Complex Number Addition and the Complex (Argand) Plane, Activity 2

Learning Goals:

1) See that it is natural to view complex numbers as vectors well as points, and that this interpretation leads to a geometric view of complex number addition in terms of the parallelogram law for vector addition.

2) Learn basic applications of the *Mathematica* functions ListPlot, Graphics, Arrow, Point, Table, and Show in this setting, as well as various *Mathematica* formatting options.

Prerequisites:

1) Familiarity with vectors: in physics, linear algebra, multivariable calculus, and/or differential equations

2) *Mathematica* content from "Complex Number Addition and the Complex (Argand) Plane, Activity 1"

Introduction:

Vector quantities are very important in many applications of mathematics, the most prominent of which is physics, though vectors can come up in many other situations, such as in economics, where there is more than one variable to think about and work with. Complex numbers can be viewed as two-dimensional vectors, and this view provides our first insights into the importance of the geometric interpretation of complex arithmetic. To be more specific, we will learn that complex number addition can be seen to follow the geometric parallelogram law for the sum of two vectors and its "head-to-tail" generalizations to the sum of more than two vectors. We will eventually see that this allows us to understand complex-valued functions, such as $e^{i\theta}$, more deeply.

Content:

In introductory physics classes, a vector is a typically described as a quantity that has both a magnitude ("size") and a direction associated with it. When you push a table horizontally across a floor at a con-

stant speed in one direction, you are applying a constant force to the table that has a magnitude and a direction — the magnitude is related to how hard you are pushing. When the speed is constant, the floor, through friction, is also applying an opposing force to the table whose magnitude is the same as the magnitude of the force you are applying, but whose direction is exactly opposite yours.

An arrow is a natural way to visually represent a vector, since a given arrow has a certain magnitude ("length") and a certain direction. We are also free to visualize the arrow to be based (started) with it's "tail" at any point we want. In other words, the location of the arrow does not change the vector it represents. We are free to rigidly translate the arrow wherever we want. However, when using vectors to analyze physical models, the choice of location of the base of the arrow can help or hinder problem-solving as well as understanding the physical meaning of the quantities involved.

Discussion 1: Whether you've had physics or not, discuss and then describe other physical quantities/measurements that would be vectors, having both a magnitude and a direction. Also discuss and then describe the most beneficial spot to place the base of the vectors when drawn as arrows in diagrams, depending on the physical quantities or measurements they represent. As a hint to get you started, think about vectors that would help to describe the motion of a car around a parking lot, not just in terms of its position, but also in terms of what the driver sees through the front window and on the speedometer as time goes by.

Response 1:

(You can type your thoughts and answers here formatted in text mode)

Grader/Instructor Response 1:

(The grader/instructor will give you feedback about your work here)

Given a complex number $a + b\,i$, we saw in Activity 1 that it can be associated with a point (a, b) graphed in a standard rectangular coordinate system over a plane. It is natural to then draw an arrow based at the origin (0, 0) and having a tip or "head" (terminal point) at the point (a, b) in this plane. Because of this natural correspondence, we can think of the arrow itself as being yet another way to geometrically interpret the complex number $a + b\,i$. In other words, we can think of $a + b\,i$ as literally being a vector, the vector with the given magnitude ("length") and direction that we have just drawn.

In *Mathematica*, the most basic way to draw a vector is through the combined use of Graphics and Arrow, as illustrated in the following line of code.

```
Graphics[Arrow[{{1, 2}, {4, 3}}]]
```

The lists {1, 2} and {4, 3} represent the base point (1, 2) and terminal point (4, 3) of the vector. The given vector can be visualized, via change in base point, in infinitely many ways; though the most natural way, absent any physical applications, is to start it at the origin. The following code therefore produces two copies of the same vector, with the lower one being the one that is based at the origin.

```
Graphics[{Arrow[{{1, 2}, {4, 3}}], Arrow[{{0, 0}, {3, 1}}]}]
```

