Sabbatical Leave Report

Michael McCain, Professor
Department of Mathematics
Sabbatical Leave: Spring 2017 Semester

My sabbatical project was to expand my knowledge of the Wolfram programming language, also known as Mathematica. I also had the task of creating a database of animations that I share with my department. The idea for this Sabbatical stemmed from a talk from Eric Schulz who I saw at a math and technology conference a few years back. He is a professor of mathematics at Walla Walla Community College in Washington state. I learned that he had been a programmer at Wolfram many years ago and still works closely with the programming engineers.

His talk was about his electronic textbook which is quickly taking over the math education of Calculus throughout the U.S. I had been to many conferences over the years to develop my teaching ability and there many useful things that I had learned. However, what I saw in his talk was different and change the way I thought. He found a new way of teaching mathematics and I had been enlightened.

The central idea of his talk was the importance of visualization in mathematics to illustrate mathematical concepts. He pointed out that we are now at a point in time where we will start to change how we teach mathematics.

For hundreds of years, the ideas of mathematics has been transported through books. Textbooks have advanced in their capabilities over time. Today we have high quality printing presses, that display beautiful text and graphics on paper. Some textbooks even incorporate animations through additional software. However, in my experience, although the animations are helpful, they are limited. The reason is because I did not have a say in designing them. There are things I would have different. I had hoped that I would have an experience that was more fluid and incorporate into classes better to increase the value of my students education. It fell short.

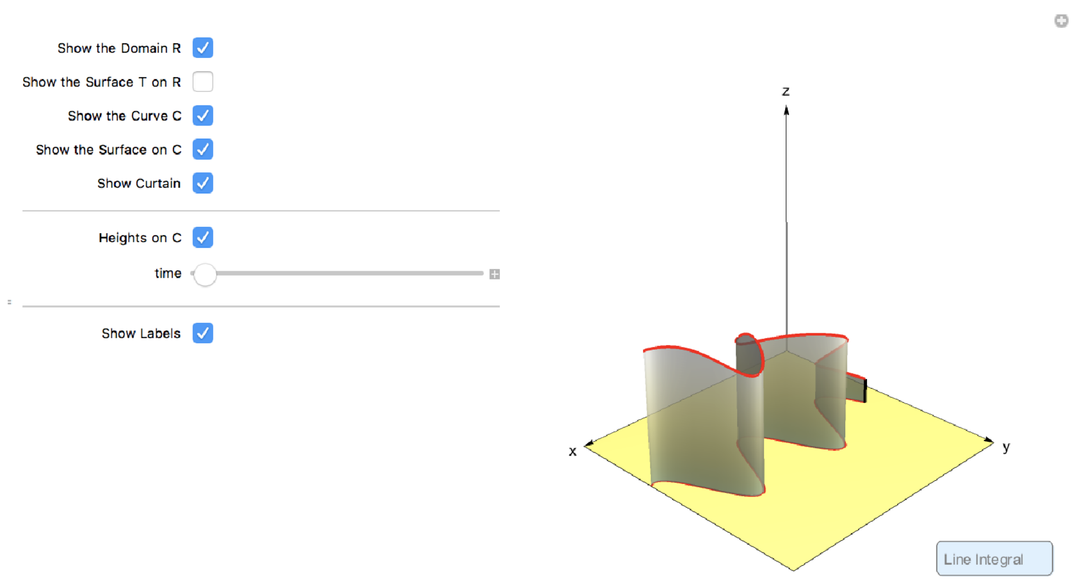
The ability to have quality interactive animations designed by instructors with little to no programming experience resonated with me. They were by far the most advanced user friendly animations I had ever seen and there were hundreds of them. With the Wolfram language these customized interactive widgets were well within the grasp of the any college math professor. The Wolfram language has, built inside of it, a structure that specifically works to allow the customization of advanced interactive graphics. The code for some of these animations would take hundreds or even thousands lines of code using other languages. It would also require a deep understanding of that language. The problem is in general it’s not useful for the work of a mathematician outside of the animations. However, the Wolfram language is different. It was created specifically for Mathematics and the Sciences.

