Journal of
open research software

Glaser-Opitz, H and Budajová, K 2016 MATH: A Scientific Tool for Numerical Methods Calculation and Visualization. *Journal of Open Research Software,* 4: e6, DOI: http://dx.doi.org/10.5334/jors.55

SOFTWARE METAPAPER

MATH: A Scientific Tool for Numerical Methods Calculation and Visualization

Henrich Glaser-Opitz[1] and Kristína Budajová[2]

[1] Department of Avionics, Faculty of Aeronautics, Technical University Košice, Slovak Republic
h.glaseropitz@gmail.com

[2] Department of Aerodynamics and Simulations, Faculty of Aeronautics, Technical University Košice, Slovak Republic
kristina.budajova@tuke.sk

MATH is an easy to use application for various numerical methods calculations with graphical user interface and integrated plotting tool written in Qt with extensive use of Qwt library for plotting options and use of Gsl and MuParser libraries as a numerical and parser helping libraries. It can be found at http://sourceforge.net/projects/nummath. MATH is a convenient tool for use in education process because of its capability of showing every important step in solution process to better understand how it is done. MATH also enables fast comparison of similar method speed and precision.

Keywords: Qt; numerical methods; nonlinear equations; system of linear equations; integration; derivative; LMS approximation; error estimation; step-by-step solutions

(1) Overview

Introduction

The MATH was originally created as open-source software to help in education process of undergraduate and graduate students to better understand numerical methods and to speed up the learning process as itself. Throughout its development process shows up many new capabilities that can be achieved with it. MATH application was designed to achieve several goals:

- Provide all calculation steps of each numerical method with easy to use graphical interface;
- Enable graphical visualisation of calculated data as well as option of exporting gathered data and graphs in suitable formats.

There are several open-source web-based applications with similar capabilities (that the author is aware of), but they are usually not designed for understanding the very basics of various numerical methods which are very important and so not well suited for education process. MATH provides so far these numerical methods [6–9]:

1) Nonlinear equations – Bisection method
   – Regula Falsi method
   – Secant method
   – Newton method
   – Iteration method
2) Definite integration – Rectangular method
   – Trapezoid method
   – Simpson's method
   – Monte Carlo hit or miss method
   – Monte Carlo average method
3) Double definite integration
4) Numerical differentiation – up to 4. Degree of differentiation
5) System of linear equations – Jacobi iteration method
6) Matrix calculation – basic numerical operations, transpose, inverse, etc.
7) Least square approximation

Numerical analysis is a subject of extreme interest to mathematicians, computer scientists and engineers who will welcome this software application. For engineers there is often a question which method will be the best to use, the most accurate, the fastest etc. To be able to decide, there is important to know the basics and core of chosen methods. MATH is designed to help engineering students to better understand how it works and to learn faster and more easily.

At this time MATH is still in an early stage of development. However it is being already used as an education tool by various professors at Technical University Košice and is spreading further. There are still many numerical methods and graphical capabilities that will be implemented throughout the time.

Implementation and architecture

MATH is written in Qt 4.8.1 [1], which is a C++ graphical toolkit for cross – platform application development. Qt uses two more libraries MuParser [2] and Gsl [3] for parsing expressions and matrix calculations. For plotting is used Qwt library [4]. The graphical user interface (GUI) shown in **Figure 1** provides users to choose from presented numerical methods, visualize chosen numerical method and data export. Data and graphs can be exported in most common formats (PDF, PNG, JPEG, TXT, etc.). Plotting tool has most of commonly used options like changing line's color, width, zooming, grids, axis, title, labels, etc.

As you can see on **Fig. 1**, you can choose from 14 numerical methods implemented so far, with graphical representation of definite integration. LMS method has its own graphical representation that will help students to understand what necessary steps need to be done to correctly solve the problem, what is the final result, and compare the performance of similar numerical methods.

