The bondgraphs package*

Geert Folkertsma
g.a.folkertsma@ieee.org

March 23, 2015

Abstract

The bondgraphs package is used to draw bond graphs in LaTeX. It uses a recent version (3.0+) of TikZ for the drawing, hence, it is mainly a set of TikZ styles that makes the drawing of bond graphs easier.

1 Introduction

A bond graph is a graphical representation of a physical dynamic system. The graph consists of nodes—junctions and elements—and edges—bonds—just like any other graph. This package, bondgraphs[1], is made to facilitate the drawing of bondgaphs in LaTeX. For more information about bondgraphs, we suggest the original introduction by Henry Paynter ("Analysis and Design of Engineering Systems," ISBN 0-262-16004-8) or the PhD thesis "Physical systems theory in terms of bond graphs" by P.C. Breedveld, ISBN 90-9000599-4.

Note that this package relies on a very recent version of TikZ/pgf, namely 3.0. This version is shipped with recent (2014+) versions of LaTeX.

To conclude this introduction, Figure 1 shows two analogous physical systems and the corresponding bondgraph.

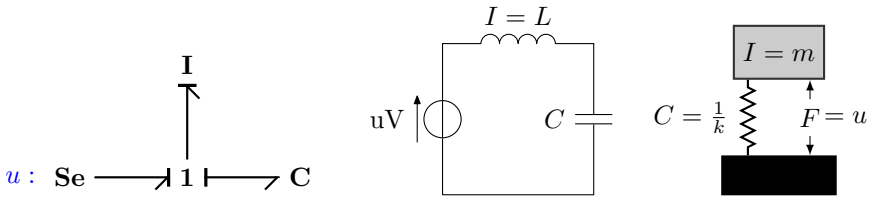


Figure 1: Two analogous physical systems: in the electrical and mechanical domain; and the bondgraph that describes both.

*This document corresponds to bondgraphs v1.0.1, dated 2015/03/23.
[1]From now on, we will write "bondgraphs" as a single word.

2 Usage

The bondgraphs package provides styles for complete bondgraph drawing, as well as text-mode commands to draw or typeset single bonds or elements, useful e.g. when explaining concepts or referring to elements in a complete bondgrahp.

2.1 Text-mode

2.1.1 Bonds

`\bond` `\bond [⟨drawing options⟩]`

The `\bond` command draws a single bond from left to right, like so:  . The drawing options can be used to specify causality and various colours and labels, as demonstrated in the table below. Note: the `diff` and `error` are redefined to grey when using the package option `grey`.

Description		Option	Result
Causality	effort in / flow out	`e_in`	
		`f_out`	
	effort out / flow in	`e_out`	
		`f_in`	
Causal stroke	differential causality	`e_out={diff}`	
	erroneous causality	`f_out={error}`	
	arbitrary colour	`f_in={blue}`	
Labels	effort	`effort={$F_2$}`	
	flow	`flow={$v_2$}`	
	both	`flow={f},effort={e}`	
Multi-dimensional bonds		`mbond`	

Note that the bond label colour can be set with the `bondlabelcolour` package option, which is dark green by default, or grey when the `grey` package option is used.

Any option that is not recognised as one of the options listed in the above table is passed on to the TikZ drawing command. This allows for example the colouring of a bond, or the changing of a causal stroke width[2]. A (rather extreme) example:

```
\bond[green,ultra thick,e_in={line width=5pt,width=20pt,orange!60}]
```

gives .

