An Introduction to XY-pic

Cameron McLeman

Department of Mathematics
The University of Arizona

October 17, 2007

Intro

- LaTeXpackage.
- Developed principally by Kris Rose and Ross Moore.
- Large community support base.
- Preamble: `\usepackage[all]{xy}`.
- Extra output on compile.

References

- K.H. Rose: Xy-pic User's Guide (16 pages)
- K.H. Rose & R. Moore: Xy-pic Reference Manual (81 pages)
- A. Perlis: Axis Alignment in Xy-pic diagrams.
- Aaron Lauda: Xy-pic tutorial with an archive of examples:
  `http://www.dpmms.cam.ac.uk/~al366/xytutorial.html`

All (and many more) available online.

Objects

```
This diagram is very important:
\begin{equation*}
\xymatrix{
M&c&L\\
E&M&A&N}
\end{equation*}
```

This diagram is very important:

$$\begin{array}{cccc} M & c & L & \\ E & M & A & N \end{array}$$

Some Xy-pic options

Compare `\xymatrix{M&c&L\\E&M&A&N}`

$$\begin{array}{cccc} M & c & L & \\ E & M & A & N \end{array}$$

to `\xymatrix@C=1pc@R=1pc{M&c&L\\E&M&A&N}`

$$\begin{array}{cccc} M & c & L & \\ E & M & A & N \end{array}$$

Arrows

Arrow (`\ar`) directions are relative, using $\{d, l, u, r\}$ to navigate, e.g.,

```
\begin{equation*}
\xymatrix@C=2pc@R=2pc{
M & c\ar[dr]\ar[drr] & L\\
E\ar[u] & M\ar[l] & A & N}
\end{equation*}
```

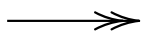
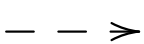
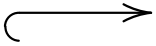
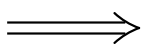
gives

$$\begin{array}{ccccc} M & & c & & L \\ \uparrow & & & \searrow & \\ E & \leftarrow & M & & A \qquad N \end{array}$$

Note that a `\ar[r]` from the "L" or a `\ar[d]` from the "A" give errors.

Arrows II: Style

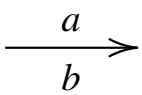
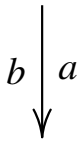
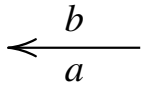
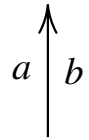
Use an `@`-modifier to change the arrow style (tail, shaft, and head):

Code	Style	Comments
`\ar@{->>}`		Surjection
`\ar@{--}>}`		Implied Existence
`\ar@{|->}`		Defined on Elements
`\ar@{^(->}`		Injection
`\ar@{_(->}`		Bizarro Injection
`\ar@{-}`		Field Extension
`\ar@{~>}`		Functorial Correspondence
`\ar@{=>}`		Implies
`\ar@3{~>>}`		Strongly sort of implies
`\ar@{||~<}`		Umm...quasipseudoisomorphism...

Note: Design your own!

Arrows III: Labels

Use ^ for labels "above the arrow," and _ for "below":

`\ar[r]^a_b`	`\ar[d]^a_b`	`\ar[l]^a_b`	`\ar[u]^a_b`
			

Combine with styles via `\ar@{->>}^a_b[r]` (order counts!):

$$A \overset{a}{\underset{b}{\twoheadrightarrow}} B$$

Arrows for Nit-pickers

$$0 \longrightarrow A \xrightarrow{\sigma_p+\psi_p} \prod_{\mathfrak{p}|p} B_{\mathfrak{p}} \oplus B' \xrightarrow{\tau} C \longrightarrow 0$$

vs.

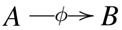
$$0 \longrightarrow A \xrightarrow{\sigma_p+\psi_p} \prod_{\mathfrak{p}|p} B_{\mathfrak{p}} \oplus B' \xrightarrow{\tau} C \longrightarrow 0$$

Used `@C` for spacing, `\ar[r]^(.33){\sigma_p+\psi_p}` for label placement, and Dr. Alex Perlis' command `\entrymodifiers={+!!<0pt,\fontdimen22\textfont2>}` for axial alignment instead of center alignment.