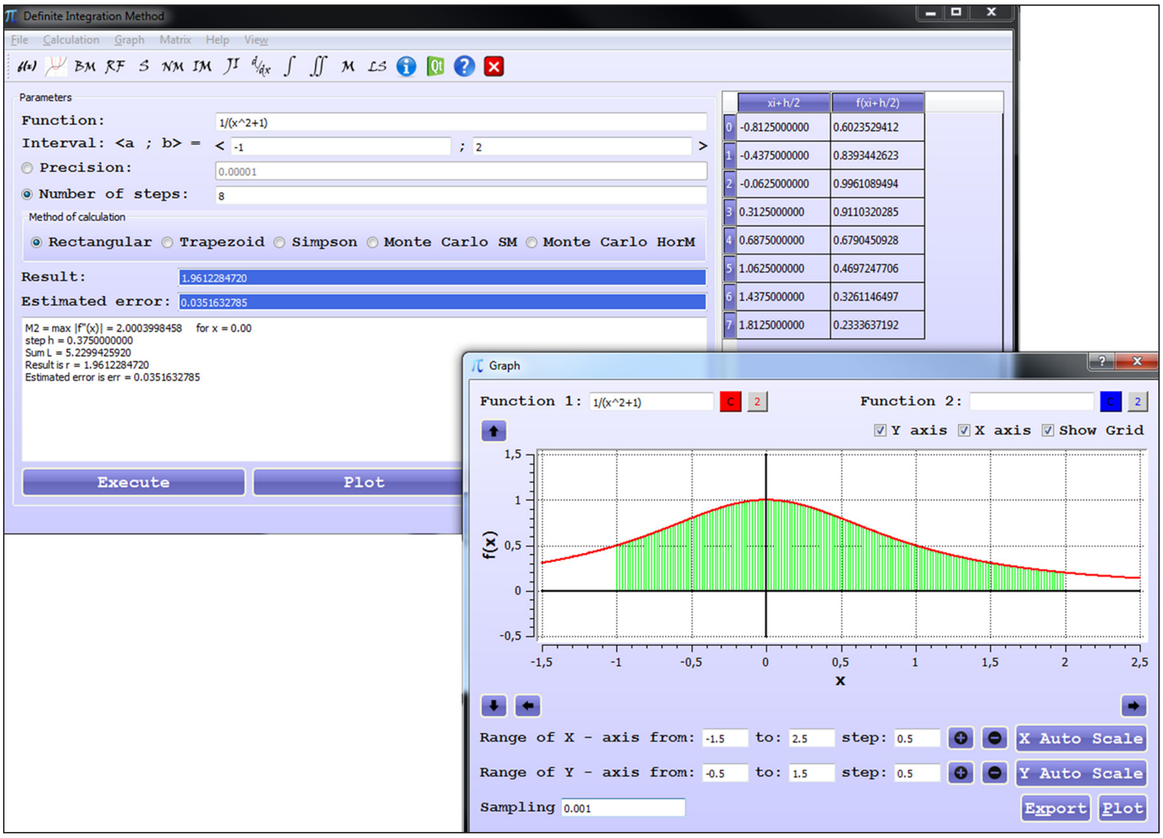
All numerical methods are implemented in a way of numerical analysis, to study of algorithms that use numerical approximation for the numerical analysis problems with arbitrary-precision arithmetic. The goal of numerical analysis is to give approximate but accurate solutions to hard problems. Iterative methods are not expected to terminate in a finite number of steps. Starting from initial guess, iterative methods form successive approximations that converge to the exact solution only in the limit. Convergence test is specified in order to decide when a sufficiently accurate solution has been found. Examples include Bisection method, Newton's method or Jacobi iteration method. MATH implements more of them. These iterative methods are generally needed for large problems. The regression takes into account that data are imprecise, and given some points (mainly measured data) we determine the unknown function in a way of minimalization of squared distance between point and function (Least mean square – LMS method). Numerical differentiation is a process of finding the numerical value of a derivative of a given function at a given point. MATH supports numerical differentiation up to 4. degree so far, and are implemented in a way of defining limit function as follows:

$$f' \equiv \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

With raising degree of numerical differentiation, the limit function gets slightly more difficult. More detailed implementation of each separate numerical method and how to use it can be found in MATH documentation at: http://sourceforge.net/projects/nummath/files/Math_documentation/

Of course, one can use already created, well documented and robust libraries for scientific applications like SciPy, Maxima, Octave, Matlab, and many others, but these libraries or applications offers only final results and plotting.

