Journal of Advances and Scholarly Researches in Allied Education
Vol. 16, Issue No. 5, April-2019, ISSN 2230-7540

Graphical Representation of Mathematical Equations Using Open Source Software

Dr. Yogeesh N.*

Assistant Professor of Mathematics, Government First Grade College, Tumkur, Karnataka, India

Abstract – It is proverbial in arithmetic exploration that all means of a contention or verification are open to investigation. Nonetheless, a proof put together even to some extent with respect to business software is difficult to survey, on the grounds that the source code—and now and then even the calculation utilized - may not be made accessible. There is the further issue that a peruser of the confirmation will most likely be unable to check the writer's work except if the peruser approaches a similar software. Therefore open-source software frameworks have consistently partaken in some utilization by mathematicians, yet as of late have frameworks of adequate force and profundity become accessible which can contend with—and at times even outperform—business frameworks. Mathematicians and science teachers might incline toward business frameworks part of the way in light of the fact that such frameworks are better advertised, yet in addition in the view that they might partake in some degree of help. Yet, this comes at the expense of starting buy, in addition to yearly authorizing charges. The present status of tertiary financing in a large part of the world implies that for everything except the extremely top level of colleges, the cost of such frameworks is more enthusiastically to legitimize. For teachers, an issue is making the framework accessible to understudies: it is realized that understudies get the most use from a framework when they have unhindered admittance to it: at home just as at their establishment. Once more, the utilization of an open-source framework makes it paltry to give access.

Keywords – Graphical, Mathematical, Software

- X -

INTRODUCTION

It have been utilizing numerical software apparatuses for a very long while at this point, beginning with handheld mini-computers as an understudy, and moving into PC polynomial math frameworks, dynamic calculation delicate product, mathematical frameworks and online evaluation frameworks. Table 1 records the greater part of the software I have utilized, either for myself, or with my understudies:

Table 1: Software used by the author

Computer Algebra Systems	Dynamic Geometry Software	Numeric Software	Assessment Systems	CAS Calculators
Axiom	GeoGebra	GNU Octave	MAA WeBWorK	Casio ClassPad
Derive		Matlab®	Pearson MyMathLab®	HP Prime
Maple®		Scilab	Wiley Assist®	TI-nspire CAS
Mathematica®		Julia		
Maxima		C++		
MuPAD®		Python		
Pari/GP				
Sage				

A portion of these frameworks are business, some are open source, and others have another permit. For the reasons for this article, we make the accompanying differentiations:

Business Software (or shut source software) will be software disseminated by the designers in executable structure just, and for which admittance to a full and unhindered variant requires the client to pay.

Open-source software is disseminated for nothing, and with the total source code, which the client can alter voluntarily. The most outrageous open source permit is "GNU CopyLeft" which guarantees that not exclusively is the first software free, however so will be any further adjustments.

Free exclusive software some software (like Geogebra) is free "for non-business use", yet requires installment for business use.

There are numerous varieties and shades of dim in the adaptability and openness of software licenses, yet for the motivations behind this paper I will consider GNU Copylefted software, yet free exclusive software: essentially any software which is accessible free of charge with its source code, and which has a permit which permits the client to utilize a full and unhindered rendition. The term FOSS for Free and Open Source Software is quite utilized, albeit a few scholars guarantee that there is a philosophical distinction between "free" which should be totally unhindered, and "open source" which could conceivably have a few limitations. As an issue of basic logic, I won't be brought into this discussion.

