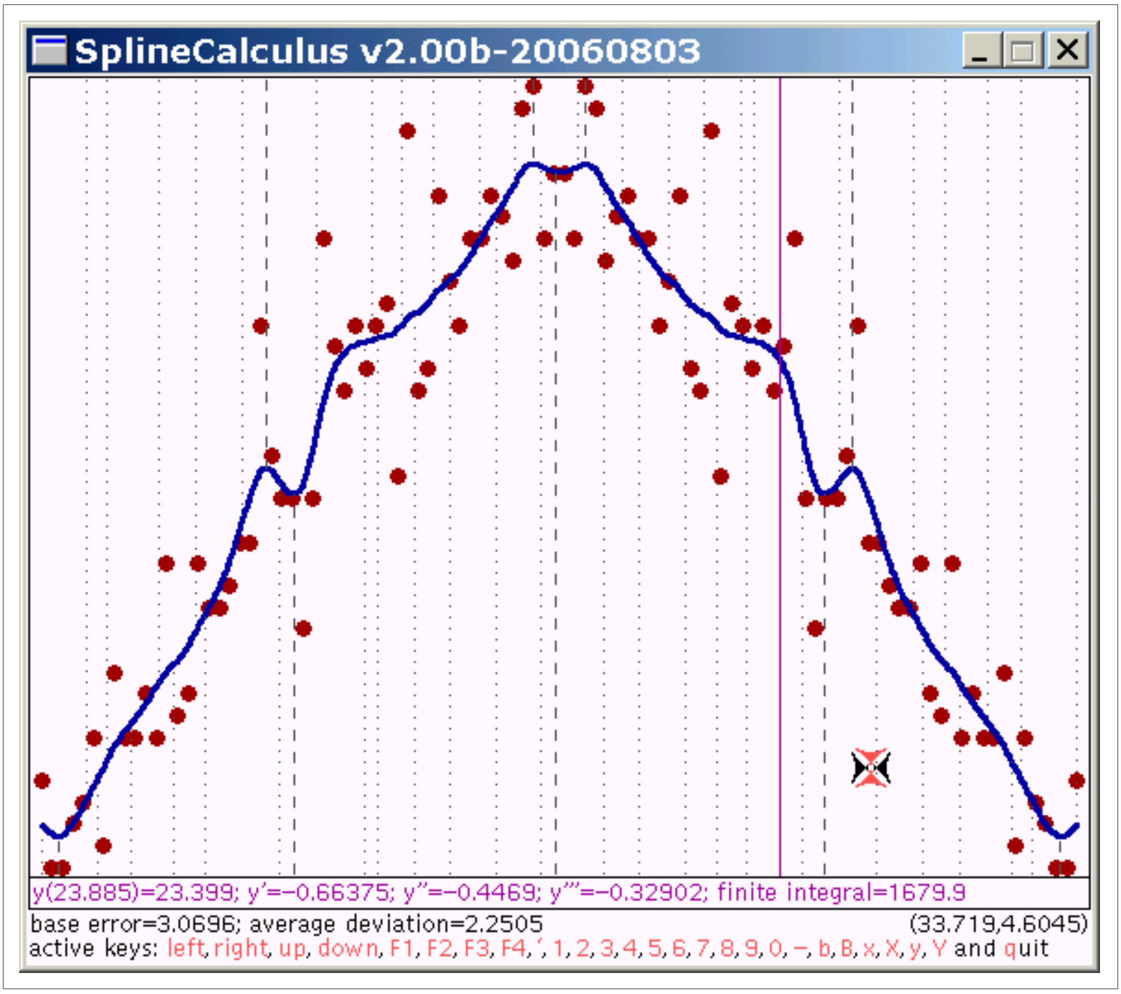
Parameter 2 is the column number of the independent values in the *data file* so this value corresponds to the value of keyword `ColumnX` (*see* page 9).

