Abstract

Using GeoGebra for conceptual understanding of numerical methods in calculus. Illustrations of Newton’s method, Euler’s plus other methods for solving differential equations, and summation (both approximations of definite integrals and convergence of sequences and series). Discussion of lists versus the spreadsheet for carrying out repetitive calculations. Introduction of basic syntax. Detailed instructions. Opportunity to work through instructions individually during session.
Outline

1. Using this Presentation
2. Lists vs Spreadsheet
3. Zeros: Newton’s Method
   - Background and Motivation
   - GeoGebra Steps - Experienced
   - GeoGebra Steps - Beginner
4. Differential Equations
   - Euler’s Method
     - Background and Motivation
     - GeoGebra Steps - Experienced
     - GeoGebra Steps - Beginner
     - Built-in Tools
   - Improved Euler’s Method
   - Runge-Kutta Method
5. Summation
   - Integration: Riemann / Trapezoid / Simpson
     - GeoGebra Steps
   - Convergence
Part I:
- when to use `Sequence[]` and when to use the Spreadsheet view
- places numerical methods show up in introductory calculus classes
- how GeoGebra might be useful in teaching/understanding numerical methods

Part II:
- build your own GeoGebra applet for one or more concepts with the aid of instructions in the slides
- use prebuilt applets to pursue suggested explorations or to create explorations for use in your own classroom
These slides (as pdf) and GeoGebra applets referenced in the slides can be downloaded from http://padlet.com/wall/GGB2013-Session104. Both may be used or modified freely by individuals for use in their own classroom teaching.

One the right side of the slides is a dynamic table of contents. Click a section header and you will go to the beginning of that section and then see more detailed subdivisions. Click on the subdivision you want. If you have the pdf and the GeoGebra applets downloaded to the same folder, clicking on the light blue links to the applets should start GeoGebra and open the applet.
**Lists**

**usage**

non-iterative: items depend only on where in the order they are and not on other items

\[ x_n(n) \]

**Ex:** First 7 terms of the harmonic series

<table>
<thead>
<tr>
<th>terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>input: Sequence([1/n,n,1,7]);</td>
</tr>
<tr>
<td>output: list1 = {1, 0.5, 0.33, 0.25, 0.2, 0.17, 0.14}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>partial sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>input: Sum[Sequence([1/n,n,1,7])];</td>
</tr>
<tr>
<td>output: a = 2.59</td>
</tr>
</tbody>
</table>
usage

iterative: items depend on other item(s) and may also depend on where in the order they are

\( x_n (n, x_{n-1}, x_{n-2}, \ldots) \)
**Spreadsheet Ex**

**Ex:** First 5 terms of Fibonacci sequence

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>A2 + A1</td>
</tr>
<tr>
<td>4</td>
<td>A3 + A2</td>
</tr>
<tr>
<td>5</td>
<td>A4 + A3</td>
</tr>
</tbody>
</table>

1. Type 1 in cell A1
2. Type 1 in cell A2
3. Type = in cell A3 and then click on cell A2, type +, and click on cell A1 and then hit return
4. Select cell A3 and move mouse to bottom right corner. Click on the corner and drag down to row 5 (fill-down).
Math Steps

1. Guess a zero of a function $x_0$.
2. Linearize the function through the point $(x_0, f(x_0))$.
3. Find where linearization intersects the $x$-axis and call that the new guess.
4. Repeat.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Should we use a spreadsheet or the Sequence function for this?
What Students Gain Using GeoGebra

1. **Step-by-step visualization of mathematical recipe**

2. **If students construction the GeoGebra applet, they have to think about what the mathematical recipe means to construct the visualization.**

3. **Dragging starting guess gives a visual of how rapidly the method converges and of how the method fails.**
Possible Explorations

- When multiple zeros exist, where should the initial guess be so the method converges to each of the zeros?
- What starting value(s) cause the method to fail?
- \( f(x) = (x - 3)^2 - 4 \)
- Identify the multiple failures of the method for
  \( f(x) = (x - 3)^3 - 4. \)

For an example of a finished GeoGebra applet applying Newton’s method, see Newton.ggb.
Create the function.

Create a point on the $x$-axis to serve as the initial guess for the zero of the function.