I began utilizing Maple in probably the most punctual adaptation, and furthermore Mathematical not long get-togethers, since I approached them. An early investigation at my college, in which I was a member, yet not the pioneer, was to utilize Mathematical in our first year classes. This flopped essentially on the grounds that the lead experimenter was excessively energetic, and expected a lot of the understudies, yet of the other staff. Sometime later we began with Derive, with painstakingly scaffold activities and lab sheets, and which the understudies appeared to appreciate. We covered this at that point. Sometime later we moved to Maple with a site permit, which we and the understudies for the most part appreciated. It was finding then a significant issue: my eagerness will in general have me ask a lot of the understudies, so that as opposed to being urged to investigate they become overpowered by every one of the new orders and their boundaries. Later the site permit turned out to be excessively costly; this was a period of low understudy numbers in science, so we moved two or three lab licenses and individual licenses, and later, when the University was going through one of its numerous hierarchical clean-ups, we lost those, as well. A third year subject in cryptography which I'd instructed at first with Maple required new software, so I went through a year exploring different avenues regarding Maxima and Axiom. Inquisitively, despite Axiom (under MS Windows) having just a content based interface, the understudies didn't appear to mind.

As Sage developed I began moving towards it, and throughout the previous few years of this present subject's presence we utilized it only. We are presently utilizing CAS Calculators (TI-nspire CAS and Casio ClassPad) as in my home province of Victoria, Australia, the utilization of such adding machines is commanded in cutting edge auxiliary arithmetic, so understudies show up at the University as of now with some knowledge of their utilization. We are utilizing them in both first year subjects and furthermore in a third year subject in mathematical strategies. Albeit not free, such adding machines are surprisingly amazing, and in addition can go with the understudies anyplace. Simultaneously, we have been exploring different avenues regarding on the web appraisal frameworks, beginning with Pearson MyMathLab, then, at that point Wiley Assist, and presently MAA WeBWorK. Both business frameworks have their shortcomings: Pearson required a permit which was just substantial for one year—this was an issue as numerous understudies require year and a half or more to finish their two center units of math; and Wiley was connected to a solitary reading material, which implied that any deviation from the course book would not be upheld. Additionally, their composing frameworks appeared to be extremely muddled and antagonistic. Be that as it may, they had exceptionally attractive and all around planned UIs.

OBJECTIVE

1. To study graphical representation of mathematical open source software

Why use open source?

Albeit the underlying expense of open source software (zero!) is now and then seen as its greatest benefit, this must be offset with the expenses of organization, upkeep, upkeep and overhauls, investigating, and backing. In a huge climate, like a college, the software should be either introduced on all lab PCs, just as on staff PCs, or on a focal worker. There will be "covered up costs" (in help and support) over the existence pattern of the software. Open-source software likely could be modest to introduce, however it is no less expensive to run, and in light of the fact that there will be no help other than client discussions, will require nearby time and work to manage any issues which emerge. That being said, there are still amazing and convincing motivations to think about open-source software:

1. No merchant lock-in. Lock-in can be deceptive: you discover you become increasingly more reliant upon software or assistance, to where it is practically difficult to change. And afterward just as the underlying expenses, there are yearly authorizing costs, just as perhaps additional expenses for redesigns, expansions, or bundles.

2. Known bugs. Arithmetic software is intricate and muddled, and clients can put extraordinary requests on it. No software is bug-free2 except for a discipline requiring precision and exactness a bug in math, the software can be unfortunate. A new model including business software [4] has gotten significant consideration; no one knows the bugs in any business framework since their organizations don't advance them. The clients just need to believe that the appropriate responses they are getting will be correct3. Open-source software engineers will keep a freely available information base of known bugs.

3. Correspondence. A lot of numerical composing now, in schooling as in research, will include some code tests. I accept that this is a significant issue. A conversation about another approach to show a specific point, like demonstrating,

can be utilized by any instructor, following the statutes and thoughts introduced by the creators. A decent late record of simply such a methodology is given by Wendelin et al [12]; the article examines teaching method, course plan, and instances of issues. However, in the event that the writers choose rather to depict how a PC framework is utilized, and on the off chance that they utilize a business framework, the readership is essentially restricted to those with admittance to a similar framework.

