TYPSETTING GRAPHS IN LATEX USING XY

TERRY LORING

1. Graphs with XY-pic

I use a package of latex macros called XY-pic. The homepage is

http://www.maths.mq.edu.au/~ross/Xy-pic.html

I am not wanting to replicate the documentation that exists for the XY-pic package. I just want to give some simple examples that are done in a style I find comfortable.

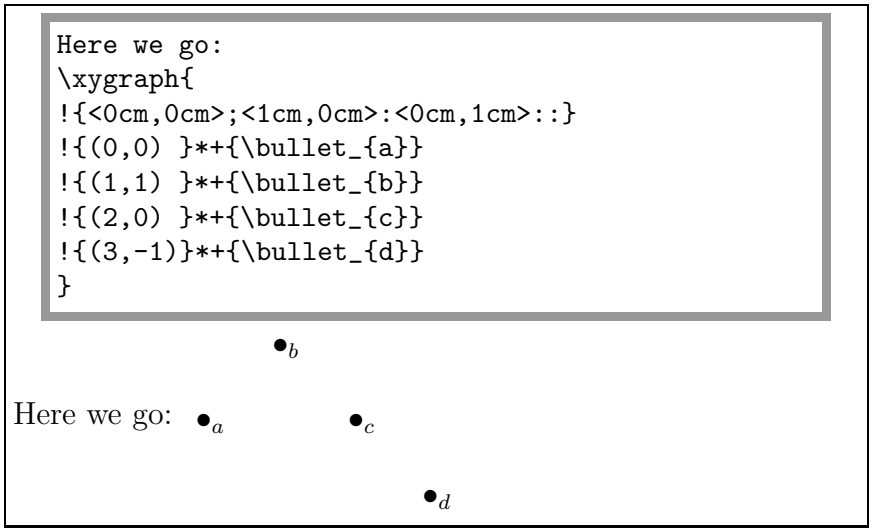
I have tested these examples using teTeX version 3.0 on a linux system. XY-pic is included in teTeX. It is also distrubuted with miktex, and is naturally available via CTAN.

2. From xygraph to PS to PDF

The easiest way I find for making a graph is the xygraph feature. The only part I find annoying is you need to use lots of symbols to get to work in cartesian coordinates. Once you learn the syntax, it is not a problem.

Example 1. Here is a small bit of xy-pic code that puts vertices in specific locations.

```
Here we go:
\xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!{(0,0) }*+{\bullet_{a}}
!{(1,1) }*+{\bullet_{b}}
!{(2,0) }*+{\bullet_{c}}
!{(3,-1)}*+{\bullet_{d}}
}
```

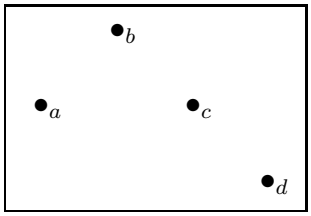


One almost never wants to put a graph in-line, so let's wrap this in an equation environment. Also, let's draw a frame around the graph, which is important if the graph is not connected.

Example 2.

```
Here we go:
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!{(0,0) }*+{\bullet_{a}}
!{(1,1) }*+{\bullet_{b}}
!{(2,0) }*+{\bullet_{c}}
!{(3,-1)}*+{\bullet_{d}}
} } \]
```

Here we go:



The previous fragement would be much good if we don't have a complete latex file and the correct commands to convert it from latex to PDF. Figure 1 shows a complete latex document, with the previous example and something a bit more exotic.

In the line

```
\usepackage[all,dvips,arc,curve,color,frame]{xy}
```

I am loading the XY package and a bunch of its subpackages. Most importantly, the option `dvips` is telling XY-pic that I plan to use the program dvips to convert from a dvi file to a ps file. This means that the lines and curves will end up as lines and curves in the PDF file, not simulated lines and curves made from multiple characters form special fonts.

