

The **sankey** package

Draw Sankey diagrams via TikZ

Paul Gaborit (paul.gaborit@gmail.com)

v3.0 – 2021/03/14

Abstract

The **sankey** package provides macros and environments to build *Sankey diagrams*¹, i.e. *flow diagrams* in which the width of the arrows is proportional to the flow rate. The initial idea for the first implementation came out from [this question](#) on [TeX.StackExchange](#).

This manual contains three parts: [User manual](#) (p.1), [Examples](#) (p.22) and [Implementation](#) (p.43).

Note: the **sankey.dtx** and **sankey.ins** files are attachments of the current PDF document.

Part I

User manual

Contents

1	Preamble	2
2	The sankeydiagram environment	2
3	Sankey diagram options	2
3.1	Keys to choose the scale	3
3.2	Keys to define rotate offset	3
3.3	Keys to define drawing parameters	3
3.4	Keys to choose drawing styles	4
3.5	Keys to define new <i>start</i> and <i>end</i> styles	4
3.6	The <i>debug</i> key	4

¹https://en.wikipedia.org/wiki/Sankey_diagram

4	Sankey nodes and flows	5
4.1	Create Sankey nodes	5
4.1.1	Choose default parameters	6
4.1.2	Create starting and ending nodes via macros	6
4.1.3	Create starting and ending nodes via options	7
4.2	Retrieve information from Sankey nodes	8
4.3	Move nodes	8
4.3.1	Macro to move straight (forward or backward)	9
4.3.2	Macro to turn forward or backward	10
4.3.3	Macros to turn left (forward or backward)	11
4.3.4	Macros to turn right (forward or backward)	13
4.4	Links between nodes	15
4.4.1	Comparison between <code>outin</code> and <code>dubins</code> paths	16
4.4.2	Examples of <code>dubins</code> paths	17
4.5	Pure filling/drawing macros	18
4.6	Forked node	19
4.6.1	Create and fork a Sankey node	19
4.6.2	Fork a Sankey node	20
5	Miscellaneous	20
5.1	The <code>debug</code> layer	20
5.2	The <code>dubins</code> TikZ library	20
5.3	How to duplicate a Sankey node	21
5.4	How to define new start and end styles	21
6	Todo	21

1 Preamble

To use the `sankey` package, insert the following line in your preamble:

```
\usepackage{sankey}
```

Note: the `sankey` package requires automatically the `xparse`, `etoolbox`, `xfp` and `tikz` packages, and the `calc`, `decorations.markings` and `dubins` (cf. 5.2 on page 20) TikZ libraries.

2 The `sankeydiagram` environment

`sankeydiagram` A `sankeydiagram` environment nested in a `tikzpicture` environment activates the `sankey` macros:

```
\begin{tikzpicture}
  \begin{sankeydiagram}[... options ...]
    ... sankey macros ...
  \end{sankeydiagram}
\end{tikzpicture}
```

3 Sankey diagram options

The `sankey` package uses `pgfkeys` to set options via `key=value` pairs with default path `/sankey` (and `/sankey/node` parameters for Sankey node parameters).

The options can be defined via the optional argument of the `sankeydiagram` environment:

```
\begin{sankeydiagram}[debug=true]
  \sankeynode{name=a,quantity=1,angle=0,at={0,0}}
\end{sankeydiagram}
```

`\sankeyset` The options can also be modified via the `\sankeyset` macro:

```
\begin{sankeydiagram}
  \sankeyset{debug=true}
  \sankeynode{name=a,quantity=1,angle=0,at={0,0}}
\end{sankeydiagram}
```

The options can be temporarily modified for a single macro:

```
\begin{sankeydiagram}
  \sankeynode[debug=true]{name=a,quantity=1,angle=0,at={0,0}}
\end{sankeydiagram}
```

At the beginning of each Sankey diagram, all options are initialized with initial values then the **every diagram** style is applied.

```
/sankey/every diagram (initially: empty)
```

This style is installed at beginning of each Sankey diagram.

For instance, to use a ratio of 5mm/10 by default (instead of 1cm/10) for all Sankey diagrams, add the following line:

```
\sankeyset{every diagram/.style={ratio=5mm/10}}
```

3.1 Keys to choose the scale

The scale or ratio of the Sankey diagram is the ratio between the **ratio length** and the **ratio quantity**.

```
/sankey/ratio quantity=<number> (initially: 10)
```

Quantity (in units of flow) to define ratio. The $\langle number \rangle$ can be any math expression.

```
/sankey/ratio length=<distance> (initially: 1cm)
```

Distance (a graphical distance) to define scale.

```
/sankey/ratio=<distance>/<number> (initially: 1cm/10)
```

Fix the ratio to $\langle distance \rangle / \langle number \rangle$.