The issue is exacerbated in research when a paper vigorously relies upon the utilization of the software. Jacob Neub¨user, the underlying maker of the GAP framework for bunch hypothesis, guaranteed in 1993:

"You can peruse Sylow's Theorem and its verification in Huppert's book in the library without purchasing the book and afterward you can utilize Sylow's Theorem for the remainder of your life for nothing, yet . . . for some PC variable based math frameworks permit expenses must be paid consistently for the absolute season of their utilization. To ensure what you pay for, you don't get the source, however just an executable, for example a black box. You can squeeze catches and you find solutions similarly as you get the splendid pictures from your TV however you can't handle how they were put forth in one or the other defense. With the present circumstance, two of the most essential guidelines of direct in math are disregarded. In math, data is passed on for nothing and everything is exposed for checking. Not having any significant bearing these guidelines to PC polynomial math frameworks that are made for numerical exploration [...] implies moving a most unwanted way. Generally significant: Can we anticipate that somebody should accept an aftereffect of a program that he isn't permitted to see?"

Individuals who are sharing thoughts, either on paper, or straightforwardly, should have a shared view with which to convey, and this incorporates a concurred PC framework as much as a typical language. Not very many individuals or establishments can bear the cost of the expenses of buying and keeping a few diverse business frameworks

Graphical Representation

There has been a change as of late in math instructors' perspectives on the job of drawn representations. As introduced in Monk (2003), charts can be seen in two particular manners. To begin with, and all the more generally, a diagram is a device for correspondence. That is, diagrams depict a bunch of information or an answer of an issue to the peruser. Notwithstanding, Monk presents the thought that there is a subsequent method to utilize diagrams – as apparatuses for producing meaning. Priest explains saying, "Though a chart had before been seen only as a conductor, a transporter of data, for instance, about the movement of a vehicle, it can now likewise be viewed as a focal point through which to investigate that movement." (p. 251, accentuation in the first). Priest keeps on bringing up that these are not alternate extremes, nor is one ideal, rather that they are two distinct ways to deal with utilizing devices that appear to be identical. Steady with Inter Math's objectives and vision, it was normal that members would utilize charts (and other visual representations) in both of these ways. Further, it had been expected that the members were utilizing the representations as critical thinking devices since that was the methodology demonstrated for them in the course.

All the more explicitly, by utilizing visual representations as critical thinking devices, members would have the option to see a few advantages – especially in their capacities to tackle the sorts of complex issues they were regularly confronted with in Inter Math. Predictable with Monk's perspectives, the Inter Math group thought about various advantages to utilizing charts and realistic components thusly. These included

- Using illustrations to investigate parts of a setting that may some way or another not be clear;
- Fostering a more profound comprehension of a setting using illustrations that inspire specific inquiries regarding those unique situations; and
- Developing a more profound comprehension of the sorts of data that can be passed on through illustrations (Monk, 2003).

Furthermore, expanding on Gagatsis and Shiakalli (2004), we attest that teachers should have the option to work with these representations bothly – as imparting and critical thinking. While Gagatsis and Shiakalli were more worried about moving between representations, their point applies to InterMath educator members. That is, deciphering among representations and inside representation frameworks is an imperative part of educating. In the event that an educator can't decipher a realistic representation that has been created by her understudies, she or he has lost one method of figuring out (a) regardless of whether the understudy comprehends an idea and (b) where the understudy might in any case require extra help in refining their agreement. In their statement that understudies frequently need nonstandard representations to help their numerical critical thinking, Greeno and Hall (1997) featured this requirement for instructor improvement much more. In case instructors are to completely uphold their understudies, they should have the option to see how understudies are utilizing graphical components to clarify their answers as well as to take care of issues.

We accept that InterMath furnishes members with freedoms to foster these sorts of miens toward graphical representations just as to refine their capacity to decipher a wide scope of representations. While the examinations and advancements utilized in InterMath do innately uphold more customary types of representation, they additionally advance different types of representation. In classes, members are supported, however not needed, to utilize at least one advances for their examinations; InterMath educators regularly exhibited a few unique ways to deal with tackling the examinations, each with their own utilization of representations. Members encountered similar sorts of educating and learning openings we trust they will create for their understudies.

