The Official Guide to Easy Graphing

ParaGraph

Ver. 6.3.5

Congratulations – Graphing just got easier.

Here's to putting a parametric grapher in *your* hands.

Matthew Vande Burgt

Vande Burgt Productions

8/8/2010

Table of Contents

A. **Introduction**

1. What is ParaGraph?
2. System Requirements
3. Terms of Use

B. **Getting Started with ParaGraph**

1. Installing ParaGraph
2. Creating a GPH File Association

C. **ParaGraph as a…**

1. Parametric Graphing Utility
    a. Plotting a Graph
    b. Adjusting the Viewing Range
    c. Performing Calculus
    d. Showing an Animation
    e. Transforming an Axis
    f. Changing the Visual Options
    g. Default Settings
    h. Printing
2. Calculator
    a. Evaluating an Expression
    b. Changing the Answer Output
    c. Using Variables
    d. Using Equations
    e. Creating Equations
3. Function / Data List
    a. Function Entry

b. Data Entry

c. Plotting Data

d. Importing Lists

e. Exporting Lists

f. Data Fitting

4. Vector Calculator

D. Program Functionality at a Glance

1. Menu Options
2. Mathematical Functions
3. Mathematical Operators
4. Mathematical Constants
5. Shortcut Keys
6. Tips and Tricks

E. Concluding Remarks

1. A History of ParaGraph
2. Sources
3. Special Thanks

A. Introduction

Mathematics and science are two ever-growing bodies of knowledge, becoming increasingly complex in today's age of computing. The passing of time brings about more and more tasks to perform in order to get important things accomplished, including both learning and doing math and science. However, many of these tasks can become time consuming and even difficult for those trying to perform mathematical operations without the use of proper computing software.

Over time, the drive for computer programmers to create free software for others to use is becoming greater and greater. This is partly due to the fact that these programmers need their own software to get their own work done and that they enjoy the feeling of solving complex problems. Another reason is that they enjoy watching others benefit freely from a homemade creation, thus being able to use, without cost, a new and enjoyable program that influences their lives in a positive way.

This user's guide is designed to help those who are both new to the world of mathematics as well as those already knowledgeable in math, and simply need to know how to use the program. ParaGraph, however, is designed to be an intuitively simple program, so users who are comfortable with computers are encouraged to treat this document as a reference for program specific items, such as function/constant lists specific to this program. However, all users are recommended to read the installation section of this manual, as well as the Terms of Use, as they contain vital information for the use of this program.

1. What is ParaGraph?

ParaGraph is a free two-dimensional parametric graphing utility, written with Microsoft Visual Basic 6.0, and is geared toward the ease of plotting and working with parametric equations. While most people are accustomed to using dependant/independent variable equations (y = f(x)), parametric equations are more versatile sets of parameters – one parameter describing x, and one describing y. In ParaGraph, all parameters are functions of z, so the two parameters are x = f(z) and y = f(z). This will be discussed more later on.

Besides acting as a graphing utility, ParaGraph comes with a versatile calculator, data storing abilities, and a vector calculator. These four functions of ParaGraph (graphing, calculating, data storage, and vector math) trump most graphing calculators' quality due to ParaGraph's speed, the pixel resolution of any standard computer, and the additive functionality of many, many features not seen in most graphing software.

2. System Requirements

One of the great qualities of ParaGraph is that it demands very little from a computer's resources. ParaGraph runs best on Windows Vista and Windows 7, and can run on Windows XP, but will go much slower when performing demanding tasks, like animations.

The ParaGraph executable is only 157 KB, which is quite small considering what it is capable of. Likewise, the files ParaGraph creates are very small as well; a file saved with purely the default settings is barely over half a kilobyte, and will only increase significantly when extensive data is stored in ParaGraph's data lists or in the calculator history box.

3. Terms of Use

By using this software, you are agreeing to the following Terms of Use. If you do not agree with these terms, do not use this software and remove any material related to this software from your computer. In the following items, any person owning and/or using ParaGraph is referred to as "the user," while those behind the creation of ParaGraph (where applicable) and ParaGraph as a program is referred to as "ParaGraph."

a. ParaGraph is a free program. ParaGraph may be freely distributed, but under no exceptions shall copies of ParaGraph or this ParaGraph help file be sold for a profit.
b. ParaGraph may be used for any desired purpose (besides the formerly mentioned restriction).

c. The GPH and TXT files ParaGraph creates are property of the user. These files may be created, edited, and/or deleted as the user sees appropriate. These files may also be sold or given to any other person at the discretion of the user.
d. ParaGraph is not responsible for any damage done to the user or the user's computer. ParaGraph is not a malicious program, but any form of damage resulting from ParaGraph will not hold ParaGraph responsible.
e. ParaGraph is not responsible for any incorrect answers calculated.
f. Although the user may alter ParaGraph, the source code for ParaGraph is not directly provided with the software. To obtain the source code for ParaGraph, please contact the author of ParaGraph.
g. Any copies or alterations of ParaGraph or ParaGraph's code should but are not required to give credit to the original author of ParaGraph, and permission from the author of ParaGraph is not necessary to edit or copy all or parts of ParaGraph.
h. This help file is encouraged, but not required, to accompany any copies of ParaGraph. However, any user is subject to these Terms of Use regardless of whether he or she has read them.
i. Any alterations of ParaGraph must be noted as such, not claiming to be part of the original program.

B. Getting Started With ParaGraph

The following items are beneficial to know before using ParaGraph.

1. Installing ParaGraph

ParaGraph was designed to be easy to install. In essence, you could merely copy the executable file from whoever gave you ParaGraph and run it immediately without any trouble, even from a memory stick. However, for best results, ParaGraph should be copied to a place on your computer where it will not be moved, like placing it in a folder called "ParaGraph" in the directory C:\Program Files. This will ensure the success of some of the features available in ParaGraph. After placing ParaGraph in a permanent location, you may create shortcuts and pin them to the Start Menu or on your desktop.

2. Creating a GPH File Association

ParaGraph saves its files using a GPH extension. If you want to be able to open the files you save by double clicking on the file instead of opening the file from the application, you need to create a file association. This can be done very easily. To create a file association for GPH files, you must do the following:

- Open ParaGraph
- Click File -> Save
- Save a file in any place on your computer
- Close ParaGraph and locate your saved file in My Computer
- Double-click on your saved file
- Choose to select a program from a list
- Choose “Browse…”
- Locate and select ParaGraph.exe
- Make sure that the check box that asks whether you want to use this program every time is checked, and click OK
- ParaGraph should open, and if you find your saved file again, it should have a blue ball on a white page for an icon

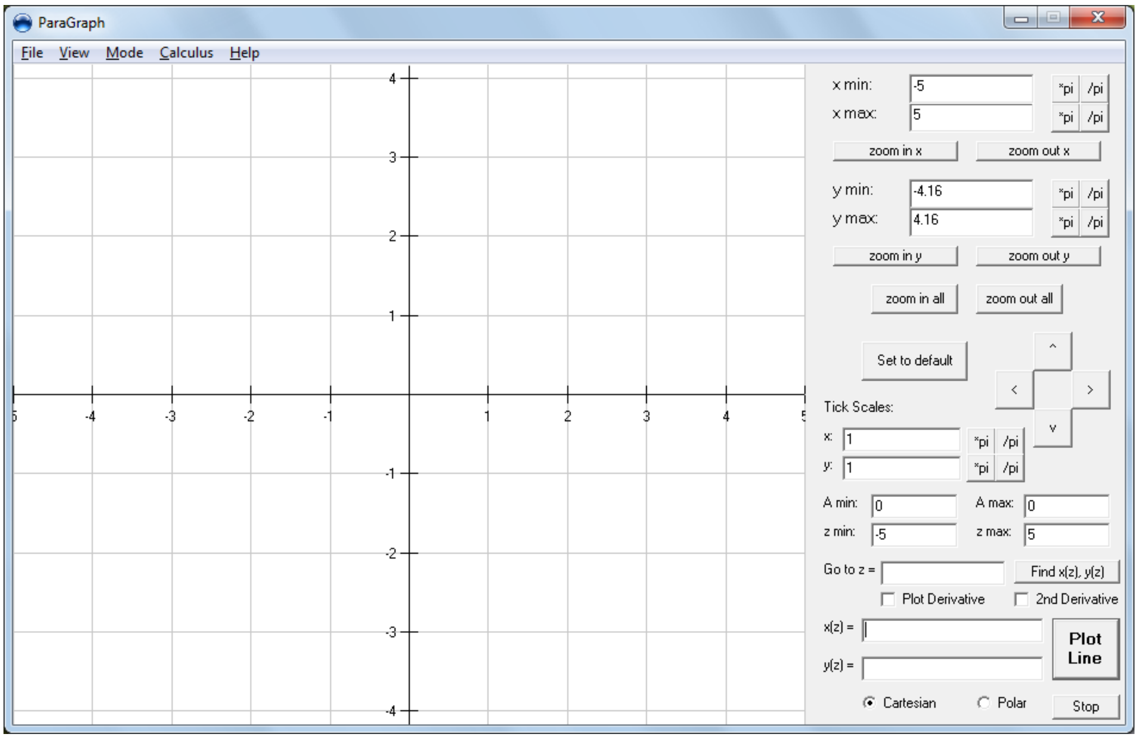
Now, every time you double click on any saved file, ParaGraph will start up with all of your data ready to go!