Main goal of this application is to help understand (not explain) how those numerical methods work. That means, MATH implements the exact procedure and steps as one would do it in the process of learning. The main reason, why was MATH developed is to allow comparing the whole procedure with final result and graphs needed to



Function: 1/(x^2+1)

Interval: <a ; b> = < -1 ; 2 >

○ Precision: 0.00001

● Number of steps: 8

● Rectangular ○ Trapezoid ○ Simpson ○ Monte Carlo SM ○ Monte Carlo HorM

Result: 1.9612284720

Estimated error: 0.0351632785

```
M2 = max |f''(x)| = 2.0003998458    for x = 0.00
step h = 0.3750000000
Sum L = 5.2299425920
Result is r = 1.9612284720
Estimated error is err = 0.0351632785
```

	xi+h/2	f(xi+h/2)
0	-0.8125000000	0.6023529412
1	-0.4375000000	0.8393442623
2	-0.0625000000	0.9961089494
3	0.3125000000	0.9110320285
4	0.6875000000	0.6790450928
5	1.0625000000	0.4697247706
6	1.4375000000	0.3261146497
7	1.8125000000	0.2333637192

Function 1: 1/(x^2+1) Function 2:

☑ Y axis ☑ X axis ☑ Show Grid

Range of X – axis from: -1.5 to: 2.5 step: 0.5

Range of Y – axis from: -0.5 to: 1.5 step: 0.5

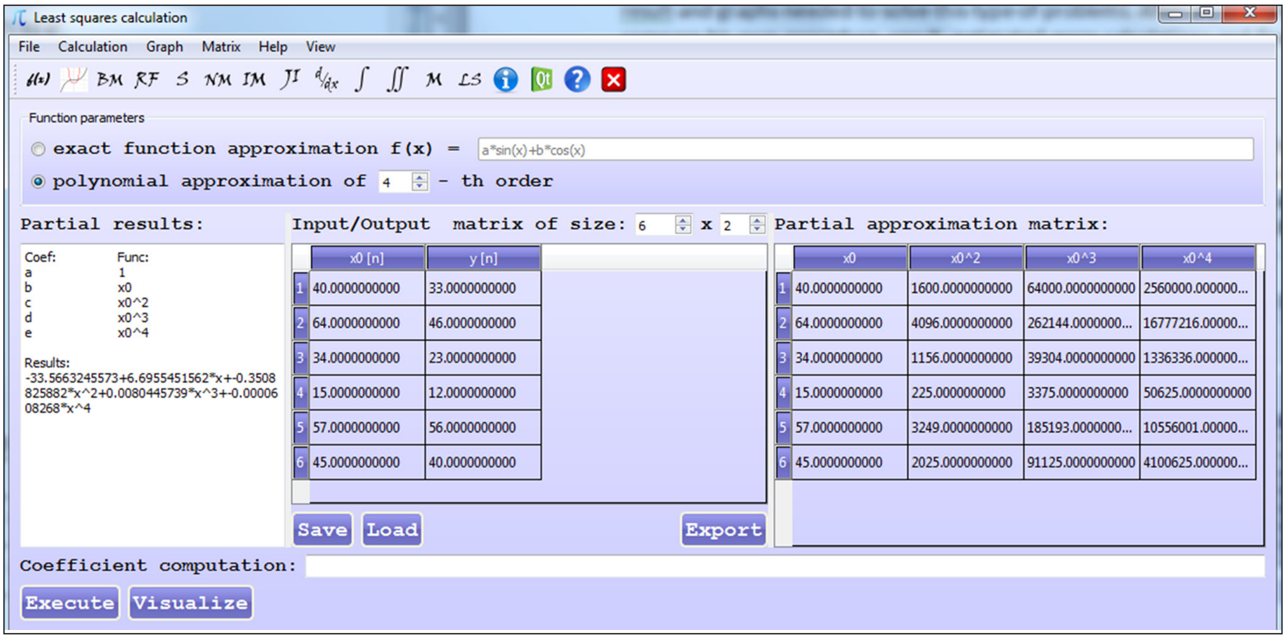
Sampling 0.001

Figure 1: MATH graphical user interface and its main components.

