

Experiences in using Sage to integrate good study habits and problem solving techniques in engineering mathematics

Linda Fahlberg-Stojanovska, Vitomir Stojanovski, Tim Fahlberg

Faculty of Technical Sciences, University St. Clement of Ohrid, Bitola, FYR Macedonia

E-mail: lfahlberg@gmail.com

Abstract

In this paper, we discuss informally some practical experiences and results of getting students to use Sage while studying and solving problems in engineering mathematics within an existing curriculum and with minimum effort on our part. We had no lofty goals. We were simply trying to see if technology now existed for our students to effectively and efficiently check their by-hand solutions to problems in single and multivariable calculus in order to help them pass written exams. With Sage, our students could quickly and freely start by just checking calculations and/or create interactive 3D plots of all types to help them visualize and explore. This did seem to help them with their exams. And we found that encouraging students to use this method of study also helped them to think about the process of problem solving in an environment particularly well adapted for students of engineering. We explain by detailing an example.

2000 Mathematics Subject Classification. 97D40. 97D50.

Keywords. Sage, education, calculus, technology, STEM.

1 Introduction

To state an obvious, as teachers of engineering, we must teach both knowledge and thinking skills. No one disputes that our students need to understand the physical processes taking place in the problems we give them. But they also need to learn the art of "problem solving". There are multitudes of studies, plans and programs about how and why to do this. We are not discussing that here.

Instead, our goal was deeply practical and basic. Find good, free, reasonably easy-to-use tools for our students to help them work through problems while studying for exams in ways that would improve their total learning experience (including passing the exams). That is, our goal was to get our students to actively use free, instant feedback internet technology tools in constructive ways while studying.

What were our parameters? Can we teach them how to quickly check their results so that they know they are on the right track as they progress through the solution process? Can we teach them how to use technology to see that they have either misunderstood a technique or they are trying to use it improperly? Can we teach them how to creatively, visually try something when they get stuck and so be able to keep going and not just give up in frustration?

Equally as important - can we do this effectively and immediately within our current teaching and assessment system?

We point out that we have been encouraging our students (and fellow teachers) for years to do this in lower level mathematics using e.g. GeoGebra [1] and WolframAlpha, both freeware

technologies that are exceptionally useful. Quick example - instead of telling students to check their solution of a 2x2 linear system of equations by hand (which they will rarely do just because it is so boring and fraught with the possibility of even more mathematical errors), we have them quickly enter the equations in GeoGebra and check the solution visually. They still have to know how to solve by hand and get the right answer for the exam, but now they have instant feedback on their skills. It takes the teacher very little time to explain this process; it requires no change at all in the curriculum. It takes the student very little time to do and it seems to improve their problem solving and thinking skills. (For example, try changing the variable names in the original problem and see if they can adapt this in say GeoGebra. Aha! Do the names of variables matter?) However, many challenges have contributed to not being able to generalize this practice and methodology to higher level mathematics and to the mathematics used inside of engineering courses themselves. With Sage, we found that both we and our students could make significant progress in this direction. We would like to make clear that many of the suggestions here will work in other environments [2] and with other applications, particularly the major commercial applications. It is indeed nice that Sage is freeware. But Sage also does seem to have an environment particularly well adapted to the nature of engineering mathematics and this again helps us as mathematicians communicate with our engineering students.

We discuss our experiences below.

2 Problem Solving in Engineering

2.1 Structure of Problems in Engineering and Problem Solving

Typically, problems in engineering

- can be stated in many different equivalent notations,
- use mathematically interrelated, but substantively different techniques,
- require several different levels of understanding in mathematics and science and
- cannot be solved directly using technology (plug and chug).

Here are some easy-to-understand first-year problems of this nature that we that could not easily or adequately translate into mathematical applications.