This is an iterative process so go to the spreadsheet.

Find the value of the function at $x_0$. Create a point there.

Create the tangent to the function at that point.

Find the new guess from the $x$-coordinate of the intersection of the tangent line and the $x$-axis: $x(\text{Intersect}[D1,x\text{Axis}])$. 
Prettification

- Use the Color tab of Object Properties to change color of function and separately on column D to change the color of the tangent lines.
- Use the Style tab of Object Properties on column D to make these lines very thin (because there will be lots of them).
- Hide the Spreadsheet and Algebra views by unchecking them in the View menu.
- Insert an Input Box to let the user set the function. Use Control-Click to call up the Object Properties and in the Basic section select Fix Object so the input box will stay in one place on the screen as you pan and zoom the graph.
- We’d probably really like to see the intersection points so add a column E to the spreadsheet starting in row 2 that creates a point with column A as the $x$-coordinate and 0 as the $y$-coordinate.
Create a function

Type exactly what is shown in the input bar at the bottom of the screen and then hit return (or enter). The result will be what shows in the algebra window (left) and graphics window (right).

```
Input: (x-3)^2-4
```

```
Function

\[ f(x) = (x - 3)^2 - 4 \]
```
Manipulating the Graphics Window

Your graphics window may not look the same as mine. Ctrl-click on the graphics window to hide/show axes. Shift+Drag on the graphics view to move the axes. Scroll up and down or use the zoom tools to zoom in and out. (Clicking the triangle at the bottom right of a tool icon reveals a drop-down menu of related tools. If the tool you want is on top, just click the top icon.)
Create a First Guess for the Zero

1. Go to the Points menu and select New Point.
2. Click somewhere on the $x$-axis.

If you haven’t changed your defaults, the point will be light blue indicating that you can move the point but that the movement is restricted to the object the point is currently on.
Most books call the first guess at the location of the zero \( x_0 \), so we’ll rename the point we just created to match that convention.

- Control-click the point in the graphics view or its name and other info in the algebra view and select Rename.

- Type \( x_0 \), using shift+– to get the underscore character. Click the okay button or hit return when you are finished typing.
Notes on Names and Renaming

- Points default to upper case names but we can assign lower case names. (However, if you create a point in the input bar and use lower case, you will get a vector to the point rather than the point.)

- Renaming anything correctly executes the renaming everywhere the object has been referred to in the file so you can type cryptic-but-quick names when creating and then change them to comprehensible-but-long at the end.
**First Iteration**

- Show the spreadsheet.

- Get the value of our initial guess. Find the corresponding $y$-value. Make a point there. Draw tangent line to function at that point.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$x(x_0)$</td>
<td>$f(x_1)$</td>
<td>$(A_1, B_1)$</td>
<td>$\text{Tangent}$(C1, f)</td>
</tr>
</tbody>
</table>

What doesn’t show is that I started the typing in each cell with =.

To get this view, use.

- Using **Options -> Algebra View -> Value** shows

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.22</td>
<td>13.84</td>
<td>(7.22, 13.84)</td>
<td>$y = 8.45x - 47.18$</td>
</tr>
</tbody>
</table>
Second Iteration

- Commands:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(x(x_0))</td>
<td>(f(A1))</td>
<td>(A1, B1)</td>
<td>Tangent[C1, f]</td>
</tr>
<tr>
<td>2</td>
<td>(x(\text{Intersect}[D1, x\text{Axis}]))</td>
<td>(f(A2))</td>
<td>(A2, B2)</td>
<td>Tangent[C2, f]</td>
</tr>
</tbody>
</table>

- Only the first column needs to be typed. Select cells B1-D1 and then move mouse to the bottom right corner of the selection. Click on the corner and drag down one row (fill-down).

- Values:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.22</td>
<td>13.84</td>
<td>(7.22, 13.84)</td>
<td>(y = 8.45x - 47.18)</td>
</tr>
<tr>
<td>2</td>
<td>5.59</td>
<td>2.68</td>
<td>(5.59, 2.68)</td>
<td>(y = 5.17x - 26.2)</td>
</tr>
</tbody>
</table>
Showing Our Work

- Notice that only the points are showing up in the graphics view.