The initial ratio is 1 cm/10 units.

Note: the **sankey** package uses the **xfp** package to evaluate all math expressions that use quantities (in units of flow). You can therefore use quantities of a very large or very small order of magnitude. In contrast, for graphic distances, the **sankey** package uses the **pgfmath** package (all calculations must not exceed ± 16383.99999).

3.2 Keys to define rotate offset

```
/sankey/rotate=<angle> (initially: 0)
```

The **rotate** key stores an offset angle applied to all Sankey nodes. This is useful when using the **rotate** option within a **tikzpicture** or a **scope**. This TikZ option is only applied to coordinates and not to TikZ nodes (remember that Sankey nodes are TikZ nodes). It's up to the author to keep the **rotate** option of the **tikzpicture** and that of the **sankeydiagram** synchronous.

3.3 Keys to define drawing parameters

```
/sankey/minimum radius=<distance> (initially: 5mm)
```

The minimum radius used by **\sankeyturn** and **\sankeydubins**.

```
/sankey/outin steps=<integer> (initially: 10)
```

Number of steps used by the **\sankeyoutin** macro to simulate flow lanes with constant width.

3.4 Keys to choose drawing styles

`/sankey/fill/.style=<style>` (initially: `line width=0pt,fill=white`)

This TikZ style is used to *fill* all sankey paths.

`/sankey/draw/.style=<style>` (initially: `draw=black,line width=.4pt`)

This TikZ style is used to *draw* all sankey paths.

`/sankey/start style=<style name>` (initially: `none`)

There are three predefined *start* styles: `none`, `simple`, `arrow`.

`/sankey/end style=<style name>` (initially: `none`)

There are three predefined *end* styles: `none`, `simple`, `arrow`.

3.5 Keys to define new *start* and *end* styles

`/sankey/new start style={<name>}{<fill path>}{<draw path>}`

Define the new start style named `<name>` with its `<fill path>` and its `<draw path>`.

`/sankey/new end style={<name>}{<fill path>}{<draw path>}`

Define the new end style named `<name>` with its `<fill path>` and its `<draw path>`.

Fill and *draw* paths are build in a TikZ scope where the origin is the center of the current Sankey node (its name is accessible via `\name`) and the coordinate system is rotated by its orientation.

3.6 The *debug* key

`/sankey/debug=<boolean>` (default: `true`) (initially: `false`)

To debug a sankey diagram.

4 Sankey nodes and flows

4.1 Create Sankey nodes

`\sankeynode[<options>]{<node parameters>}`

`\sankeynode` The `\sankeynode` macro defines a Sankey node. The *<options>* can be any Sankey diagram keys. To define a Sankey node, you must provide a *name*, a *quantity* and an *angle* as *<node parameters>*.

`/sankey/node parameters/name=<name>`

The *<name>* of the new Sankey node (and the associated TikZ node).

`/sankey/node parameters/quantity=<quantity>`

The quantity (in flow unit) of the new Sankey node. The *<quantity>* can be any math expression.

`/sankey/node parameters/angle=<angle>`

The orientation of the flow (0 points to the right) of the new Sankey node.

`/sankey/node parameters/at=<at>` (initially: 0,0)

The position of the new Sankey node (a TikZ coordinate *without* round brackets or parentheses).

`/sankey/node parameters/anchor=<anchor>` (initially: center)

Specify the anchor of the Sankey node. Possible values are *center*, *left* or *right*.

`/sankey/node parameters/as=<name>`

Copy the *name*, the *quantity*, the *angle* and the *position* of the Sankey node named *<name>*.