At that point, I decided to take on this project and start thinking about a Sabbatical. I had been to working to develop my knowledge the language for about 2 years prior. However, learning was slow as I have had very little programming experience. This sabbatical gave me the time to push my learning rate into high gear. I have learned quite a bit in this time and have developed material that I currently use every day in my classroom and have shared with other instructors who also use it in their classroom.

To help develop my understanding of Mathematics I read through “Programming with Mathematica” by Paul Wellin. I quickly realized that the language was so vast and I didn’t have a firm foundation to learn everything I wanted to know in one semester.   I decided to learn the basics and start deconstructing the code for existing animations in the hopes that I would learn faster. It was rough at first. I spent much of my time figuring out options that I find simple now. There are thousands of options and options on those options. I needed to expand my knowledge in a different way. This is when I discovered stackex-change.com. This website is a network of question-and-answer type content on topics in various fields, one of them being the Wolfram Language.

I joined the community and spent much of time asking questions. I learned slowly but overtime my learning rate increased and I was able to contribute. I am now able to create animations that I would characterize as mid level difficult.   Because this page is static, and you cannot see how the interactive animations work. It’s ironic that the thing I want to show is the very thing a static page cannot express. And its exactly the thing that I want to change for everyone.

Below is an example of an interactive animation I built for Multivariable Calculus. I designed every aspect of the graphic and buttons, and I did it with a relatively small amount of code.   The graphic can be rotated to achieve any viewpoint in three dimensions by dragging the mouse over the graphic. The checkboxes and sliders on the left allow the user to turn on and off different features. The slider near the button let’s the user alter some variables in the graphic in real time. The interaction is instant, interesting and fun. My student enjoy them I believe they learn more from them then they do anything else.



I believe interactive animations are the future of teaching in mathematics. With only a small about of programming knowledge, as an instructor I can customize any interactive animations any way I want. The instructor can now illustrate complicated concepts with interactive dynamic controls.

One of the advantages to using the Wolfram Language is that there is no need for students to purchase Mathematica or even know the Wolfram language at all. A student only needs to download the free player and they will be able to interact with any animation I created. All their focus is on math now and right at their fingertips. They have buttons, sliders, checkboxes, drop down menus and beautiful graphics. This is invaluable to math eduction and now a permanent part of the way I teach.