- Select all of column D by clicking in the shaded box labeled D.
- Now Control-Click in the same box and select Object Properties.
- Click so the Show Object checkbox is blue and has a checkmark.
More Iterations

Select cells A2-D2 and drag the bottom right corner down several rows to get more iterations. Don't get too carried away - Geogebra will get slow and this method converges quickly when it works at all. Since we've already told GeoGebra to show the lines, the rest will show automatically.
Prettification

- Use the Color tab of Object Properties to change color of function and separately on column D to change the color of the tangent lines.
- Use the Style tab of Object Properties on column D to make these lines very thin (because there will be lots of them).
- Hide the Spreadsheet and Algebra views by unchecking them in the View menu.
- We’d probably really like to see the intersection points so add a column E to the spreadsheet starting in row 2 that creates a point with column A as the $x$-coordinate and 0 as the $y$-coordinate.
Insert an Input Box to let the user set the function. Use Control-Click to call up the Object Properties and in the Basic section select Fix Object so the input box will stay in one place on the screen as you pan and zoom the graph.
Euler’s method uses linearization to approximate the solution of a differential equation given a particular initial condition. Effectively, Euler’s method is the algebraic version of what we do when we sketch solutions in slope fields.
Steps

1. Start from the initial condition, $x_0$, - a point given to be on the solution curve.

2. Decide how far to move in the $x$ direction, $\Delta x$.

3. Approximate the resulting change in $y$ by assuming the slope of the secant line between $(x_0, y_0)$ and $(x_1, y_1)$ is equal to the slope of the tangent line at $x_0$.

\[
\frac{\Delta y}{\Delta x} = \frac{dy}{dx}
\]  

(1)

\[
\frac{y_1 - y_0}{\Delta x} = \left. \frac{dy}{dx} \right|_{x_0,y_0}
\]  

(2)

\[
y_1 = y_0 + \left. \frac{dy}{dx} \right|_{x_0,y_0} \Delta x
\]  

(3)

generally: $y_{i+1} = y_i + \left. \frac{dy}{dx} \right|_{x_i,y_i} \Delta x$  

(4)
What Students Gain Using GeoGebra

1. step-by-step visualization of mathematical recipe
2. If students construction the GeoGebra applet, they have to think about what the mathematical recipe means to construct the visualization.
4. See the effects of error accumulation vs big step sizes.
Possible Explorations

- How do we know when to stop making the step size smaller?
- \( y' = 1 + y \) with positive \( y \)-values in the initial condition.
- \( y' = 1 + y \) with negative \( y \)-values in the initial condition.

For an example of a finished GeoGebra applet applying Euler’s method, see Euler.ggb.
Rough Version

1. Create a function of two variable $x$ and $y$.
2. Create a starting point.
3. Set a step size (by hand or by calculation from an ending point).
4. This is an iterative process, so show the Spreadsheet view.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x(IC)</td>
<td>y(IC)</td>
<td>(A1, B1)</td>
<td>yprime(C1)</td>
</tr>
<tr>
<td>2</td>
<td>A1 + stepsize</td>
<td>B1 + D1 stepsize</td>
<td>(A2, B2)</td>
<td>yprime(C2)</td>
</tr>
</tbody>
</table>

5. Fill down a few 10s of rows.
Hiding Some Steps

1. Create a new slider running from 1 to the number of rows you have in your spreadsheet. Call it MaxSteps.
2. Show the Spreadsheet view.
3. Make column E count the rows. (Put 1 in first row, then put =E1+1 in the second, and then fill down.)
4. Pull up Object Properties for C2 and go to the Advanced tab.
5. By Condition to Show Object type E2<=MaxSteps and hit return and then close the dialog box.
6. Now select C2 and fill down the C column again.
7. Hide the Spreadsheet view again.

Now you can click on the MaxSteps slider and use the arrow keys on your computer keyboard to step through the iterations. You can also hide iterations beyond the number needed to reach the target \(x\) value.
Showing the Effect of Changing Step Size

Using the stepsize slider we can see the effect of step size roughly. We can make it easier though.