A Sankey node is also a Tikz node but with only three anchors: *left*, *center* and *right*²:



```
\begin{tikzpicture}
\begin{sankeydiagram}[debug]
\sankeynode{name=a,quantity=10}
\sankeynode{name=b,quantity=5,angle=45,at={2,1},anchor=left}
\sankeynode{name=c,quantity=20,angle=-90,at={5,1},anchor=right}

\foreach \nodename/\pos in {a/left,b/below left,c/above}{
\foreach \ancname in {left,center,right}{
\node[node font=\ttfamily\footnotesize,\pos=1mm of \nodename.\ancname,
inner sep=0pt,rotate=\sankeygetnodeorient{\nodename},anchor=east]
{\ancname\vphantom{g}};
\fill[black] (\nodename.\ancname) circle(1pt);
}
}
\end{sankeydiagram}
\end{tikzpicture}
```

²In fact, to be able to use the TikZ *fit* library, the *north*, *north east* and *north west* anchors exist and are equal to *left*, the *east* and *west* anchors exist and are equal to *center* and the *south*, *south east* and *south west* anchors exist and are equal to *right*.

4.1.1 Choose default parameters

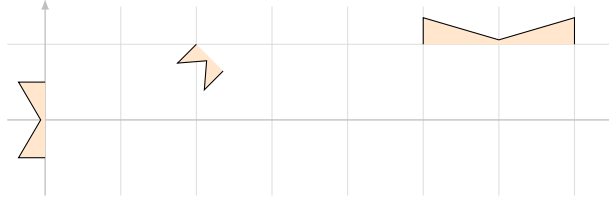
`/sankey/every node/.style={\langle node parameters \rangle}` (initially: `empty`)

The `\langle node parameters \rangle` defined by the `every node` style is installed at the creation of every Sankey node.

4.1.2 Create starting and ending nodes via macros

`\sankeynodestart[\langle options \rangle]{\langle node parameters \rangle}`

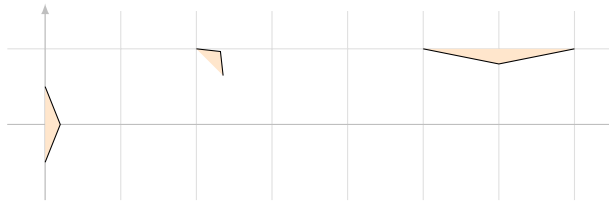
`\sankeynodestart` The `\sankeynodestart` creates and fills/draws a starting Sankey node:



```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,fill/.style={fill=orange!20}]
    \sankeynodestart{name=a,quantity=10}
    \sankeynodestart{name=b,quantity=5,angle=45,at={2,1},anchor=left}
    \sankeynodestart{name=c,quantity=20,angle=-90,at={5,1},anchor=right}
  \end{sankeydiagram}
\end{tikzpicture}
```

`\sankeynodeend[\langle options \rangle]{\langle node parameters \rangle}`

`\sankeynodeend` The `\sankeynodeend` creates and fills/draws an ending Sankey node:

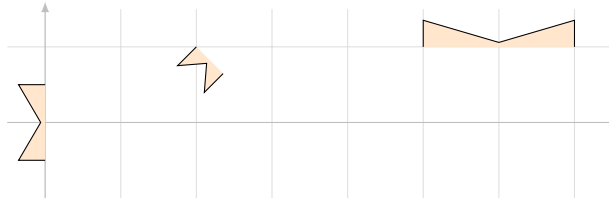


```
\begin{tikzpicture}
  \begin{sankeydiagram}[end style=simple,fill/.style={fill=orange!20}]
    \sankeynodeend{name=a,quantity=10}
    \sankeynodeend{name=b,quantity=5,angle=45,at={2,1},anchor=left}
    \sankeynodeend{name=c,quantity=20,angle=-90,at={5,1},anchor=right}
  \end{sankeydiagram}
\end{tikzpicture}
```

4.1.3 Create starting and ending nodes via options

/sankey/node parameters/**start**=*<boolean>* (default: **true**) (initially: **false**)

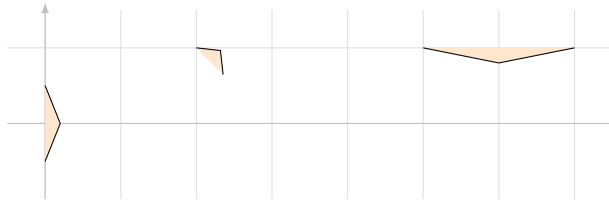
The `\sankeynode` macro acts as the `\sankeynodestart` macro if you add the **start** option to its options.



```
\begin{tikzpicture}
\begin{sankeydiagram}[start style=arrow,fill/.style={fill=orange!20}]
\sankeynode{name=a,quantity=10,start}
\sankeynode{name=b,quantity=5,angle=45,at={2,1},anchor=left,start}
\sankeynode{name=c,quantity=20,angle=-90,at={5,1},anchor=right,start}
\end{sankeydiagram}
\end{tikzpicture}
```