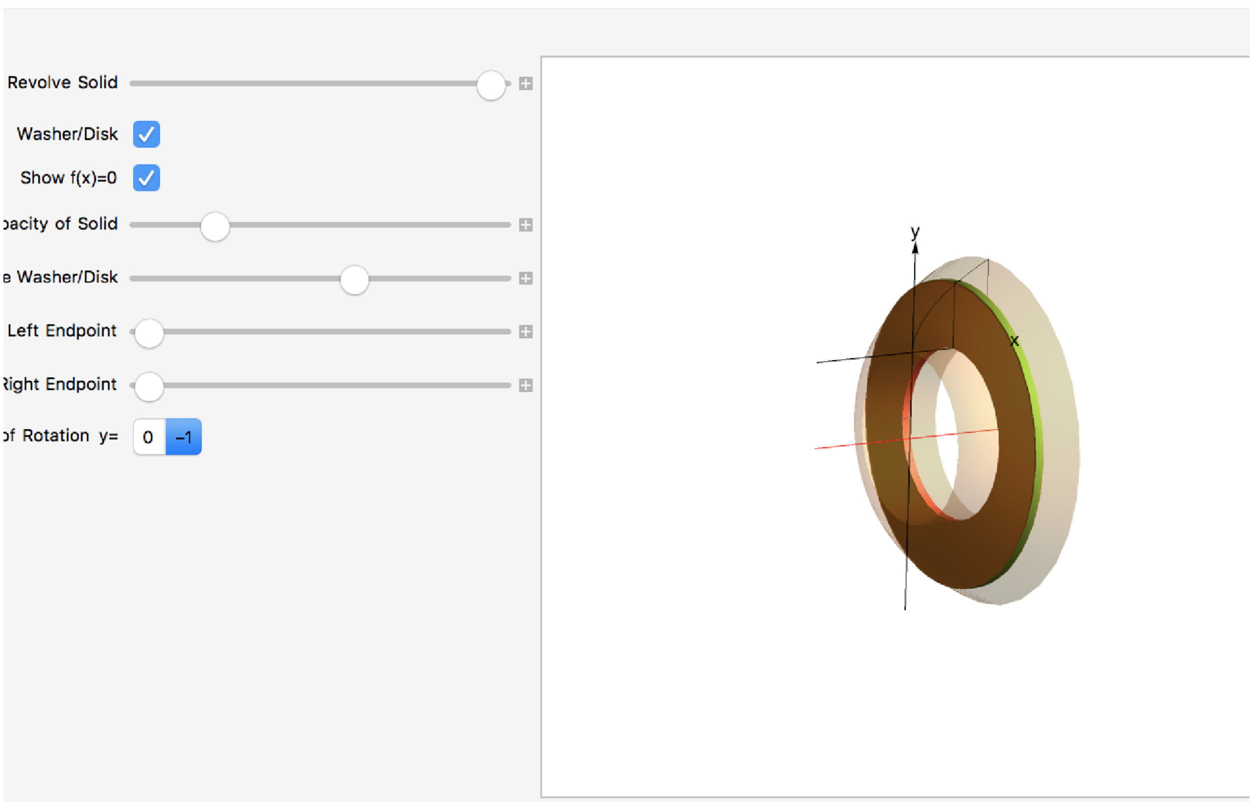
This is especially true in Calculus which is the study of motion. For hundreds of years we have been using static books to express motion. With the creation of animations now in the hands of instructors, we finally have the ability to teach motion with motion. Below is the code that generates the interactive graphic above. As you can see it relatively short, and can do quite a bit.

```
DynamicModule[{x,y,u,v,z},Manipulate[

g7=Graphics3D[{Black,Thick,Line[{Append[curve,0]/.x→t,Append[curve,surface/.{x→curve[[1]]

Show[{
If[a,g1,Graphics3D[]],
If[b,g2,Graphics3D[]],
If[c,g3,Graphics3D[]],
If[d,g4,Graphics3D[]],
If[e,g5,Graphics3D[]],
If[f,g7,Graphics3D[]],
axis3d},

ImagePadding→None,
PlotRange→{{xmin,xmax},{ymin,ymax},{zmin,zmax}},
Boxed→False,
Axes→False,
SphericalRegion→True,
BoxRatios→{1,1,1},
PlotRangePadding→None,

Epilog→{If[ep,Inset[Framed[Pane[RawBoxes@
 FormBox[RowBox[{UnderscriptBox["lim",RowBox[{"x", "→", "3"}]],RowBox[{"f", "(", "x", ")"
 ,70],Background→LightBlue,RoundingRadius→4,BaseStyle→Gray],{Right,Bottom},{Right,Bottom}

ViewPoint→{Pi,Pi/2,2}],
{{a,True,"Show the Domain R"},{True,False}},
{{b,True,"Show the Surface T on R"},{True,False}},
{{c,True,"Show the Curve C"},{True,False}},
{{d,True,"Show the Surface on C"},{True,False}},
{{e,True,"Show Curtain"},{True,False}},Delimiter,
```

```
{{f,True,"Heights on C"},{True,False}},

{{t,tstart,"time"},tstart,tend},Delimiter,

{{ep,True,"Show Labels"},{True,False}},

Initialization:→{
(surface=Sin[x y]),
(domain=Function[{x,y},-2<x<2&&-2<y<2]),
(curve={x,.3Cos[8x]+1}),
(tstart=.2),
(tend=3),
(xmin=0),
(xmax=2),
(ymin=0),
(ymax=2),
(zmin=0),
(zmax=2),

(*Domain*)
(g1=Plot3D[0,{x,xmin,xmax},{y,ymin,ymax},RegionFunction→domain,
Mesh→None,
PlotPoints→50,
Lighting→"Neutral",
PlotStyle→"LightGreen",
PlotStyle→Opacity[.95]]),

(*Surface on Domain*)
(g2=Plot3D[surface,{x,xmin,xmax},{y,ymin,ymax},RegionFunction→domain,
Mesh→None,
ColorFunction→"LightTerrain",
ClippingStyle→None,
PlotPoints→50,
PlotStyle→Opacity[.95]]),

(*Curve in Domain*)
(g3=ParametricPlot3D[Append[curve,0],{x,tstart,tend},PlotStyle→Red]),

(*Curve in Domain onto Surface*)
(g4=ParametricPlot3D[Append[curve,surface/.{x→curve[[1]],y→curve[[2]]}]/.x→p,{p,tstart,te

(*Curtain*)
(g5=ParametricPlot3D[Append[curve/.x→u,v],{u,tstart,tend},

{v,0,surface/.{x→curve[[1]],y→curve[[2]]}/.x→u},
```

```
PlotPoints→50,ColorFunction→"LightTerrain",PlotStyle→Opacity[.8],Mesh→None]),

(axis3d=Graphics3D[
{Arrowheads[Small],Arrow[{{xmin,0,0},{xmax,0,0}}],
Arrowheads[Small],Arrow[{{0,ymin,0},{0,ymax,0}}],
Arrowheads[Small],Arrow[{{0,0,zmin},{0,0,zmax}}],
Text["x",{xmax+.1,0,0}],Text["y",{0,ymax+.1,0}],Text["z",{0,0,zmax+.1}]
}]),TrackedSymbols:→{t}
},
ControlPlacement→Left,
Paneled→False,ContinuousAction→True]]
```