- Insert a checkbox.
- Just create the caption for now.
- In cell F2 type segment[C1,C2] and hit return or enter.
- Go to the Advanced tab of Object Properties and type E2<=MaxSteps under Condition to Show Object. The ^ character needs to come from the symbols table which shows if you click on the α at the right side of the entry area.
- Go to the Basic tab of Object Properties and check Show Trace.
- Fill down so there is a value in F for every row that has other values except the first one.
Set Up Differential Equation

1. Type \( y' = 1 + y \) into the input field at the bottom of the GeoGebra screen.

2. Hit the return or enter key.

3. I've used the up-arrow key in the input field to show both what I typed and what happens after I hit return.

\[
y'(x, y) = 1 + y
\]
Unlike functions of one variable where GeoGebra allows us to enter only the function, with multivariable functions we must explicitly tell GeoGebra what the variables are by typing something along the lines of $f(x, y) =$ into the input field before typing the actual function.

The graphics view really is empty. We would need the Slopefield command to visualize this differential equation.
Establish Initial Condition

The initial condition is a point, so create a free point.

1. Go to the Points menu and select New Point.

2. Click in the Graphics window anywhere but on an axis. (We can move the point onto an axis later if we like, but we don’t want to be limited to initial conditions on one of the axes.)
Naming Initial Condition

This point is the initial condition so relabel it.

1. Control-click the point in the graphics view or its name and other info in the algebra view and select Rename.

2. Type IC and hit return. At the end we can rename again to the clearer InitialCondition but meanwhile we have less typing to do.
Moving

1. Set the stepsize $\Delta x$ by typing `stepsize=0.1` in the input bar and hitting return.

2. Choose the $x$ value at which we want to approximate the $y$-value by attaching a moveable point to the $x$-axis.
   - Go to the Points menu and select New Point.
   - Click somewhere on the $x$-axis.

   By default, the point will be light blue indicating that you can move the point but that the movement is restricted to the object the point is currently on. (Compare to the color of the IC point.)
Euler’s Method so far

Multivariable Function

\[ y'(x, y) = 1 + y \]

Point

\[ A = (5.33, 0) \]
\[ IC = (2.52, -1.82) \]
First Iteration

- Show the spreadsheet.

- Get the value of our initial guess. Find the corresponding $y$-value. Make a point there. Draw tangent line to function at that point.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x(I0)</td>
<td>y(I0)</td>
<td>(A1, B1)</td>
<td>yprime(I0)</td>
</tr>
</tbody>
</table>

What doesn’t show is that I started the typing in each cell with $=$. To get this view, use

Using Options -> Algebra View -> Value shows

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.52</td>
<td>-1.82</td>
<td>(2.52, -1.82)</td>
<td>-0.82</td>
</tr>
</tbody>
</table>
Second Iteration

- Commands:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x(IC)</td>
<td>y(IC)</td>
<td>(A1, B1)</td>
<td>yprime(C1)</td>
</tr>
<tr>
<td>2</td>
<td>A1 + stepsize</td>
<td>B1 + D1 stepsize</td>
<td>(A2, B2)</td>
<td>yprime(C2)</td>
</tr>
</tbody>
</table>

- Only the first two columns needs to be typed. Select cells C1-D1 and then move mouse to the bottom right corner of the selection. Click on the corner and drag down one row (fill-down).

- Values:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.52</td>
<td>-1.82</td>
<td>(2.52, -1.82)</td>
<td>-0.82</td>
</tr>
<tr>
<td>2</td>
<td>2.62</td>
<td>-1.9</td>
<td>(2.62, -1.9)</td>
<td>-0.9</td>
</tr>
</tbody>
</table>
More Iterations

Select cells A2-D2 and drag the bottom right corner down several rows to get more iterations. Don’t get too carried away - GeoGebra will get slow.
If No New Points Appear in Graphics

- Select all of all but the first row of column C by clicking in C2 and dragging down.

- Now Control-Click in the resulting blue box and select Object Properties.

- Click so the Show Object checkbox is blue and has a checkmark.
Cleaning Up

- The labels on the estimates aren’t very useful. Hide them by selecting all but the first row of column C and then Control-Clicking inside the selection region and clicking Show Label to unselect it. (If some show and some don’t show labels, you may have to repeat this or go to the Basic tab of Object Properties to get everything to behave.)

