# MathType Orientation

MathType is Word plugin that allows us to enter math in a format that is compatible with our braille formatting program, Duxbury. Duxbury reads equations in the Word document that are encoded with MathType and converts them into Nemeth code braille. The document is then proofread and embossed.

Although one can use the MathType WYSIWYG (what you see is what you get) editor, we found that it is often much more efficient, clean, and reliable to use a markup language called TeX to enter math into MathType. LaTeX (and TeX) is a common markup language used for typesetting documents and offers some advantages for people who are blind or low-vision by allowing them to see and access math.

## Marking and Converting Equations

For inline equations, you will need to surround LaTeX/TeX code in dollar signs like this:

$y=x^2$

To convert a single equation, you will need to put your cursor inside of or just outside of the equation and then press “Alt+\”. To convert multiple equations, select all the text of the equations you want to convert and press “Alt+\”; any equation you have selected will convert (or uncovert if it was already converted). Be aware that if your cursor is not inside of an equation and you have no equation selected, pressing “Alt+\” will still initiate the convert/unconvert shortcut; MathType will search for the last equation before the point of your cursor and attempt to convert/unconvert it, so if you accidentally press “Alt+\”, be sure to go back and check the state of the last equation.

## Spaces

* Equations must contain spaces after any uses of a term that is indicated with a backslash, such as “\cap ” – note the final space.
* Other times, spaces are not necessary, though they are often preferred for clarity, such as around an equal sign (“x = y”).
* If MathType is refusing to convert something you have written in Tex, such as a decimal, putting a slash before a space (“\ ”) inside of the Tex is a handy way of getting it to convert what you’ve typed.
* Spaces without a backslash will not effect the braille output, but "\ " with a space will, so use them as seldom as possible. They are not needed in indices or before units, such as kg, m, degrees, etc.

## MathPix

MathPix is a math-specific OCR software that we use for converting snipped screenshots of visual math into LaTex. Remember that although this software does an amazing job at picking things up, it is similar to ABBYY in that we always have to carefully check its work.

In order to install MathPix, visit [MathPix.com](https://mathpix.com/) and click on the windows install button. Run the downloaded mathpix\_snipping\_tool\_setup.exe and proceed with “next” through the installation. It should not ask you to provide administrator credentials. After the software is installed, you can bring it up with Ctrl+Alt+M. Drag a box around whatever you would like to convert and when you release, the visual math will be converted into LaTex. Pick the one that is enclosed in single $ signs.

Note that MathPix generally uses “\begin{array}{<cell alignment code>}” instead of “\begin{matrix}”, but this does not pose any problems for Duxbury and results in identical braille output. You do not need to replace the arrays MathPix produces with the more specific matrix formats at the end of this document.

MathPix will also use “\mathrm” to mark text in math. All instances of “\mathrm” should be removed because it causes unpredictable results when converted to braille.

## LaTeX Source Files

Occasionally we receive LaTeX source files. Although a PDF source file is still necessary so we can check formatting, a LaTeX source file can save some time since one can simply copy the LaTeX math code from the source file instead of typing it in. However, not every LaTeX source file follows the conventions we use, so one must check the LaTeX code carefully to make sure it meets our editing standards. This includes making sure that the LaTeX code used to produce particular symbols matches the code we generally use to produce those symbols. Some code produces a result that looks visually similar, but will not convert to braille properly in Duxbury. In some rare cases, the LaTeX source files we receive are computer-generated. Computer-generated LaTeX code should be reviewed the same way as MathPix-generated LaTeX code. Computer-generated LaTeX code can often be identified by extra manual spacing including \quad, \qquad, and manual line skips. These extra spaces should be removed from the LaTeX code before it is converted into MathType, otherwise Duxbury will often produce extra spaces.

## Before you start

### Autocorrect

In order to prevent Word from autocorrecting your work in ways that could directly impact the MathType equations, go to File -> Options -> Proofing -> Autocorrect Options -> Autocorrect tab and turn off “Replace straight quotes with smart quotes” and “Replace hyphens with dashes” in both the “Autoformat as you type” and “Autoformat” tabs.

### Clearing Formatting

It is often best to erase all formatting before beginning work. You can find the clear formatting button on the ribbon (Home > Font section > “Clear all formatting” button).

## Formatting Standards and Quirks

### Breaking Up Documents

If a document has over 600 math objects, it would be better to split it so that Duxbury can process it properly. To find out the number of math objects are in a single document, type ^g in the search field in the navigation pane.

### Breaking Up Math Objects

MathType will not convert math objects with more than around 464 characters (including spaces). If possible, break math objects up that are going to exceed this limit.

### Equation Labels

Sometimes equations are assigned labels in a textbook, and we should indicate these labels by marking them up as a heading on the previous line. For example, you may see an equation and then (1.1) at the end of the same line; it should be moved to the previous line and marked as a heading since the text references the equations by these labels.

### Vectors

Some math documents mark vectors by using bold-faced text. This includes both general variables (e.g.  ) and the zero vector . MathPix, if you are using it, does not generally recognize these and mark them correctly as vectors. Whenever bold-faced text is used to mark vectors, one must be careful to make sure they are marked with “\vec”. Oftentimes there will be clues in the document to help you figure out whether a bold-faced variable is a vector, but if unsure please ask.