To get this document from graphExample.tex to graphExample.pdf you need to run the following:

```
latex graphExample.tex
latex graphExample.tex
dvips -o graphExample.ps graphExample.dvi
ps2pdf graphExample.ps
```

If the look craggy, you may have an old version of tex installed. If the lines are craggy, you should turn on smoothing for line art in acrobat/acroread under edit > preferences > general.

FIGURE 1. A sample Latex File with Two Graphs

```
%% graphExample.tex:  A small latex file with an XY graph

\documentclass[12pt]{amsart}

%% Here I load the macros I need from XY-pics, including
%% options I may not use.
\usepackage[all,dvips,arc,curve,color,frame]{xy}

%% Declaring a new color for use in a XY graph.
\newxyColor{pink}{1.0 0.4 0.5}{rgb}{}

\begin{document}

Here we go:
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!{(0,0) }*+{\bullet_{a}}
!{(1,1) }*+{\bullet_{b}}
!{(2,0) }*+{\bullet_{c}}
!{(3,-1)}*+{\bullet_{d}}
} } \]

Now we get fancy:
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!~-{@{-}@[|(2.5)]@[pink]}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a"-"b"  "a"-"d"
"b"-"c"
"b"-@/_/@[green]"d"
"b"-@/^/"d"
"b"-@/^1cm/"d"
} } \]

\end{document}
```

3. Setting your units

In Example 2, the line

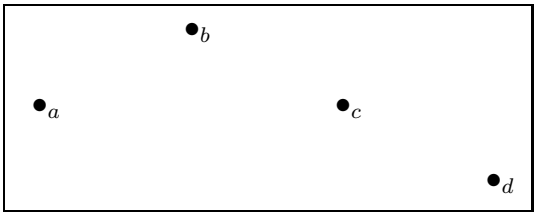
```
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
```

sets up the coordinate system. This is done by giving three vectors in the "real" coordinate system, where x increases as we move right and y increases as we move *up*.. The first specifies where you want the origin, and I always set this to `<0cm,0cm>`. The next two vectors specify the new x and y unit vectors.

We can make things bigger, or smaller, or flip things upside-down by adjusting this.

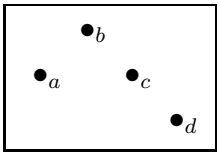
Example 3. Wider:

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<2cm,0cm>:<0cm,1cm>::}
!{(0,0) }*+{\bullet_{a}}
!{(1,1) }*+{\bullet_{b}}
!{(2,0) }*+{\bullet_{c}}
!{(3,-1)}*+{\bullet_{d}}
} } \]
```



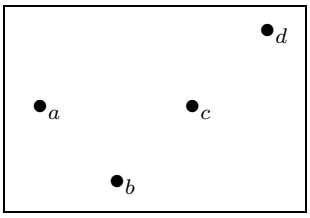
Example 4. Smaller:

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<0.6cm,0cm>:<0cm,0.6cm>::}
!{(0,0) }*+{\bullet_{a}}
!{(1,1) }*+{\bullet_{b}}
!{(2,0) }*+{\bullet_{c}}
!{(3,-1)}*+{\bullet_{d}}
} } \]
```



Example 5. Upside-down:

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,-1cm>::}
!{(0,0) }*+{\bullet_{a}}
!{(1,1) }*+{\bullet_{b}}
!{(2,0) }*+{\bullet_{c}}
!{(3,-1)}*+{\bullet_{d}}
} } \]
```



4. Adding Edges

Now suppose we want to make a graph with these four vertices. All we need to do is add internal names for the vertices using