Parameter 3 is the column number of the dependent values in the *data file* so this value corresponds to the value of keyword `ColumnY` (*see* page 9).

Parameter 4 (or its absolute value) is the number of the redundant rows at the beginning of the *data file* so this value corresponds to the value of keyword `OmittedRows` (*see* page 9).

If this parameter is a negative integer number, the extrema and the points of inflection are not be denoted on the screen and they are not written into the *result file*. This behavior corresponds to the `ShowCalculus=no` setting (*see* page 10) in the *configuration file*. If there is no row to be omitted in the *data file*, this behavior can be achieved by the "`-0`" parameter value.

Parameter 5 is the column number of the deviation values in the *data file* so this value corresponds to the value of keyword `ColumnError` (*see* page 9).



```
y(23.885)=23.399; y'=-0.66375; y''=-0.4469; y'''=-0.32902; finite integral=1679.9
base error=3.0696; average deviation=2.2505                              (33.719,4.6045)
active keys: left, right, up, down, F1, F2, F3, F4, ', 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, -, b, B, x, X, y, Y and quit
```

Figure 2.1: The graphics screen of the program.

Parameter 6 corresponds to the integer number given in keyword `ErrorRatio`. See page 10 for the interpretation of this number.

2.4 Operating on the Graphics Screen

If the calculation of the smoothing spline is ended, the program plots the results on the screen (*see* Figure 2.1 as an example). The displayed window consists of two parts. The upper, framed one shows the results graphically. The lower part include some actual numerical information and it enumerates the shortcut keys for operating on the screen. The nominations on the graph are the followings:

- The input data are denoted by solid circles.
- The calculated smoothing spline is drawn by a solid line. The points of this curve is calculated at 200 or more x-values, depending on the actual slope. These calculated points are connected.
- The extrema are denoted by vertical dashed lines. The line is below the calculated curve at a minimum and it is above the curve at a maximum. If both the first and second derivatives are zero, the dashed line fills the whole displayed y-range.
- The points of inflection are denoted by vertical dotted lines. The line below the calculated curve means that the values of the third derivative is negative while a line is drawn above the calculated curve in case of positive third derivative.

- If keyword `ShowCalculus` is set to "`no`", the extrema and the points of inflection will not be denoted on the figure.
- A vertical solid line denotes the *focus line* which is always drawn at the actual x-value. Immediately below the graph, some results are displayed: the function value, the first three derivative and the definite integral are written at the actual x-value. Pressing the left button of the mouse on the area of the figure changes the actual x-value and moves the *focus line* to the clicked x-position.

The lower part consists of two rows. The left part of the first one gives two values: *(1)* the actual value of the base error (*see* the first paragraph of Subsection 2.2.3) and the average deviation from the last calculation. The latter one is the average of the absolute values of the differences between the input and calculated y-values, i.e., $\sum_{i=1}^{n} \left| y_i^{\text{calc}} - y_i^{\text{input}} \right|$ using the nominations given in the appendix.

The right part of the first row displays the actual position of the mouse in parentheses.

The second row of the lower part gives the shortcut keys. They have unique color defined by keyword `ColorKeys`. Their actions can be achieved either by pressing the appropriate key or by left-clicking the name of the key with the mouse. The meaning of the shortcut keys are the following:

left, right, up and down denotes the arrows. They move the graph to the corresponding direction without resizing. The extent of this movement can be set by keyword `SizingStep` (*see* page 14).

F1 and F2 decreases and increases the actual x-value, respectively. The *focus line* follows the actual x-value. The extent of this change is one pixel by default but it can be modified by keyword `Jumpsize` (*see* page 14). A more flexible change of the actual x-value is possible through the left button of the mouse.

F3 writes the actual x-value itself, its function value, the first three derivatives and the definite integral at the actual x-value into the *user output file*. Each pressing of this key adds one new row to the file but restarting the program erases the existing *user output file*.