We can see this within a wider window in the given coordinate system by using the options Axes, AxesStyle, GridLines, and PlotRange. We can also color and thicken the arrows with and Thick and

Red.

```
Graphics[{Thick, Red, Arrow[{{1, 2}, {4, 3}}], Arrow[{{0, 0}, {3, 1}}]},
 Axes → True, AxesStyle → Thick, GridLines → Automatic, PlotRange → 5]
```

We can now graph the same diagram as in Activity 1, but with the natural origin-based vectors for each complex number shown as well, allowing us to match Figure [1] (Ch. 1.I .1, p. 2) of Tristan Needham's Visual Complex Analysis more fully. The user-defined variable **Points** is being used to store a list {{4, 0}, {4, 3}, {0, 3}, {−7, 1}, {−2, −3}, {2, −2}} of the points being plotted with ListPlot. The expression **Points**[[k]], which uses the Part [[ ]] operator, then represents the k^{th} element of the list **Points** — for instance, **Points**[[4]] = {−7, 1}.

```
Points = {{4, 0}, {4, 3}, {0, 3}, {-7, 1}, {-2, -3}, {2, -2}};
InitialFigure1 = ListPlot[Points, PlotStyle → {Black, PointSize[.02]},
   PlotRange → {{-9, 9}, {-4, 4}}, AxesLabel → {"real", "imaginary"},
   AxesStyle → Thick, GridLines → {Table[i, {i, -9, 9}], Table[i, {i, -4, 4}]},
   AspectRatio → Automatic]; Show[InitialFigure1,
 Graphics[{Text[4, {4.5, .5}], Text[4 + 3 ⅈ, {4.5, 3.5}], Text[3 ⅈ, {.5, 3.5}],
    Text[-7 + ⅈ, {-7.3, 1.5}], Text[-2 - 3 ⅈ, {-2.5, -2.5}], Text[2 - 2 ⅈ, {2.5, -1.5}]}],
 Graphics[{Thick, Red, Arrow[{{0, 0}, Points[[1]]}], Arrow[{{0, 0}, Points[[2]]}],
    Arrow[{{0, 0}, Points[[3]]}], Arrow[{{0, 0}, Points[[4]]}],
    Arrow[{{0, 0}, Points[[5]]}], Arrow[{{0, 0}, Points[[6]]}]}], ImageSize → Large]
```

A slightly more efficient way to create the last picture is to make use of Table and Length, the last of which returns the number of elements of a list — in this case, the elements are themselves lists with two elements. Recall from Activity 1 that Table creates lists of objects in *Mathematica* that have "formulas". In this case, the formula is **Graphics[{Thick, Red, Arrow[{{0, 0}, Points[[*i*]]}]}]**, which creates an arrow from the origin (0, 0) to the i^{th} entry of **Points**. Text, Style, Large, and Italic have also been combined to make the labeling of the axes look nicer.

```
Points = {{4, 0}, {4, 3}, {0, 3}, {-7, 1}, {-2, -3}, {2, -2}};
InitialFigure1 = ListPlot[Points,
   PlotStyle → {Black, PointSize[.02]}, PlotRange → {{-9, 9}, {-4, 4}}, AxesLabel →
    {Text[Style["real", Large, Italic]], Text[Style["imaginary", Large, Italic]]},
   AxesStyle → Thick, GridLines → {Table[i, {i, -9, 9}], Table[i, {i, -4, 4}]},
   AspectRatio → Automatic]; Show[InitialFigure1,
 Graphics[{Text[4, {4.5, .5}], Text[4 + 3 ⅈ, {4.5, 3.5}], Text[3 ⅈ, {.5, 3.5}],
    Text[-7 + ⅈ, {-7.3, 1.5}], Text[-2 - 3 ⅈ, {-2.5, -2.5}], Text[2 - 2 ⅈ, {2.5, -1.5}]}],
 Table[Graphics[{Thick, Red, Arrow[{{0, 0}, Points[[i]]}]}], {i, 1, Length[Points]}],
 ImageSize → Large]
```