- Make it easier to see whether we have gotten to our target \( x \)-value by putting a vertical line through point \( A \).
  - Click in the Graphics view.
  - Select the Perpendicular Line tool.
  - Click on point \( A \). Then click on the \( x \)-axis.
  - Control-click on either the line definition in the Algebra view or the line itself in the Graphics view and select Object Properties to change color or line thickness (under the Style tab).
Prettification

- Insert an Input Box to let the user set the function. Use Control-Click to call up the Object Properties and in the Basic section select Fix Object so the input box will stay in one place on the screen as you pan and zoom the graph.

- Insert a slider for the step size.
  
  1. Find stepsize in the Algebra view and click on the empty radio button next to it. A slider will appear somewhere in the graphics view.
  
  2. Move the slider where you want it to be by clicking on the bar (not the dot) and dragging into position.
  
  3. Control-click on the slider and choose Object Properties. Go to the slider tab. Fix the Min, Max, and Increment values. (Negatives are fine, but an increment of 1, the default, is too big.)

- Hide the Spreadsheet and Algebra views by unchecking them.
**Hiding Some Steps**

1. Create a new slider running from 1 to the number of rows you have in your spreadsheet. Call it MaxSteps.

2. Show the Spreadsheet view.

3. Make column E count the rows. (Put 1 in first row, then put =E1+1 in the second, and then fill down.)

4. Pull up Object Properties for C2 and go to the Advanced tab.

5. By Condition to Show Object type E2<=MaxSteps and hit return and then close the dialog box.

6. Now select C2 and fill down the C column again.

7. Hide the Spreadsheet view again.

Now you can click on the MaxSteps slider and use the arrow keys on your computer keyboard to step through the iterations. You can also hide iterations beyond the number needed to reach the target $x$ value.
Showing the Effect of Changing Step Size

Using the stepsize slider we can see the effect of step size roughly. We can make it easier though.

- Insert a checkbox.

- Just create the caption for now.

- Show the Spreadsheet view.
- In cell F2 type \texttt{segment[C1,C2]} and hit return or enter.
- Go to the Advanced tab of Object Properties and type \( b \ \backslash \ E2<\text{MaxSteps} \) under Condition to Show Object. The \( \backslash \ E2 \) character needs to come from the symbols table which shows if you click on the \( \alpha \) at the right side of the entry area.
- Go to the Basic tab of Object Properties and check Show Trace.
- Fill down so there is a value in F for every row that has other values except the first one.
The built-in numerical solver is probably (documentation is unclear) based on Runge-Kutta rather than Euler or Improved-Euler because Runge-Kutta is more accurate. However, Runge-Kutta is more complicated so we usually teach the other two first and leave Runge-Kutta for a class specifically on differential equations (typically the 4th semester in the calculus sequence). For an example of a finished GeoGebra applet using slopefield and solveODE commands see Slopefields.ggb. For a version that does better with zeroes in the denominator of the derivative, see SlopefieldsSafer.ggb.
Conceptual Differences

1. Use Euler’s method to predict the next point.
2. Average the slopes at the previous point and the Euler’s new point.
3. Calculate and improved next point still as a linearization but using the average slope just calculated.
Revised Formulas

\[ z_n = y_{n-1} + y'(x_{n-1}, y_{n-1}) \Delta x \]  \hspace{1cm} (5) \\
\[ y_n = y_{n-1} + \frac{y'(x_{n-1}, y_{n-1}) + y'(x_n, z_n)}{2} \Delta x \]  \hspace{1cm} (6)
### Revised Spreadsheet

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x(IC)</td>
<td>0</td>
<td>y(IC)</td>
<td>(A1, D1)</td>
<td>yprime(E1)</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>A1 + stepsize</td>
<td>D1 + F1 stepsize</td>
<td>yprime(A2, B2)</td>
<td>D1 + (F1 + C2) / 2 stepsize</td>
<td>(A2, D2)</td>
<td>yprime(E2)</td>
</tr>
</tbody>
</table>
Conceptual Differences