F4 saves the current graph into an image file using Portable Network Graphics format. The image includes the upper part of the screen, the base error and the average deviation values but it does not contain the current mouse position and the list of the shortcut keys. The name of the image file is automatically chosen by the program as it is described at keyword `ImageBaseFileName` (*see* page 9).

‘, 1, 2, 3, 4, 5, 6, 7, 8, 9 and 0 shortcut keys multiply the actual base error by 1.01, 1.1, 2, 3, 4, 5, 6, 7, 8, 9 and 0.5, respectively. The deviation are recalculated with the new base error (*see* the first paragraph of Subsection 2.2.3), the smoothing spline is calculated again, the *result file* is rewritten and the graphics screen is redrawn.

− restores everything with the previous base error. i.e., this shortcut key is the undo function.

b, B, x, X, y and Y change the displayed ranges of the axes. "`b`" increases and "`B`" decreases the ranges of both axes. The center of the displayed portion does not move. The extent of resizing can be set by keyword `SizingStep` (*see* page 14). "`x`", "`X`" and "`y`" and "`Y`" keys work similarly as "`b`" and "`B`", but they influence only the x- and y-axis, respectively.

q or Q terminates the program.

2.5 Result File

The result file is a simple ASCII-file consisting three parts.

- The first part consists of the first two rows. The first row gives the version of the used Spline Calculus while the second one gives the conditions of the current task. The filename of input data, the column and error definitions are given here.
- The second part includes the data of the calculated curve with fine resolution. This part is omitted if keyword `WriteCurve` is set to "`no`". This part consists of six columns. They contain the calculated independent (`X`) and dependent (`Y`) data; the first (`Der1`), second (`Der2`) and third (`Der3`) derivatives and finally the definite integral (`Int`) values. For the definite integral, the starting point is the smallest input x-value and the last point is the actual x-value. These values are drawn as calculated curve on the graphics screen. This table is useful *(1)* for quick checking of the values and *(2)* for calculating y-values, derivatives and definite integrals by interpolation.
- The last part is also a table beginning with the "`Data of Ax^3+Bx^2+Cx+D-s of spline:`" title. Each row of this table belongs to a range of x-values surrounded by two successive points from the *data file*. The table consists of 12 columns: The lowest (`Xmin`) and highest (`Xmax`) x-value of the range; the coefficients (`A`, `B`, `C` and`D`) of the third degree polynomial which gives the calculated curve inside the range;[8] the function value (`Y(Xmin)`), the first three derivatives (`Der1(Xmin)`, `Der2(Xmin)` and `Der3(Xmin)`) and the definite integral (`Int(Xmin)`) at the lower boundary of the range. Finally, the last column gives the deviation (`Err(Xmin)`) at the lowest x-value of the range.

  If keyword `ShowCalculus` is set to "`yes`" (*see* page 10), the x-positions of the minimums, maximums and the points of inflection are also given below the row of the range in question.

  The last row of this table belongs to only the highest x-value given in the *data file*. It does not contain coefficients but the other columns are indicated.

2.6 User Output File

This file makes possible that the calculated results can be stored at only those independent values which are chosen by the user. The name of this file is `spcalc.cal` by default which can be changed by keyword `CalcDataFileName` in the *configuration file* (*see*p]age 9). The first two rows of this file include the version of the program and the most important conditions of the calculation. These conditions are given in the same form as they are given in the first part of the *result file*.

The structure of the rest of this file is the same as that of the second part of the *result file*. If the user presses the F3 key, the program writes the function value, the first three derivatives and the definite integral into this file at the actual x-value. Each new use of F3 key writes a new row into the file. Restarting the program erases this file when the F3 key is pressed for the first time. Keyword `WriteCurve` does not influence this file.

2.7 Message File

This file contains the runtime messages and error descriptions. The old version of this file is always erased at the start of the program. In case of automatic graphics driver and graphics mode detection (*see* Subsection 2.2.5) the *message file* includes the available drivers and their modes.