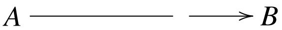
Holes and Breaks

Some final decoration tricks:

```
\xymatrix{A\ar[r]|\phi&B}
```

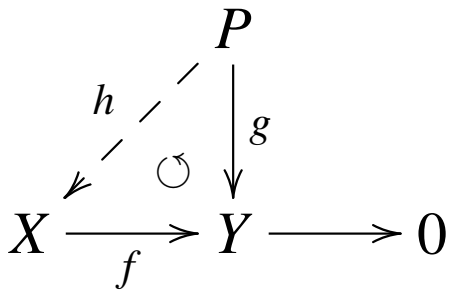


```
\xymatrix{A\ar'[rr][rrr]&&&B}
```



Example 1: Basic Commutative Diagram

The projective module diagram:

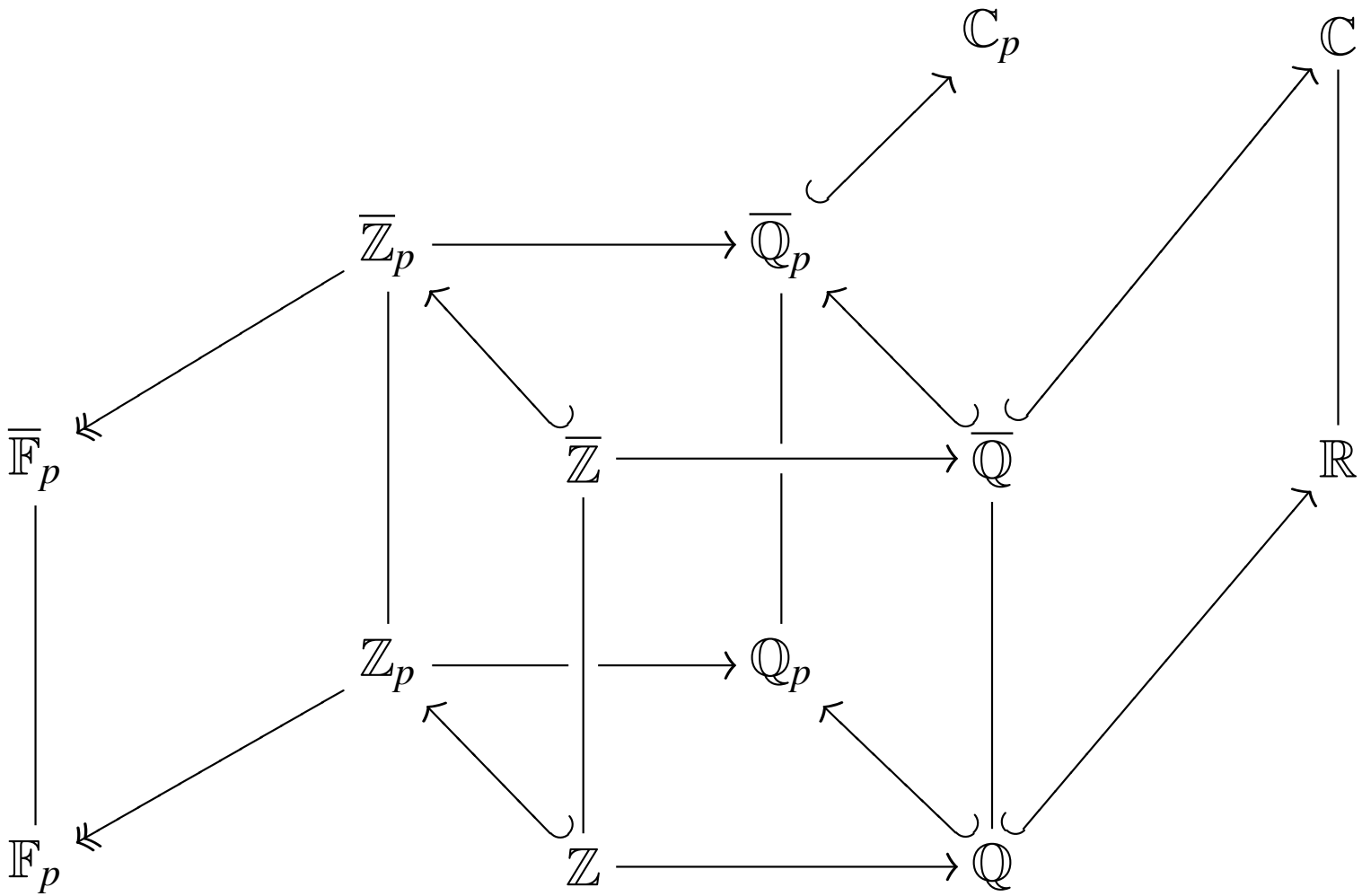


was typeset using

```
\xymatrix{
\ar@{}|(.7)\cal[dr]&P\ar@{-->}[dl]_h\ar[d]^g\\
X\ar[r]_f&Y\ar[r]&0
}
```