[2] This is actually already used in the table, with the `f_in=blue` option

2.1.2 Elements

`\bgelement` `\bgelement [⟨options⟩] {⟨element⟩}`

Like `\bond`, `\bgelement` typesets a single bondgraph element in text, which is most useful when referring to elements in a complete bondgraph drawing. In contrast to the `\bond` command, `\bgelement` does not use TikZ for drawing, but LaTeX typesetting. The following table shows the various options:

Element type	Extra options	Examples	Result
Normal	–	`\bgelement{R}`	**R**
		`\bgelement{1}`	**1**
	Multiport	`\bgelement[multiport]{C}`	$\mathbb{C}$
		`\bgelement[multiport]{MTF}`	$\mathbb{MTF}$
	n-dimensional	`\bgelement[n=3]{Se}`	$\mathbf{Se}_3$
		`\bgelement[n=1]{Gy}`	**Gy**
	combination	`\bgelement[n=2,multiport]{I}`	$\mathbb{I}_2$
Word-bondgraph	(none allowed)	`\bgelement[wordbg]{Sys}`	Sys

Note that the word-bondgraph element is exception to the no-TikZ rule: it is drawn as an elliptical TikZ node, with argument `{⟨element⟩}` passed as node contents. Hence, it is possible to do something like:

`\bgelement[wordbg]{\bgelement[n=4]{Environment}}`,

which gives **Environment**$_4$.

2.2 Drawing mode

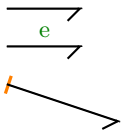
While the text-mode commands have their uses, the most important part of this package is the drawing of actual bondgraphs. Because the drawing of bonds has been implemented using TikZ styles, the drawing of bondgraphs comes down to drawing nodes and connecting edges using TikZ. There are examples in the next section (Section 3: Examples); here only the basic structure is explained.

Note that, because all drawing is done in TikZ, it is very easy to make use of TikZ's advanced positioning features and other libraries. By default, the TikZ-libraries `shapes` and `positioning` are loaded; the example section demonstrates their use.

2.2.1 Bonds

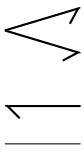
Bonds are drawn using the `bond` style on a path or an edge. Because the text-mode `\bond` command actually draws an edge with the `bond` style appended to the optional `[`⟨*drawing options*⟩`]` (see Section 2.1.1: Bonds), all the options explained there are applicable in drawing mode, too. The example below shows how to draw bonds inside a `tikzpicture` environment:

```
\begin{tikzpicture}
  \draw[bond] (0,0) -- (1,0);
  \draw (0,-.5) edge[bond,effort={e}] (1,-.5);
  \draw[bond,e_in={diff}] (0,-1) to (2,-1.5);
\end{tikzpicture}
```



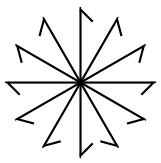
`bonds` A special drawing style is defined, `bonds`, which appends the `bond` style to all `edge`s on the path. It is useful when drawing a lot of bonds in one go—just remember to use `edge` rather than `to` or `--` between nodes or coordinates:

```
\begin{tikzpicture}
  \draw[bonds]
  (0,0)  edge (1,0.3)
         edge (1,-0.3)
  (1,-1) edge (0,-1)
  (0,-1.5) -- (1,-1.5); %won't produce a bond
\end{tikzpicture}
```



Finally, a silly drawing that shows the half-arrow stroke always points down:

```
\begin{tikzpicture}
  \foreach \a in {0,30,...,350}
  {\draw[bond] (0,0) -- (\a:1);}
\end{tikzpicture}
```



There are two more things to mention on the subject of bonds drawing:

1. the package option `curly` will change the straight half-arrow into a curly one;

2. some people prefer the half-arrow stroke to point left or down; this can be tweaked at the top of the generated `.sty` file (see Section 4.3: Bonds), which *might* become a package option in the future.

2.2.2 Elements

Elements and junctions are drawn as TikZ nodes, with the `bgelement` style. This style does two things: it typesets the label text in bold (`\bfseries`) and defines a special label style for element labelling. Some examples:

```
\begin{tikzpicture}
  \node[bgelement] at (0,0) {C};
  \node[bgelement] at (0,-0.5) {Se};
\end{tikzpicture}
```

C
Se

```
\begin{tikzpicture}
  \node[bgelement,label=east:$v_x$] at (0,0) {1};
  \node[bgelement,label=south:m] at (0,-0.5) {I};
\end{tikzpicture}
```

1 v_x
I
m

Note that the element label colour can be set with the `labelcolour` package option, which is blue by default, or grey when the `grey` package option is used.

