

GRAPHTEX 3.0

SEVERINO V. GERVACIO
Mathematics and Statistics Department
De La Salle University
2401 Taft Avenue
0922 Manila, Philippines

1 IMPROVEMENTS

Improvements of the \LaTeX command `\vec` and the GraphTeX 2 command `\TriLoop` are made and incorporated in GraphTeX 3 .

1.1 Vectors

\LaTeX has the command `\vec` to put an arrow on top of a letter in order to denote a vector. For example, `\vec v` produces \vec{v} . Sometimes we use two or more letters to denote a vector. In this case `\vec` does not produce a good result. For example, `\vec {xy}` produces \vec{xy} .

In GraphTeX 3 , we can create a better “vector” using `\Vec`. For example, \vec{v} , \overrightarrow{xy} , \overrightarrow{ABC} are produced using the code `\Vec v, \; \Vec {xy}, \; \Vec{ABC}`.

The macro `\Vec` adjusts to text size. For example, \overrightarrow{AB} is created by the code `\LARGE \Vec{AB}`.

1.2 TriLoop

Let us recall the `\TriLoop` in GraphTeX 2 .

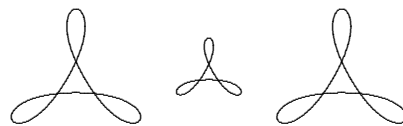


Figure 1: TriLoop

The triloops in Figure 1 are created using the following code:

```

\begin{figure}[h]
$$\pic
\TriLoop[1] (0,0)
\Align[c] ({\pic\TriLoop[2] (0,0) \cip}) (-50,0) (50,0)
\cip$$
\caption{TriLoop}
\label{fig:triloop}
\end{figure}

```

Note that the parameter #1 in `TriLoop [#1] (#2,#3)` is magnification factor. On the other hand (#2,#3) is the location of the drawing. We used `Align` to put copies of the same triloop in different locations.

In `GraphTEX 3`, we added a new triloop. The line in `\TriLoop` is made wider by replacing the line in it by a double line. Please refer to Figure 2.

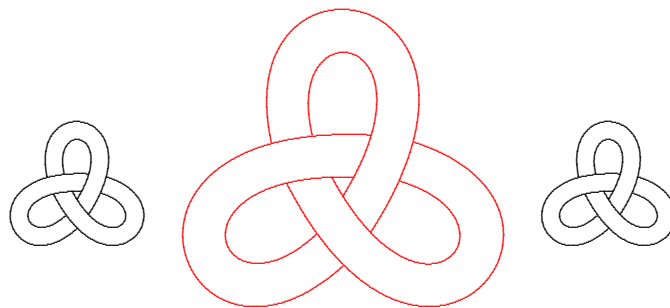


Figure 2: WTriLoop

The code that created Figure 2 is shown below:

```

\begin{figure}[h]
$$\pic
\Align[c] ({\pic \WTriLoop \cip}) (-100,0) (100,0) %Default size
\Magnify2.4
\WTriLoop
\cip$$
\caption{WTriLoop}
\label{fig:wtriloop}
\end{figure}

```

Note that the `\xunit` and `\yunit` are not specified. The default values are used, namely 1pt. We can vary the size of a shape by using the command `\Magnify #1` in `GraphTEX 2`.

Of course, the original `\TriLoop` is still available in `GraphTEX 3`.

2 NEW COMMANDS

Majority of the new commands in `GraphTEX 3` are codes to create letters, numbers, and symbols in a special form – blocks. Other new commands create some special drawings of simple objects like pencil, book, ball, *etc.*

2.1 Block letters, numbers, and symbols

Letters, numbers, and some symbols can be typeset in a special form using `GraphTEX 3` commands. Some examples are shown below.

The code used to produce the above display is

```
\noindent\Blk\M\A\bsp\T\HH\E\M\A\bsp\T\II\C\SX\ \1\1\5\
{\Magnify 1.5 \Blk\SX\T\bsp\A\bsp\T\II\SX\T\II\C\SX}\
{\Magnify 2 \Blk\G\R\A\PP\HH\SX}\
```

Let us take note of the following important points.

