

# **MATHEMATICA BASED COMPUTER LABORATORIES FOR THE ENHANCEMENT OF A CALCULUS SEQUENCE**

Raouf Boules and Mike O'Leary

Department of Mathematics

Towson University

Towson, MD 21252

boules@towson.edu and moleary@towson.edu

## **Introduction**

During the past fifteen years there has been a strong interest in the United States and elsewhere in the improvement of mathematics education at all levels. A strong widespread movement has called for change in the way introductory calculus courses are taught. At Towson University, we have decided to incorporate a *Mathematica* based computer laboratory component in each of the three courses that compose our calculus sequence. In 1998 and 1999, four faculty members (Raouf Boules, Geoff Goodson, Ohoe Kim, and Mike O'Leary) wrote a sequence of computer laboratories that were gradually introduced to the calculus curriculum. Later revised and published by Harcourt, these calculus laboratories now form a required component of all of our calculus courses. In this note, we shall describe the results of that process.

## **Structure of Laboratories**

There are three courses in our calculus sequence; each carries four credit hours, and meets for four hours each week for lecture and discussion, and for one hour each week for the computer laboratory. Classes are limited to 24 students. Apart from the computer laboratory component, the courses are fairly traditional with the primary textbook being Stewart's *Calculus: Early Transcendental Functions*. The primary audience for the courses are students with majors in mathematics, mathematics education, computer science, and computer information sciences, with majors in chemistry and physics present to a lesser degree. Once each week for fifty minutes the class meets in a specially equipped computer laboratory classroom. Equipment includes 24 computers for the students, together with a computer and projection equipment for the instructor.

We assume that our students have no prior experience with computer algebra systems in general or *Mathematica* in particular. During the first laboratory meeting of the course, students work through a tutorial designed to introduce them to the necessary concepts and syntax of *Mathematica*. Since we use *Mathematica* as a teaching tool, we only use a small subset of the commands available in *Mathematica*. Further, because we have a number of students who transfer to Towson from other institutions, this introductory material is covered at the start of each course in the calculus sequence.

Each laboratory is written in the form of a *Mathematica* notebook. This is an interactive document, which contains expository and explanatory material together with a collection

of problems for the student. Each laboratory focuses on a particular topic from the course; examples include Newton's method, the integral test, and partial derivatives. Each laboratory begins with an introduction, which describes the type and scope of the problems to be considered. The remainder of the laboratory consists of matched pairs of examples and experiments. Each example/experiment pair discusses a single subtopic. For example, in the laboratory on the integral test, there are five example/experiment pairs discussing series as areas, overestimates, underestimates, applying the integral test to divergent series, and using the integral test to estimate the sum of series. The example contains the necessary mathematics and any new *Mathematica* commands needed to solve the problems posed in the experiments. The experiments range from the purely computational to ones that require the students to make and support conjectures about general situations.

At the beginning of each laboratory period, the instructor will introduce the laboratory to be studied, and discuss how it fits the material that is being covered in class. This time is also used to discuss any new *Mathematica* syntax that is introduced in the laboratory. During the remainder of the class hour, students work their way through the assigned portion of the laboratory. Students are encouraged to work in small groups, however each student must hand in an individual lab report. Students are not expected to finish the entire laboratory during the class hour, but complete the work on their own in one of our open computer laboratories. The final laboratory report is due one week after the initial assignment. Like the assignment, the laboratory report is a *Mathematica* notebook. Students are expected to produce a well-written report, with appropriate textual explanatory material.

## **Philosophy**

We view the inclusion of *Mathematica* based computer laboratories as a supplement to the material being taught in the traditional portion of the course, rather than as a replacement for it. In particular, we use the rich graphing, animation, and programming abilities of *Mathematica* to promote meaningful exploration, learning by discovery, asking what-if questions, and the making and testing of conjectures. Our goal in the use of the laboratories is to illustrate the concepts and ideas taught in a traditional calculus course, not to teach the effective use of *Mathematica*. For example, although *Mathematica's* FindRoot command uses Newton's method, this approach is not useful for a class on Newton's method. We prefer to use the capabilities of *Mathematica* to create a set of modules that emphasize the geometric meaning as well as the limitations of Newton's method. The laboratory engages the learner in an interactive way and leads him/her step by step in the learning process.