Below is another graphic that I created for a fellow math instructor. He will be using this semester in teaching volumes of solids of revolution in his Calculus II course. He has the code and he can make any changes he wants.



Again, notice you can view the solid by from different viewpoints in three dimensions by dragging the mouse across the graphic. You can zoom in and out and move the solid in space. In this particular animation, we are showing how to revolve a two dimensional region about an axis. By moving the slider back and forth the rotation is generated, taken away and regenerated. To draw these types of diagrams on a white board is incredibly difficult and time consuming. It's also only going to be as good as the artistic ability of the instructor. On top of that, the instructor cannot show movement. The interactive animations are sharp, detailed and allow students to manipulate motion with controls that are natural to understand and in real time. Again, the student sees no code. They see buttons, checkboxes, sliders and beautiful graphics to play with. Below is the code for the animation above.

```
Manipulate[
```

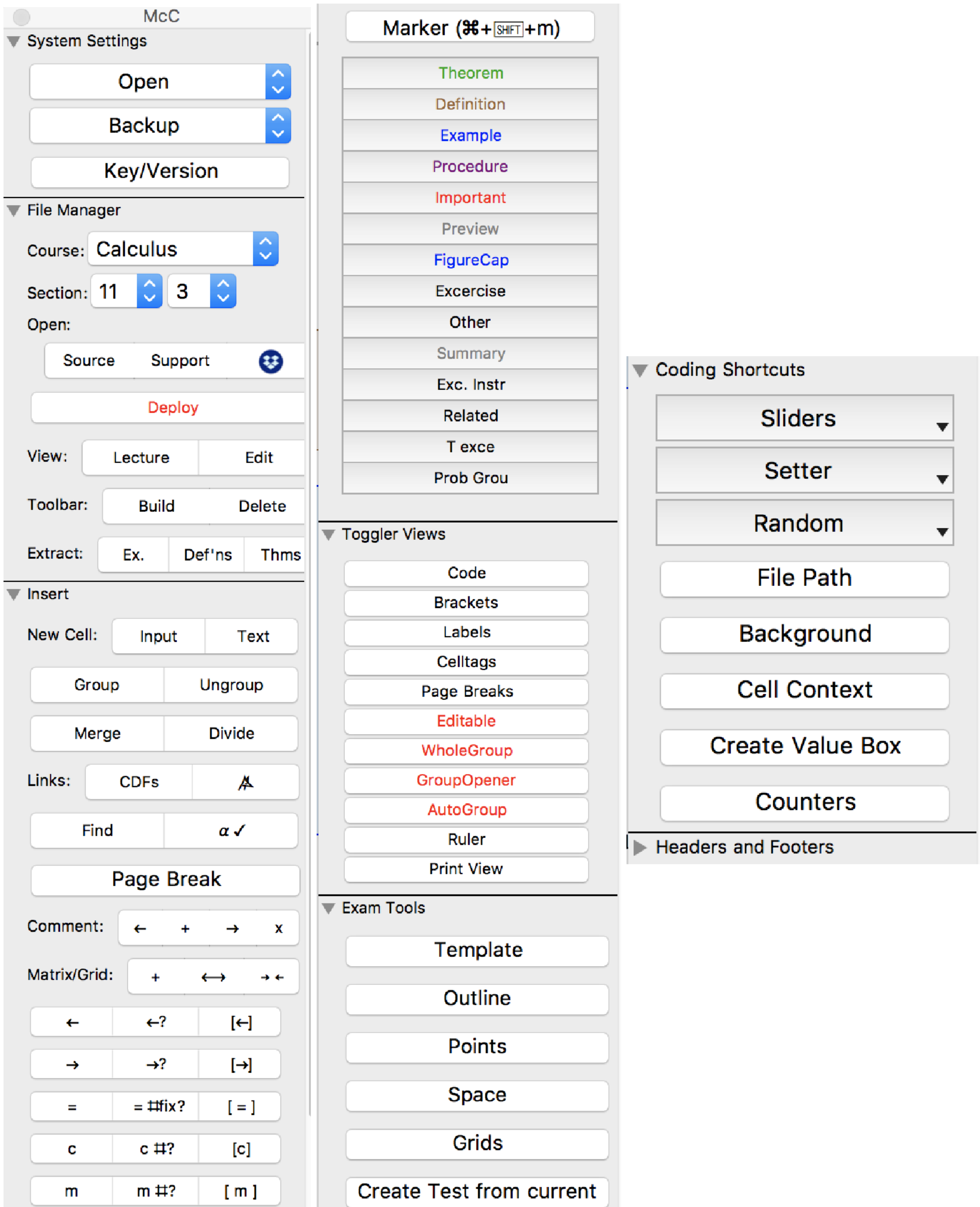
```
If[l, d[x_] := 0, d[x_] := x^2];

Show[{

  axis3d,

  Graphics3D[{Red, InfiniteLine[{0, yt, 0}, axis]}, PlotRange -> {{xmin, xmax}, {ymin, yı

  ParametricPlot3D[{
    {t, c[t], 0},
    {tstart, ((t - tstart)/(tend - tstart))*(c[tstart] - d[tstart]) + d[tstart], 0},
    {tend, ((t - tstart)/(tend - tstart))*(c[tend] - d[tend]) + d[tend], 0}},
   {t, tstart, tend}, PlotStyle -> {{Black, Thin}}, PlotRange -> {{xmin, xmax}, {ymin, yı

  ParametricPlot3D[{
    {t, d[t], 0}},
   {t, tstart, tend}, PlotStyle -> {{Black, Thin}}, PlotRange -> {{xmin, xmax}, {ymin, yı
  If[k, Graphics3D[Polygon[{
      {Dynamic@n - rx, c[Dynamic@n], 0},
      {Dynamic@n + rx, c[Dynamic@n], 0},
      {Dynamic@n + rx, d[Dynamic@n], 0},
      {Dynamic@n - rx, d[Dynamic@n], 0}
      }], PlotRange -> {{xmin, xmax}, {ymin, ymax}, {zmin, zmax}}], {}],