```
="place"
```

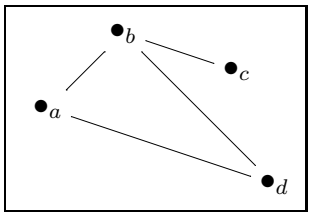
and then put in

```
"here"-"there"
```

for each edge.

Example 6.

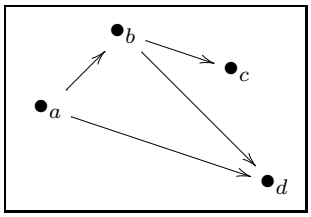
```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a"-"b"  "a"-"d"
"b"-"c"
"b"-"d"
} } \]
```



If you want to draw a directed graph, you replace each - by a :.

Example 7.

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a":"b"  "a":"d"
"b":"c"
"b":"d"
} } \]
```



5. Bending Edges

Sometimes moving the vertices around is not enough to get a nice drawing of a graph. Often one needs to bend some edges. Changing

```
"here" - "there"
```

to

```
"here" -@/^/ "there"
```

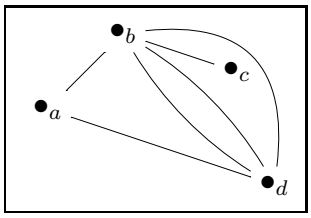
will bend an edge in one direction or the other (generally moving the center in the direction 90-degrees counter-clockwise from the direction from "here" to "there.") You can bend it the other way with

```
"here" -@/_/ "there"
```

Furthermore, you can specify how far you want to move the middle of an edge by adding a tex-stye length before the closing slash.

Example 8.

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a"-"b"  "a"-"d"
"b"-"c"
"b"-@/_/"d"
"b"-@/^/"d"
"b"-@/^1cm/"d"
} } \]
```

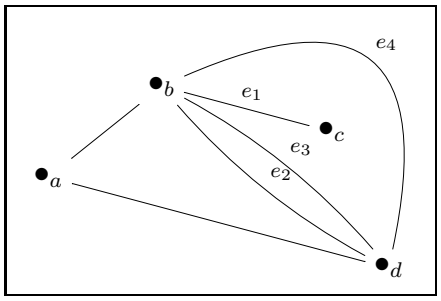


6. Labeling Edges

You can put a label "above" or "below" a line. Here "above" is moved off the edge in the same direction as `@/^/` bends an edge.

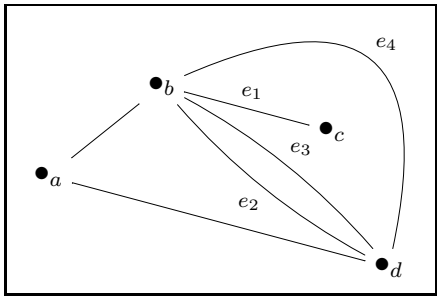
Example 9.

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1.5cm,0cm>:<0cm,1.2cm>::}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a"-"b"  "a"-"d"
"b"-"c" ^{e_1}
"b"-@/_/"d" ^{e_2}
"b"-@/^/"d"  ^{e_3}
"b"-@/^2cm/"d" ^{e_4}
} } \]
```



Example 10.

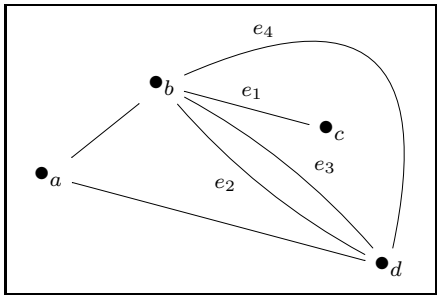
```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1.5cm,0cm>:<0cm,1.2cm>::}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a"-"b"  "a"-"d"
"b"-"c" ^{e_1}
"b"-@/_/"d" _{e_2}
"b"-@/^/"d"  ^{e_3}
"b"-@/^2cm/"d" ^{e_4}
} } \]
```



You can move the labels:

Example 11.