solve this type of problems. After all, student can compare his own procedure, result, estimated error calculations and find out if a mistake was made and if, where exactly.

Example of use (LMS – Least Mean Square method): For approximation of measured data with polynomial of 4th order, one has to choose the LS method from toolbar, then select "polynomial approximation" and set 4 as the order of approximation. In the input/output matrix should be written or loaded desired data and then executed, what leads to approximation matrix and after visualization, one can see the data approximated as selected.

On **Fig. 2** one can see the implementation of LMS method and its output in graphical form on **Fig. 3**. Each of implemented method work in a similar way. With this



Function parameters

○ exact function approximation f(x) = a*sin(x)+b*cos(x)

● polynomial approximation of 4 - th order

Partial results:

Coef: Func:
a 1
b x0
c x0^2
d x0^3
e x0^4

Results:
-33.5663245573+6.6955451562*x+-0.3508825882*x^2+0.0080445739*x^3+-0.0000608268*x^4

Input/Output matrix of size: 6 x 2

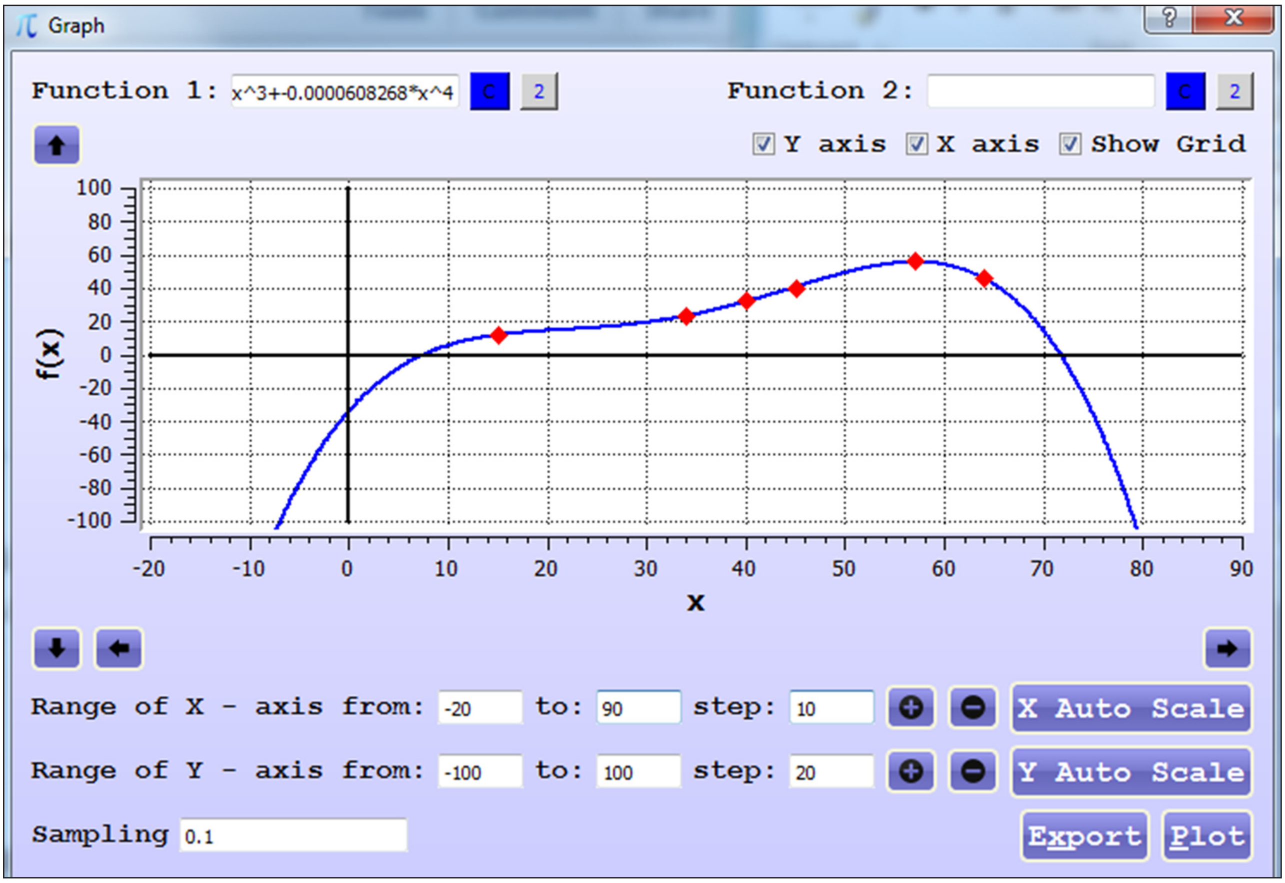
	x0 [n]	y [n]
1	40.0000000000	33.0000000000
2	64.0000000000	46.0000000000
3	34.0000000000	23.0000000000
4	15.0000000000	12.0000000000
5	57.0000000000	56.0000000000
6	45.0000000000	40.0000000000