1. Find the length of the closed curve defined by the polar function $r = 5 \sin(\varphi)$.
2. Define the line segment from point A(1,0,2) to point B(2,4,1).
3. Explain the meaning of and then calculate the work done by the constant force $F=kx$ along the distance $x \in [0, m]$ in a way that is extendible to a 2D or 3D force.
4. Use integration to find the volume inside the sphere $x^2 + y^2 + z^2 = 3$.

Before reading further, we invite you to be a student studying for an exam and try to solve and check your solutions to these problems (with or without technology). What problems might he encounter? For example, in the first problem - what is the interval of integration; in the second problem - what is the equation of a line in 3D; in the third problem - how do they understand work - is it really the area under the curve, in the fourth problem - do we need a parameterization of the surface or a change of variables?

2.2 Our Sample Problem

Let C^+ be that part of the positively oriented curve that is the intersection of the surfaces: $\frac{x^2}{4} + \frac{y^2}{9} = 1$ and $z = 1$ in the first octant. Find the work done by the force $\vec{F} = (xy, 3z, 1)$ on C^+ .

Notice that we have deliberately made this an exam problem without hints. We did not describe the surfaces (elliptical cylinder or plane). We did not say "Solve the line integral of work type" and put everything in standard P,Q,R format for direct application of the line integral formula. Our experience has been that this type of "helping" often happens in the classroom (where we so very often give verbal if not actual written hints), but never on exams. [3] And students go crazy.

Also, this problem cannot be solved directly (plug and chug) just using technology since we did not give the vector-parametric form of the curve (which given their magnificent power would allow a simple single command line solution in say MatLab or Mathematica, but not however in Sage).

On the other hand, in our opinion, this problem cannot be well understood by the typical engineering (or mathematics) student without technology. We make this statement baldly based on our own experiences as students in the classroom, our many, many years of teaching without technology and now our experiences in integrating technology into education. There is just too much going on. The problem requires physics, 3D geometry with multivariable, vector-parametric functions, differential geometry, multiple coordinate systems and multivariable calculus.

And so the student sitting at his desk at home tries to make sense of this problem, futilely trying to plugin the information given in the problem into one of the formulas he has to work with, desperately hoping to get the answer at the back of the book. How do we help him use technology improve this situation?

3 Solving the Problem

Again our emphasis is on getting students to actively use Sage while studying for their exams in order to improve their study experience (and at the same time learn problem solving techniques). We have tried to keep the narrative as typical as possible of our students (and our own) experiences.

3.1 Approaching the problem with technology

A typical first step of a student given the example problem is to see whether the information provided matches any of his formulas. *Hmm. Looks like $W = \int_{C^+} \vec{F} \cdot d\vec{s}$, but we don't have a vector s , we don't have a function for the curve, ...*

Now what? *Hmm. Surfaces. I can draw surfaces. Let's try that.* He starts to think about the surfaces. He sketches and scratches. Nothing works well. After all, he is just learning this stuff. So he thinks -can I plot this? Quickly dashes to WolframAlpha and types in one or both of the two equations. *Ahh, an ellipse and a plane. Plane is good. But wait - an ellipse is not a surface. What's missing here? Where is z?* Hopefully the technology is making the student think about the types of surfaces involved and how they become 3D. But now it is harder to use WolframAlpha. He needs a program that will actively plot in 3D.

Sage has all of the standard 3D plot commands including `plot3d`, `implicit_plot3d` and `parametric_plot3d`.

- We declare our variables and set (changable) minimums and maximums.
- The first surface S1 is implicit (not solved for z) so we use `implicit_plot3d`
- The second surface S2 is explicit so we use `plot3d`

Sage code is shown in red

```
var ('x y z')
xmin=-3; xmax=3; ymin=-4; ymax=4; zmin=-0.5; zmax=1.5
S1=implicit_plot3d(x^2/4 + y^2/9 == 1, (x,xmin,xmax), (y,ymin,ymax),
    (z,zmin,zmax), color='green', opacity=.5)
S2=plot3d(1, (x,xmin,xmax), (y,ymin,ymax), opacity=.5)
show(S1+S2)
```