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1.5cm,0cm>:<0cm,1.2cm>::}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a"-"b"  "a"-"d"
"b"-"c" ^{e_1}
"b"-@/_/"d" _(0.4){e_2}
"b"-@/^/"d"  ^(0.6){e_3}
"b"-@/^2cm/"d" ^(0.2){e_4}
} } \]
```

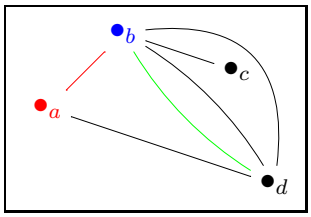


7. Color and Width

Color helps in a lot of cases. You can make a vertex or an edge a color if you put in `[red]` `[green]` or `[blue]` in the right place.

Example 12.

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!{(0,0) }*+[red]{\bullet_{a}}="a"
!{(1,1) }*+[blue]{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a"-@[red]"b"  "a"-"d"
"b"-"c"
"b"-@/_/@[green]"d"
"b"-@/^/"d"
"b"-@/^1cm/"d"
} } \]
```



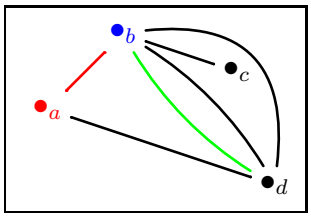
This is hard to see, and if we use colors like pink it will get worse. We need a way to make an edge wider. The modifier `[|(n)]` makes the line n-times the normal width.

If you want to change the default width, you can do so with the following line of mubletypeg (where n is the desired multiple).

```
!~-{@{-}@[|(n)]}
```

Example 13.

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!~-{@{-}@[|(2.5)]}
!{(0,0) }*+[red]{\bullet_{a}}="a"
!{(1,1) }*+[blue]{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a"-@[red]"b"  "a"-"d"
"b"-"c"
"b"-@/_/@[green]"d"
"b"-@/^/"d"
"b"-@/^1cm/"d"
} } \]
```

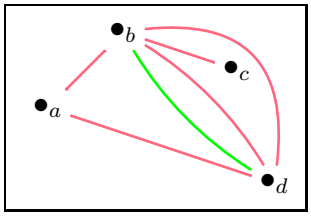


If you want a custom color, one way to get it is to describe a new color in terms of red, green and blue light. Here 0 means no light in that color and 1 means as much light as possible in that color.

Here I define a pink and then define that as the default color.

Example 14.

```
\newxyColor{pink}{1.0 0.4 0.5}{rgb}{}
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!~-{@{-}@[|(2.5)]@[pink]}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a"-"b"  "a"-"d"
"b"-"c"
"b"-@/_/@[green]"d"
"b"-@/^/"d"
"b"-@/^1cm/"d"
} } \]
```

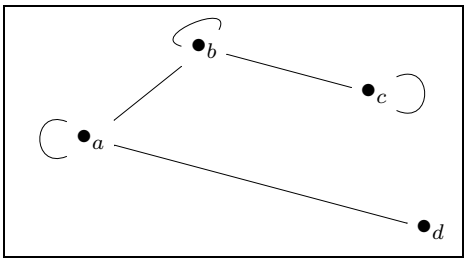


8. Loops

Loops are a bit tricky. One loop is easy enough. You modify the edge command (the minus sign) with `@(d1,d2)` where d1 and d2 are the in and out directions, where your choices are `l r u d` for left, right, up and down, and `dr dl ur ul` for the four diagonals.

Example 15.

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1.5cm,0cm>:<0cm,1.2cm>::}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a"-"b"  "a"-"d"
"b"-"c"
"a" -@(lu,ld) "a"
"b" -@(l,ru) "b"
"c" -@(ru,rd) "c"
} } \]
```

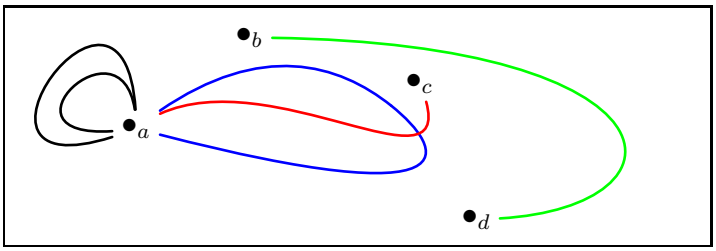


The result is asthetically challenged, and there is no good way to put in lots of loops at the same vertex. You can specify control points that determine the shape of any arrow. If you don't know bezier curves, or a graphics program like Illustrator (by Adobe, TM TM), then just think that the curve will move smoothly toward the first control point, and before it gets there, moves toward the second, and so forth.