Changing the Thick directive to Thickness, and giving a numerical input for Thickness (such as .01) that gives a fraction of the horizontal plot range (such as 1%), allows us to make the vectors even thicker.

```
Points = {{4, 0}, {4, 3}, {0, 3}, {-7, 1}, {-2, -3}, {2, -2}};
InitialFigure1 = ListPlot[Points,
   PlotStyle → {Black, PointSize[.02]}, PlotRange → {{-9, 9}, {-4, 4}}, AxesLabel →
    {Text[Style["real", Large, Italic]], Text[Style["imaginary", Large, Italic]]},
   AxesStyle → Thick, GridLines → {Table[i, {i, -9, 9}], Table[i, {i, -4, 4}]},
   AspectRatio → Automatic]; Show[InitialFigure1,
 Graphics[{Text[4, {4.5, .5}], Text[4 + 3 ⅈ, {4.5, 3.5}], Text[3 ⅈ, {.5, 3.5}],
    Text[-7 + ⅈ, {-7.3, 1.5}], Text[-2 - 3 ⅈ, {-2.5, -2.5}], Text[2 - 2 ⅈ, {2.5, -1.5}]}],
 Table[Graphics[{Thickness[.01], Red, Arrow[{{0, 0}, Points[[i]]}]}],
   {i, 1, Length[Points]}], ImageSize → Large]
```

In physics, when a vector is drawn from the origin to a given point, that vector is called the position vector of the given point (relative to the given coordinate system), though we could once again translate this vector to be based at any spot we want, and it would still be called the position vector of the given

point.

Mathematica* Exercise 1:** Use ListPlot to make a plot, in the complex plane, of the points $2-3i$, $-5+8i$, $1+2i$, $-4i$, and 10. Make sure your axes are labeled appropriately, your window is chosen well, your points are larger than the default (and make them colored Blue), the grid-lines are included, the scales on the axes are the same, the axes are thick, and the points are labeled. ***Also make use of Graphics and Arrow to make position vectors for these points (and make them colored Green). Experiment with your code to strive to make it as efficient as possible (as short as possible, and possibly just use one cell of code rather than multiple cells). Note that a semicolon ";" will suppress output in *Mathematica* but can also be used to separate distinct lines of code within the same cell if you wish.

Mathematica Work 1:

(Enter your code under this cell when *Mathematica* is in "Input mode" — make sure a horizontal line is showing before you start typing)

(You can type your thoughts and answers here formatted in text mode)

Grader/Instructor *Mathematica* Assessment 1:

(The grader/instructor will give you feedback about your work here)

It can be experimentally verified that when two forces are applied to an object, the resultant ("net" or "total") force can be determined by the so-called parallelogram law. To be more specific, if two vectors **v** and **w** are not parallel — they are neither in the same nor the opposite direction — and neither is a "zero vector" (represented as a point rather than an arrow so that it has length zero and has no direction), then a parallelogram can be formed by first picking a common point *P* to use as the base for both **v** and **w**, then drawing another copy of **v** at the tip of **w** and another copy of **w** at the tip of **v**. These last two vectors will meet at a common point *Q*. If **v** and **w** represent forces acting on one object, then the arrow drawn from *P* to *Q* will represent the total force acting on the object, and it is natural and common to represent this force symbolically as $\mathbf{v}+\mathbf{w}$, implying that we are thinking of the two individual forces "adding" to give the total force.