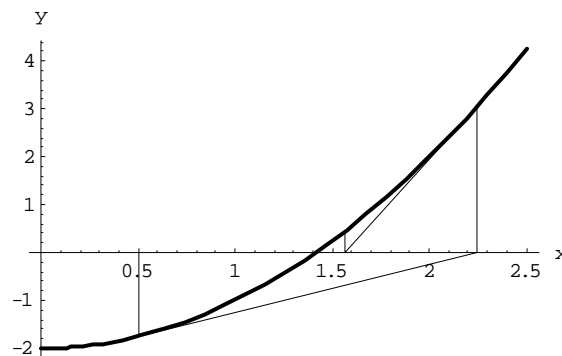
## **Example Laboratory 1: Newton's Method**

After a short introduction about the purpose of the Newton's method, the student begins by studying the *Mathematica* commands Nest and NestList, which are used to iterate functions. Next, students learn how to apply these commands to generate the list of Newton iterates for the function  $f(x) = x^2 - 2$ . In the corresponding experiment, the

function  $f(x) = \sin(\ln x) - 0.2$  is studied. Note that the students do not use the built-in *Mathematica* command `FindRoot` described above, but rather use the commands `Nest` and `NestList` to perform the needed iterations. The third example/experiment pair is devoted to the graphical interpretation of Newton's method. We introduce the following module.

```
newton[f_, a_, b_, x1_, n_] := Module[{}, g[x_] := x - f[x]/f'[x];
  data = NestList[g, x1, n];
  data2 = Flatten[Table[{data[[i]], 0, data[[i]], f[data[[i]]}],
    {i, Length[data]}]];
  data3 = Partition[data2, 2];
  data4 = Drop[data3, -1];
  p1 = Plot[f[x], {x, a, b}, PlotStyle -> {{RGBColor[1, 0, .1], Thickness[.008]}},
    AxesLabel -> {"x", "y"}, PlotRange -> All, DisplayFunction -> Identity];
  p2 = Graphics[{Hue[.6], Line[data4]}]; Show[{p1, p2},
    DisplayFunction -> $DisplayFunction];]
```

As input, this module takes a function  $f(x)$ , the endpoints  $a$  and  $b$  of an interval containing the root, an initial guess  $x_1$  for the root, and the number of iterations  $n$  of Newton's method to perform. The module returns a graph of the function, together with a graphical representation of the process used to generate the Newton iterates; it also keeps the list of iterates generated in a variable for later retrieval. This module, when applied to the function  $f(x) = x^2 - 2$  on the interval  $[0, 2.5]$  with the initial guess  $x_1 = 0.5$  and 3 iterations yields the following graph.



**Figure 1**  
The result of the command `newton[f,0,2.5,0,3]` applied to  $f(x) = x^2 - 2$

Another module is also presented, which animates these graphs. In the corresponding experiment, students study the geometric properties of Newton's method. The remaining two example/experiment pairs study how Newton's method may diverge or enter into a cycle. This is done both symbolically and graphically, with the aid of the module `Newton` described above.

### Example Laboratory 2: Integral Test

Our laboratory on the integral test begins by describing how the value of an infinite series can be visualized as the area of an infinite sequence of rectangles. A module is presented that takes as input the general term in a series, and returns the graph of the corresponding

family of rectangles. In the next example/experiment pair the use of a function whose area is an overestimate for the value of the series is discussed, while in the subsequent example/experiment pair, the corresponding results for underestimates are presented, resulting in a precise statement of the integral test. Essential to this process are a pair of *Mathematica* modules that generate the needed graphs. The following module takes as input the general term of an infinite series  $a_n$ , a comparison function  $f(x)$  and a list of terms, and returns a graph of the function as well as the graph of the set of rectangles whose area corresponds to the value of the infinite series.

```
OverestimateGraph[a_,f_,{n,left_,right_}]:=Module[{g1,g2,g3,g4,x,n},
  g1=Plot[f[x],{x,left-1,right},PlotStyle->{{RGBColor[1,0,.1],
  Thickness[.0045]}},Frame->True,Axes->False,
  DisplayFunction->Identity];
  g2=Graphics[{Hue[0.17],Table[Polygon[{{x,0},{x+1,0},
  {x+1,f[x+1]},{x,f[x+1]}]},{x,left-1,right-1}]}];
  g3=Graphics[Table[Text["a"x+1},{x+1/2,a[x+1]+0.05*f[left]},
  TextStyle->{FontSize->12,FontSlant->"Italic"}],{x,left-1,right-1}]];
  g4=Graphics[Table[Line[{{x,0},{x+1,0},{x+1,a[x+1]},
  {x,a[x+1]},{x,0}}],{x,left-1,right-1}]];
  Show[g1,g2,g4,g1,g3,DisplayFunction->$DisplayFunction,
  ImageSize->600,PlotRange->{{left-1,right},{0,1.1 f[left}}];]
```

This module, when applied to the sequence  $\sum_{n=1}^{\infty} \frac{1}{1+n^{3/2}}$  and the function  $f(x) = \frac{1}{1+x^{3/2}}$  for the first 20 terms yields the following graph.