There are two control points you don't need to mention. They are `c` for the source of the edge (really an arrow with no head) and `p` for the target. You can add positions to these. You can also use the names of there vertices.

Example 16.

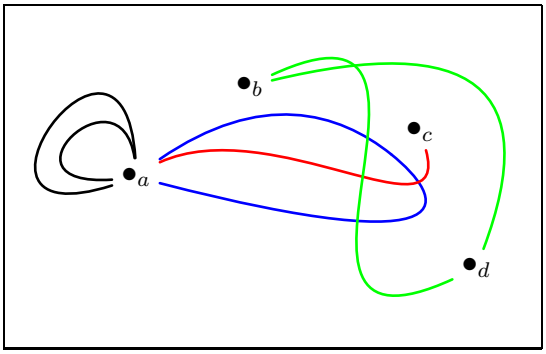
```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1.5cm,0cm>:<0cm,1.2cm>::}
!~-{@{-}@[|(2.5)]}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a" -@`{"b", "c", "d"}@[blue] "a"
"a" -@`{p+(0,2.5),p+(-2,-1)} "a"
"a" -@`{"a"+(0,1.5),"a"+(-1.5,-0.3)} "a"
"a" -@`{"b", "d"}@[red] "c"
"b" -@`{c+(4,0),p+(2,0)}@[green] "d"
} } \]
```



If you want to refer to the mid-point, or any other point on the line between the source and target, you can, but the syntax is again worse. (Things seem backward here, so see the reference manual.)

Example 17.

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1.5cm,0cm>:<0cm,1.2cm>::}
!~-{@{-}@[|(2.5)]}
!{(0,0) }*+{\bullet_{a}}="a"
!{(1,1) }*+{\bullet_{b}}="b"
!{(2.5,0.5) }*+{\bullet_{c}}="c"
!{(3,-1)}*+{\bullet_{d}}="d"
"a" -@`{"b", "c", "d"}@[blue] "a"
"a" -@`{p+(0,2.5),p+(-2,-1)} "a"
"a" -@`{"a"+(0,1.5),"a"+(-1.5,-0.3)} "a"
"a" -@`{"b", "d"}@[red] "c"
"b" -@`{**{}?(0.5)+(2,2)}@[green] "d"
"b" -@`{**{}?(0.8)+(2,2),?(0.2)+(-2,-2)}@[green]"d
} } \]
```

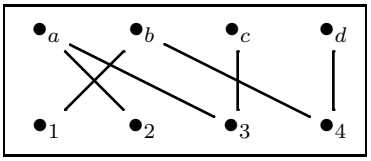


9. Matrix Moves

The reason for the strange syntax for putting vertices down using cartesian coordinates is that `\xygraph` was designed to work like a matrix. You put in vertices separated by `&` along rows separted by `\\`.

Example 18.