1. We always begin a line of block letters/numbers/symbols with `\Blk`.
2. `\Magnify #1` magnifies everything after it #1 times
3. The effect of `\Magnify 1.5` gives a magnification of 1.5 times. To restrict the effect to the second line only, we enclose the entire line in braces. If the braces in lines 2 and 3 are removed, the magnification in line 3 would be 3 times, which is the product of 1.5 and 2.
4. The `\bsp` means *back space*. The amount of backspace is 1.5 pt. This is optional. Its use is to fine-tune the spacing of letters. Try to remove `\bsp` in the above code and you will still produce the display. Compare the new display with the original one to find the effect of `\bsp`.

5. There is a command similar to `\bsp`. This command is `\fsp` which means forward space by 1.5 pt.

Here is another example of the block form.

$$5.8 - (2.5 + 0.4) = 2.9$$

The code that created the above display is

```
\begin{center}
\Blk\5\DOT\8\Minus\LeftPar\2\DOT\5\Plus\0\DOT\4\RightPar\Equals\2\DOT\9
\end{center}
```

Spaces between commands may be provided. For example, we can type the above code as follows.

```
\begin{center}
\Blk \5 \DOT \8 \Minus \LeftPar \2 \DOT \5 \Plus \0 \DOT \4 \RightPar
\Equals \2 \DOT \9
\end{center}
```

The tables that follow summarize the different commands for the block type.

2.1.1 Letters

Command	<code>\A</code>	<code>\B</code>	<code>\C</code>	<code>\D</code>	<code>\E</code>	<code>\F</code>	<code>\G</code>	<code>\HH</code>	<code>\I</code>	<code>\II</code>
Block	A	B	C	D	E	F	G	H	I	II

Command	<code>\J</code>	<code>\K</code>	<code>\LL</code>	<code>\M</code>	<code>\N</code>	<code>\OO</code>	<code>\PP</code>	<code>\Q</code>	<code>\R</code>	<code>\RR</code>
Block	J	K	L	M	N	O	P	Q	R	RR

Command	<code>\SX</code>	<code>\T</code>	<code>\U</code>	<code>\V</code>	<code>\W</code>	<code>\X</code>	<code>Y</code>	<code>\Z</code>
Block	S	T	U	V	W	X	Y	Z

2.1.2 Numbers

Command	<code>\0</code>	<code>\Zero</code>	<code>\1</code>	<code>\One</code>	<code>\2</code>	<code>\3</code>	<code>\4</code>	<code>\5</code>	<code>\6</code>	<code>\7</code>	<code>\8</code>	<code>\9</code>
Block	0	0	1	1	2	3	4	5	6	7	8	9

2.1.3 Symbols

Command	\Period	\DOT	\Comma	\LeftPar	\RightPar	\LeftSB	\RightSB
Block	.	◦	,	()	[]

Command	\Colon	\Dash	\Minus	\Plus	\Equals
Block	:	—	-	+	=

2.2 Peso Sign

Sometimes we need to type the peso sign in our document. You can produce ₱ 123.45 and ₱ 123.45 using the codes `\peso 123.45` and `\Peso 123.45`. The space between `\peso` or `\Peso` and the number is optional. The commands also adjust to text size. Examples are shown below.

₱ 35.20 ₱ 48.15
 ₱ 47.13 ₱ 169.28

The commands `\peso` and `\Peso` also work in mathematics mode, *i.e.*, you can type `$_\Peso 123.45$` to obtain ₱ 123.45 in mathematics mode. In non-mathematics mode, it is `\Peso 123.45` which gives the same result ₱ 123.45.

2.3 Some Simple Objects

Three examples of simple objects are drawn using the commands `\Tree`, `\Cloud`, and `\NipaHut`. A drawing using these three commands is shown in Figure 3.



Figure 3: Nipa hut, tree, and cloud

The code for the above drawing is as follows:







```
\begin{figure}[h]
$$\pic
\Align[c] (\NipaHut) (0,0)
\Align[c] (\Magnify 0.6 \Tree) (36,12)
\Align[c] (\Cloud) (-20,30)
```







```


\cip$$
\caption{Nipa hut, tree, and cloud}
\label{fig:view}
\end{figure}






```

The following tables summarize the commands for drawing other simple objects.