Important note: the `bgelement` TikZ style is *not* the same as the `\bgelement` text-mode macro. There *is* a `multiport` option/style that works by a lucky stroke, typesetting the node text in blackboard bold font, but it is recommended to instead use the `\bgelement` macro inside the node text for more advanced elements:

`\tikz \node {\bgelement[multiport,n=2]{RS}} ;` gives $\mathbb{RS}_2$

2.3 The `bondgraph` environment

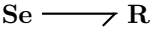
`bondgraph`

The `bondgraphs` package currently defines only one environment: the `bondgraph` environment. It is actually an alias for the `tikzpicture` environment, with two convenient options passed to it automatically:

1. every node gets the style `bgelement`;
2. every edge gets the style `bond` appended to it.

Any options passed to the `bondgraph` environment are passed along to the `tikzpicture` environment. Note that while this environment is a convenient shorthand to quickly draw bondgraphs, the fact that the styles get appended to *every* node and edge mean that it is only suitable for pure bondgraphs—no signal lines, no block diagrams, et cetera, since they would be warped by the bondgraph-specific styles.

```
\begin{bondgraph}
  \node (Se) {Se};
  \node[right=1 of Se] (R) {R};
  %use 'edge': 'to' or '--' has no bond style
  \draw (Se) edge (R);
\end{bondgraph}
```

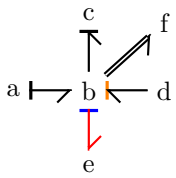


3 Examples

This section contains some basic and more advanced examples of the bondgraphs package usage.

3.1 Various bonds

Example picture, showing bonds from and to several nodes, using the `bond` style and causality (`{e,f}_{in,out}`) options, including bond colours and causal stroke colours:



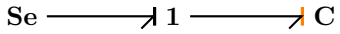
Source:

```
\begin{tikzpicture}
  \node (a) at (0,0) {a};
  \node (b) at (1,0) {b};
  \node (c) at (1,1) {c};
  \node (d) at (2,0) {d};
  \node (e) at (1,-1) {e};
  \node (f) at (2,0.9) {f};

  \draw[bond,e_in] (a) -- (b);
  \draw[bond,e_out] (b) -- (c);
  \draw[bond,e_out={diff}] (d) -- (b);
  \draw[bond,red,f_out={blue}] (b) -- (e);
  \draw[mbond] (b) -- (f);
\end{tikzpicture}
```

3.2 bgelement nodes

Example showing the `bgelement` node style and again bonds between them:



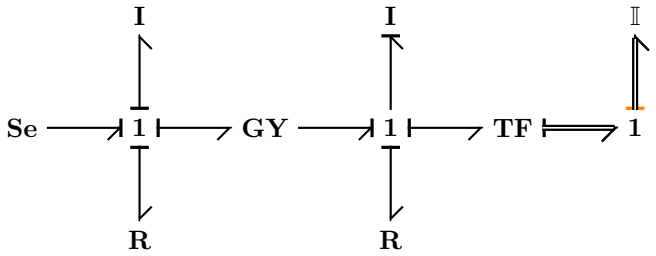
Source:

```
\begin{tikzpicture}
  \node[bgelement] (Se) at (0,0) {Se};
  \node[bgelement] (one) at (2,0) {1};
  \node[bgelement] (C) at (4,0) {C};

  \draw[bond,f_in] (Se) -- (one);
  \draw[f_in={orange},bond] (one) -- (C);
\end{tikzpicture}
```

3.3 Large motor model

Large motor model, with a scope that sets the `bgelement` style for each node; and uses a short-hand `edge` for all the bonds, with the `bonds` style that sets the `bond` style for each edge. Uses the `positioning` TikZ-library.