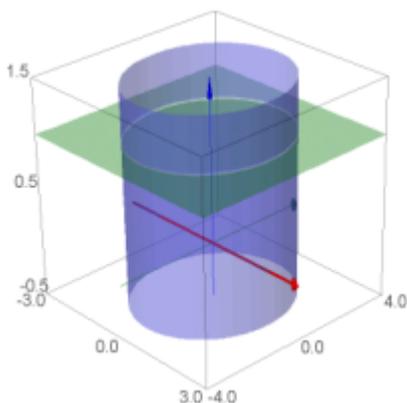


FIGURE 1. Plotting the surfaces

Tip: Sage does not label the axes, so usually we use different values for minimums and maximums so that by these values we can also determine which axis is which.

This yields the plot in Figure 1 which we can click and drag to rotate. *Ahh, the curve is an ellipse up there at $z=1$. Yes, that makes sense. Let's look at our formulas again. Hmm. Can I define the curve s now that I see it?*

3.2 Think, Work and then Check with Technology

At this point in solving, the student cannot use technology to continue since he needs to be able to determine a vector-parametric function for the curve. But presumably this HAS been explained in class. Creating the plot has helped him understand and visualize what is happening. His confidence is returning. He works out the definition of the ellipse $\vec{s}(t) = (2 \cos(t), 3 \sin(t), 1)$ and he can write up this work using proper mathematical notation as text within the Sage worksheet. This helps him keep a narrative of what he is doing.

Now he thinks: *Is this right? Does this function give the curve?* He quickly plots his function to see if gives the same curve as above. Oh yes! Instant feedback.

Back to the text and the formula and still working within Sage. *Hmm. The integral needs an interval of integration? Probably, $t \in [0, \pi/2]$*

Again he thinks: *Is this right? Might be a trick interval. Let me spend a few seconds to check.* He quickly graphs his parametrization on this interval (see Figure 2). *Great! Hey, let me add some*

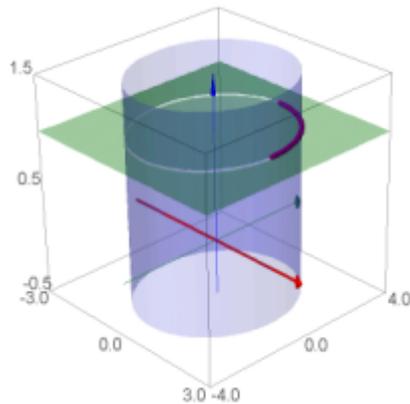


FIGURE 2. Check the Parametrization and Interval

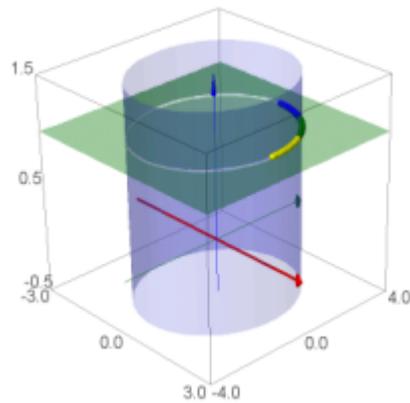


FIGURE 3. Check the Orientation

colors to check that the orientation is good. He checks the orientation by splitting the interval into 3 parts and plotting each part of the curve with a different color (see Figure 3). (Tip: You can use 2 colors if it is not a closed curve.)

Input and check the plot of the orientation preserving vector-parameterization of the curve C on the interval.

```

var ('t')
s=vector((2*cos(t),3*sin(t),1))
t1=0;t2=pi/2;t3rd=(t2-t1)/3;tmid1=t1+t3rd; tmid2=t1+2*t3rd;
C1=parametric_plot3d(s,(t,t1,tmid1),color='yellow',thickness=10)
C2=parametric_plot3d(s,(t,tmid1,tmid2),color='green',thickness=10)

```

```
C3=parametric_plot3d(s,(t,tmid2,t2),color='orange',thickness=10)
show(S1+S2+C1+C2+C3)
```

Notice that the only part of this code that changes with the problem is the input for s and for t_1 and t_2 . The code is reusable!