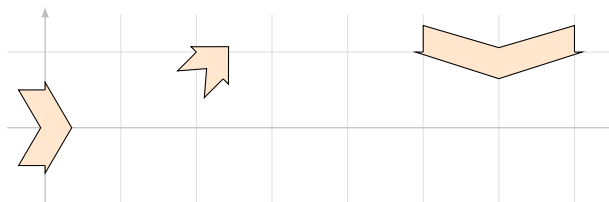
/sankey/node parameters/**end**=*<boolean>* (default: **true**) (initially: **false**)

The `\sankeynode` macro acts as the `\sankeynodeend` macro if you add the **end** option to its options.



```
\begin{tikzpicture}
\begin{sankeydiagram}[end style=simple,fill/.style={fill=orange!20}]
\sankeynode{name=a,quantity=10,end}
\sankeynode{name=b,quantity=5,angle=45,at={2,1},anchor=left,end}
\sankeynode{name=c,quantity=20,angle=-90,at={5,1},anchor=right,end}
\end{sankeydiagram}
\end{tikzpicture}
```

Although rarely necessary, you can mix these two parameters:



```
\begin{tikzpicture}
\begin{sankeydiagram}
\sankeyset{
end style=arrow,
start style=arrow,
fill/.style={fill=orange!20}
}
\sankeynode{name=a,quantity=10,start,end}
\sankeynode{name=b,quantity=5,angle=45,at={2,1},anchor=left,start,end}
\sankeynode{name=c,quantity=20,angle=-90,at={5,1},anchor=right,start,end}
\end{sankeydiagram}
\end{tikzpicture}
```

4.2 Retrieve information from Sankey nodes

`\sankeygetnodeqty{<node name>}`

`\sankeygetnodeqty` The expandable command `\sankeygetnodeqty` returns the quantity assigned to the Sankey node named `<node name>`.

`\sankeyqtytolen{<quantity>}`

`\sankeyqtytolen` The expandable `\sankeyqtytolen` macro converts `<quantity>` to graphical length using the current ratio.

`\sankeygetnodeorient{<node name>}`

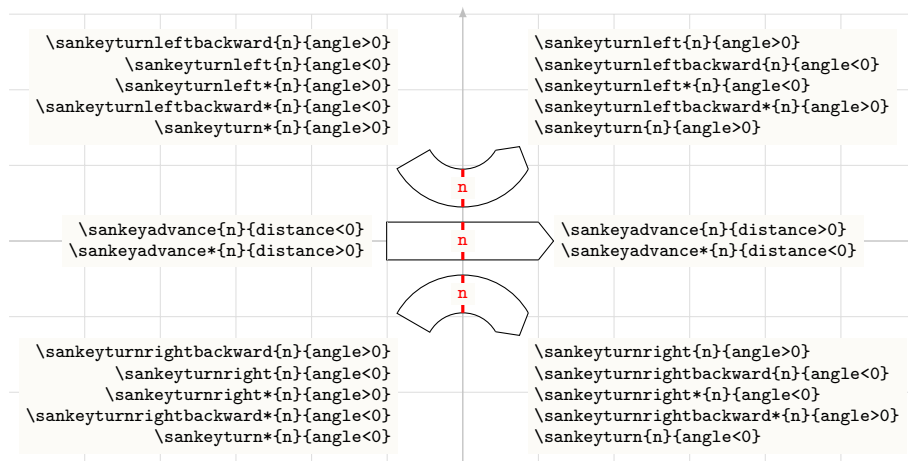
`\sankeygetnodeorient` The expandable command `\sankeygetnodeorient` returns the angle (orientation) assigned to the Sankey node named `<node name>`.

4.3 Move nodes

All the macros of this section move a Sankey node and fill/draw a portion of the Sankey flow. Then the previous position of the Sankey node is accessible via the `-old` suffix (i.e. if you move the `a` node, its previous position is the `a-old` node).

The starred version of each of these macros moves in the opposite direction to their non-starred version.