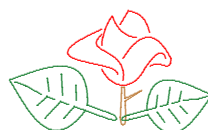
Command	<code>\Lightning</code>	<code>\Coin</code>	<code>\Ball</code>	<code>\OpenBox</code>	<code>\Leaf</code>	<code>\Happy</code>
Object						

Command	<code>\PentaStar</code>	<code>\Gift</code>	<code>\Heart</code>	<code>\Pencil</code>	<code>\Rose</code>	<code>\Sad</code>
Object						

Note that there are lines with color red in `\Gift`. This red color is permanent. The command `\textcolor{cyan}{\Gift}` gives the result .

Command	<code>\Book</code>	<code>\Laptop</code>	<code>\SailBoat</code>	<code>\Cross</code>	<code>\RubiksCube</code>
Object					

We can make a more complicated drawing by combining some objects in the table similar to that in Figure 3. Here is an example.



We need to reorient and resize the `\Leaf` in two different ways and combine them with the `\Rose`. You can see how this is done by examining the code that follows.

```

\begin{figure}[h]
$$\pic

```

```

\newbox\leftleaf
\newbox\rightleaf
\setbox\leftleaf=\hbox{\Magnify 1.1 \Rotate[70] (0,0) \Leaf}
\setbox\rightleaf=\hbox{\Magnify0.6 \Rotate[-75] (0,0) \Leaf}
\Align[b] (\textcolor{brown}{\Rose}) (0,0)
\Align[lr] (\copy\leftleaf) (-1,-5)
\Align[ll] (\copy\rightleaf) (1,-6)
\cip$$
\end{figure}

```

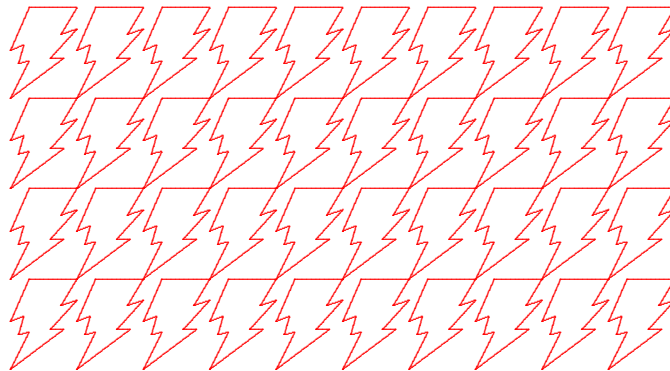
One special method of drawing by replication of a fixed drawing is done by the use of

```
\RectangularArray[#1] (#2) (#3,#4) (#5,#6)
```

For example,

```
\RectangularArray[c] (\textcolor{red}{\Lightning}) (4,10) (0,0)
```

produces

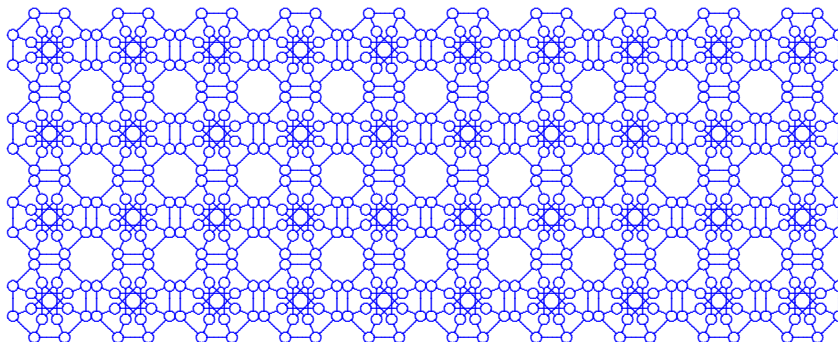


The argument *c* means center. Other options are *b* bottom, *t* top, *l* left, *r* right, *ul* upper left, *ll* lower left, *ur* upper right, and *lr* lower right. In our example we produced an array of 4 rows and ten columns as specified in the argument (4, 10). The center of the array is placed at (0, 0)

Rectangular arrays may use, aside from the objects in the table, any set of texts or any graphtex drawing. For example, the command

```
\RectangularArray[c] (\pic \textcolor{blue}{\Petersen(8,3)}
[0.4cm] (0,0) } \cip) (4,10) (0,0)
```

produces the following drawing.