3.3 Finishing the problem with and without technology

Now what? He sees that he almost has all the information to either solve by hand or using the technology. That is, he has all of the letters in the formula: $W = \int_{C^+} \vec{F} \cdot d\vec{s}$. He doesn't have \vec{F} as a vector function of t ; it is still a function of (x,y,z) , but he probably has an example that shows the components of \vec{s} are the functions of \vec{F} . He doesn't have the derivative vector of \vec{s} , but that should be an easy calculation. Then, the dot product and finally, the integration itself.

All of these steps can be done in "study mode", i.e. they can be done first by hand. Then they can be checked using Sage technology **comparatively**. To the authors, this latter was very important. Sage is primarily a programming and not a mathematics environment as many of the commercial applications tend to be with the result that they are often **too** powerful. In order for the student to check his work, he must use the appropriate Sage command that matches the actual mathematical step he would do by hand. This helps him both understand the step itself and get a sense of common notation across the STEM languages (not to mention improving his programming and logical thinking skills).

Find the derivative vector

```
ds=diff(s,t)
view(ds)
```

Create a mini programming function to substitute the component functions and then do the substitution

```
def changevar(h, eqn, newvar):
    return h.substitute(eqn)
F=changevar(F,x==s[0],t)
F=changevar(F,y==s[1],t)
F=changevar(F,z==s[2],t)
view(F)
```

Complete the integrand by calculating the dot product

```
integrand=F.dot_product(ds)
view(integrand)
```

Integrate this integral analytically and/or numerically

```

result=integral(integrand,(t,t1,t2))

view(result)

n(result)

```

The result is 5. *Check the back of the book. That's right! Phew!*

4 Understanding our Example - Some Sage Extras

This above solution process clearly has several good points in and of itself. But another key bonus with Sage is that we can plot the force on the curve and look at what is happening and then interactively change the solution parameters to see if we really do understand. (This is actually easier in Sage than in any other application we tried and so we include it here to show Sage's capabilities.)

```

vs=2; num=10

vf=sum([arrow3d((2*cos(c),3*sin(c),1),(2*cos(c)+2*cos(c)*3*sin(c)/vs,
3*sin(c)+3*1/vs,1+1/vs), color=hue((c+8)/8), width=0.8) for c
in [t1..t2,step=(t2-t1)/num]])

show(C1+C2+C3+vf,aspect_ratio=(1,1,pi), figsize=5, frame=False)

```

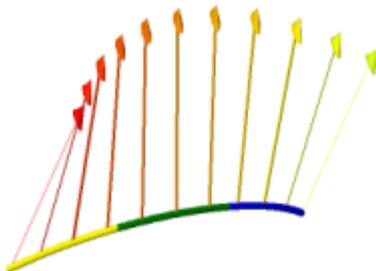


FIGURE 4. The Force F on the Curve C

Looking carefully, we might be able to see in Figure 4 that the projection of the force vectors onto the curve gradually get smaller and smaller so perhaps there is much more work being done at the beginning of the curve then towards the end. We can then test this hypothesis by changing the interval on t and take the Sage action 'Evaluate All' to recalculate everything. We could then test what work is done on the whole ellipse (see Figure 5) and find that the net work done is 0, that is, that the force goes forwards and backwards and the work is cancelled out.

5 Using Sage - Some Observations

The authors of this paper have worked extensively with many different applications for encouraging interactive study and problem solving in engineering mathematics as observed above. We have found that Sage is good for the following reasons.