It is our view that the utilization of graphical components ought to enormously upgrade the issue solver's capacity to effectively finish an examination. However in our example, this was not really evident. This investigation, consequently, thinks about why instructors who were utilizing at least one visual representations in their reviews utilized numerically improper methodologies as well as found wrong solutions. For the motivations behind this examination, we think about the accompanying inquiries: How did members utilize graphical representations in their critical thinking measures? How did the graphical representations permit the members to wander from right or fitting numerical methodologies as well as neglect to arrive at right or proper arrangements?

Graphics and Interfaces

These frameworks support designs in two and three measurements: capacities characterized unequivocally, verifiably, parametrically, unique shapes, for example, polyhedra, diagrams and organizations, and all with some kind of intelligence. So the client can move a three dimensional shape going to acquire the best perspective on it, zoom in and out, change tones, change delivering. Figure 1 shows instances of illustrations created in every one of the frameworks.

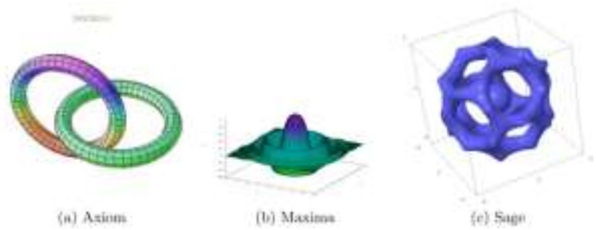

Figure 1: Graphics examples

One of the impediments of open source frameworks has been in the interfaces: rich publishable journal interfaces with skimming ranges don't exist in the open-source world. Nonetheless, there have been late advances: Maxima have since a long time ago had its wxMaxima interface, which runs under a wide range of working frameworks, and both Sage and presently FriCAS can run in a program utilizing the iPython framework. Figure 2 shows instances of the interfaces of every framework. Furthermore, obviously everything frameworks can run in a control center, without illustrations, and without typeset yield.

Numeric Software

In this section we shall briefly investigate numeric software. The standard commercial offering is Matlab R which is beloved of engineers the world over, with Mathcad R a close second.

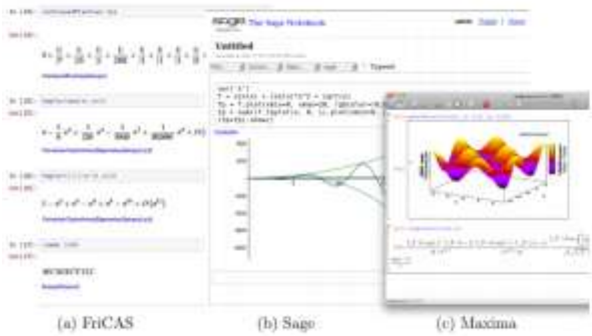

Figure 2: Interfaces

The Two primary open-source competitors are GNU Octave and Scilab. GNU Octave is intended to be Matlab-viable, for certain little contrasts. Projects written in Matlab, as long as they don't depend on explicit extra tool stash, should run with almost no alteration on Octave. Scilab isn't so worried about similarity, albeit quite a bit of its punctuation is like that of Matlab. Scilab likewise accompanies a graphical proofreader called Xcos to plan and mimic dynamical frameworks; comparable here and there to Matlab's Simulink. Octave doesn't have a particularly graphical subsystem, albeit the majority of this reproduction can be accomplished by different means. To show the force of these frameworks, we will tackle a straightforward mathematical issue: to fit the SIR model of infection spread to information of a flu flare-up in an English school. This is a notable contextual analysis; the quantity of tainted understudies from the very first moment to 14 of the flare-up was