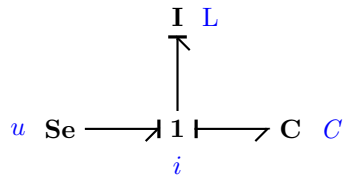
Source:

```
\begin{tikzpicture}[scale=0.7]
  \begin{scope}[every node/.style={bgelement}]
  \node (Se) at (0,0) {Se};
  \node[right=1 of Se] (i) {1};
  \node[above=1 of i] (Iel) {I};
  \node[below=1 of i] (Rel) {R};
  \node[right=1 of i] (GY) {GY};
  \node[right=1 of GY] (w) {1};
  \node[above=1 of w] (Im) {I};
  \node[below=1 of w] (Rm) {R};
  \node[right=1 of w] (TF) {TF};
  \node[right=1 of TF] (ww) {1};
  \node[above=1 of ww,multiport] (Iw) {I};
  \end{scope}

  \draw[bonds]
  (Se)  edge [e_out] (i)
  (i)   edge [e_in] (Iel)
        edge [e_in] (Rel)
        edge [e_in] (GY)
  (GY)  edge [e_out] (w)
  (w)   edge [e_out] (Im)
        edge [e_in] (Rm)
        edge [e_in] (TF)
  (TF)  edge [mbond,e_in] (ww)
  (ww)  edge [mbond,e_in={diff}] (Iw);
\end{tikzpicture}
```

3.4 `bondgraph` environment

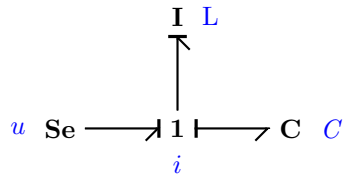
Example of the same bondgraph, twice: once manually with a `tikzpicture` environment and the `every node/.style=bgelement`; `every edge/.append style=bond`; once with the `bondgraph` environment that is an alias for tikzpicture but sets these styles.



Source:

```
\begin{tikzpicture}[every node/.style={bgelement},every edge/.append style={bond}]
  \node[label=left:$u$] at (0,0) (Se) {Se};
  \node[label=below:$i$,right=1 of Se] (i) {1};
  \node[label=right:L,above=1 of i] (L) {I};
  \node[label=right:\textit{C},right=1 of i] (C) {C};

  \draw
    (Se)  edge[e_out] (i)
    (i)   edge[e_out] (L)
          edge[f_out] (C);
\end{tikzpicture}
```



Source:

```
\begin{bondgraph}
  \node[label=left:$u$] at (0,0) (Se) {Se};
  \node[label=below:$i$,right=1 of Se] (i) {1};
  \node[label=right:L,above=1 of i] (L) {I};
  \node[label=right:\textit{C},right=1 of i] (C) {C};

  \draw
    (Se)  edge[e_out] (i)
    (i)   edge[e_out] (L)
          edge[f_out] (C);
\end{bondgraph}
```

3.5 Bondgraph in text

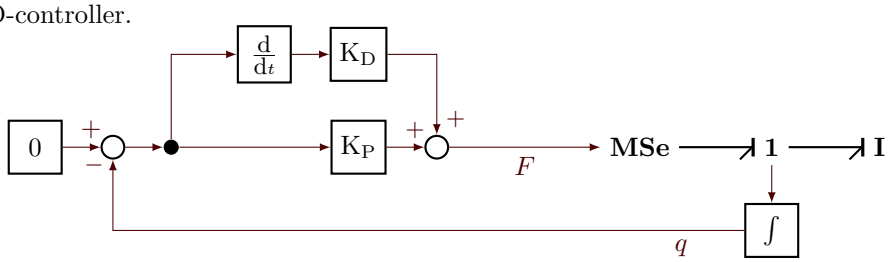
Bondgraph in a line of text looks like this:  and here the sentence continues. Notice the use of TikZ's `baseline` option to align the text with the baseline of the **Se**-element.