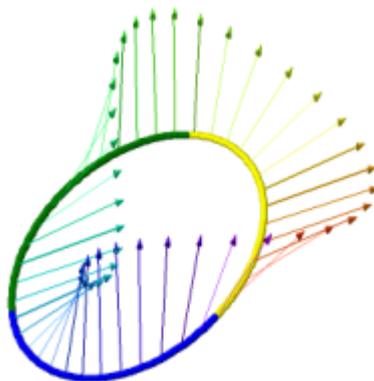


FIGURE 5. The Force F on the Whole Ellipse

- It is freeware and can be used online or offline. The worksheets can be made public. This allows for open-ended collaboration between educators and students.
- Because it is built into a common Python-based interface, the structure, syntax and notation of Sage seems to be easier for engineering and programming students to use than say Mathematica or MatLab. (We have found Sage incredibly more natural to use as well as more helpful/forgiving of syntax errors than other applications including the expensive commercial applications.
- The notebook interface is both linear and non-linear in format. This means that it can be used for both the exploration and testing stages of problem solving (working backwards and forwards) as well for the complete final step-by-step solution. It has a wide variety of file and edit commands useful for testing and restructuring the projects.
- It is readable. It allows for explanatory text and by-hand solutions to be interspersed with programming, i.e. technology solutions. It allows for direct input of text from webpages, nice mathematical formatting (latex), embedding of html code (and thus interactive GeoGebra worksheets), development of animated gifs, etc.
- It does 3d plots incredibly well in a wide variety of formats.
- It is as powerful and/or as easy to use as the user wants.
- There is a good online forum with active participants who answer questions.

To be fair, Sage has its limitations.

- While the forum and documentation are good, they are not great.
- It too can have quirky syntax, e.g. requiring double parenthesis around vector functions.
- Regular backups and downloads are a very good idea.

- The online notebook has regular ups and downs; the offline version requires more than basic computer installation knowledge.

6 Summary

In summary, we would like to emphasize several additional points about using Sage for solving problems in engineering. The authors are a theoretical mathematician, a mechanical engineer and a technology in mathematics education pioneer and yet with minimal effort we were able to produce worksheets that both do the mathematics (perform the technology properly) and communicate the mathematical process [4] (in a step-by-step understandable and readable format) - a complete package deal.

Throughout this process of encouraging our students to actively use Sage, we ourselves learned a great deal - indeed a more complete understanding of the subject matter. More importantly though is that we learned as educators about how technology such as Sage can help us integrate the often seemingly disparate points of view of science, technology, engineering and mathematics a vital task for our students. It is our hope that through the Sage forum and discussion groups that this process can continue and we are incredibly interested in discussing ideas and experiences of effective, practical, free educational developments in this field.

References

- [1] L. Fahlberg-Stojanovska and V. Stojanovski, *GeoGebra: Freedom to Explore and Learn*, Teaching Mathematics and its Applications. Oxford Journals, Oxford, UK, Vol. 26(1) (2009), pp.17-22. <http://teamat.oxfordjournals.org/cgi/content/abstract/hrp003>
- [2] M. Legua, J. Morano and L. Sanchez Ruiz, *Computer Aided Line Integral Calculus in Engineering*, ICEE 2001, Session 7D5, Oslo/Bergen Norway (2001). <http://www.ineer.org/Events/ICEE2001/Proceedings/papers/345.pdf>
- [3] United States Airforce Academy, Department of Mathematical Sciences, *A Mathematica Notebook Library* (2000). <http://www.nku.edu/~longa/classes/mat320/mathematica/multcalc.htm>
- [4] N. Idris N, *Enhancing Student's Understanding in Calculus Through Writing*, IEJME. International Electronic Journal of Mathematics Education Vol. 4(1) (February 2009).
- [5] Work defined as "area under curve". <http://hyperphysics.phy-astr.gsu.edu/hbase/wint.html>
- [6] Line Integral of Work Type - Example 2 <http://sage.math.canterbury.ac.nz/home/pub/194/> (Our open source published Sage page for the given example).
- [7] sage-support@googlegroups.com (This is the sage forum).
- [8] <http://www.linkedin.com/groups/Sage-Math> (This is the linkedin sage group).