3, 6, 25, 73, 222, 294, 258, 237, 191, 125, 69, 27, 11, 4

With no deaths Thus the total population remained constant, and so the disease model:

$$\frac{dS}{dt} = -\beta IS, \ \frac{dI}{dt} = \beta IS - \gamma I, \ \frac{dR}{dt} = \gamma I$$

where S, I, R are the quantity of helpless, tainted, and recuperated people separately, should fit the information, for fittingly picked upsides of β and γ. With Octave, the initial step is to make a capacity to display the differential conditions:

Function xdot = f (x, t)

Global B;

Global G;

xdot = zeros(3,1);

xdot(1) = -B*x(1)*x(2);

xdot(2) = B*x(1)*x(2)-G*x(2);

xdot(3) = G*x(2);

End function

Then, at that point the lsode capacity can be utilized to give a mathematical arrangement. For instance, we will utilize the boundaries β = 0.01 and γ = 0.1 (which we will allude to as B and G):

```
octave:  t = linspace(0,14,127);
octave:  data = [ 6 25 73 222 294 258 237 191 125 69 27 11 4];
octave:  B = 0.01; G = 0.1;
octave:  y = lsode("f",[760;3;0],t);
octave:  S = y(:,1);I=y(:,2),R=y(:,3);
octave:  plot(t,S,"b","linewidth",2,t,I,"g","linewidth",2,...
> t,R,"r","linewidth",2,[1:14],data,"*k","markersize",10)
```

The plot is displayed on the left in figure 3. Note that the green bend—addressing the contaminated numbers, is an extremely helpless fit for the real information.

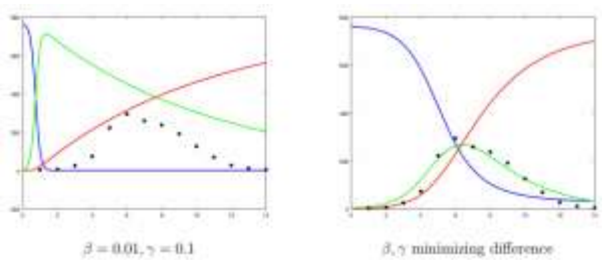

Figure 3: The SIR model with different parameters

To discover the boundaries that best fit the information, first we need a capacity which creates an amount of squares between the information and the registered I esteems:

```
function out = ss(b)
    global B;
    global G;
    B = b(1);G=b(2);
    t = linspace(0,14,127);
    y = lsode("f",[760;3;0],t);
    data = [3 6 25 73 222 294 258 237 191 125 69 27 11 4]';
    out = sum((data-y(10:9:127,2)).^2);
endfunction
```

Now we can use the nelder mead min function from Octave's "Optim" package: octave: nelder_mead_min(@(x) ss(x),[0.001;0.001])

ans =

0.0018868

0.4192210

Presently if these boundaries are utilized for β and γ in the model, the subsequent chart is displayed on the right in figure 3, and the bend addressing the tainted numbers is an awesome fit to the underlying information. The projects and orders for Scilab are almost indistinguishable. First the projects:

```
function ydot = f(t,y)
  global('B', 'G')
  ydot(1) = -B*y(1)*y(2);
  ydot(2) = B*y(1)*y(2)-G*y(2)
  ydot(3) = G*y(2);
endfunction
```

```
function out = ss(b)
  global('B','G')
  B = b(1);G=b(2);
  t = linspace(0,14,127);
  x = ode([760;3;0],[0;0;0],t,f);
  data = [3 6 25 73 222 294 258 237 191 125 69...
  27 11 4];
  out = sum((data-x(2,10:9:127)).^2);
endfunction
```

and the commands used:

```
-->t = linspace(0,14,127)
-->B=0.01; G=0.1;
-->x = ode([760;3;0],[0;0;0],t,f);
-->data = [3 6 25 73 222 294 258 237 191 125 69 27 11 4]
-->plot(t,x(1,:),'r',t,x(2,:),'g',t,x(3,:),'b',[1:14],data,'.k')
-->fminsearch(ss,[0.001;0.001])
 ans =
```

0.0018869 0.4192225

There are some phenomenal conversations and examinations of numeric instruments which test both free and business software against an assortment of mathematical and computational issues.