```
\[ \fbox{ \xygraph{
!~-{@{-}@[|(2.5)]}
*+{\bullet_{a}}="a"      & *+{\bullet_{b}}="b"
   & *+{\bullet_{c}}="c" & *+{\bullet_{d}}="d" \\
*+{\bullet_{1}}="1"      & *+{\bullet_{2}}="2"
   & *+{\bullet_{3}}="3" & *+{\bullet_{4}}="4" \\
"a"-"2" "a"-"3" "b"-"1" "b"-"4" "c"-"3" "d"-"4"
} } \]
```



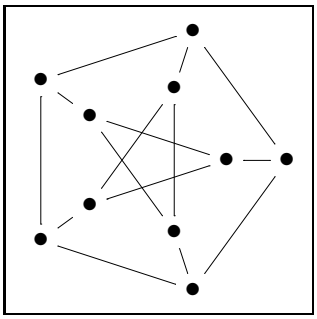
10. Polar Coordinates

Polar Coordinates are very useful when drawing graphs. The syntax is a bit odd. Basically you drop a null object at the origin, then an null object on the unit circle, create the line that they determine and then move along the line the the correct place.

If you use large unit-vectors in your coordinate system and thus only use raddii less than one, latex will become confused by the null objects on the unit circle.

Example 19.

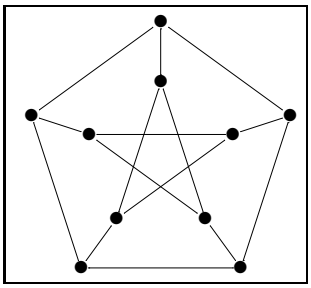
```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<1cm,0cm>:<0cm,1cm>::}
!{(0,0);a(0)**{}?(1.0)}*+{\bullet}="a1"
!{(0,0);a(72)**{}?(1.0)}*+{\bullet}="a2"
!{(0,0);a(144)**{}?(1.0)}*+{\bullet}="a3"
!{(0,0);a(216)**{}?(1.0)}*+{\bullet}="a4"
!{(0,0);a(288)**{}?(1.0)}*+{\bullet}="a5"
!{(0,0);a(0)**{}?(1.8)}*+{\bullet}="b1"
!{(0,0);a(72)**{}?(1.8)}*+{\bullet}="b2"
!{(0,0);a(144)**{}?(1.8)}*+{\bullet}="b3"
!{(0,0);a(216)**{}?(1.8)}*+{\bullet}="b4"
!{(0,0);a(288)**{}?(1.8)}*+{\bullet}="b5"
"a1"-"a3" "a3"-"a5" "a5"-"a2" "a2"-"a4" "a4"-"a1"
"b1"-"b2" "b2"-"b3" "b3"-"b4" "b4"-"b5" "b5"-"b1"
"a1"-"b1" "a2"-"b2" "a3"-"b3" "a4"-"b4" "a5"-"b5"
} } \]
```



This looks better if we rotate the whole thing, which we do by changing the coordinate system. Also, Let's change the spaceing between the vertices and edges by using `*{...}` instead of `*+{...}`.

Example 20.

```
\[ \fbox{ \xygraph{
!{<0cm,0cm>;<0cm,1cm>:<-1cm,0cm>::}
!{(0,0);a(0)**{}?(1.0)}*{\bullet}="a1"
!{(0,0);a(72)**{}?(1.0)}*{\bullet}="a2"
!{(0,0);a(144)**{}?(1.0)}*{\bullet}="a3"
!{(0,0);a(216)**{}?(1.0)}*{\bullet}="a4"
!{(0,0);a(288)**{}?(1.0)}*{\bullet}="a5"
!{(0,0);a(0)**{}?(1.8)}*{\bullet}="b1"
!{(0,0);a(72)**{}?(1.8)}*{\bullet}="b2"
!{(0,0);a(144)**{}?(1.8)}*{\bullet}="b3"
!{(0,0);a(216)**{}?(1.8)}*{\bullet}="b4"
!{(0,0);a(288)**{}?(1.8)}*{\bullet}="b5"
"a1"-"a3" "a3"-"a5" "a5"-"a2" "a2"-"a4" "a4"-"a1"
"b1"-"b2" "b2"-"b3" "b3"-"b4" "b4"-"b5" "b5"-"b1"
"a1"-"b1" "a2"-"b2" "a3"-"b3" "a4"-"b4" "a5"-"b5"
} } \]
```



E-mail address: loring@math.unm.edu