[8] These coefficients are denoted by a, b, c and d in the appendix.

Appendix

Mathematical Background

Given a set of (x_i, y_i) $(i = 1 \dots n)$ data pairs where $x_j \neq x_k$ $(j = 1 \dots n, k = 1 \dots n)$, i.e., duplicated independent values are not allowed. Supposing that x_i values are sorted in increasing order, the aim of any interpolating procedure is to estimate the dependent y-values at any x in the $[x_1, x_n]$ interval.

Using splines is one possibility for the interpolation. It can be proved mathematically that the natural cubic spline is the smoothest interpolating curve which means that this curve has the smallest

$$\int_{x_1}^{x_n} (f''(x))^2 \, dx$$

value for a specific data set.

Natural Cubic Spline

This expression means that a cubic polynomial ($y = a{\cdot}x^3 + b{\cdot}x^2 + c{\cdot}x + d$) is used to connect two successive points. Figure 2 illustrates the natural cubic splines. Since the $[x_1, x_n]$ interval are divided into $(n-1)$ ranges by the $x_2, \dots, x_{n-1}$ values, $4\cdot(n-1)$ coefficients should be calculated (they are the a_i, b_i, c_i and d_i ($i = 1 \dots n{-}1$) values). This can be done by applying the following restrictions and assumptions:

- The function values at the inner endpoints of the ranges can be calculated from any of the neighboring cubic polynomials:

$$\begin{aligned} y_i &= a_i{\cdot}\, x_i^3 + b_i{\cdot}\, x_i^2 + c_i{\cdot}\, x_i + d_i \quad (i = 1 \dots n-1) \qquad \text{and} \\ y_i &= a_{i-1}{\cdot}\, x_i^3 + b_{i-1}{\cdot}\, x_i^2 + c_{i-1}{\cdot}\, x_i + d_{i-1} \quad (i = 2 \dots n)\,. \end{aligned} \tag{1}$$

  It also means that the overall interpolating function is continuous. The left and right limits of the function values are the same, so there is no point of discontinuity in the $[x_1, x_n]$ interval.

- The left and right limits of the values of the first derivative should be the same at the inner endpoints of the used interval:

$$3a_i{\cdot}\, x_i^2 + 2b_i{\cdot}\, x_i + c_i = 3a_{i+1}{\cdot}\, x_i^2 + 2b_{i+1}{\cdot}\, x_i + c_{i+1} \quad (i = 2 \dots n{-}1)\,. \tag{2}$$

  As a consequence of this equation, there is no break point inside the $[x_1, x_n]$ interval.

- The left and right limits of the values of the second derivative should be the same at the inner endpoints of the used interval:

$$6a_i{\cdot}\, x_i + 2b_i = 6a_{i+1}{\cdot}\, x_i + 2b_{i+1} \quad (i = 2 \dots n{-}1)\,. \tag{3}$$

- Finally, two restrictions must be set to the second derivative at the endpoints of the overall interval:

$$6a_1{\cdot}\, x_1 + 2b_1 = 0 \quad \text{and} \quad 6a_{n-1}{\cdot}\, x_n + 2b_{n-1} = 0\,, \tag{4}$$

  i.e., there are points of inflection at both ends of the overall interval.

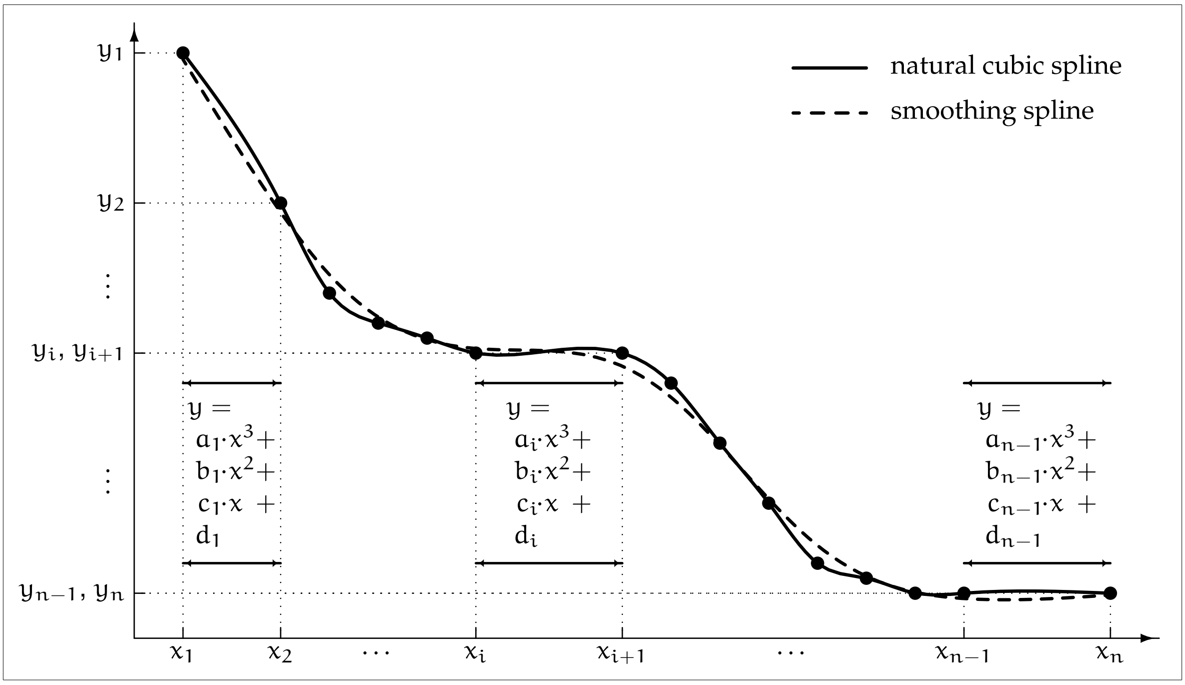


Figure 2: Illustration for the splines and the used notations.

Equations (1–4) forms an exact linear equation system (the number of equations is the same as the number of parameters to be determined), so the coefficients of the cubic polynomials can unambiguously be determined. With these coefficients, the function values, the first, second and third derivatives and the definite integrals can be calculated.

Smoothing Spline

Smoothing spline is a modified case of the natural cubic spline. The smoothing spline does not go through the (x_i, y_i) $(i = 1 \dots n)$ data points, it just appropriately nears to them around their deviations (δ_i) as it can be seen in Figure 2. It means mathematically that the

$$\frac{\sum_{i=1}^{n-1}\left(\frac{y_i - a_i \cdot x_i^3 - b_i \cdot x_i^2 - c_i \cdot x_i - d_i}{\delta_i}\right)^2 + \left(\frac{y_n - a_{n-1} \cdot x_n^3 - b_{n-1} \cdot x_n^2 - c_{n-1} \cdot x_n - d_{n-1}}{\delta_n}\right)^2}{n} \leq 1$$

inequality must also be fulfilled together with Equations (2–4) and with the following equations:

$$a_{i-1} \cdot x_i^3 + b_{i-1} \cdot x_i^2 + c_{i-1} \cdot x_i + d_{i-1} = a_i \cdot x_i^3 + b_i \cdot x_i^2 + c_i \cdot x_i + d_i \quad (i = 2 \dots n-1)$$

This equation system usually has unique solution for a_i, b_i, c_i and d_i $(i = 1 \dots n-1)$ at reasonably small δ_i values. However, if the δ_i values are large then infinite number of lines (the regression line is also among them) can be the solution.

It should be emphasized that the δ_i values are constant input data together with the x_i and y_i ones. They are not minimized during the calculation of the coefficients since the splines are not related to the non-linear parameter estimating procedures in any way.