C. <u>ParaGraph as a…</u>

ParaGraph is a multi-functioned program, capable of performing the basic mathematical functions and operations up to and including methods of basic numerical calculus. There are four main components to ParaGraph: a parametric graphing utility, a calculator, a data storage unit, and a vector calculator. Each of these components is fully described in the sections below.

1. Parametric Graphing Utility

When opening ParaGraph with no initialize file, you should see this:



From this graphing page, you can perform the majority of the tasks ParaGraph is capable of. If ParaGraph is ever not in Parametric Grapher mode, it can be directed there either by pressing the *F1* key on your keyboard or by clicking the menu option *Mode -> Parametric Grapher.*

While in parametric graphing mode, if you move your mouse over the plotting area, a live coordinate display of where the mouse is will appear in the lower left corner of the plotting area. To show your coordinates in multiples of π, click on the text that says “Mouse Coordinates” and move your mouse around. The display coordinates should now be shown as multiples of π. To change back to regular display, click the text again.

a. Plotting a Graph

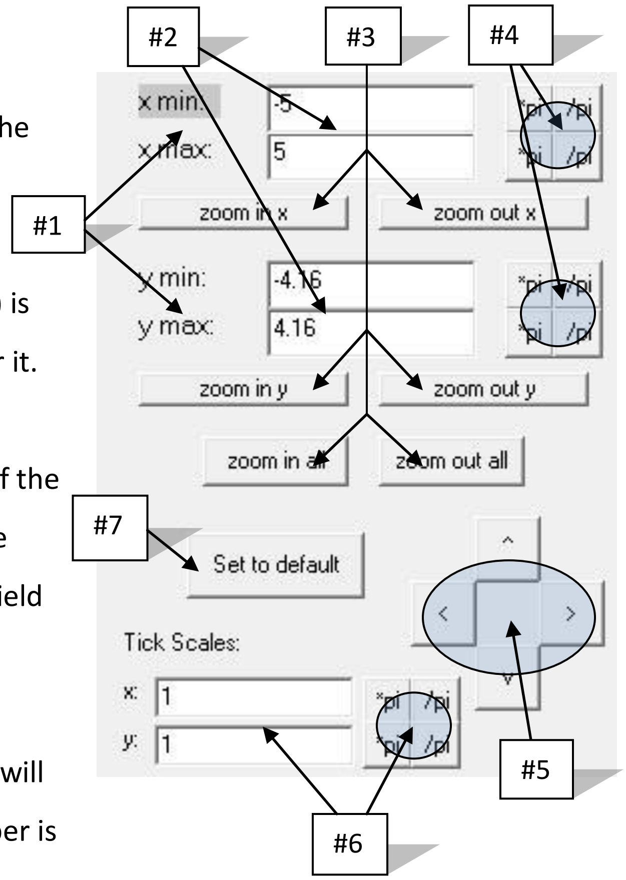
To plot any line, two items must be entered: a function for x and a function for y. The independent variable for these functions is the letter z (although if you are used to using an "x", don't worry; ParaGraph will replace that with a "z"). For example, to plot the line $y = 2sin(2x)^{3}$, you would need to type "z" in the x(z) box, and "2sin(2z)3" in the y(z) box, like the picture at right. Now, either press the Enter key on your keyboard, or click the "Plot Line" button at right. ParaGraph will insert multiplication signs and power signs (^) where needed, and proceed to plot the line.

To find the value of a certain point on your plot, enter a number in the text box marked "Go to z =" and press the Enter key on your keyboard or click the button that says "Find x(z), y(z)". A red crosshair will be drawn at the point you entered, and the coordinates for that point will display in the bottom right corner of the plotting area. Just like the display of the mouse coordinates, you can show these coordinates in terms of π. Click the text that says "Freeze x(z), y(z)" and hold the mouse there. For this case, the numbers will be shown in terms of π as long as you are holding the mouse button down on the specified text.

To plot in polar graphing mode, click the option button at the bottom of the window that says "Polar". You graph will be automatically changed into polar coordinates, with θ (theta) as a function of z and the radius (r) as a function of z.

You may adjust the z range by entering numbers in the z min and z max text boxes. This will tell ParaGraph how much of a range it should change the independent variable by. The quality of the graph is also dependant on the size between the minimum and maximum z values. If you ever think your graph looks a bit like a "connect the dots" picture, just adjust the z range values to decrease the space between them. This will lead to a smaller step between each calculated point on the graph.

ParaGraph can also set the z range for you. By default, this setting is active, but you can turn this off by clicking *View -> Auto z Range.* If this setting is on, ParaGraph will only set the z range when the x(z) function is just "z". Otherwise, no automation will occur. Therefore, any y = f(x) function will be able to have an automated z range, since for any y = f(x) function the equivalent parametric function for x(z) is equal to z. The only time ParaGraph will not automatically set a z range when x(z) is equal to z is if the polar graphing mode is turned on, or an axis transformation is active.



b. Adjusting the Viewing Range

ParaGraph has many ways of adjusting the viewing range. The picture at right shows the various controls that allow you to change the plotting range. The text displaying "x min:" (#1) is highlighted because the mouse is currently over it. This is because clicking that text is one way to change the viewing range. By left clicking any of the four text fields in group #1, you will increase the range by one tick mark. Right clicking the text field will do the opposite.

In group #2 on the picture, the user may manually type in a number range. The window will automatically re-plot the graph when any number is changed, so pressing the "Plot Line" button is not necessary.

Group #3 shows various zooming buttons. Each caption on the button describes what it will do, so zooming in your plot range is fairly straightforward.

Group #4 allows you to multiply or divide your current range numbers by π. For example, suppose you were graphing the line y = sin(x). To include exactly two repetitions of this wave, enter "-2" in the x min box and "2" in the x max box. Then, click the buttons that say "*pi" on them. Your plotting range should now be from -2π to 2π along the x axis. This includes two waves from the function y = sin(x).

Group #5 consists of a set of buttons you can use to shift the graph up, down, left, or right. Clicking one of these buttons will cause the graph to shift one tenth of its dimension range.

Group #6 controls how you scale your tick markings. For example, if you entered “0.5” for the x tick value, twice as many tick marks would appear along the x axis, since there would now be one tick mark for every half of a unit.

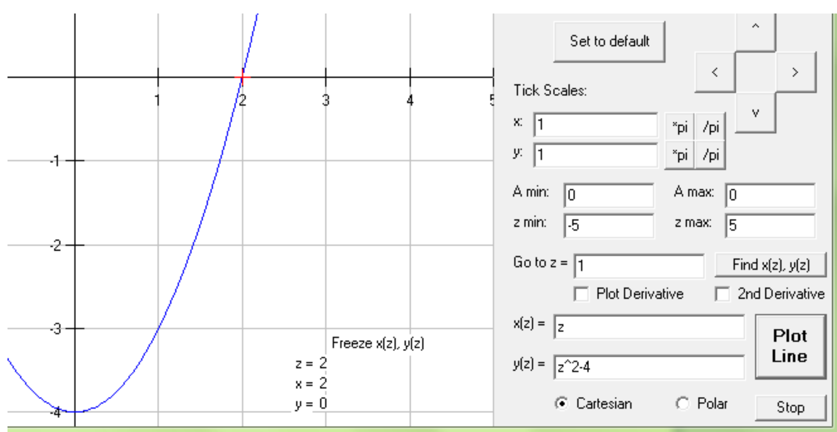
Another way to scroll the viewing range is to drag the mouse through the plotting region while holding down the left mouse button. This will scroll the plotting range according to where you move your mouse until you release the mouse. Alternatively, you may double-click the left mouse button anywhere on the graph to center the graph at that point.

Clicking and dragging the RIGHT mouse button across the graph display will create a zooming box that will activate once the mouse is released. This is a convenient way to quickly zoom in to a desired portion of the graph. Also, if you ever start dragging the mouse but then decide you don’t want to zoom in after all, simply click the left mouse button before you release the right mouse button, and the zooming box will disappear.

Finally, if you want to easily return your display back to its default values, simply click the “Set to default” button (#7). This button will change your x and y minimums and maximums, but your tick mark spacing settings will remain unchanged.

c. Performing Calculus

ParaGraph can do a few simple calculus based routines. The first thing you can do is to find where a function goes to zero. Although this is considered a non-calculus based routine, ParaGraph uses calculus to efficiently find the roots of a function. To find a root, enter a starting number in the text box marked “Go to z =”. Then, click the menu option *Calculus -> Find Root.* ParaGraph will follow the slope of the plotted line and converge where the function has a root. Here is an example image of ParaGraph finding where the equation y = x^2-4 goes to zero with a starting value of z =



1. The freeze coordinates show the z, x, and y values for the point. A red crosshair has also been drawn at the root. Note that sometimes the y value reported in the freeze text area will not quite be zero, but rather very close to zero. This is because the value didn't converge to zero with the allowed number of steps, but the z value is still very close and can be trusted to several decimal places.