Take note that there is a required space (ore more) after $(0,0)$ and before the right brace of $\{\backslash\text{Petersen} \text{ --- } (0,0) \}$.

2.4 Special Drawings

2.4.1 Power of a Cycle

To draw the graph of the power of a cycle C_n^k , we use $\backslash\text{CyclePower}(n,k) [s] (x,y)$ where n is the length of the cycle, s is the length of one side, and (x,y) is the location of the center. For example $\backslash\text{CyclePower}(10,4) [0.8\text{cm}] (0,0)$ produces the graph shown in Figure 4.

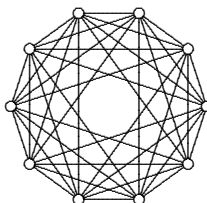


Figure 4: The graph C_{10}^4

The parameter 0.8cm is the length of one side of the cycle in the drawing.

Make sure that you do not subject this command to the effect of `Translate` or `Reflect` or `Rotate`.

2.4.2 Popeye

Do you know the cartoon character Popeye, the sailor man? Here, I made a drawing of Popeye using Beziér curves. The `GraphTEX 3` command is $\backslash\text{Popeye}$.



Figure 5: Popeye the sailor man

Here is the code for the drawing in Figure 5.

```

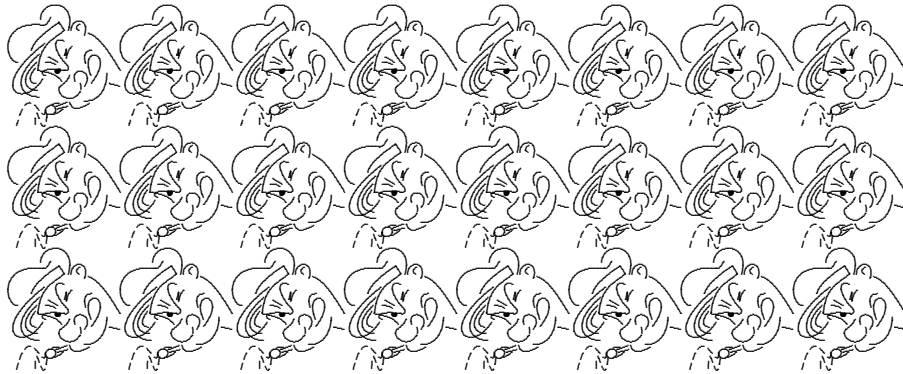
\begin{figure}[h]
$$\pic
\Align[c] (\Popeye) (0,0)
\Align[c] (\Magnify 0.8 \Popeye) (100,0)
\Align[c] (\Magnify 0.5 \Popeye) (60,-50)
\cip$$
\caption{Popeye the sailor man}
\label{fig:popeye}
\end{figure}

```

How do you produce the drawing below? (Use `\RectangulaArray`)



A drawing can be reflected or rotated before putting it in a rectangular array. Here is an example.



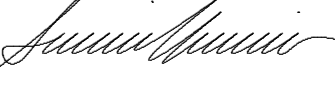
Here is the code that produces the above array.

```

 $\$$  $\$$ \pic
\newbox\tempbox
\setbox\tempbox=\hbox{\Magnify0.5 \Reflect(0,0) (-1,1) \Popeye}
\RectangularArray[c] (\copy\tempbox) (3,8) (0,0)
\cip $\$$  $\$$ 

```

2.4.3 My Signature

The command `\svg` produces  and the size can be varied as usual using `\Magnify`. The line thickness may also be varied using `\PlotSize`. Some

examples are 

This command is for my personal use only and so I removed it from the distribution file of Graph_{TEX} 3.

I would like to end with a very important information. Always remember that Graph_{TEX} 3 automatically loads the package `xcolor` and so it is not necessary that you include `xcolor` in the `\usepackage` of your document. In case you still include `xcolor` in the `\usepackage` of your document, there is actually no problem. If there are no errors in your document you can still compile your file. However, the computer memory usage will be more.