1. The approximation is parabolic rather than linear (Simpson’s rule for estimating integrals).

2. The slope at the middle point is estimated using both Euler’s method and Improved Euler’s method.
Revised Formulas

\[ k_{n1} = y'(x_n, y_n) \]  \hspace{1cm} (7)

\[ k_{n2} = y' \left( x_n + \frac{\Delta x}{2}, y_n + \frac{\Delta x}{2} k_{n1} \right) \]  \hspace{1cm} (8)

\[ k_{n3} = y' \left( x_n + \frac{\Delta x}{2}, y_n + \frac{\Delta x}{2} k_{n2} \right) \]  \hspace{1cm} (9)

\[ k_{n4} = y' \left( x_{n+1}, y_n + k_{n3} \Delta x \right) \]  \hspace{1cm} (10)

\[ y_{n+1} = y_n + \frac{\Delta x}{6} (k_{n1} + 2k_{n2} + 2k_{n3} + k_{n4}) \]  \hspace{1cm} (11)
Revised Spreadsheet

Column B contains the $x$-coordinates $\Delta x$ apart.

- yprime(B1, N1)
- yprime(B1 + dx / 2, N1 + J2 dx / 2)
- yprime(B1 + dx / 2, N1 + K2 dx / 2)
- yprime(B2, N1 + L2 dx)
- N1 + dx / 6 (J2 + 2K2 + 2L2 + M2)

To see how much more quickly Runge-Kutta converges than Euler or Improved Euler, use Euler-ImpEuler-RungaKutta.ggb.
Explorations

- When can you guarantee that a left (right) Riemann sum will underestimate (overestimate) the actual value of the definite integral?
- Compare the behavior of the various sums as the number of intervals increases.
- When is Simpson’s rule NOT an improvement on the trapezoid rule?
- How would you decide when to use Simpson’s rule and when to use the trapezoid rule?

An example GeoGebra applet containing left, right, upper, lower, trapezoid rule, and Simpson’s rule approximations is NumericalIntegration.ggb.
Function and Endpoints

1. Create the function. Type what shows in the input box and the pieces in the Algebra view and the Graphics view will show when you hit return (or enter).

2. Create left and right edges of the interval.
   1. Go to the Points menu and select New Point.
   2. Click somewhere on the $x$-axis.
   3. Repeat to the right of the previous point. (Which side only matters for making the instructions simpler.)
**Break into Subintervals**

1. Choose the slider tool.

2. Click somewhere in the Graphics view that is likely to be out of the way.

3. Set the minimum value to 2 and the increment to 2 as well.

4. Control-click on the slider and make sure that Absolute Position on Screen is checked.
Built-in Methods

Type the following one at a time into the input bar at the bottom of the GeoGebra window.

- `LeftRiemann=LeftSum[f,x(A),x(B),subintervals]`
- `LowerRiemann=LowerSum[f,x(A),x(B),subintervals]`
- `UpperRiemann=UpperSum[f,x(A),x(B),subintervals]`
- `TrapRule=TrapezoidalSum[f,x(A),x(B),subintervals]`
Cleaning Up 1

The view is a mess now. We’d like to compare the values of the 4 methods and to control how many of them we see at one time. It might also help to vary the colors.

- Insert text in the Graphics view.

- For each of the four methods so far type a description in the top of the dialog box and then use the Object menu to select the relevant object. The bottom of the dialog box gives a preview.

- Control-click the text in the graphics view and select Object Properties. Go to the Position tab and check the Absolute Position on Screen box to prevent losing the text as you pan and zoom.
Cleaning Up II

- Control-click each of the sums and turn off Show Label.
- Click the arrow by the label at the top of the Graphics view. Now click each sum in turn and choose colors. The same will work for the function.
- Create a checkbox for displaying each of the four sums.
- Go to the Basic tab of Object Properties and click Fix Checkbox for each checkbox so they don’t get lost.
Right Sum from Scratch

1. Create points on the axis for the bases of the rectangles by typing `bottoms=Sequence[(x(A)+n*dx,0),n,0,subintervals]` in the input bar.