Entering the following line of code will help you visualize the ideas in the last paragraph.

```
Show[Graphics[
  {Thickness[.01], Red, Arrow[{{0, 0}, {1, 3}}], Arrow[{{2, 2}, {3, 5}}], Blue,
   Arrow[{{0, 0}, {2, 2}}], Arrow[{{1, 3}, {3, 5}}], Black, Arrow[{{0, 0}, {3, 5}}]}],
 Graphics[Text[Style["v", Large], {1.3, 1}]],
 Graphics[{Text[Style["v", Large], {1.6, 4}],
   Text[Style["w", Large], {.4, 2}], Text[Style["w", Large], {2.7, 3.2}],
   Rotate[Text[Style["v+w", Large], {1.3, 2.5}], 60 Degree]}],
 Axes → True, PlotRange → {{-.1, 5.1}, {-.1, 5.1}}]
```

Actually, half the diagram above is superfluous, unless our goal is to do a (visual) "proof without words" of the commutative property for vectors. All we need to focus on is the fact that we can first draw one copy of **v** and then draw one copy of **w**, based at the tip of **v**, and we will be able to draw $\mathbf{v}+\mathbf{w}$ correctly. Such a process can be more easily extended to the sum of more than two vectors — we just keep

placing the vectors "tip to tail" (put the base (tail) of the next vector at the tip of the preceding one). This is illustrated with the code below.

```
Show[Graphics[{Thickness[.01], Red, Arrow[{{0, 0}, {1, 3}}],
    Blue, Arrow[{{1, 3}, {3, 5}}], Green, Arrow[{{3, 5}, {4, 2}}],
    Black, Arrow[{{0, 0}, {4, 2}}], Text[Style["u", Large], {.4, 2}],
    Text[Style["v", Large], {2, 4.3}], Text[Style["w", Large], {3.7, 3.7}],
    Rotate[Text[Style["u+v+w", Large], {2, 1.3}], 27 Degree]}],
 Axes → True, PlotRange → {{-.1, 5.1}, {-.1, 5.1}}]
```

Discussion 2: Does this way of interpreting vector addition make sense in other contexts? For instance, what if you travel from a point *P* to a point *Q* and then to a point *R* and then let **v** and **w** represent the corresponding "displacement vectors"? What about if you are swimming at a nonzero angle with the straight current of a straight river, where **v** represents your constant velocity relative to the river and **w** represents the constant velocity of the current relative to the river banks?

Response 2:

(You can type your thoughts and answers here formatted in text mode)

Grader/Instructor Response 2:

(The grader/instructor will give you feedback about your work here)

We saw in the Activity 1 that it makes symbolic sense to add complex numbers as if they were linear polynomials in the indeterminate i. That is, it seems that defining $(a + b\,i) + (c + d\,i)$ to be $(a + c) + (b + d)\,i$ might be a good idea. Indeed, algebraically-speaking, this is part of what is necessary to make the set of all complex numbers $\mathbb{C}$ into an algebraic structure called a field (***not*** to be confused with a vector field). The other necessary part to make $\mathbb{C}$ a field is to define multiplication in a "good" way. What does all this mean? In a nutshell, with respect to addition, the collection of all complex numbers satisfies every algebraic property you could hope for: associativity, commutativity, existence of an additive identity (the number $0 = 0 + 0\,i$) and existence of additive inverses — the additive inverse of $a + b\,i$ is $(-a) + (-b)\,i$. Once multiplication has been defined, similar properties will hold for it, as well as the distributive property, which relates multiplication and addition. The number $1 = 1 + 0\,i$ will be the multiplicative identity and it will also turn out that nonzero complex numbers will have multiplicative inverses (at the moment, we'll leave it to you to figure out what the multiplicative inverse (reciprocal) of a nonzero number $a + b\,i$ will be, if you wish).

To more fully justify this definition of addition of complex numbers as a good idea, we need to explore what it means geometrically. Let's consider a particular example: the fact that $(1 + 2\,i) + (4 + i) = (1 + 4) + (2 + 1)\,i = 5 + 3\,i$. If we visualize these complex numbers as points in the complex plane, we get the picture generated by the following code. For variety and to illustrate *Mathematica*'s flexibility, we note that we are now making these points by combining Graphics and Point rather than using ListPlot. Note also the use of PointSize and how it behaves in a way similar to Thickness.

```
DotPlot = Graphics[{PointSize[.03], Red, Point[{{1, 2}}], Point[{{4, 1}}],
    PointSize[.06], Blue, Point[{{5, 3}}]}, Axes → True, AxesStyle → Thick,
  AxesLabel → {Style["real", Large, Italic], Style["imaginary", Large, Italic]},
  PlotRange → {{-1, 6}, {-1, 6}}]
```