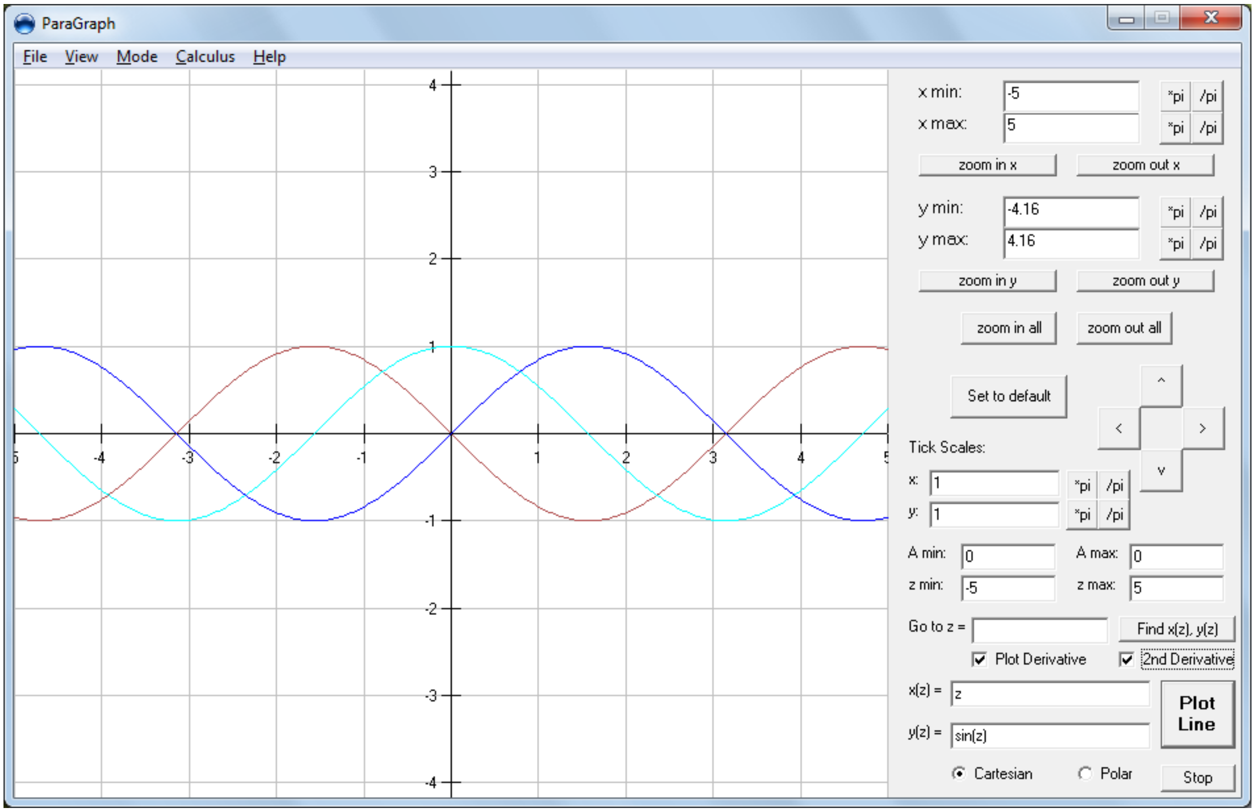
The next thing you can do with ParaGraph is to find where the slope of the function goes to zero. This is useful because a zero slope value most likely means a local maximum or a local minimum on your graph. Do this by once again making sure you have a starting value in the "Go to z =" text box, and then clicking *Calculus -> Find Local Extremum.* In the same way as finding a root, ParaGraph will follow the slope of the line searching for a zero slope point. When it finds one, it will put the freeze coordinates for that point up, and draw a red crosshair at the point.

ParaGraph can also find infection points. An inflection point is where the change of the slope of a function is zero. Inflection points are useful because they tell where the slope of the line has a maximum or a minimum. To find an inflection point, click *Calculus -> Find Inflection Point.* Finding an inflection point involves stepping through the same process as finding roots or extrema.

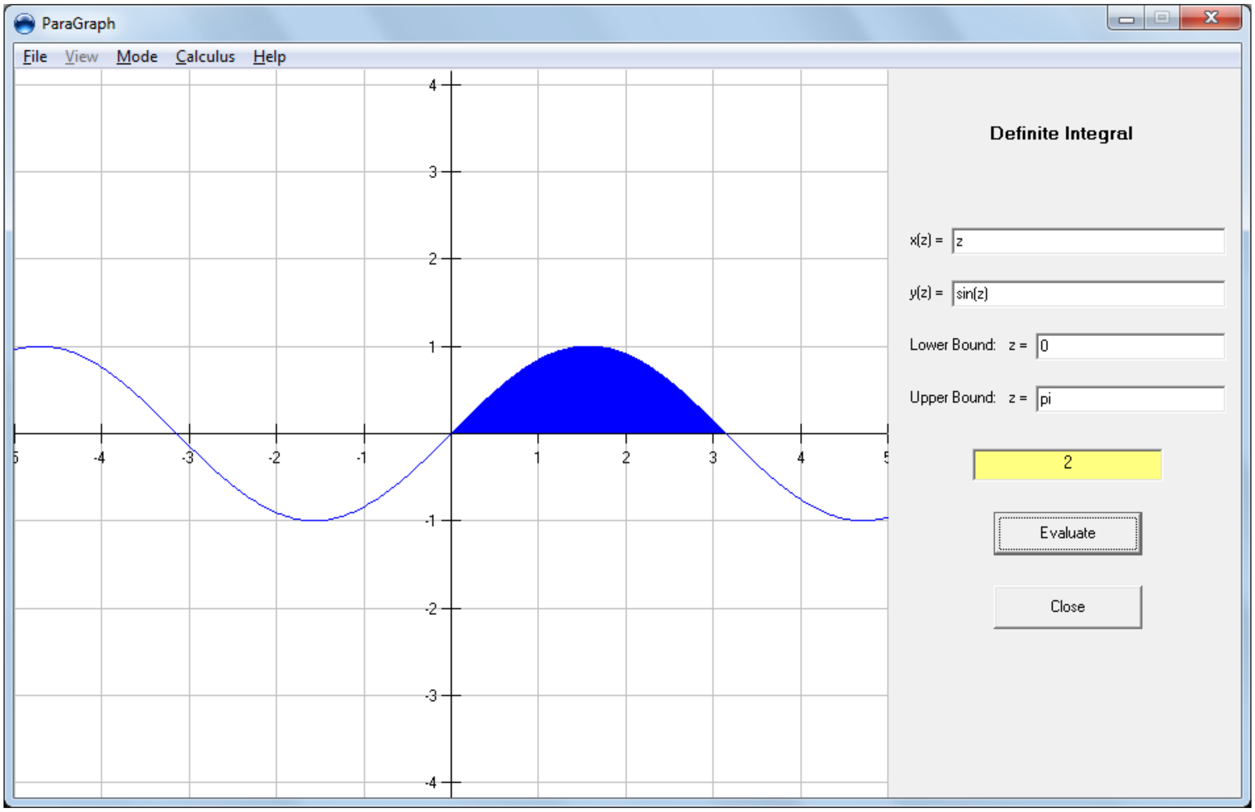
In the above processes, you may once again use the π multiplied display for the freeze coordinates by clicking the text that says "Freeze x(z), y(z)". Also, if, for example, a local maximum cannot be found, an appropriate message box will appear, and instead of plotting a point at an extremum, ParaGraph will plot a point where you specified your z value. If you know that there is indeed an inflection point (or the like), but ParaGraph cannot find it, try changing your starting value to something closer to what the final value might be.

Next, you can evaluate the first and second derivatives of a function. To do this, type a number in the text box marked "Go to z =" and click *Calculus -> Evaluate Derivative*, or *Calculus -> Evaluate Second Derivative*. Your evaluation will be displayed in a message box, and the point at which you calculated your derivative will be plotted with a red crosshair.

You may also graph the first and second derivatives of your function. To do this, click one or both of the check boxes saying "Plot Derivative" or "2nd Derivative". The new line will automatically be plotted. The screenshot below shows the function y = sin(x) with the first and second derivatives plotted. Notice that the two check boxes for plotting derivatives are checked.



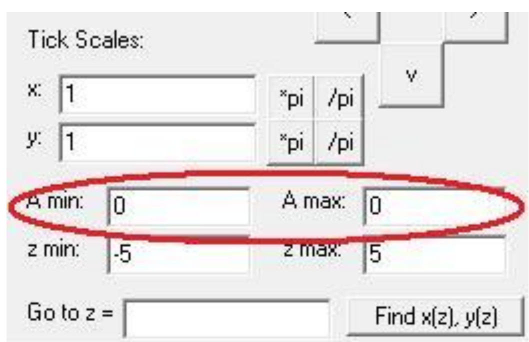
In addition to the above methods of calculus, ParaGraph can approximate the numeric integration of a function using Simpson's method of integration. To open the integration mini-page, press F5 on your keyboard or click the menu option *Calculus -> Evaluate Integral.* The buttons controlling the plotting area will be replaced with a simple integration page. If you had functions entered in the x(z) and y(z) boxes previously, they will be shown in the new x(z) and y(z) boxes. To find an integral, enter the lower and upper limits in the appropriate boxes and click the "Evaluate" button. The result will be displayed in the yellow text box, and the area you just integrated will be filled in on the graph. To find your result in terms of π, click and hold the yellow text box. Below is an example of finding the area under the curve y = sin(x) from 0 to pi.