  ParametricPlot3D[
   {Evaluate[RotationTransform[θ, axis, {0, yt, 0}][{
       {t, d[t], 0}
       }]]}
   ,
   {t, tstart, tend}, {θ, 0, a}, Axes -> None, Boxed -> False, Mesh -> None, Performance(

  ParametricPlot3D[
   Evaluate[RotationTransform[θ, axis, {0, yt, 0}][
     {{t, c[t], 0},
      {tstart, ((t - tstart)/(tend - tstart))*c[tstart], 0},
      {tend, ((t - tstart)/(tend - tstart))*(c[tend] - d[tend]) + d[tend], 0}}

     ]],
   {t, tstart, tend}, {θ, 0, a}, Axes -> None, Boxed -> False, Mesh -> None, PlotStyle -:

   PlotRange -> {{xmin, xmax}, {ymin, ymax}, {zmin, zmax}}],


  If[k, ParametricPlot3D[
    Evaluate[RotationTransform[θ, axis, {0, yt, 0}][
```

```
      {{n - rx, ((t - tstart)/(tend - tstart))*(c[n] - d[n]) + d[n], 0},
       {n + rx, ((t - tstart)/(tend - tstart))*(c[n] - d[n]) + d[n], 0},
       {n + (2 ((t - tstart)/(tend - tstart)) - 1)*rx, c[n], 0},
       {n + (2 ((t - tstart)/(tend - tstart)) - 1)*rx, d[n], 0}}

      ]]



    ,
    {t, tstart, tend}, {θ, 0, a}, Axes -> None, Boxed -> False, Mesh -> None, Performanc




  }, Boxed -> False, SphericalRegion -> True, PlotRange -> {{xmin, xmax}, {ymin, ymax},

{{a, .01, "Revolve Solid"}, .01, 2 Pi},
{{k, True, "Washer/Disk"}, {True, False}},
{{l, True, "Show f(x)=0"}, {True, False}},
{{o, .3, "Opacity of Solid"}, 0, 1},
{{n, .6, "Move Washer/Disk"}, tstart, tend},
{{tstart, 0, "Left Endpoint"}, 0, 1},
{{tend, 1, "Right Endpoint"}, 1, 2},
{{yt, 0, "Axis of Rotation  y="}, {0, -1}},

TrackedSymbols :> {a, tstart, tend, n, o, yt, k, l},

Initialization :> {
  (c[x_] := Sqrt[x]),
  (d[x_] := x^2),
  (rx = .05),
  (xmin = -1.3),
  (xmax = 1.3),
  (ymin = -3),
  (ymax = 1.3),
  (zmin = -3),
  (zmax = 3),
  (tstart = 0),
  (tend = 1),
  (origin = {0, yt, 0}),
  (axis = {1, 0, 0}),

  (axis3d = Graphics3D[
     {Arrowheads[Small], Arrow[{{xmin, 0, 0}, {xmax, 0, 0}}],
      Arrowheads[Small], Arrow[{{0, ymin, 0}, {0, ymax, 0}}],
```

```
Text["x", {xmax + .1, 0, 0}], Text["y", {0, ymax + .1, 0}]
}])}]
```

In learning to create these animations, I became curious on other topics that Eric Schulz had addressed in his talk. I took another look and found an idea incorporated into his electronic textbook that I had missed. I discovered that I could create my own platform similar to his that I could use to create lectures in the classroom. I spent much time learning to mimic what we had created in the hopes that I could make this work and help teachers create lectures. I have been working on creating this platform ever since. My idea was to create an interface that allows instructors to create beautifully written typeset lectures with customized styles and interactivity animation throughout.

The first area I focused on was creating a palette that instructors could use that contained everything they would need. Below is the palette I created and currently use. The code is far too long to include here.



In the creation of my palette, I spent much time figuring out things that teachers need to have in their written lectures. The palette contains too many buttons to discuss, however, I will discuss two of them.

First, I created a buttons that navigate between courses and content sections. Those buttons can be seen near the word “Calculus” on the palette above. My goal was to allow instructors to create content that students could easily navigate within. This turned out to be difficult endeavour. The code to write was more complicated than I had anticipated, so I had to learn a lot of new language. I drew knowledge mostly from videos of Eric Schulz at a couple talks he gave. I am not finished with the palette, however, I have most of the functionality developed at this point.

In the middle of the palette there are buttons that insert pre-styled blocks of content that a teacher would use. For example, a “Theorem” box. Below you can see that the “Theorem” box is outlined in a dark green. The background is light green. The title is bold amongst other things. In addition there is an auto-numbering system that renumbers theorems as new theorems are inserted any where in the document.