Now imagine drawing line segments from the origin (0, 0) to (1, 2), from (1, 2) to (5, 3), from (5, 3) to (4, 1), and from (4, 1) back to the origin (0, 0). It appears that we have created a parallelogram. Thinking in terms of vectors, it appears that if we let **v** be a vector pointing from (0, 0) to (4, 1) and let **w** be a vector pointing from (0, 0) to (1, 2), then after an appropriate translation, **v** will point from (1, 2) to (5, 3) and **w** will point from (4, 1) to (5, 3). This can be confirmed visually with the next line of code (make sure the preceding line of code has been entered before entering this line so that the variable **DotPlot** has a "value").

```
Show[DotPlot, Graphics[{Thickness[.01], Brown, Arrow[{{0, 0}, {4, 1}}],
    Arrow[{{1, 2}, {5, 3}}], Orange, Arrow[{{0, 0}, {1, 2}}], Arrow[{{4, 1}, {5, 3}}]}]]
```

In fact, we can "quantify" **v** and **w** by describing their horizontal and vertical displacements; any vector is completely determined, within a given rectangular coordinate system, by how much it points to the *right* (***positive*** horizontal displacement) or *left* (***negative*** horizontal displacement) and by how much it points *upward* (***positive*** vertical displacement) or *downward* (***negative*** vertical displacement). For the example above, **v** has a horizontal displacement of 4 and a vertical displacement of 1; while **w** has a horizontal displacement of 1 and a vertical displacement of 2. It's no accident that these numbers match the coordinates of the points (4, 1) and (1, 2), respectively; and it's also no accident that these numbers match the real and imaginary parts of the complex numbers $4 + i$ and $1 + 2i$, respectively. Finally, note that we can also obtain these displacements by subtracting the coordinates of (1, 2) and (4, 1), respectively, from the point (5, 3). This is essentially a geometric proof that the figure in the picture is a parallelogram, and it leads us to write **v** and **w** in a new way. One common notation used for these vectors is $\mathbf{v} = \langle 4, 1 \rangle$ and $\mathbf{w} = \langle 1, 2 \rangle$, and this is the notation we will use. In physics, the standard way to represent these vectors is in terms of the "standard unit vectors". We will ***not*** use that notation, because of the obvious confusion that could arise when using it within the study of complex numbers through the use of the "i" in that notation (also called "i hat" and written $\hat{i}$).

What we have just done, for the given example, is confirmed that complex number addition also satisfies the parallelogram law, just as vector addition does. In fact, the extended view of vector addition works when we add three or more complex numbers. We will therefore want visualize complex numbers as vectors when we add them.

***Mathematica* Exercise 2:** Use *Mathematica* to visually illustrate the sum of the complex numbers $3 + 2i$ and $-5 + 4i$ as equivalent to the addition of the vectors $\langle 3, 2 \rangle$ and $\langle -5, 4 \rangle$. What is different about the parallelogram so generated compared to the parallelogram created with the example above? Use the distance formula to confirm that opposing sides of this parallelogram have the same length and use facts from trigonometry, such as the Law of Cosines, to confirm that opposing angles of this parallelogram have the same measure.

Mathematica Work 2:

(Enter your code under this cell when *Mathematica* is in "Input mode" — make sure a horizontal line is showing before you start typing)

(You can type your thoughts and answers here formatted in text mode)

Grader/Instructor *Mathematica* Assessment 2:

(The grader/instructor will give you feedback about your work here)

In Activity 3 of this learning module on complex addition and the complex plane, we will use *Mathematica*'s Manipulate function to create dynamic (interactive) versions of these diagrams. We will also delve into the algebraic and geometric meanings of complex number subtraction.