CONCLUSION

In this short article we have just start to expose the open source world, and took a gander at a couple of items. We have not addressed powerful math software, of which Geogebra, C.A.R/CarMetal and Cinderella are the foremost current free contributions. In every one of the three regions we talked about: PC polynomial math frameworks, numeric software, evaluation, there are numerous different items. There are additionally programming dialects intended for specific numerical use, or language libraries, like SymPy for Python (which incorporates phenomenal emblematic preparing, just as math), and furthermore for Python the numeric and logical libraries SciPy and NumPy. Clients of C or C++ can utilize the GNU Multiple Precision Library (which has coverings for different dialects), or PARI. Julia is another dialect intended to have the force of Matlab and the speed of C. Accordingly the client is ruined for decision. Except if there are quite certain prerequisites which must be met by a business framework, I see no requirement for math teachers not to emphatically uphold open source software. Sharing some software practically speaking implies that it is far simpler to share thoughts; regardless of whether instructing or research. Notwithstanding, on the grounds that there are frequently various items to pick it doesn't really imply that any one individual will be knowledgeable about more than one. Notwithstanding, eliminating the value component implies that downloading (and here and there not even that) and testing should be possible at no expense. This isn't the situation with material including business software.

REFERENCES

1. Eliana S de Almeida, Antonio C Medeiros, and Alejandro C Frery. “How good are MatLab, Octave and Scilab for computational modeling?” In: Computational & Applied Mathematics 31.3 (2012), pp. 523–538.

2. Kevin Charlwood. “Integration on Computer Algebra Systems”. In: Electronic Journal of Mathematics and Technology (2008).

3. Wolfram Decker et al. Singular 4-0-2 — A computer algebra system for polynomial computations. http://www.singular.uni-kl.de. 2015.

4. Antonio J Dur´an, Mario P´erez, and Juan L Varona. “The Misfortunes of a Trio of Mathematicians Using Computer Algebra Systems. Can We Trust in Them?” In: Notices of the AMS 61.10 (2014).

5. Bur¸cin Er¨ocal and William Stein. “The Sage Project: Unifying free mathematical software to create a viable alternative to Magma, Maple, Mathematica and MATLAB”. In: Mathematical Software–ICMS 2010. Springer, 2010, pp. 12–27.

6. Themistoklis Glavelis, Nikolaos Ploskas, and Nikolaos Samaras. “A computational evaluation of some free mathematical software for scientific computing”. In: Journal of Computational Science 1.3 (2010), pp. 150–158.

7. Donald Ervin Knuth. “Literate programming”. In: The Computer Journal 27.2 (1984), pp. 97–111.

8. Alasdair McAndrew et al. “Using a computer algebra system to facilitate the transition from secondary to tertiary mathematics”. In: World Conference on Computers in Education VI Abstracts. 1995, p. 287.

9. Chris Sangwin. Computer Aided Assessment of Mathematics. OUP Oxford, 2013.

10. Neeraj Sharma and Matthias K. Gobbert. A comparative evaluation of Matlab, Octave, FreeMat, and Scilab for research and teaching. Tech. rep. available at http : / / ciep . itam.mx/~rtorres/progdin/comparative-evaluation-of-matlab-octave-scilabfreemat.pdf. Department of Mathematics & Statistics, University of Maryland, 2010.

11. Richard Stallman. Why Open Source misses the point of Free Software. Accessed September 11, 2015. 2015. url: https://www.gnu.org/philosophy/open-source-missesthe-point.html.

Corresponding Author

Dr. Yogeesh N.*

Assistant Professor of Mathematics, Government First Grade College, Tumkur, Karnataka, India