### Spacing

Equations that are inline with text should have spaces before and after the equation to separate it from the text, just like a word would. And also like a word, no space is needed between the MathType object and any following punctuation. Do NOT try to combine math instances that are separated by obvious sentence structures or punctuation.

#### Example

##### Incorrect (look at the spacing and placement of commas):

Emily has 200 marbles, andof them were purple with blue stripes. If, then how many mumbo jumbos does she have? Is it , or just 42?

##### Correct:

Emily has 200 marbles, and  of them were purple with blue stripes. If , then how many mumbo jumbos does she have? Is it , , or just 42?

### MathType or Text

#### Rule

If a variable or number appears on its own, leave it as normal text without any special formatting, even if the number is negative. Decimals, however, should be marked up as math, adding a backslash (\) and a space before or after the number within the dollar signs in order to get it to render. This rule does not apply to figure or section references or titles.

##### Example

If X is equal to , and c is equal to -9, what is  and what is ? Find out in Figure 1.1 below.

#### Rule

If a currency amount is in the text, leave it as text. There is no need to try to make it “math.”

##### Example

I want to add $500 and $700 in order to come up with $1,300 but it doesn’t seem to be working.

#### Rule

(x, y) indices should be marked as math and an explicit space has to be placed after the comma using a backslash and a space (“\ “).

##### Example

If you go to point , you’ll find the treasure!

#### Rule

If you see math inside a block of code, leave it alone. Code has to be conveyed to the client character by character.

##### Example

We shall present the following code as if it is interesting, but in reality we are well aware of how boring it is:

[begin code]

1 set launchSeq[x]{

2 rocket.booster.go()

3 f(x)=c+10

4 }

[end code]

#### Rule

If a mathematical equation uses words with spaces that need to be preserved, put those words and spaces in a \text{} enclosure.

##### Example

In pseudo-sciences, we like using fancy word equations like 

#### Rule

If a list of math objects are in a sentence, separated by words and/or commas, make sure each math object is separate; commas used by the math should be a part of the math object and commas used by an enclosing sentence should not be within the math objects:

##### Example (look at where each math object begins and ends)

Go to the following indices in order: , , and .

#### Rule

If Chemical formulas containing numbers appear in a document for braille, treat them like math.

##### Example

Wouldn’t you like some  with your KN?

If so, get some this way: 

### Parenthesis

In Math Type all types of parentheses should be inserted only from the keyboard, even if they look bigger in the hardcopy and lots of options exist in Math Type to do them.

#### Example

##### Incorrect:



##### Correct:

 

### Feet and Inches

MathType does not work well with trying to put in ‘ and “ for feet and inches in the math, and so instead we need to put the shortened words “ft” and “in”, including the unit of measure inside of math that may proceed before, e.g. .

#### Example

##### Incorrect:



##### Correct:



### Display Code vs. Clean Code

Since Duxbury reads the encoded equations, it is important for the LaTeX code to be “clean.” There are ways of entering in LaTeX code that looks fine visually, but does not convert properly. Entering equations into the WYSYWIG MathType editor can sometimes result in display code. MathType’s backtranslator does not always translate cleanly. An example of this would be:

#### Example 1

##### Code that looks fine but does not work:

$f(x,y)=

\begin{cases}

\frac{3x^2y}{x^2+y^2} & \text{if } (x,y) \ne (0,0),\\

0 & \text{if } (x,y) = (0,0)

\end{cases}$

###### Result:



##### Code that works properly:

$f(x,y) = \frac{3 x^2y}{x^2 + y^2}$ if $(x,y) \neq (0,0)$ and

$f(x,y) = 0$ if $(x,y) = (0,0)$

###### Result:

 if  and

 if 

#### Example 2

##### Code that looks fine but does not work:

$\underset{h\to 0}{\mathop{\lim }}\,\frac{f(96, 70 + h) - f(96, 70)}{h}$

###### Result:



##### Code that works properly:

$\lim\_{h \to 0} \frac{f(96, 70 + h) - f(96, 70)}{h}$

###### Result:



## Common Symbols

### Math Operator Symbols

| ***Source*** | ***TeX*** |
| --- | --- |
|  | + |
|  | - |
|  | \times |
|  | \cdot |
|  | \div |
|  | \pm |

### Equality symbols

| ***Source*** | ***TeX*** |
| --- | --- |
|  | = |
|  | \neq or \ne |
|  | \approx |
|  | < |
|  | \leq or \le |
|  | > |
|  | \geq or \ge |
|  | \equiv |

### Set symbols

| ***Source*** | ***TeX*** |
| --- | --- |
|  | \cap |
|  | \cup |
|  | \land OR \wedge |
|  | \lor OR \vee |
|  | \in |
|  | \not\in |
|  | \ni or \owns |
|  | \subset |