Source:

```
Bondgraph in a line of text looks like this: \begin{bondgraph}[baseline=(Se.base)]
  \node(Se){Se};
  \node[right=1 of Se] (C) {C};
  \draw (Se) edge[e_out={orange}] (C);
\end{bondgraph} and here the sentence continues.
Notice the use of \tkz{}'s |baseline| option to align the text with
the baseline of the \bgelement{Se}-element.
```

3.6 Bondgaphs mixed with block diagrams

This is a drawing of a simple mechanical system (a force actuator on a mass), where the flow is integrated to find the position, which is then used as input for a PD-controller.



Source:

```
\tikzstyle{block}=[minimum width=7mm,minimum height=7mm,draw,thick]
\tikzstyle{plusminus}=[circle,thick,draw,inner sep=0pt,minimum size=3mm]
\tikzstyle{splitter}=[circle,fill,minimum size=2mm,inner sep=0pt]
\tikzstyle{signal} = [-latex, color=red!30!black]
\begin{tikzpicture}
  % The bond graph of a mechanical system
  \begin{scope}[every node/.style={bgelement}]
    \node (one) {1};
    \node[left=1 of one] (MSe) {MSe};
    \node[right=1 of one] (I) {I};
  \end{scope}
  \draw[bonds]
    (MSe) edge [e_out] (one)
    (one) edge [e_out] (I);

  % Position sensor integrator
  \node[block,below=.5 of one] (int) {$\int$};
```

```
  % The controller
  \node[plusminus,left=2 of MSe] (pp) {};
  \node[block,left=.5 of pp] (K) {K$_\text{P}$};
  \node[splitter,left=2 of K] (split) {};
  \node[block,above=.5 of K] (D) {K$_\text{D}$};
  \node[block,left=.5 of D] (ddt) {\small $\frac{\mbox{d}}{\mbox{d}t}$};
  \node[plusminus,left=.5 of split] (pm) {};
  \node[block,left=.5 of pm] (setpoint) {0};

  % Draw all signals
  \begin{scope}[every path/.style={signal}]
    \draw (setpoint) -- node[near end,above]{$+$} (pm);
    \draw (pm) -- (split);
    \draw (split) |- (ddt);
    \draw (ddt) -- (D);
    \draw (D) -| node[pos=.9,right]{\small $+$} (pp);
    \draw (split) -- (K);
    \draw (K) -- node[near end,above]{\small $+$} (pp);
    \draw (pp) -- node[below]{$F$} (MSe);
     % The position signal
     \draw (one) -- (int);
     \draw (int) -| node[pos=0.05,below]{$q$} node[pos=.96,left]{\small $-$} (pm);
  \end{scope}
\end{tikzpicture}
```

4 Implementation

4.1 Dependencies

The package heavily relies on TikZ, but also uses some special fonts to typeset bondgraph elements. Note that bondgraphs uses kvoptions to process package options; that dependency is listed below under Section 4.2: Options.

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{bondgraphs}[2015/03/23 v1.0.1 TikZ-based Bond graphs formatting package]
```

TikZ

TikZ does all the drawing for the bondgraphs package, using a number of libraries: the bond half-arrow is from `arrows/arrows.meta` and is drawn using `pathreplacing`; `positioning` isn't really required, but very useful when aligning bondgraph elements; `shapes` is used for the ellipse around word-bondgraph elements.

```
\RequirePackage{tikz}[2013/12/13]
\usetikzlibrary{arrows,arrows.meta,decorations.pathreplacing}
\usetikzlibrary{positioning}
\usetikzlibrary{shapes}
```

As mentioned, amsfonts is used for the mathbb font (multi-port elements) and bm is used to typeset bold math (bold mathbb symbols).

```
\RequirePackage{amsfonts}
\RequirePackage{bm}
```

4.2 Options

kvoptions

The kvoptions package is used to parse package options.

```
\RequirePackage{kvoptions}
\SetupKeyvalOptions{family=bondgraphs,prefix=bondgraphs@}
\DeclareStringOption[blue]{labelcolour}
\DeclareStringOption[green!50!black]{bondlabelcolour}
\DeclareBoolOption[false]{grey}
\DeclareComplementaryOption{colour}{grey}
% Options for the bond drawing
% TODO: curly on double line (multibond) are longer
\DeclareBoolOption[false]{curly}
\ProcessLocalKeyvalOptions{bondgraphs}
```

colours

The bondgraphs package uses colours for labels on bonds and elements, and also to allow the user to easily indicate a differential or erroneous causality, by colouring the causal stroke. Package option "grey" overrides all these colours to grey.