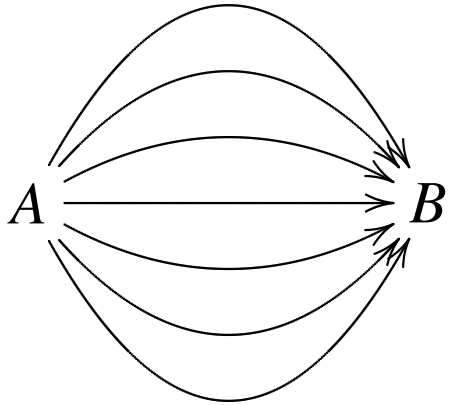
(\cal = \circlearrowleft)

Example 2: Mixing and Matching



Advanced Arrows: Easy Curving

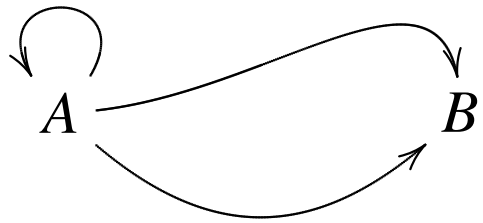
Use `@/_<curve amount>/` or `@/^<curve amount>/`



The above was generated by:
`\ar@/^npc/[r]` for $n \in \{0, 1, 2, 3\}$
`\ar@/_npc/[r]` for $n \in \{1, 2, 3\}$.

Advanced Arrows: Easy Curving

Alternatively, specify outgoing and incoming directions with `\ar@(<out>,<in>)`:

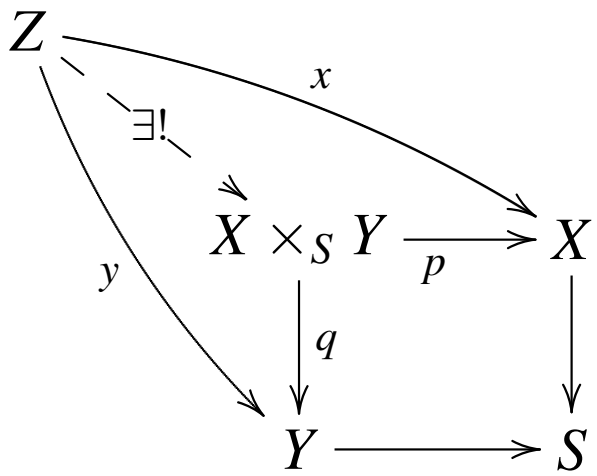


The three arrows are:

- `\ar@(r,u)[r]`
- `\ar@(dr,dl)[r]`
- `\ar@(ur,ul)[]`

Example 3: Pull-Back Diagrams

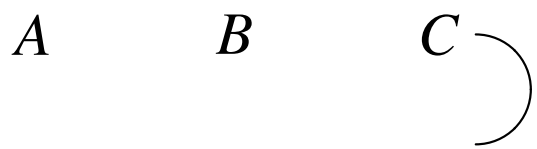
Universal properties:




```
\xymatrix{Z \ar@/_/[ddr]_y \ar@/^/[drr]^x
\ar@{-->}[dr] |(.45){\exists !} \\
& X \times_S Y \ar[d]^q \ar[r]_p & X \ar[d]
\\ & Y \ar[r] & S}
```

Advanced Arrows: Harder Curving

Make detours using ` and turn commands. A simple curved arrow:

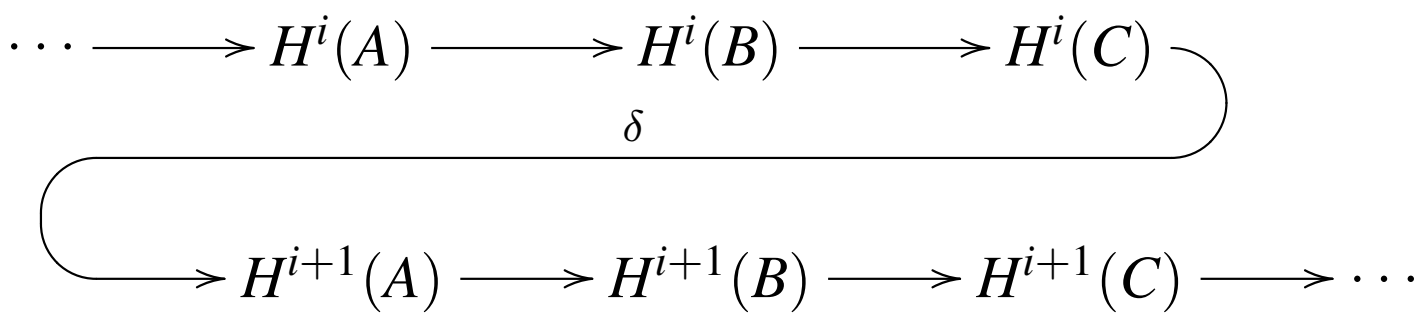


"Start out of the r side of C, make a $\frac{1}{4}$-turn towards [d], continue the same direction and then make a quarter turn toward [l]".

```
\xymatrix{A&B&C
\ar@{-} `r[d] `[l] \\
&&}
```