### Grouping Symbols

| ***Source*** | ***TeX*** |
| --- | --- |
|  | ( |
|  | ) |
|  | \left[ or \lbrack |
|  | \right] or \rbrack |
|  | \lbrace |
|  | \rbrace |
|  | \langle |
|  | \rangle |
|  | \left[\!\left[ |
|  | \right]\!\right] |
|  | \left| |
|  | \right| |
|  | |\*Looks the same as absolute value lines but comes alone; use the pipe character above the enter key. |
|  | \left \lceil \right \rceil |

### Fractions

| ***Source*** | ***TeX*** |
| --- | --- |
|  | \frac{a}{b} |
|  | \frac{a + 2}{\sqrt{\frac{3b}{c}}} |
|  | a \frac{b}{c} |

### Subscripts and Superscripts

| ***Source*** | ***TeX*** |
| --- | --- |
|  | a^2 |
|  | a^3 |
|  | a^4 |
|  | a\_1 |
|  | a^{12} |
|  | a\_{12} |
|  | a\_{12}^{24}\*subscript first |

### Roots

| ***Source*** | ***MathSpeak*** |
| --- | --- |
|  | \sqrt{a} |
|  | \sqrt{a\sqrt{b}} |
|  | \sqrt[a]{b} |

### Character Modifiers

| ***Source*** | ***TeX*** |
| --- | --- |
|  | \bar{a} |
|  | \bar{(\bar{a})} |
|  | \vec{a} |
|  | \hat{a} |
|  | \ddot{a} |
|  | a’ or a \prime |

### Summations

| ***Source*** | ***TeX*** |
| --- | --- |
|  | \sum\_{a}^{b} c |

### Limits

| ***Source*** | ***TeX*** |
| --- | --- |
|  | \lim\_{a} b |

### Integrals

| ***Source*** | ***TeX*** |
| --- | --- |
|  | \int\_{a}^{b} c |
|  | \iint \limits\_{a} b |

### Functions

| ***Source*** | ***TeX*** |
| --- | --- |
|  | \log |
|  | \cos |
|  | \sin |
|  | \tan |

### Greek Letters

|  |  |
| --- | --- |
| ***Source*** | ***Tex*** |
|  | \Sigma |
|  | \pi |
|  | \theta |
|  | \Delta |
|  | \delta |
|  | \alpha |
|  | \lambda |
|  | \mu |
|  | \gamma |
|  | \rho |
|  | \varepsilon |
|  | \nabla |
|  | \chi |

Notes: The LaTex for the upper and lower case Greek letters differ only by capitalization, e.g. “\Delta” and “\delta.”

### Miscellaneous symbols

Certain characters perform special functions in LaTeX/TeX such as the ampersand and the percent sign. The ampersand tells LaTeX/TeX how to visually align equations, and the percent sign is used to tell LaTeX to ignore whatever comes after it. Because of this, there are specific ways of entering ampersand and percent signs.

| ***Source*** | ***TeX*** |
| --- | --- |
|  | \o or \emptyset or \varnothing |
|  | ^\circ |
|  | \angle |
|  | \measuredangle |
|  | \R |
|  | \And |
|  | \% |
|  | \partial |
|  | \infty |
| : | : |
|  | , |
|  | \dots |
|  | \cdot |
|  | \cdots |
| ? | ? |
|  | \to or \rightarrow |
| ~ | \sim |
|  | \forall |
|  | \exists |
|  | \oint |
|  | \oplus |
|  | \lnot |
|  | \Box |
|  | /\*leave ambiguous fractions with a slash instead of a \frac{}{} |

## Various Matrices in MathType

Here are examples of different types of matrices in MathType.

### Simple Matrix without parenthesis

#### Code:

$M\_R=\begin{matrix}

1 & 0 & 1 \\

0 & 1 & 0 \\

1 & 1 & 0 \\

\end{matrix}$

#### Output:



### Matrix enclosed with parenthesis

#### Code:

$M\_R= \begin{pmatrix}

1 & 0 & 1 \\

0 & 1 & 0 \\

1 & 1 & 0 \\

\end{pmatrix}$

#### Output:



### Matrix enclosed with square brackets

#### Code:

$M\_R=\begin{bmatrix}

1 & 0 & 1 \\

0 & 1 & 0 \\

1 & 1 & 0 \\

\end{bmatrix}$

#### Output:



### Matrix enclosed with curly braces

#### Code:

$M\_R=\begin{Bmatrix}

1 & 0 & 1 \\

0 & 1 & 0 \\

1 & 1 & 0 \\

\end{Bmatrix}$

#### Output:



### Matrix enclosed in vertical lines (determinant of matrix)

#### Code:

$M\_R=\begin{vmatrix}

1 & 0 & 1 \\

0 & 1 & 0 \\

1 & 1 & 0 \\

\end{vmatrix}$

#### Output:



### Matrix enclosed in double vertical lines (absolute value matrix)

#### Code:

$M\_R=\begin{Vmatrix}

1 & 0 & 1 \\

0 & 1 & 0 \\

1 & 1 & 0 \\

\end{Vmatrix}$

#### Output:



### Inline matrix:

#### Code:

$M\_R = \bigl[ \begin{smallmatrix}

1 & 0 & 1 \\

1 & 1 & 0 \\

\end{smallmatrix} \bigr]$

#### Output:



[end of document]