Partial approximation matrix:

	x0	x0^2	x0^3	x0^4
1	40.0000000000	1600.0000000000	64000.0000000000	2560000.000000...
2	64.0000000000	4096.0000000000	262144.0000000...	16777216.00000...
3	34.0000000000	1156.0000000000	39304.0000000000	1336336.000000...
4	15.0000000000	225.0000000000	3375.0000000000	50625.0000000000
5	57.0000000000	3249.0000000000	185193.0000000...	10556001.00000...
6	45.0000000000	2025.0000000000	91125.0000000000	4100625.000000...

Coefficient computation:

Figure 2: MATH graphical user interface for LMS method.



Function 1: x^3+-0.0000608268*x^4

Function 2:

☑ Y axis ☑ X axis ☑ Show Grid

Range of X - axis from: -20 to: 90 step: 10

Range of Y - axis from: -100 to: 100 step: 20

Sampling 0.1

Figure 3: MATH graph output for LMS method.

output, students can compare their partial and final results with MATH results to ensure the correct procedure of calculation.

Quality control

Functionality has been tested on Microsoft Windows (7 and XP) where everything worked as expected. Qt provides also easy cross – platform compatibility, and MATH can be compiled on Linux or MacOS as well.

Before publishing a trial run with students from Technical University was performed using input data (problems to solve) from lectures of Applied Mathematics in order to test its usability and detect any bugs. All final results were compared with results from Matlab to ensure the precision of all used methods, as Matlab includes many more advanced methods with higher precision outputs. With Matlab we performed tests of all numerical methods available in MATH also for limit values that could cause problems based on the principle of individual algorithms methods. These tests verify that all methods work correctly with desired precision. Although various limit cases were tested, there still may occur some bugs considering special cases. For such occasion, there is support available through mail or MATH website.

(2) Availability

Operating system

MATH has been tested on Microsoft Windows (7 and XP). On Linux it can run using Wine tool.

Programming language

Qt 4.8.1 (should work also with Qt 5 – depends on addition libraries – currently untested).

Additional system requirements

N/A.