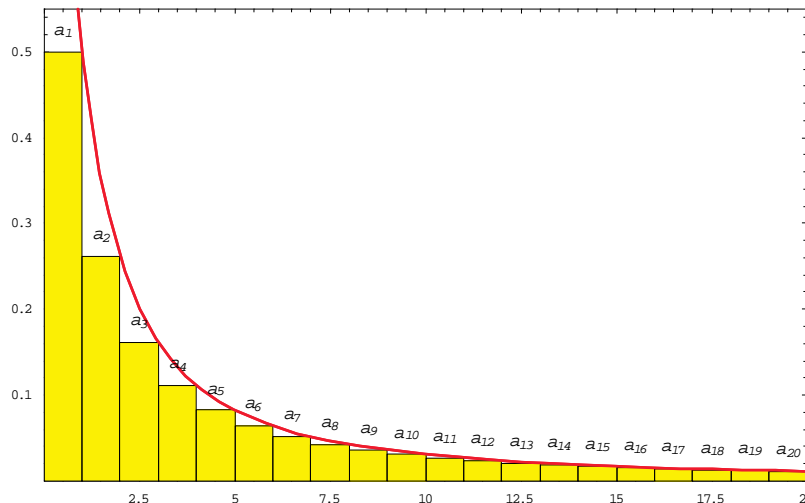


Figure 2

The result of the command `overestimate[f,a,{n,1,20}]` applied to  $a_n = (1+n^{3/2})^{-1}$  and  $f(x) = (1+x^{3/2})^{-1}$

The remaining two example/experiment pairs in this laboratory are devoted to applications of the integral test. In the first, students determine the rate at which a divergent series diverges. In the second, students use the integral test to approximate the value of an infinite series by calculating an appropriately chosen partial sum. Students need to use the integral test to determine the number of terms needed in their partial sum

to obtain a specified degree of accuracy and then find the partial sum; they do not use the capability of *Mathematica* to numerically evaluate the sum of the infinite series.

### Example Laboratory 3: Partial Derivatives

Our laboratory on partial derivatives begins with a graphical introduction to the idea. This is done with the aid of a module *YSlice*, described below, and a corresponding version called *XSlice*. *YSlice* takes as input a function, a *y* value, and a viewing region, and returns two graphs: a graph of the three dimensional surface sliced along the given *y* value, and a graph of the function restricted to that value.

```
YSlice[f_,ys_,{x,x0,x1},{y,y0,y1},{z,z0,z1}]:=
Module[{a, b, c, d},
  a=Plot3D[f[x, y],{x,x0,x1},{y,y0,y1},
  PlotRange->{z0,z1}, BoxRatios->{1,1,1}, AxesLabel->{"x","y","z"},
  DisplayFunction->Identity];
  b=ParametricPlot3D[{x,ys,z},{x,x0,x1},{z,z0,z1}, BoxRatios->{1,1,1},
  DisplayFunction->Identity];
  c=Show[a,b,PlotRange->{{x0,x1},{y0,y1},{z0,z1}},
  DisplayFunction->Identity];
  d=Plot[f[x,ys],{x,x0,x1},PlotRange->{z0,z1},Axes->False,
  FrameLabel->{StyleForm["x"],StyleForm["z"]},
  DisplayFunction->Identity, Frame->True, PlotStyle->RGBColor[0, 0, 1]];
  Show[GraphicsArray[{c,d}], DisplayFunction->${DisplayFunction},
  ImageSize -> 600];]
```

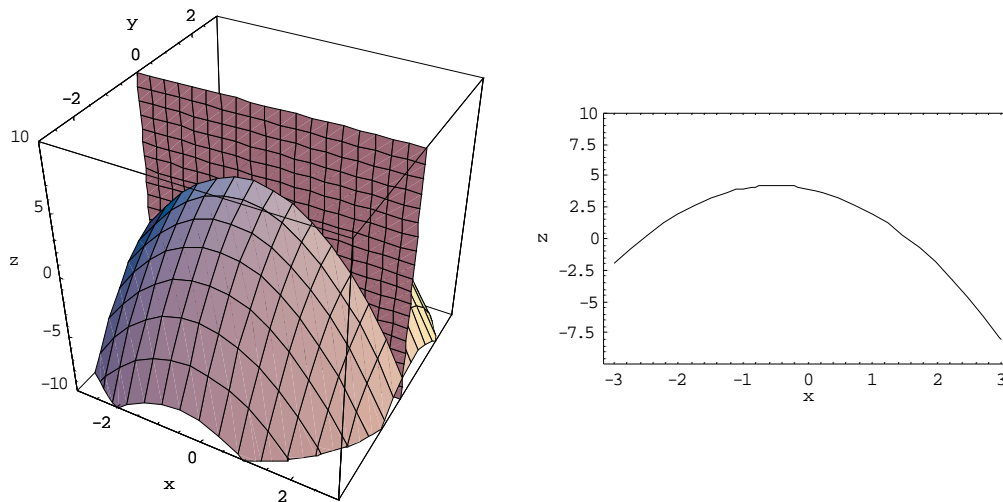


Figure 3

The result of the command `YSlice[f,0,{x,-3,3},{y,-3,3},{z,-10,10}]` applied to  $f(x) = 4 - x + y - x^2 - y^2$

Students use this module and its sibling, *XSlice* to graphically estimate the value of partial derivatives at different points. The second example/experiment pair is devoted to the algebra and analysis of the difference quotients for partial derivatives, while in the final example/experiment pair, students study how to use partial derivatives to construct tangent planes to surfaces.