To get back to the regular graphing page, press F5 again, click the "Close" button, or click the menu option *Calculus -> Evaluate Integral.*

d. Showing an Animation

To show an animation, you need to enter a low and high value for a variable called A. This can be done from the main window, as is seen in the picture at right. When ParaGraph plots a line, it will run an animation if either of these values is not zero, so if you do not want an animation to run, ensure that these value are zero.



ParaGraph's animation works by changing a constant within your functions periodically to produce a changing line. For example, say your x(z) function was equal to z, your y(z) function was equal to z + A, and your A values ranged from 0 to 3. When you plot your line, you will get a diagonal line that starts with a y intercept at 0, but moves up until the y intercept is 3, where it stops.

While the animation is running, you may pause it by holding down the “Plot Line” button (during an animation this button says “Pause”). To stop the animation, click the “Stop” button. This will stop the animation and also reset your current A value so that when you start the animation again, the animation will start from the beginning.

Since animating a function causes that function to be plotted multiple times, a bit of a demand is placed on your computer when using this procedure. Computers using Windows Vista or Windows 7 will step through these animations much more quickly than computers running XP. Whether this is due to the efficiency of the Windows graphics calls or simply because computers with newer operating systems have better processing power is currently unknown.

e. Transforming an Axis

Some types of data or functions are better represented when an axis transformation has taken place. ParaGraph has two types of axis transformations: log and inverse. When these are used, either a logarithm (base ten) or an inverse (1/x) is applied to the function and/or data being graphed. These transformations may be in any combination of the x and y axis, but two transformations may not be used on the same axis at once. To transform an axis, click the menu option *View -> Axis Transformation*, and then choose from the four options. If you transform an axis with a logarithm, the tick marks will be spaced according to a log scale, and you will not be able to change the tick spacing for that axis. Also, the plotting ranges on the right side of the window will always “act” like you have no axis transformations turned on, even though the real ranges will be different. For example, if your x range is from -5 to 5 with no axis transformation, it will still read -5 to 5 when you use a logarithmic transformation, but your real range will be from 0.00001 to 10000, which are the values for 10^{-5} and 10^{5}, respectively.

f. Changing the Visual Options

You can change many of the colors ParaGraph uses when plotting various things. A color selection box becomes available if you click the menu option *View -> More Visual Options.* From here, you may change eleven of the color settings ParaGraph uses. Also, you may set the line thickness by entering a number from 1 to 11 in the text box on the right. As you may notice, using odd numbers for the line thickness gives better image quality than using even numbers, since odd numbers guarantee a symmetric pixel pattern within the lines. For an interesting color display, try typing "color" into the x(z) text box and pressing the "Plot Line" button.

Underneath the entry box for the line thickness is a check box controlling whether the numbers by each tick mark are visible. Uncheck this box if you do not want to see each number by its corresponding tick mark. There is also a second check box that controls the visibility of the grid lines displayed on the graph.

g. Default Settings

When you click the "Set to default" button, the plotting range sets itself back to the default settings for ParaGraph. However, you can change what those reset values will be. If you want values other than the ones provided, click the menu option *File -> Set Default Ranges.* ParaGraph will ask you if this is what you want to do. If you proceed, your current range values will be stored for whenever you click the "Set to default" button. These values will remain in your current file only, so if you save that file, those values will be saved, but they will not be the default ranges when you start a new ParaGraph document. In this way, you can have multiple documents with their own unique default ranges.

If you want to change ParaGraph's default ranges so that they are there even on startup, or if you ever find yourself constantly changing ParaGraph to other settings when you run the program, you may want to create an initialize file. This can be done by clicking the menu option *File -> Create Initialize File.* When you click this option, ParaGraph will prompt for a confirmation, and if you choose to proceed, a text file called INIT.txt will be created in the same directory ParaGraph.exe is in. However, you do not need to see this document (unless you wish to delete the file to restore the default settings). Upon startup, ParaGraph will check for this document to load its settings. If INIT.txt exists, your settings will automatically be loaded every time ParaGraph is opened. Everything you can change in ParaGraph will be recorded here; in fact, INIT.txt exists with exactly the same data format as a .GPH file, but it is created as a text file to set it apart from any other ParaGraph related document.

If you choose to create another initialize file, the old one will be replaced with the new one, so the only way to have two initialize files would be to rename INIT.txt before creating a new one, but only the text file specifically named INIT.txt will be loaded upon startup.

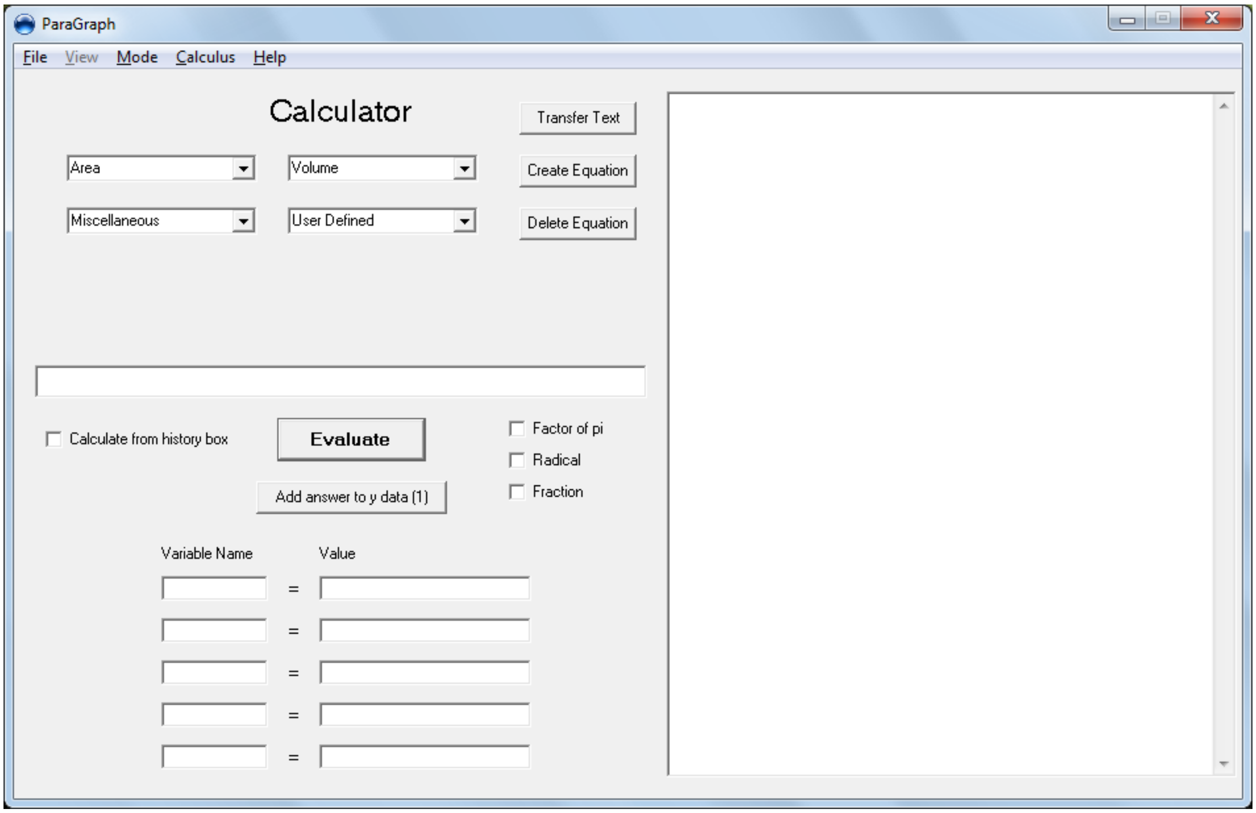
h. Printing

To print directly to a printer, click the menu option *File -> Print.* A simple printing dialogue box will appear, with the option of which printer you wish to print from, how many copies you would like, as well as whether you want your x(z) and y(z) equations to be printed along with the picture of your graph. The printer you are printing from must be directly connected to your computer; network printers will not print from this dialogue box.

This method of printing is quite limited. If you want to print to a network printer, or if you are having trouble printing from the *Print* menu option for whatever reason, it is recommended that you print the save the image as a bitmap file. This can be done by clicking the menu option *File -> Save as Bitmap.* ParaGraph will let you choose where to save the file, and you may print it off outside of ParaGraph. Bitmaps tend to take up quite a bit of memory, so you may want to open that saved bitmap and resave it as a JPEG file.

2. Calculator

The second main component of ParaGraph is its calculator page. To set ParaGraph to calculator mode, click the menu option *Mode -> Calculator,* or press F2 on your keyboard. Upon entering calculator mode, you should see this:



a. Evaluating an Expression

To find an answer to any expression, simply type your calculation in the text box on the left side of the window. Then, either press the Enter key on your keyboard or click the "Evaluate" button directly underneath the text box. Your expression and answer will be displayed in the history box at right. Also, you may type notes or anything else in the history box besides merely showing your calculations.