Dependencies

– Qwt 6.1.0.
– MuParser 2.2.3.
– Gsl 1.8.
– MinGW 4.8.1.

List of contributors

– Henrich Glaser – Opitz: coding, testing.
– Kristína Budajová: testing.
– Leonard Glaser – Opitz: artwork.

Software location:

Archive (e.g. institutional repository, general repository) (required)

Name: SourceForge
Persistent identifier: http://sourceforge.net/projects/nummath/files
Licence: GNU General Public Licence version 3
Publisher: Henrich Glaser – Opitz
Date published: 25/05/2014

Code repository (e.g. SourceForge, GitHub etc.) (optional)

Name: SourceForge
Identifier: http://sourceforge.net/p/nummath/code/ci/master/tree/
Licence: GNU General Public Licence version 3
Date published: 17/07/2014

Language

English.

(3) Reuse potential

MATH as an application provides easy to use tool for calculating roots of nonlinear equations, roots of system of linear equations, differentiate, integrate, approximation, matrix calculation using mentioned numerical methods for calculating results, estimated errors and much more. MATH is completed with graphical visualization capable of producing publication-quality figures. Future development will be aimed on expanding new numerical methods and graphical capabilities. Code can be expanded in an easy way, because application is constructed in a way where each method has separate source code which is then implemented in mainwindow source code.

In order to build from source code, the mingw compiler must be used, following the instructions at: http://sourceforge.net/p/nummath/code/ci/master/tree/README.txt.

Application has already been implemented into education process on Technical University Košice and we are trying to spread it further away. So far there has been published publication with examples from numerical mathematics using MATH as a tool [5]. MATH's reuse potential lies in every university where undergraduate or graduate students need to understand how all these numerical methods work from a fundamental level and not just how to use them. Engineers, computer scientists and others often need to know which method will be most effective.

With MATH it is possible to compare their performance very quickly and easy. Graph dialog window can be used also as an individual tool, which is providing plotting options as any other plotting tool that can and will be extended throughout further development process.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgements

In this place, author would like to thank his colleagues Kristína Budajová and Štefan Berežný for their support, many advices and help with debugging the application.

References

1. **Qt documentation.** Available at http://qt-project.org/doc/ (Last accessed 01 January 2016).
2. **MuParser documentation (fast math parser library).** Available at http://muparser.beltoforion.de (Last accessed 01 January 2016).
3. **Gsl documentation (Numerical library).** Available at http://www.gnu.org/software/gsl/ (Last accessed 01 January 2016).
4. **Qwt documentation (Qt widget for technical applications).** Available at http://qwt.sourceforge.net (Last accessed 01 January 2016).

5. **Budajová, K** and **Glaser-Opitz, H** 2014 *Solved examples from numerical mathematics using MATH.* Technical University Košice. ISBN: 978-80-553-1727-4 Available at http://web.tuke.sk/lf-kas/ucitelia/budajova/subor_riesenych_prikladov_z_numerickej%20_matematiky.pdf (Last accessed 01 January 2016).
6. **Young, T** and **Mohlenkamp, M J** 2014 (September) *Introduction to Numerical Methods and Matlab Programming for Engineers.* Department of Mathematics, Ohio University.
7. **Hamming, R W** 1962m *Numerical Methods for Scientists and Engineers.* Second edition. ISBN: 0-486-65241-6.
8. **Bewley, T R** *Numerical Methods in Science and Engineering.* UC San Diego.
9. **Riccardo, S** 2013 (January) *Numerical Methods for Civil Engineering.* Milano.

How to cite this article: Glaser-Opitz, H and Budajová, K 2016 MATH: A Scientific Tool for Numerical Methods Calculation and Visualization. *Journal of Open Research Software,* 4: e6, DOI: http://dx.doi.org/10.5334/jors.55

Submitted: 04 August 2014 **Accepted:** 13 January 2016 **Published:** 10 February 2016

]u[*Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS