

Advanced Visualization for Polynomial Regression Data Fit

Anton Vrdoljak¹

¹*Faculty of Civil Engineering, University of Mostar, Bosnia and Herzegovina*

ABSTRACT

A new and advanced visualization for polynomial regression data fit is presented. The visualization based on Flash – an advanced multimedia authoring and development environment for creating animations and interactive applications inside Web pages – will present a model for better understanding of curve fitting. This model ensures to engineering students, as well as other researchers, an interactive explanations which helps them to achieve some main learning goals: how the range and uncertainty and number of data points affect correlation coefficient, or how correlation coefficient and chi squared can be used to indicate how well a curve describes the data relationship. Another objective of this paper is to explore the ways to enhance motivational impact and outcomes of such Web-based learning activities.

Keywords: *Visualization, Advanced Visualization Tools, Polynomial Regression, Best Data Fit Curve, Web-based learning.*

INTRODUCTION

Visualization is a process that generates visual representations, such as imagery, graph, and animation, of information that is otherwise more difficult to understand through other forms of representations, such as text and audio. Visualization is essential component in modeling and simulation and is widely utilized at various stages of modeling and simulation applications [1]. Visualization is especially important and useful for conveying large amount of information and dynamic information that varies with time. Various levels of software and tools have been developed to facilitate visualization development, but most of them require significant programming skills and experience. Fortunately, there are many software packages which have been built so that users with less or no programming experience can make use of visualization for various purposes, such as 3D modeling, scientific visualization, and animations. Such software packages, that are widely used, are MS Excel, MATLAB and Adobe Flash. The most known package is MS Excel, and it is very hard to find someone, especially at university level, who never used this great tool, so in the next few rows will other two packages have a short introduction.

MATLAB is a high-level computing language and interactive environment for numeric computation, algorithm development, data analysis, and data visualization. It is the most widely used tool for teaching science, technology, engineering, and mathematics (STEM) in universities and colleges and for research and rapid prototype development in the industry [1]. The core of MATLAB is the MATLAB language, which is a high-level language optimized for matrix, vector, and array computations. It provides many toolboxes for various applications, including math and optimization, statistics and data analysis, control system

design and analysis, signal processing and communications, image processing, test and measurement, computational biology, computational finance, and databases. The MATLAB desktop environment consists of a set of tools that facilitate algorithm development and debugging. Users of MATLAB can build applications with graphic user interfaces. MATLAB provides powerful visualization capabilities, and it has an extensive set of functions for matrix and vector visualization, graph annotation and presentation, 3D visualization, displaying images of various formats, as well as camera and lighting controls. The major advantage of using MATLAB for visualization is simplicity and rapid implementation. For many scientific and engineering applications, the visualization capabilities provided by MATLAB are sufficient. In other applications, MATLAB can be used to develop prototype visualizations in order to obtain quick understanding of the problem. Then, a full-featured stand-alone application can be developed, which does not need MATLAB in order to implement and run the visualizations.

Flash is an advanced multimedia authoring and development environment for creating animations and interactive applications inside Web pages. It was originally a product of Macromedia but was recently acquired by Adobe. Flash supports vector graphics, digital images and videos, and rich sound effects, and it contains a comprehensive set of tools for creating, transforming, and editing objects, visual effects, and animations for different hardware and software platforms. Animations can be developed quickly using optimized animation tools, including object-based animation, motion presets, motion path, motion editor, and inverse kinematics. Procedural modeling and filtering can be used to generate various visual effects more quickly and easily. 2D objects can be animated in 3D space using 3D transformation and rotation tools. In addition to the graphic authoring environment, Flash has its own scripting language, ActionScript, which is an object-oriented programming language that allows for more flexibilities and control in order to generate complex interactive Web applications. Flash applications can run inside Web pages using plug-ins, such as Flash player. Flash can also embed videos and audio in Web pages, and it is used by many Web sites, such as YouTube. Flash is currently the leading technology for building interactive Web pages, and it now integrates well with other Adobe products. Many interactive and educational applications including games built using Flash are available on the Internet.

In this paper we will look at background to data fitting (curve fitting) as implemented by new and advanced visualization based on Flash. What we mean by data (curve) fitting in this context? Data fitting is the mathematical analysis of a data set. Specifically we used it to fit the “best line” or “best curve” through our data and give us fitted parameter values [2]. Thanks to modern computer technology, especially in past three decades, we are able to do computer based data fitting, what obviously has a number of advantages, for example it is a faster than a manual methods, and gives values that are not subjective. Next, the data can be fitted to complex equations including non-linear ones, and the fit also gives the error estimates that can be used to assess data quality. However, most computer based data fitting will not help students or other researchers to understand how well a curve describes a set of data. Fortunately, there are small team of researchers at University of Colorado [3], who recognized such faults, and who tried to develop a new model which will be able to handle with these teaching activities. Their very interactive simulation for curve fitting served as a model for our new and advanced visualization for polynomial regression data fit. So, as we noticed in the abstract, the main goal of this paper is presenting a new model that ensures to engineering students, as well as other researchers, an interactive explanations which helps them to achieve some main learning goals: students will be able to explain how the range and uncertainty and number of data points affect correlation coefficient and chi squared; describe how correlation coefficient and chi squared can be used to indicate how well a curve describes the data relationship; apply understanding of curve fitting to designing experiments.

POLYFIT: ONE PART OF THE DEVELOPED MODEL

The tool which we mainly utilized for the development of our model is Adobe Flash. Reasons are multiple. For example, its potential of creating rich media or multimedia for the World Wide Web with user friendly interface, then especially its possibility in combination of drawing tool and programming at the same time, or its availability and compatibility. These features make contents more accessible and most attractive for almost all online users. This is very important because students are the majority of user population.

We present here PolyFit – one part of the developed model based on the advantage of learning with ICT, especially Web-based learning. The example, polynomial regression data fit, is a part of “Interactive Lessons” of the curve fitting in course applied mathematics (this example or activity could be used in any course where students are asked to understand how well a curve describes a set of data). To make best use of PolyFit, it is helpful to know some of the theory behind regression analysis. Regression analysis is a technique used to calculate the “best fit” line or curve through a data set by minimizing the deviation of the data from the curve[2]. Mathematically, it finds values for various unknown parameters in an equation that produce the smallest possible sum of the squares deviations of the experimental values from the calculated ones. Regression analysis is the mathematical equivalent of finding the best line by eye – it has the advantage, however, of being non-subjective. It is also possible to fit the “best curve” through the data by non-linear regression. This has no manual equivalent (because humans are very poor at judging accurate curvature), but is equally simple mathematically using computational methods. Regression analysis is probably the most powerful method to analyze data. However, such analysis involves several assumptions about the function chosen and the error structure of the data [4].

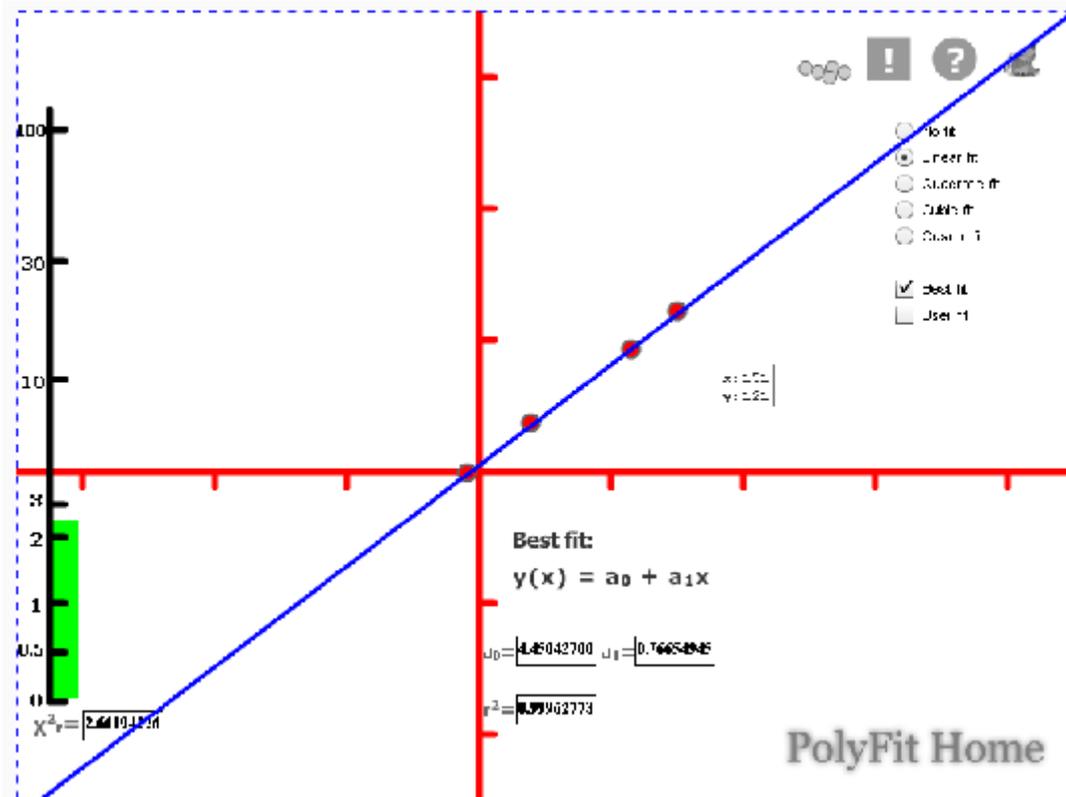


Figure 1 Red dots can be dragged and the changes of values are visible

The dynamic plane has a button for adding some mobile dots, which are our data points (Figure 1). With your mouse, drag data points and watch the best-fit polynomial curve update instantly. At the same time, the values of coefficients for polynomial regression and the value for chi-squared are given as a feedback or information. Next, by adding even more new data points, it's possible to dynamically change shape of best curve and coefficient values too, which are represented numerically and graphically. This is the most important function of this interactive lesson, because immediate giving of the values of coefficients for polynomial regression and value for chi-squared, after adding new data points, has the potential to change the student's activities in a very impressive manner. Probably the most important aspect of Web-based learning is not the access to information, but the ability to customize learning approaches to match student's learning styles. So, in our research and development, we have tried to develop a model which will support above indicated aspect.

You can choose these types of fit: *linear*, *quadratic*, *cubic*, or *quartic*. The reduced chi-square statistic shows you when the fit is good. Or you can try to find the best fit by manually adjusting fit parameters. Finally, you are able to erase all data points at once by clicking on a proper button in the dynamic plane. All commands are available with right mouse click too.

Besides main learning goals, we expect here too that the learner finds, by exploring and observing numerical data and graphic representation, some optional but advanced learning goals: differentiate between correlation coefficient and chi-squared; use the equation for chi-squared to explain why a curve fit with a value near one describes the data well.

Teaching experiment, later kept at Faculty of Civil Engineering in Mostar (first year of graduate study), shows that students were able to give very well and correct answers on a question: *how well does the curve describe the data?* Next, they were able to predict which data will have a better curve fit, and explain way (Figure 2). Evident difference of student's activity was observed as they were motivated to actively participate during the lesson. Also they were enthusiastic throughout the lesson.

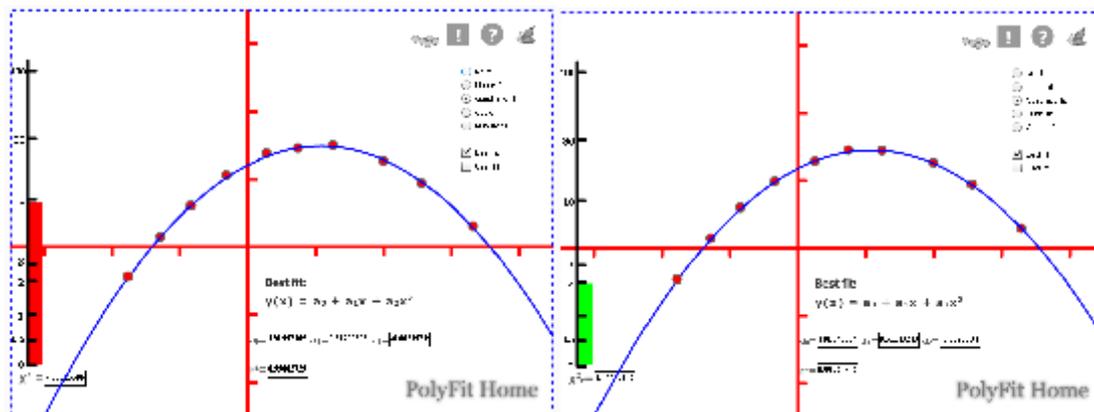


Figure 2 Which is best and way?