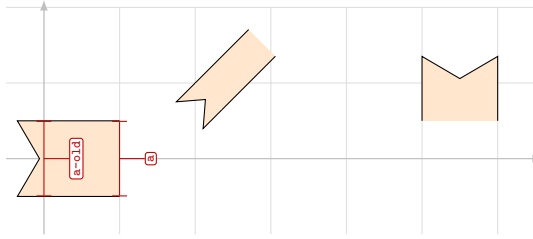
Except for the `\sankeyturn` macro, a negative value (distance or angle) moves in the opposite direction (the `\sankeyturn` macro is an exception: a negative angle turns right while a positive value turns left).



4.3.1 Macro to move straight (forward or backward)

\sankeyadvance[*<options>*]{*<node name>*}{*<distance>*}

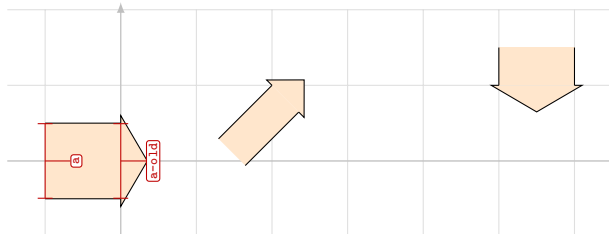
\sankeyadvance The **\sankeyadvance** moves the sankey node straight ahead and fills/draws this portion of the sankey path. A positive *<distance>* moves forward while a negative *<distance>* moves backward.



```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,fill/.style={fill=orange!20}]
    \sankeynodestart{name=a,quantity=10}
    \sankeyadvance{a}{1cm}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeynodestart{name=b,quantity=5,angle=45,at={2,1},anchor=left}
    \sankeyadvance{b}{1cm}
    \sankeynodestart{name=c,quantity=10,angle=-90,at={5,1},anchor=right}
    \sankeyadvance{c}{5mm}
  \end{sankeydiagram}
\end{tikzpicture}
```

\sankeyadvance*[*<options>*]{*<node name>*}{*<distance>*}

\sankeyadvance* The **\sankeyadvance*** moves the sankey node straight back and fills/draws this portion of the sankey path. A positive *<distance>* moves backward while a negative *<distance>* moves forward.

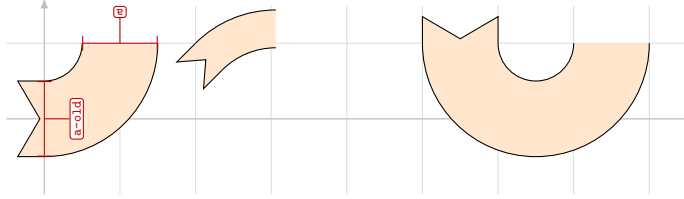


```
\begin{tikzpicture}
  \begin{sankeydiagram}[end style=arrow,fill/.style={fill=orange!20}]
    \sankeynodeend{name=a,quantity=10}
    \sankeyadvance*{a}{1cm}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeynodeend{name=b,quantity=5,angle=45,at={2,1},anchor=left}
    \sankeyadvance*{b}{1cm}
    \sankeynodeend{name=c,quantity=10,angle=-90,at={5,1},anchor=right}
    \sankeyadvance*{c}{5mm}
  \end{sankeydiagram}
\end{tikzpicture}
```

4.3.2 Macro to turn forward or backward

`\sankeyturn[<options>]{<node name>}{<angle>}`

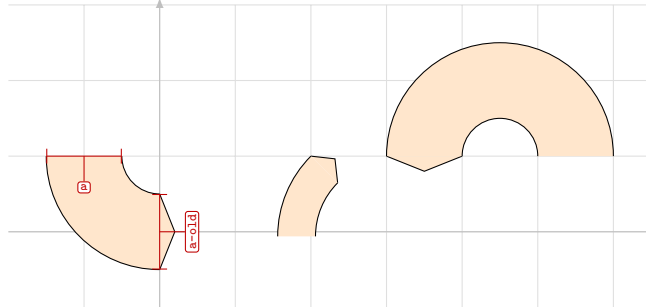
`\sankeyturn` The `\sankeyturn` macro moves the sankey node by turning to one side or the other and fills/draws this portion of the sankey path. A *positive* *<angle>* turns left while a *negative* *<angle>* turns right.



```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,fill/.style={fill=orange!20}]
    \sankeynodestart{name=a,quantity=10}
    \sankeyturn{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeynodestart{name=b,quantity=5,angle=45,at={2,1},anchor=left}
    \sankeyturn[minimum radius=1cm]{b}{-45}
    \sankeynodestart{name=c,quantity=10,angle=-90,at={5,1},anchor=right}
    \sankeyturn{c}{180}
  \end{sankeydiagram}
\end{tikzpicture}
```

`\sankeyturn*[<options>]{<node name>}{<angle>}`

`\sankeyturn*` The `\sankeyturn*` macro moves the sankey node backward by turning right or left and fills/draws this portion of the sankey path. A *positive* *<angle>* turns left while a *negative* *<angle>* turns right.