```
\ifbondgraphs@grey
  \message{Bondgraphs package: greyscale set}
  \colorlet{diff}{black!60!white}
  \colorlet{error}{black!40!white}
  \def\bondgraphs@labelcolour{black!50!white}
```

```
  \def\bondgraphs@bondlabelcolour{black!50!white}
  %TODO: would be nice to have this as a style instead of a colour...
\else
  \message{Bondgraphs package: using colour}
  \colorlet{diff}{orange}
  \colorlet{error}{red}
\fi
```

4.3 Bonds

bonds

All bonds are drawn as a (straight or curly) half-arrow with TikZ. The switch/case statement makes sure that the half-arrow stroke always points down (credits to Mark Wibrow at http://tex.stackexchange.com/questions/208313/tikz-pgf-half-arrow-stroke-always-down-left). This orientation can be changed to make the stroke always go left or down, never right, by changing the `-1` in `\pgfdecoratedangle-1` to a `+45`. TODO: make this a package option. The argument to this TikZ stile is passed to the draw command, to allow different colour/line width/etc.

```
\tikzset{bond/.style args={#1}{
  decoration={show path construction,
    lineto code={
    \draw[#1,thick]
      \pgfextra{%       +45 i.s.o. -1 for to left or to under, always
        \pgfmathparse{int((\pgfdecoratedangle-1)/90)}
        \ifcase\pgfmathresult
          \ifbondgraphs@curly
            \tikzset{-{Classical TikZ Rightarrow[length=3 2 0.8,right]}}
          \else
            \tikzset{-{Straight Barb[length=3 2 0.8,right]}}
          \fi
        \or
          \ifbondgraphs@curly
            \tikzset{-{Classical TikZ Rightarrow[length=3 2 0.8,left]}}
          \else
            \tikzset{-{Straight Barb[length=3 2 0.8,left]}}
          \fi
        \or
          \ifbondgraphs@curly
            \tikzset{-{Classical TikZ Rightarrow[length=3 2 0.8,left]}}
          \else
            \tikzset{-{Straight Barb[length=3 2 0.8,left]}}
          \fi
        \else
          \ifbondgraphs@curly
            \tikzset{-{Classical TikZ Rightarrow[length=3 2 0.8,right]}}
          \else
            \tikzset{-{Straight Barb[length=3 2 0.8,right]}}
          \fi
        \fi
```

```
      }  (\tikzinputsegmentfirst) -- (\tikzinputsegmentlast);
    }
  },
  postaction=decorate
},
bond/.default={}
}
```

`mbond` A multi-bond is simply a bond with a double draw

```
\tikzset{mbond/.style={bond={double,#1}}}
```

`draw[bonds]` Convenience function (style) to draw lots of bonds, using `\draw[bonds]`

```
\tikzset{bonds/.style={every edge/.append style={bond}}}
```

`bond labels` Provide labelling for effort (above) and flow (below) the bond. Labels are sloped, so "above" is always defined from bond point-of-view

```
\tikzset{
  bondlabel/.style={
    font=\small,
    color=\bondgraphs@bondlabelcolour,
    sloped,
  },
  effort/.style args={#1}{
    edge node={node [bondlabel,above]{#1}}
  },
  effort/.default={},
  flow/.style args={#1}{
    edge node={node [bondlabel,below]{#1}}
  },
  flow/.default={}
}
```

`causality` The causal stroke of bonds is drawn as a thick line at the end, or start, of the bond. `e_in` / `f_out` and `f_in` / `e_out` are aliases. The optional argument gets passed on to the draw command of the stroke, e.g. to set its colour.