To use the last answer ParaGraph has calculated, you may type the text "ans" in an expression, and ParaGraph will use the last answer it has obtained for that number. Also, if you type "+" when the text entry field is blank, "ans" will be inserted before your "+" to allow you to utilize your last answer quickly. The same holds for all of the other operators, except for "-".

To have ParaGraph insert “ans” before a minus sign, type “-” twice, and ParaGraph will insert “ans” with a single minus sign. This is required because we don’t want ParaGraph to unnecessarily insert “ans” when we are merely trying to type “-2” (negative two).

Another way to evaluate an expression is to type it in the history box at right. Then, if the check box labeled “Calculate from history box” is checked (unchecked is the default value), the line of text where your cursor is will be evaluated when you press the Enter key. If you want to type something and merely have the Enter key tell the cursor to go to a new line, uncheck that box.

The button located just below the evaluation button says “Add answer to y data (1)”. If you click this button, ParaGraph will add your most recent answer to your active data list on the page from list mode. The caption on this button changes depending on which list is your active list, but by default, the caption refers to y data (1) because that is your default active list.

For some fun quotes, try typing “computers”, “mathematics”, or “physics” into the calculation text box and evaluating that.

b. Changing the Answer Output

Often times, the answer you get from a calculation will be best expressed as a fraction, a factor of π, or perhaps as a radical (a square root form). ParaGraph can do all three of these, as well as any combination of them at once. For example, try entering “4pi/6” in the calculation text box, and check the boxes that say “Factor of pi” and “Fraction”. After evaluating the expression, you can see that both the decimal number has been calculated, but the fractional number is underneath it, along with the π multiplication. That text should read “(2 / 3) * pi”. Note that if a fraction cannot be found when the check box for fractions is checked, ParaGraph will only display the decimal form of the number.

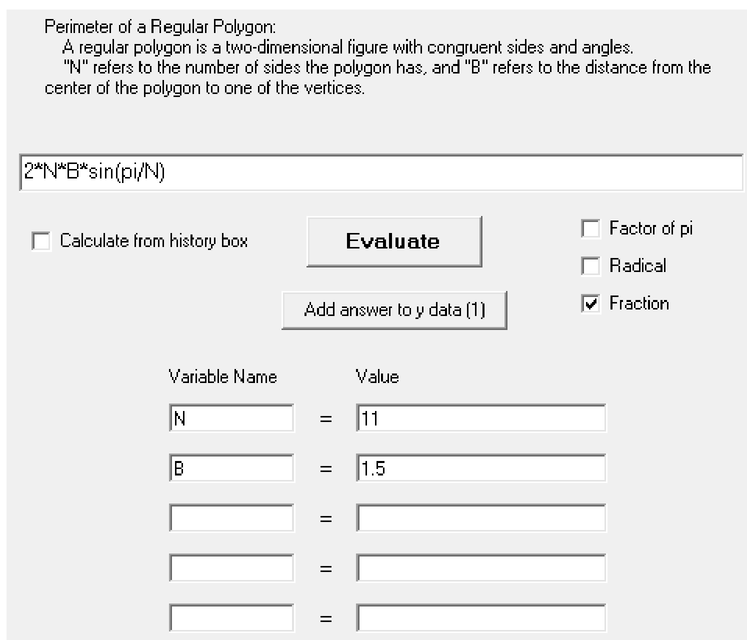
Here is another example. Try evaluating the expression “sqrt(80)” with the check box for expressing your answer as a radical checked (uncheck the other two if they are checked). When you evaluate your answer, the radical form of the answer should read “4 * sqrt(5)”, which is equivalent to the square root of 80.

c. Using Variables

The set of text boxes underneath the evaluation box is for storing numbers with variable names. To use variables, type the name of a variable in one of the left boxes, and its value in the corresponding right box. Then, use that variable in your expressions, and your answer will be evaluated accordingly. This may be useful when you have one main equation to perform with a set of changing parameters. Note that it is not a good idea to use single lowercase letters to define variables, since they will take precedence when used in cases such as “sin” or “tan”. It is a good idea to use words or uppercase letters when defining variables.

d. Using Equations

ParaGraph’s calculator comes with 44 predefined equations for your convenience. These equations cover a range of uses, like the quadratic formula, or the area of an ellipse. The equations are divided up into three groups: Area, Volume, and Miscellaneous.



Upon clicking on a desired equation, several items will appear. First, a description of which equation you are using will appear just above the calculation text box. Second, the equation with its associated variables will appear in the calculation text box. Third, the variables to which you must assign values will appear in the boxes underneath the calculation text box. The picture at right shows an example of using an equation to find the perimeter of a regular polygon. When using the parameters 11 and 1.5 for N and B, the perimeter is calculated to be 9.297 units long.

e. Creating Equations

If you aren't content with the built-in equations, ParaGraph lets you create your own equations, which work in the same way the built-in ones do once they are created. Below is an example of how to create your own equations.

Let's say you want to calculate how long it will take you to read any given book. The two things you would likely need are how many words are in a particular book, and how many words you can read each minute. Your formula could be "(pages * wpp) / wpm / 60" (the number of pages in a book times the words per page divided by the words you can read per minute, all divided by 60 to convert to hours). Say you know that you can read 300 words per minute, and a certain book you are looking at has 430 pages, with about 380 words on each page.

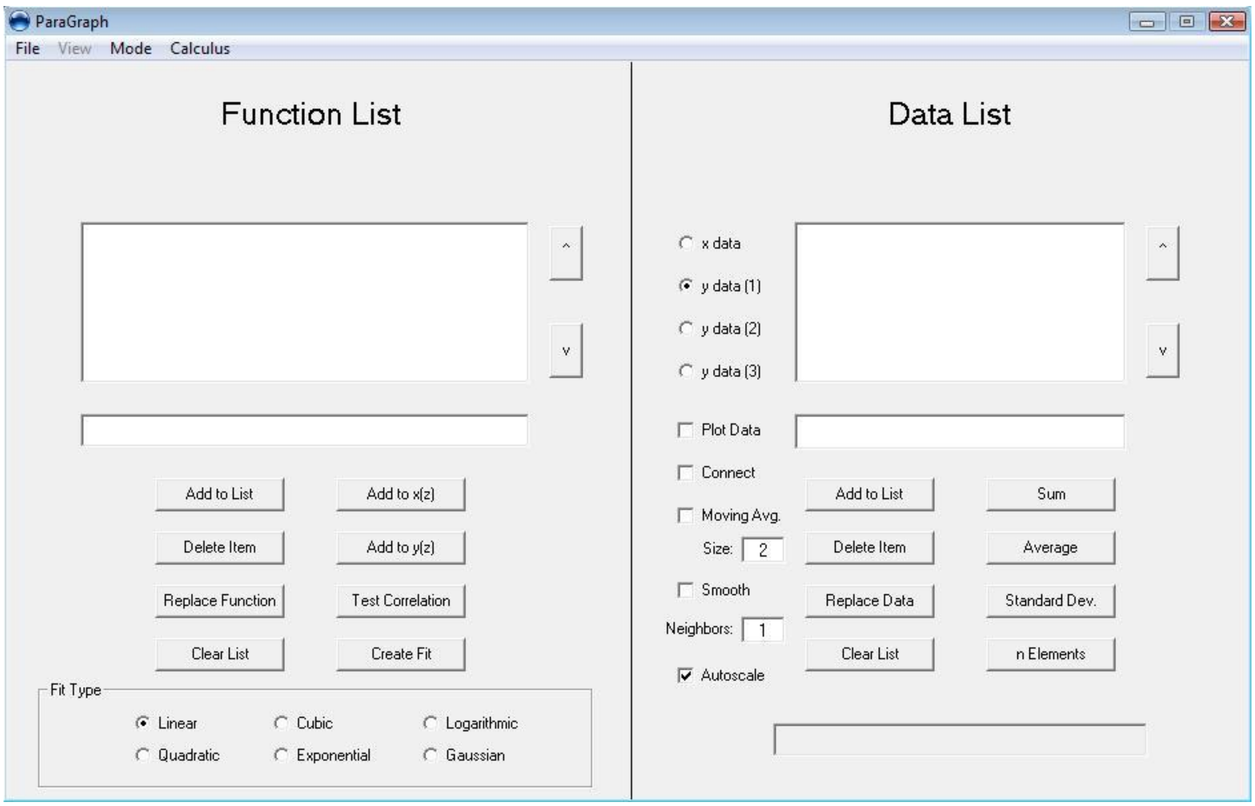
Start creating your equation by typing "(pages * wpp) / wpm / 60" in the calculation text box. Then, create your three variables below ("pages", "wpp", and "wpm"). You do not need to assign values to these variables just yet. Next, clear any text in the calculation history box, and type a description about your new equation, such as, "This equation finds how long it will take you to read a book; answer is in hours." Then, click the button that says "Transfer Text" near the top of the window. Your description will be placed above the calculation text box. If you wish to further edit your description text, drag and drop the text back over to the calculation history box. The last thing you will do (and it must be last!) is to name your equation. Do this by clicking the list box that says "User Defined" and type "Read Time". After that, click the button that says "Create Equation". All the text fields should clear themselves. To use your equation, locate it in the "User Defined" list box. Your equation should appear just like you entered it. To find out how long it would take you to read that 430 page book, enter the numbers from the above paragraph. The result should be a little over 9 hours.