Example 4: The Connecting Homomorphism

With sufficient patience:




```
\xymatrix{ \cdots\ar[r] & H^i(A) \ar[r]&
H^i(B) \ar[r] & H^i(C)\ar `r[d] `[l]
`[llld]_{\delta} `[dll]  [dll]\\
& H^{i+1}(A) \ar[r] &  H^{i+1}(B)
\ar[r] & H^{i+1}(C)\ar[r] & {\dots} }
```

Actual Xy-pic

Xy-pic:

- Name comes from xy-coordinates.
- `\xymatrix` is just a front-end.
- More flexible, less intuitive.

Basic commands:

- Set up coordinates.
- Make something and put is somewhere.
- Connect two things.

Make things and connect things:

- `<pos>*<object>`
- `**<arrow>`

```
\[\begin{xy}
(0,0)*{B};
(10,0)*{a};
**{an};
\end{xy}\]
```

$$Banana$$

Use `**{-}` for straight lines.

Use labels and reconstruct arrow shafts with `{\ar}`:

```
\[\begin{xy}
(0,0)*+{A}="A";
(20,0)*+{B}="B";
{\ar@{->} "A";"B"};
\end{xy}\]
```

$$A \longrightarrow B$$

Note: Careful about `''` vs. `"` in emacs.

Curves in \xy

Curving using Bezier curves and B-splines:

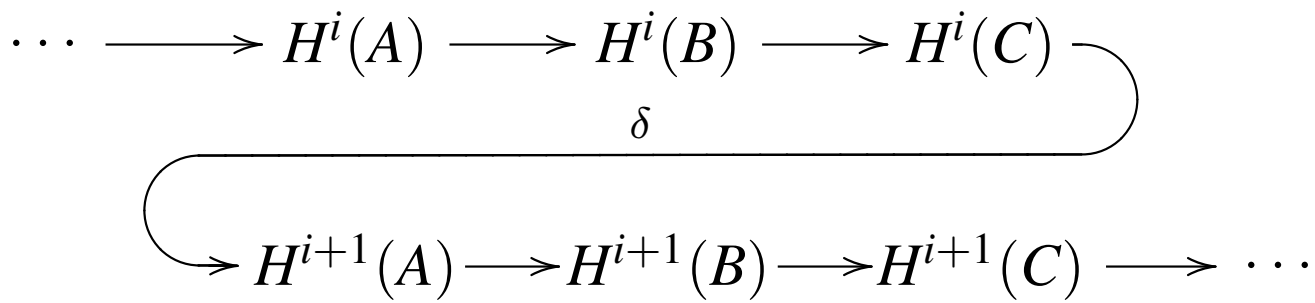
```
\[\begin{xy}
(0,0)*+{A}="A";
(20,0)*+{B}="B";
**\crv{(5,10)&(15,-10)};
\end{xy}\]
```

Example 5: Connecting Homomorphism Revisited

Curves Using \PATH:

```
\xy
(-20,0)*+{\cdots},{\ar (-16,0);(-6,0)};
(0,-14.2)*+{H^{i+1}(A)}="target",
(59,-14.2)*+{\cdots},{\ar (48,-14.2);(55,-14.2)};
{\ar (7,-14.2);(13,-14.2)};
(20,-14.2)*+{H^{i+1}(B)},{\ar (27,-14.2);(33,-14.2)};
(40,-14.2)*+{H^{i+1}(C)},(0,0)*+{H^i(A)};
(20,0)*+{H^i(B)};{\ar (6,0);(14,0)};
(40,0)*+{H^i(C)}="C";{\ar (26,0);(34,0)};
\PATH ~={**\dir{-}?>*\dir{}}~>{|>*\dir{>}}
'_d (50,-5)
'_l (50,-5)
' (-10,-7.1)_\delta
'^d (-10,-10)
'^r (-10,-10)
"target",
\endxy
```

Example 5: Connecting Homomorphism Revisited



Comparison to previous technique–

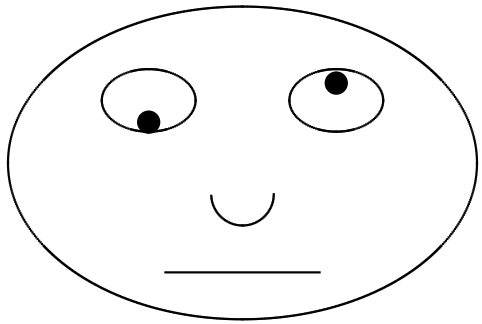
- Pros: More customizable
- Cons: Less automated.