Limit Rules

Many of these rules we will use are just extensions of rule we learned in single variable calculus.

Theorem 12.3.1 **Limit Laws**

Let a, b, and c be real numbers.

1. $\lim_{(x,y)\to(a,b)} c = c$
2. $\lim_{(x,y)\to(a,b)} x = a$
3. $\lim_{(x,y)\to(a,b)} y = b$

Theorem 12.3.2 **More Limit Laws**

Let L and M be real numbers and suppose that $\lim_{(x,y)\to(a,b)} f(x, y) = L$ and $\lim_{(x,y)\to(a,b)} g(x, y) = M$. Assume c is a constant, and m and n are integers.

1. $\lim_{(x,y)\to(a,b)} (f(x, y) + g(x, y)) = L + M$
2. $\lim_{(x,y)\to(a,b)} (f(x, y) - g(x, y)) = L - M$
3. $\lim_{(x,y)\to(a,b)} c\, f(x, y) = c\, L$
4. $\lim_{(x,y)\to(a,b)} f(x, y)\, g(x, y) = L\, M$
5. $\lim_{(x,y)\to(a,b)} \left[\frac{f(x,y)}{g(x,y)}\right] = \frac{L}{M}$, provided $M \neq 0$
6. $\lim_{(x,y)\to(a,b)} (f(x, y))^n = L^n$
7. If m and n have no common factors and $n \neq 0$, then $\lim_{(x,y)\to(a,b)} [f(x, y)]^{m/n} = L^{m/n}$, where we assume $L > 0$ if n is even.

Example 1 **Evaluating a Limit**

Evaluate $\lim_{(x,y)\to(2,8)} \left(3\, x^2\, y + \sqrt{x\, y}\right)$

Near the bottom of the image above there you can see the output for pressing the “Example” button in the palette. I also worked on typesetting of mathematics and how to make it easier for teachers to do it. Expressing math notation with a standard keyboard is difficult. There is no standard for it and there are thousands of different symbols used in mathematics and different ways of writing each of them. They have different fonts, spacings, thickness and angles. Mathematica has a prebuilt palette with buttons that will output this type of notation but it’s time consuming for the user and difficult to locate the correct button. My idea was to create a way that allowed instructors to express notation quickly and efficiently and in a way that was intuitive.

$$\lim_{(x,y)\to(2,8)} \left(3\, x^2\, y + \sqrt{x\, y}\right)$$

I designed 220 shortcuts for all of the math notation that I could think of that anyone might use in mathematics at the community college level. Again this code, is too long to include here. I have a shortcut that when you type “limitxy” and then press the space bar, you get the structure below in place with empty slots ready for input. The instructor just needs to tab through to populate them.

$$\lim_{(x,y)\to(\square,\square)} \square$$

I also learned how to incorporate stylesheets into this lecture creating platform. A stylesheet is collection of options for almost every different way anything can appear in Mathematica content. For example, by changing the stylesheet we can change the look of the “Theorem” box I discussed earlier to look like the new “Theorem” box below. The text is now bold, the font is now red, and the background is now green.

Theorem 12.3.1 **Limit Laws**

Let a, b, and c be real numbers.

1. $\lim_{(x,y)\to(a,b)} c = c$
2. $\lim_{(x,y)\to(a,b)} x = a$
3. $\lim_{(x,y)\to(a,b)} y = b$

Stylesheets allow the instructor to have the control for consistency in the appearance of all documents in one single place. This is very useful as we sometimes like to make changes to the way material looks. This gives the instructor the freedom to change things in any document while keeping consistency between all documents everywhere with one easy control source.

I have learned so much over this Sabbatical. It would take months to express everything that I have learned, and years to express them correctly as they would require a deeper understanding of the Wolfram Language. I was pleased with the results achieved and I am confident my experience and expertise will benefit me, my department, our college and most importantly, our students. This is a new way of learning and I very excited to be part of it. I will continue to pioneer the area of interactive animations and represent our college on this front.

Thank you for the time spent assigned to this task.