Note that you may create up to 50 equations for each file you save. Also, you may delete any custom equation by selecting it in the "User Defined" list box, and clicking the "Delete Equation" button.

When you create equations, they will remain specific to that file, so if you save an equation, you will only be able to use that equation on that specific file. If you want an equation to always be there whenever you use ParaGraph, you should *create an initialize file* for ParaGraph to read upon startup.

3. Function / Data List

The third main component of ParaGraph is its list page. To set ParaGraph to list mode, click the menu option *Mode -> List,* or press F3 on your keyboard. Upon entering list mode, you should see this:



a. Function Entry

This mode of ParaGraph has two parts, but these two parts can work intricately together. The left side of the window is titled “Function List”. This is a single list that may hold any number of functions you may want to save for later plotting. To add a function to the list, type it into the narrow text box underneath the list box and press the Enter key on your keyboard or click “Add to List”. To delete a function, select it in the large box and click “Delete Item”. To delete the entire list of function, click “Clear List” (once you clear a data list, it is permanently erased!). Also, if you need to change one of your functions, make sure it is selected in the list box and make any changes you need in the entry box. Then, click “Replace Function”. Your old function will be replaced with your new one. Finally, you may move your equations around by selecting one of them and clicking the up and down buttons on the right side of the list box. This will cause whichever item you have to switch places with either the function above of the function below your selected function.

If you wish to use any of your functions, select it in the list box. The text you selected will appear in the entry box. You may then click “Add to x(z)” or “Add to y(z)” to use that for your x or y function. If you do not have any item selected in your list, clicking one of these buttons will still transfer the text from your entry box to the appropriate plotting text box. In this way, you do not necessarily have to add a function to a list in order to add it to your plotting boxes back in graphing mode. Note: although the captions for the buttons will not change, you can use these buttons when you are in polar graphing mode; the x(z) button will correspond to th(z) and the y(z) button will correspond to r(z).

The button labeled “Test Correlation” will take your current function and test it against your x data list and your active y data list. The correlation value will appear on the bottom right of the window. The closer this value is to 1, the better your data fits your function. Note that this procedure assumes that you are using y = f(x), which is a “normal” function; x(z) is always z.

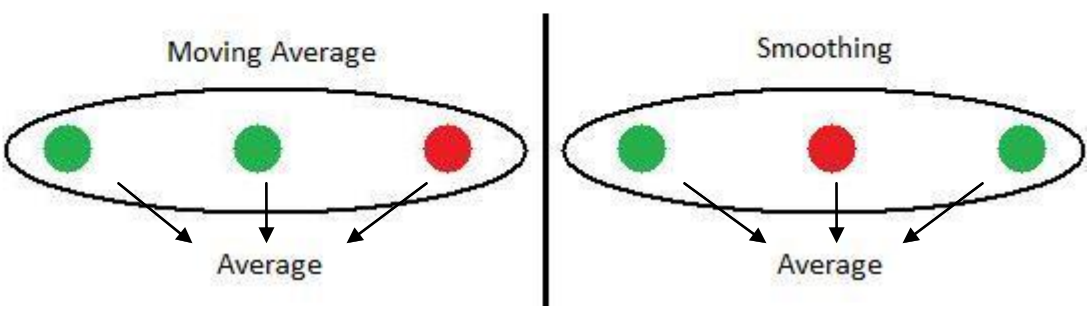
b. Data Entry

The right side of the window contains the data entry section. Entering, editing, and removing data is done in the same way as with the function list. However, you may have four data lists that are switched by clicking on the option buttons on the left side of the data list box. The four right-hand side buttons apply to whichever list is visible (active). “Sum” will add up all the numeric values of each item in the list, “Average” will calculate the average value of the active list, “Standard Dev.” will find the standard deviation of you active list, and “nElements” will find the number of items you current data list has.

c. Plotting Data

After entering data, you may plot it, provided that there are the same number of data points in your x list as in your current y list. When you plot data, ParaGraph will use the x list as the x coordinates of you data, and your active list as the y coordinates. To graph a set of data, click any number of the check boxes shown on the left side of the data list area and go back to the graphing mode of ParaGraph. The image at right shows the options to plot the points of a set of data, as well as connecting each point with a line. Also, the Autoscale box is checked. This means that when you go back to graphing mode, the viewing range of your plot window will automatically resize to fit the data you are plotting.

There are also a few other options to control how to view your data. The check box marked "Moving Avg." (moving average) will find the average of the last *n* data points of your data at each data point, where *n* is specified by the number in the text box marked "Size". The "Smooth" check box works similarly to the moving average, but it works with data before and after each data point. So, the "Neighbors" number is set to 1, there will be one data point before and one data point after the point in the middle. ParaGraph will then take these points and average them, and then plot the result. Therefore, the higher of a number this value is, the more smoothed your data will become.



In the image above, the red dots represent the data point currently being observed, and the green dots represent the dots being included in the average. For the moving average, ParaGraph looks at *n* points behind the current point; for smoothing, ParaGraph looks at *n* dots in front and *n* dots behind.

d. Importing Lists

If you have data that is stored in a text file, you can import that into one of your lists within ParaGraph. To do this, first select which list you want to import to by clicking one of the four lists on the left side of the data list area. For example, if "y data (1)" is selected, your list will be imported to y data (1). Next, click *File -> Import List* to select a text file. Your text file should consist of one number per line. If there already is data in your current list, ParaGraph will ask you if you wish to delete that data.

e. Exporting Lists

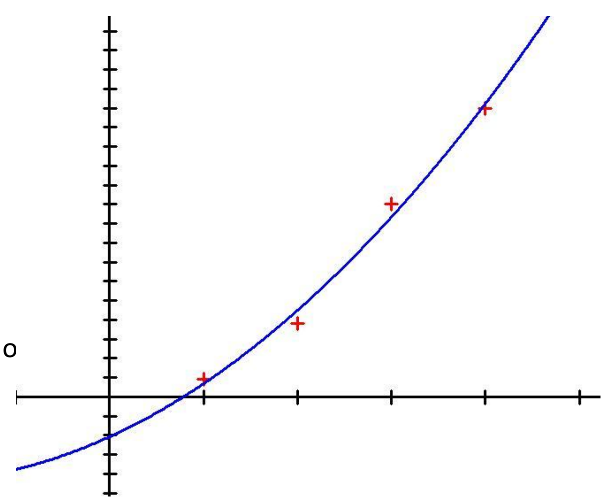
You can export a data list in the same way you import them. To export a data list to a text file, make sure the list you wish to export is your active list. Then, click the menu option *File -> Export List* and you should get a dialogue box requesting you to choose a text file to save to. You may either save over a previous text file or create a new one.

x	y
1	.9
2	2.1
3	3.3
4	4.4

Aside from simple exporting, there are a few other methods of exporting as well. Choosing *Export Log() of List* will apply a base ten logarithm to the numbers it saves to file. Note that the data in your list will remain unchanged. Another exporting method is *Export Inverse of List*. This will take the inverse of your data (1/x) and write that to file. Finally, you may also choose the option *Export List-Function Difference*. This option will take your current data list minus the points made by your x data when put into the function in the text box on the left of the window. For instance, say you had the following data in the table above. If the function in the function entry box was "z", then the list-function difference would be [-.1, .1, .3, .4]. These numbers would be saved to a text file, one number for each line.

f. Data Fitting

The button labeled "Create Fit" will use whichever fit type is selected at the bottom of



the window to create a best fit line using your x data and active y data list as a guide. When you click the button "Create Fit", ParaGraph will display an equation in the equation entry box. To use that equation, click the "Add to y(z)" button and go to graphing mode (make sure x(z) says "z"). Here is a picture of some data fitted to a quadratic curve, with the data for this plot in following table:

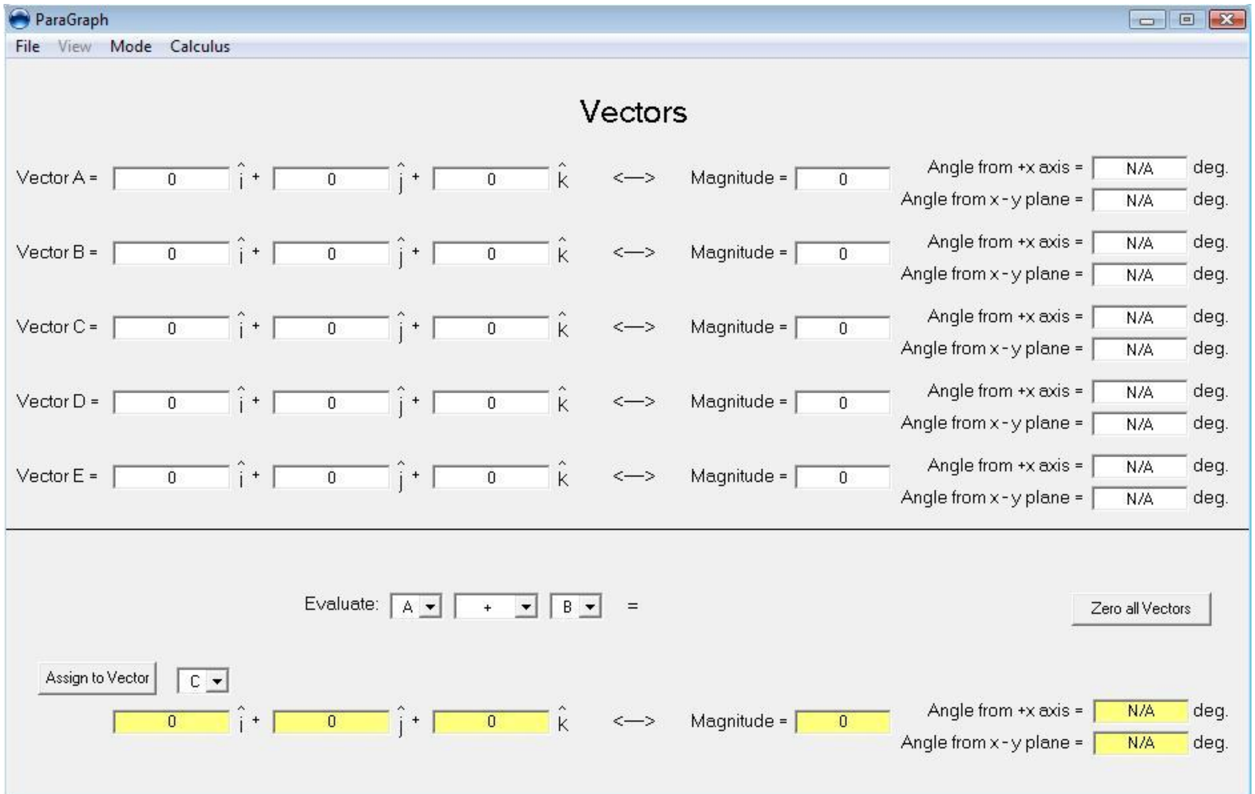
x	y
1	0.9
2	3.8
3	10
4	15

ParaGraph found the best fit line to be y(z) = .525*(z)^2+(2.22500000000009)*(z)+(-2.075), with a correlation value of 0.992, showing that this is indeed a good fit. Note again that with every best fit line, ParaGraph assumes x(z) = z, meaning that y is a function of x.

ParaGraph should find the best fit very quickly. The only fit type that may take a significant amount of time is the Gaussian fit. This fit should take about one minute for every thousand data points in your data list. If you wish to halt the process of finding a Gaussian fit, click and hold the left mouse button on any blank area of ParaGraph until the function entry text box reads "Data fit halted." Also, Gaussian curve fitting is not recommended on Windows XP operating systems, as XP will take much more time to find a Gaussian best fit line.

4. Vector Calculator

The last section of ParaGraph is its vector calculator. To get to the vector calculator page, click the menu option *Mode -> Vectors*, or press F4 on your keyboard. Upon entering Vectors mode, you should see this:



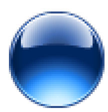
Within this page, you can do five basic operations with three-dimensional vectors: addition, subtraction, finding the angle between two vectors, dot product, and cross product. To do any of these, simply enter your vector's components in one of the five vectors. As you enter the numbers in, you will see the magnitudes and angles display according to the corresponding components. Alternatively, you may enter the magnitudes and angles and see the components adjust accordingly. If you only require two-dimensional vectors, the k component and the angle from the x-y plane will merely remain zero.

After you enter your vectors, simply change the drop-down list to one of the five options, and make sure you are using the correct vectors on either side. If your answer is scalar, the answer will be displayed next to the "=" sign near the bottom of the window, and if your answer is a vector, a series of yellow boxes will become visible with the appropriate numbers in them. Note that ParaGraph will always calculate an answer immediately when you type in numbers in any of the five vectors, so there is never any need to press any “equals” button; ParaGraph’s vector calculation is automatic!

After you calculate an answer that is in vector form, you may assign this vector to any one of your vectors by clicking the button labeled “Assign to Vector”. The default vector to which ParaGraph will assign is C, but you may choose any of the five vectors. After clicking this button, the current answer will disappear, and you may show it again by reselecting which operation you wish to perform. Finally, you may reset every value to zero if you wish to start things off with a clean slate by pressing the button labeled “Zero all Vectors”.

D. Program Functionality at a Glance

This next section provides a list of the available methods ParaGraph is capable of. Things such as menu options, mathematical functions, constants, and keyboard shortcuts are described here.

1. Menu  Options

Under *File:*

New Graph:	Restarts ParaGraph. A prompt for saving your old graph will occur if needed.
New Window:	Opens a new ParaGraph window without closing the original window.
Open:	Choose a GPH file to open. A prompt for saving your old graph will occur if needed.
Save:	Saves current graph. A save dialogue will appear if your graph has not yet been saved.
Save As:	Choose a new GPH file to save your current graph to.
Save As Bitmap:	Save a picture of your graph to a bitmap file.

Print: Opens a simple print dialog box to allow you to print the current plot to paper.

Import List: Imports a text file containing numbers into the current data list.

Export List: Exports the current data list to a text file.

Export Log() of List: Exports the log base ten of the current data list.

Export Inverse of List: Exports the inverse (1/x) of the current data list.

Export List-Function Difference: Exports the difference between the current data list and a function.

Set Default Ranges: Records the current plotting range contained in the plotting screen and sets them as the default range. When the *Set to default* button is pressed, the new values will be used. The new default ranges will remain only with your current GPH file.

Create Initialize File: Saves a text file called "INIT.txt" in the same directory as ParaGraph.exe. Upon startup, ParaGraph will check for this file from which to load its full settings. To return the startup settings to the original ones, simply delete INIT.txt from the ParaGraph.exe directory. Remember, INIT.txt must always be in the same directory as ParaGraph.exe, or ParaGraph will not find that file.

Under *View:*

Plot: Plots the current functions entered in the x(z) and y(z) text boxes.

Stop Animation: Halts the execution of the current function animation.

Axis Transformation: Lets you choose which transformation you would like to apply to an axis.

Set to Default: Sets the viewing range of the graph to the current default settings.

Zoom In: Lessens the viewing range of the graph for a closer viewpoint.

Zoom Out: Increases the viewing range of the graph for a farther viewpoint.

Shift Graph Left: Moves the viewing range of the graph to the left.

Shift Graph Right: Moves the viewing range of the graph to the right.

Shift Graph Up: Moves the viewing ran

Shift Graph Down: Moves the viewing range of the graph down.

Hide Scrolling Plot: Hides the graphics while the range is being updated (improves performance).

Smooth Plotting: Does a constant screen refresh as the function is being drawn, resulting in a more "smooth" look. The function is plotted faster without this option, but the user cannot see the process of the function being drawn.

More Visual Options: Opens a color selection dialogue box, which allows the user to select custom colors for various parts of the program, as well as line thickness.

Auto z Range: Automatically changes the z value range to fit a plot. ParaGraph only adjusts the z range if x(z) = z.

Under *Mode:*

Parametric Grapher: Switches the program to graphing mode.

Calculator: Switches the program to calculator mode.

List: Switches the program to list mode.

Vectors: Switches the program to vector mode.

Under *Calculus:*

Find Root: Finds where the current function goes to zero near a user-defined point.

Find Local Extremum: Finds where the current function's slope goes to zero near a user-defined point, likely resulting in a local maximum or minimum.

Find Inflection Point: Finds where the current function's second derivative goes to zero near a user-defined point.

Evaluate Derivative: Finds the numeric value of the derivative of the function at a user-defined point.

Evaluate Second Derivative: Finds the numeric value of the second derivative of the function at a user-defined point.

Evaluate Integral: Shows an integral page where a definite integral may be calculated.

2. Mathematical Functions

Text Entry	Function	Example
sin	Sine	sin(3) = .141120008059867
cos	Cosine	cos(3) = -.989992496600445
tan	Tangent	tan(3) = -.142546543074278
csc	Cosecant	csc(3) = 7.08616739573719
sec	Secant	sec(3) = -1.01010866590799
cot	Cotangent	cot(3) = -7.01525255143453
asin	Inverse Sine	asin(.5) = .523598775598299
acos	Inverse Cosine	acos(.5) = 1.0471975511966
atan	Inverse Tangent	atan(.5) = .463647609000806
acsc	Inverse Cosecant	acsc(1.5) = .729727656226966
asec	Inverse Secant	asec(1.5) = .84106867056793
acot	Inverse Cotangent	acot(1.5) = .588002603547566
sinh	Hyperbolic Sine	sinh(2) = 3.62686040784702
cosh	Hyperbolic Cosine	cosh(2) = 3.76219569108363
tanh	Hyperbolic Tangent	tanh(2) = .964027580075817
csch	Hyperbolic Cosecant	csch(2) = .275720564771783
sech	Hyperbolic Secant	sech(2) = .26580222883408
coth	Hyperbolic Cotangent	coth(2) = 1.03731472072755
asinh	Inverse Hyperbolic Sine	asinh(3) = 1.81844645923207
acosh	Inverse Hyperbolic Cosine	acosh(3) = 1.76274717403909
atanh	Inverse Hyperbolic Tangent	atanh(.5) = .549306144334055
acsch	Inverse Hyperbolic Cosecant	acsch(2) = .481211825059603
asech	Inverse Hyperbolic Secant	asech(.5) = 1.31695789692482
acoth	Inverse Hyperbolic Cotangent	acoth(2) = .549306144334055
rnd	Random Number	rnd = .3639033

abs	**Absolute Value**	**abs(-4) = 4**
sqr	**Square Root**	**sqr(9) = 3**
sqrt	**Square Root**	**sqrt(9) = 3**
log	**Log Base 10**	**log(7) = .845098040014257**
ln	**Log Base e (natural)**	**ln(7) = 1.94591014905531**
mod	**Modulo (remainder)**	**7 mod(4) = 3**
exp	**Exponential of e**	**exp(2) = 7.38905609893065**
E	**Exponential of 10**	**1.5 E2 = 150**
round	**Round (to nearest)**	**round(2.5) = 3**
fix	**Fix (rounds abs down)**	**fix(-2.5) = -2**
int	**Integer Part (rounds down)**	**int(-2.5) = -3**
frac	**Fractional Part**	**frac(2.5) = .5**
sgn	**Sign**	**sgn(-3) = -1**
fact	**Factorial**	**fact(6) = 720**
prime	**Prime**	**prime(6) = 13**
fib	**Fibonacci**	**fib(6) = 8**