2. Create points on the function for all the top right corners.
   
   `topRight=Sequence[(x(Element[bottoms,n]),f(x(Element[bottoms,n]))),n,2,subintervals+1]`

3. Create top left corners of rectangles.
   
   `topLeft=Sequence[(x(Element[bottoms,n-1]),f(x(Element[bottoms,n]))),n,2,subintervals+1]`

4. Draw the rectangles.
   
   `RRRect=Sequence[Polygon[Element[bottoms,n],Element[bottoms,n+1],Element[topRight,n],Element[topLeft,n]],n,1,subintervals]`

5. Add up the areas. `RightRiemann=Sum[RRRect]`

6. Hide the corners by clicking the radio button left of each list so the button is hollow. Now the view is the same as for the other sums.
Simpson’s Rule

(Don’t type the line breaks into GeoGebra.)

1. Put points on the function for all the bases.
   
   \[
   \text{tops} = \text{Sequence}[(x(A) + n \times \text{dx}, f(x(A) + n \times \text{dx})) , n, 0, \text{subintervals}]
   \]

2. Fit parabolas (quadratic functions) through sets of 3 points starting at the left of the interval. The right point on one parabola is the left point on the next parabola.
   
   \[
   \text{parabolas} = \text{Sequence}[(\text{FitPoly}[[\text{Element[tops, 2n - 1]], \text{Element[tops, 2n]], \text{Element[tops, 2n + 1]]}, 2]) , n, 1, \text{subintervals / 2}]
   \]

3. Calculate the area from Simpson’s rule.
   
   \[
   \text{Simpson} = \text{dx/3} \times (2 \times \text{Sum[Sequence[y(Element[tops, 2n - 1]], n, 1, subintervals / 2 + 1]]} + \\
   4 \times \text{Sum[Sequence[y(Element[tops, 2n]], n, 1, subintervals/2]]} - (y(\text{Element[tops, 1]}) + y(\text{Element[tops, subintervals + 1]]}))
   \]

4. Show the areas being added up.
   
   \[
   \text{Sequence[Integral[Element[parabolas, n]], x(\text{Element[tops, 2n - 1]]), x(\text{Element[tops, 2n + 1]]), n, 1, subintervals / 2]}
   \]

5. Hide the corners and the full parabolas so we just see outlines like the original four sums.

6. Add these sums to the text box (double-click to edit) and to the set of check boxes.
Better Color Coordination

We can make the text values and the check boxes color coordinate with their drawings.

1. Use the Color tab in Object Properties to make the check box captions match.
2. Double click the text box to edit it.
3. Click the box next to “LaTeX formula”.
4. Add $\textcolor{100,100,100}{}$ to the beginning of each line and $\}$ to the end of each line. Replace the values 100,100,100 with the triple of numbers that comes up in the Color tab of Object Properties Color for any object (so you can match accurately).
Difficulties

- convergence of sequence versus convergence of sum
- conditional convergence of alternating series
Illustrating an Infinite Sum

1. Create a function.
2. Create a variable that says how many terms from the beginning of the infinite sequence we will examine.
3. Create points representing the individual terms in the sum.
\[
aSequence = \text{Sequence}[(n, f(n)), n, 1, n_{\text{max}}]
\]
4. Create points representing the sum as each term in the sequence is added – the sequence of partial sums.
\[
aSequence_{\text{Partial Sums}} = \text{Sequence}[(n, \text{Sum}[\text{Sequence}[y(Element[aSequence, i]), i, 1, n]], n, 1, n_{\text{max}}]
\]
Explorations

- Find a sequence whose terms diverge. What happens to the sequence of partial sums? What does this tell us about the infinite sum?
- Find a sequence whose terms converge to a non-zero value. What happens to the sequence of partial sums? What does this tell us about the infinite sum? Does it matter whether or not the non-zero value is positive or negative?
- Find a sequence whose terms converge to zero but for which the sequence of partial sums diverges.
- Find a sequence whose terms converge to zero and but for which the sequence of partial sums converges to a non-zero value. Is it possible for the partial sums to converge to zero.
Conditional Convergence

One of the surprising properties of conditionally convergent series is that re-ordering the infinite list of terms can produce convergence to an arbitrary value. The GeoGebra applet ConditionalConvergence illustrates this, and its failure for an absolutely convergence series, with the two series $\frac{1}{n}$ and $\frac{1}{n^2}$. 