```
\tikzset{
  e_out/.style args={#1}{
    -{|[line width=1.2pt,width=7pt,#1]}
  },
  e_out/.default={black},
  f_in/.style args={#1}{
    e_out={#1}
  },
  e_in/.style args={#1}{
    {|[line width=1.2pt,width=7pt,#1]}-
  },
  e_in/.default={black},
  f_out/.style args={#1}{
    e_in={#1}
```

```
  }
}
```

`bond` A simple command to place a bond in text. Simply draw a bond from `(0,0)` to `(1,0)`, passing on any options to the edge.

```
\newcommand\bond[1][]{%
  \tikz \draw(0,0) edge[bond,#1] (1,0);
}
```

4.4 Elements

`elements` Bondgraph elements in TikZ are typeset in bold. It would be nicer to pass the node text through the `\bgelement` macro that is used for typesetting bondgraph elements in text (see below), but that's not easy. Possibly work around this by using `node contents` instead, as per http://tex.stackexchange.com/questions/209175/filter-tikz-node-text-through-macro.

```
\tikzset{
  bgelement/.style={
    font=\bfseries,
    prefix after command= {\pgfextra{
      \tikzset{every label/.style={
        % Element label style
        \bondgraphs@labelcolour,
        font={\mdseries}
      }
    }}}
  }
}
```

Multiport elements are typeset with the mathbb font. The `\bgroup` and `\egroup` actually do nothing and this construction only works by coincidence as explained by egreg in the aforementioned tex.stackexchange topic. TODO: fix this.

```
\tikzset{
  multiport/.style={
    execute at begin node=$\mathbb\bgroup,
    execute at end node=\egroup$
  }
}
```

Word-bg elements: ellipse with text inside

```
\tikzset{
  wordbgelement/.style={
    draw,ellipse,minimum size=12pt,thick,font=\mdseries
  }
}
```

`bgelement` Bondgraph elements in text are typeset much more robustly. `\pgfkeys` is used to parse all options. Currently: `multiport` to typeset in blackboard bold font; `n=x` to add a subscript `x` to indicate n-dimensional elements.

```
% Must create the if for multiport and wordbg outside of pgfkeys
\newif\ifbondgraphs@bgelement@multiport
\newif\ifbondgraphs@bgelement@wordbg
\pgfkeys{
  /bgelement/.is family, %automatically cd to /bgelement when present
  /bgelement/n/.store in=\bondgraphs@bgelement@n,
  /bgelement/n=1,
  %
  /bgelement/multiport/.is if=bondgraphs@bgelement@multiport,
  /bgelement/multiport=false,
  /bgelement/wordbg/.is if=bondgraphs@bgelement@wordbg,
  /bgelement/wordbg=false,
}
\newcommand\bgelement[2][]{%
  \pgfkeys{/bgelement,multiport=false,wordbg=false,#1}%
  \ifbondgraphs@bgelement@wordbg
```

Word-bg elements have an ellipse around them and hence are drawn with inline TikZ.

```
    \tikz[baseline=(the_elm.base)] \node[wordbgelement] (the_elm) {#2};%
  \else
    \ifbondgraphs@bgelement@multiport
      \ifnum\bondgraphs@bgelement@n=1
        \ensuremath{\mathbb{#2}}%
      \else
        \ensuremath{\mathbb{#2}_{\bondgraphs@bgelement@n}}%
      \fi
    \else
      \ifnum\bondgraphs@bgelement@n=1
        \textrm{\textbf{#2}}%
      \else
        \textrm{\textbf{#2}$_{\bondgraphs@bgelement@n}$}%
      \fi
    \fi
  \fi
}
```

4.5 Environments

bondgraph

When drawing pure bondgraphs, it is convenient to use the `bondgraph` environment, which is an alias for tikzpicture but with the proper styles set for nodes and edges.

```
\newenvironment{bondgraph}[1][]{%
\begin{tikzpicture}%
  [#1,every node/.style={bgelement},every edge/.append style={bond}]%
}{%
\end{tikzpicture}
}
```

⟨/package⟩

5 Change History

6 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.