3. Mathematical Operators

Text Entry	Operator	Example
+	Addition	2 + 5 = 7
-	Subtraction	7 – 2 = 5
/	Division	10 / 2 = 5
*	Multiplication	5 * 2 = 10
^	Power	2 ^ 3 = 8
\	Rounding Division	5 \ 2 = 2
=	Equals Comparison	(3 = 3) = True (num. val. is -1)
<	Less Than Comparison	(2 < 1) = False (num. val. is 0)
>	Greater Than Comparison	(2 > 1) = True (num. val. is -1)
(	Open Parenthesis	(2) = 2
)	Closed Parenthesis	(2) = 2
[	Open Brace	Gets changed to "("
]	Closed Brace	Gets changed to ")"

4. Mathematical Constants

Text Entry	Constant	Numerical Value
e	Base of Natural Logarithm	2.71828182845905
pi	π	3.14159265358979
G	Gravitational Constant	6.67428E-11
mol	Avogadro's Number	6.02214155E+23
deg	Degree to Radian Conversion	π/180; 1.74532925199433E-02
rad	Radian to Degree Conversion	180/π; 57.2957795130824
life	Meaning of Life	42

5. Shortcut Keys

Keyboard Entry	Function
Ctrl + N	New Graph
Ctrl + O	Open
Ctrl + S	Save
Ctrl + P	Print
Shift + Del	Set Viewing Range to Default
F11	Zoom In
F12	Zoom Out
Ctrl + J	Shift Graph Left
Ctrl + L	Shift Graph Right
Ctrl + I	Shift Graph Up
Ctrl + K	Shift Graph Down
F1	Go to Parametric Graphing Mode
F2	Go to Calculator Mode
F3	Go to List Mode
F4	Go to Vectors Mode
F5	Evaluate Integral

6. Tips and Tricks

- You can use copy and paste to quickly bring in expressions from other programs.
- The current A value for an animation is visible when the animation is running. You can find it just above the "A max" text box.
- For some text boxes, you can type an expression, and the number will be evaluated when you need it. These expressions can include the built in constants like G and pi as well. The text boxes in which you can do this are: the "Go to z =" box, the "A min" and "A max" boxes, the "z min and z max" boxes, the values for variable names on the Calculator page, the list data entry box on the List page, and the limits of integration boxes on the integration page.
- When you save a file, every single one of your settings will be saved, even things like which mode you were in when you saved your file, or whether or not your current mouse coordinates are displayed as multiples of pi.
- Try typing “warp” in the x(z) text box and plotting that.
- If you find that you are always assigning x(z) as just z every time you open ParaGraph, try this: open ParaGraph, and type a "z" in for x(z), then create an initialize file by clicking on "File -> Create Initialize File". Creating this file will cause ParaGraph to have a "z" filled in your x(z) text box every time you open ParaGraph.
- Creating an initialize file is very handy if you want to keep your custom equations from calculator mode close at hand.
- You can use the variable "A" not only in the x(z) and y(z) text boxes, but also in the "Go to z =" text box. This will change the place your function is being evaluated at when running an animation!
- The integration box, the visual options window and the printing window can be closed using the keyboard Escape (Esc) key.
- ParaGraph is always in radian mode for its trigonometry calculations. You can use the predefined constant "deg" to change something to radians. The constant "deg" is equal to pi/180, so entering "sin(90 deg)" will get you an answer of 1. Also, note that although the constant "rad" equals 180/pi, it is only useful for converting radians to degrees and should not be used inside a trig function, since ParaGraph is already in radian mode.

E. Concluding Remarks

1. A History of ParaGraph

Although ParaGraph is a multifunctional program, there were many steps involved in the progress of this program, often spaced with months of developmental dormancy. This section describes the process of how ParaGraph was created, from the first version up to this one, ParaGraph 6.1. ParaGraph was updated continuously for bugs and performance throughout its workings, and only the major features of how ParaGraph changed over the months are mentioned here.

I, the author of ParaGraph, first got the idea to create a computer interfaced graphing program around the beginning of 2007. I realized that a conventional graphing calculator was fairly limited with its graphing abilities due to the small number of pixels on the screen. A computer monitor, however, would be capable of high resolution graphs, which would be much easier to look at and learn from. From that simple idea, I created a small program capable of drawing an x – y coordinate system with tick marks automatically spaced evenly apart using a free programming language called Envelop. However, the work stopped there for a long period of time, due to various reasons.

It wasn't until February of 2008 that more programming began to get rolling towards creating the graphing system. With an upgrade to Microsoft's Visual Basic 6.0 and more determination, progress began to increase. Within a month, version 1.0 of ParaGraph was complete, which actually consisted of six different programs. Also, there was no name for the set of programs at that time; "ParaGraph" would come later. The six programs each had a separate task. First, there were two groups, three in each. One group was designed for running animations, while the other was not. In each group, one program could graph using the y = f(x) method, one program could graph using x(z) and y(z) (parametric) and one could graph in polar coordinates. These six programs had no error handling; they would crash upon any typo the user performed while entering a function. They were also incapable of saving anything to file. All in all, the programs totaled to about 400 KB of memory.

Then, in the late fall of 2008, more work began on the graphing systems. With version 2.0, the parametric graphers and polar graphers were merged into one program, with the ability to switch modes. The traditional y = f(x) grapher was eliminated since it is somewhat redundant in comparison to the parametric grapher. After version 2 was complete, there were two programs in existence; one for animations, and one for standard plotting.

The next obvious step for version 3.0 was to merge the two remaining programs into one, which was completed soon after the previous version. With version 3, the single program's size was 112 KB, which was much smaller than the previous set of six programs. Also, a greatly needed error handling system was introduced, which improved the ease of use for the graphing program. Along with the necessary error handling code, a "healing" function was written, which does things like changing the text "2z" into "2*z.

With version 4.0, ParaGraph began to accelerate its growth. While version 3 was stable in its existence, it was limited to its ability to only accomplish a few things. With version 4 came the ability to save graphs as GPH files, as well as navigating to a calculator and list page. At this point in time, the title "ParaGraph" came into existence as well. Version 4 was a major update to ParaGraph, and took a long time to program. It was completed mid spring in 2009.

Version 5.0 contained all the calculus methods ParaGraph now uses. Evaluating integrals, plotting derivates, finding local maxima, and the like were developed here.

ParaGraph 6.0 was completed in June, 2009. Version 6.0 contained the vectors page, capable of breaking vectors into their components, as well as basic vector math, such as the cross product. The most recent version, ParaGraph 6.3.5 (finished in August, 2010), shows a number near each tick mark on the graphing page as well as a grid on the viewing area. ParaGraph 6.3.5 also includes easier access to this help file, as well as a vast number of bug fixes.

Looking back, ParaGraph has come a long way from its original development, and may see more improvement in the future. As the author of this program, there have been times when I stated that I couldn't think of any possible way to improve ParaGraph, only to get revamped with ideas, leading to months of further development. ParaGraph has been a very rewarding project to work on, and will hopefully serve many people should they use it.

2. Sources

The sources of information for the construction of ParaGraph are from a variety of sources. Most things were taken from public discussion forums found online using Google searches, but there are a few specific items I would like to bring special attention to (sources are not limited to the following):

- The majority of the equations from calculator mode were found using Wikipedia.
- The code to obtain a fraction from a decimal was found online, written by Daniel Corbier.
- The code to calculate the derivative of a function with a given value was obtained from online example code used in DeadLine, a free simple graphing program.
- The code to evaluate integrals was found online, written by Vagelis Plevris.
- The name “ParaGraph” originated from Jesse Groenewold’s creative imagination.
- The icon used for ParaGraph was obtained online from Techlogica Free Icons.

3. Special Thanks

I would like to thank the following people for their encouragement, support, and creative input toward the development of ParaGraph:

- Professor John Zwart
- Jesse Groenewold
- Daniel Mahaffy

And for one final Easter Egg…

Try typing “Matthew Vande Burgt” or “ParaGraph” in calculator mode.