Circles and Ellipses

Need to include `\usepackage[arc,all]{xy}`:
`(0,5)*\ellipse(3,1){-};`

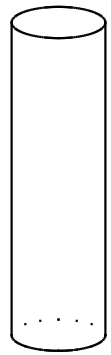
- Centered at $(0, 5)$.
- Horizontal length 3, vertical length 1.
- Use `@{style}` to change style.
- Can make partial arcs by specifying angle.

Samples:



Composite Constructions

Three ellipses and two lines:

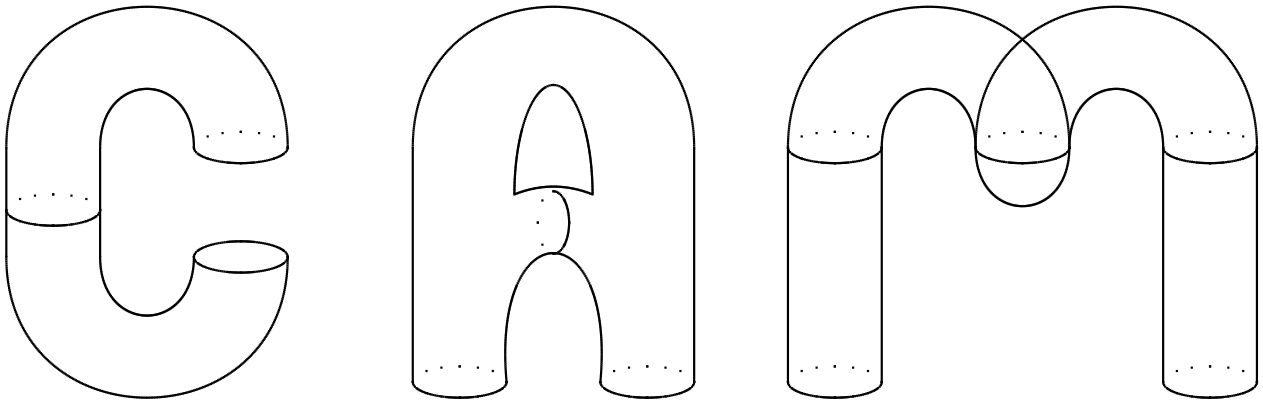


Shading via:

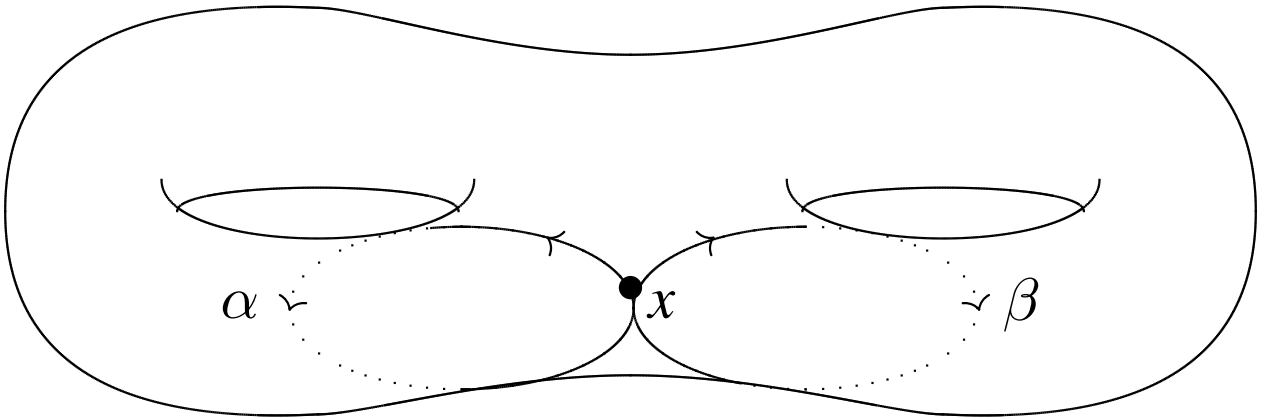
```
(0,-5)*\ellipse(3,1){.};
(0,-5)*\ellipse(3,1)__,=:a(-180){-};
```

Even Compositer Constructions

Again, some things come in very handy...



And finally, a less useful example...



$$\alpha\beta\alpha^{-1}\beta^{-1} = 1$$