This part of developed model is still under construction, but we are close to finish our project. New version of PolyFit – projected for June 2011 – will bring new features: exporting your data points to XML file, loading your data points from some text file, etc. That will be probably final version, because no other developing is projected for close future. One reason is in expecting a new release of GeoGebra – free mathematics software for learning and teaching. It will be released in the summer of 2011, and will bring many new features and tools [5]. As we know in this moment, among others, it will bring button tool, what will give new dynamical possibilities, close to ones that we can find in Adobe Flash.

CONCLUSION

The brief descriptions of activities, with a given part of developed model in last chapter, points only some of the educational potentials and possibilities afforded with a new and advanced visualization for polynomial regression data fit. Our developed model ensures to engineering students, as well as other researchers, a very interactive explanations for regression analysis, which helps them to achieve big improvement in their learning activities. The core findings of our work have shown that:

- visualization based on Flash and/or Web-based learning enables a multi-sensory approach to both teaching and learning, and many children need visual, auditory and kinaesthetic stimulation in order to enable learning;
- student's motivation has been enhanced most positively;
- students have positive attitudes to the use implemented models of PolyFit;
- and finally visualization based on Flash and/or Web-based learning enabling students to work on tasks outside classrooms.

It is obvious that for this result we can thank mainly the integration and implementation of technologies in teaching and learning of regression analysis. Technology brings to students and their teachers the opportunity to individualize learning – to generate illustrative examples, as well as a dynamic and rich presentation of a given subject, to follow interesting topics to the desire depth, to choose their own problems and appropriate tools for solving them. However, the complexity in integration and implementation of technologies in teaching and learning process, as well as the complexity in using modern and advanced technology in education implies that this is an area in need of further research about different issues. There is also a substantial need of development concerning different ways to train and educate all those who are involved in decisions about and realization of education supported by technology, especially Web-based education.

REFERENCES

- [1] Sokolowski, J.A. and Banks, C.M. (2010) Modeling and Simulation Fundamentals. *John Wiley & Sons, Inc.*, Hoboken, New Jersey, USA.
- [2] Chapra, S.C. and Canale R.P. (2006) Numerical methods for Engineers. *McGraw-Hill Book Co.*, New York, NY, USA.
- [3] Loeblein, T. and Dubson M. (2008) Interactive Simulations. <http://phet.colorado.edu/>, University of Colorado, USA.
- [4] Yakowitz, S. and Szidarovsky, F. (1986) An Introduction to Numerical Computation. *Macmillan*, New York, NY, USA.
- [5] GeoGebraWiki (2011) Release Notes GeoGebra 4.0. <http://www.geogebra.org/>, GeoGebra – Dynamic Mathematics for Everyone.

In R for fitting a polynomial regression model (not orthogonal), there are two methods, among them identical. Suppose we seek the values of beta coefficients for a polynomial of degree 1, then 2nd degree, and 3rd degree: `fit1 <- lm(sample1$Population ~ sample1$Year)` `fit2 <- lm(sample1$Population ~ sample1$Year + I(sample1$Year^2))` `fit3 <- lm(sample1$Population ~ sample1$Year + I(sample1$Year^2) + I(sample1$Year^3))`. Or we can write more quickly, for polynomials of degree 2 and 3: `fit2b <- lm(sample1$Population ~ poly(sample1$Year, 2, raw=TRUE))` `fit3b <- lm(sample1$Population ~ p` Polynomial regression is a special case of linear regression where we fit a polynomial equation on the data with a curvilinear relationship between the target variable and the independent variables. In a curvilinear relationship, the value of the target variable changes in a non-uniform manner with respect to the predictor (s).
importing libraries for polynomial transform from sklearn.preprocessing `import PolynomialFeatures` # for creating pipeline from sklearn.pipeline `import Pipeline` # creating pipeline and fitting it on data `Input=` `[('polynomial',PolynomialFeatures(degree=2)),('modal',LinearRegression())]` `pipe=Pipeline(Input)` `pipe.fit(x.reshape(-1,1),y.reshape(-1,1))`.