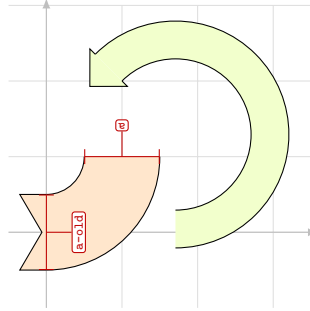


```
\begin{tikzpicture}
  \begin{sankeydiagram}[end style=simple,fill/.style={fill=orange!20}]
    \sankeynodeend{name=a,quantity=10}
    \sankeyturn*{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeynodeend{name=b,quantity=5,angle=45,at={2,1},anchor=left}
    \sankeyturn*[minimum radius=1cm]{b}{-45}
    \sankeynodeend{name=c,quantity=10,angle=-90,at={3,1},anchor=right}
    \sankeyturn*{c}{180}
  \end{sankeydiagram}
\end{tikzpicture}
```

4.3.3 Macros to turn left (forward or backward)

\sankeyturnleft[*(options)*]{*(node name)*}{*(angle)*}

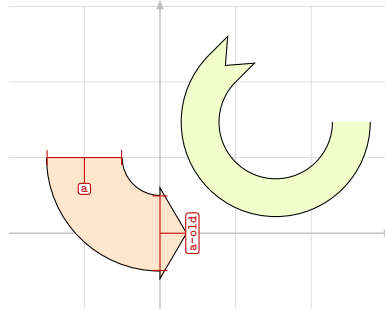
\sankeyturnleft The **\sankeyturnleft** macro moves the sankey node by turning left and fills/draws this portion of the sankey path. A *positive* *(angle)* turns forward while a *negative* *(angle)* turns backward.



```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,end style=arrow]
    \sankeyset{fill/.style={fill=orange!20}}
    \sankeynodestart{name=a,quantity=10}
    \sankeyturnleft{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeyset{fill/.style={fill=lime!20}}
    \sankeynodeend{name=b,quantity=5,angle=-135,at={1,2},anchor=left}
    \sankeyturnleft[minimum radius=1cm]{b}{-225}
  \end{sankeydiagram}
\end{tikzpicture}
```

\sankeyturnleft*[*(options)*]{*(node name)*}{*(angle)*}

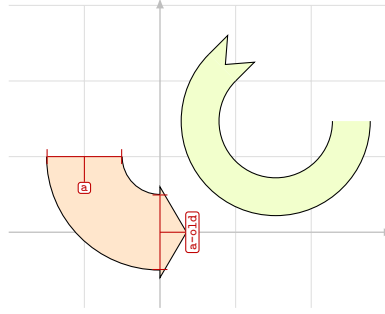
\sankeyturnleft* The **\sankeyturnleft*** macro moves the sankey node backward by turning left and fills/draws this portion of the sankey path. A *positive* *(angle)* turns backward while a *negative* *(angle)* turns forward.



```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,end style=arrow]
    \sankeyset{fill/.style={fill=orange!20}}
    \sankeynodeend{name=a,quantity=10}
    \sankeyturnleft*{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeyset{fill/.style={fill=lime!20}}
    \sankeynodestart{name=b,quantity=5,angle=-135,at={1,2},anchor=left}
    \sankeyturnleft*[minimum radius=.75cm]{b}{-225}
  \end{sankeydiagram}
\end{tikzpicture}
```

`\sankeyturnleftbackward[<options>]{<node name>}{<angle>}`

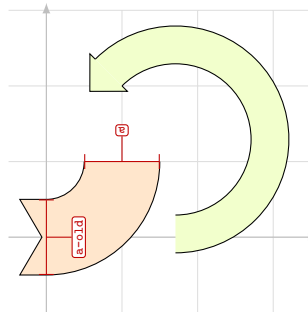
`\sankeyturnleftbackward` The `\sankeyturnleftbackward` macro moves the sankey node backward by turning left and fills/draws this portion of the sankey path. A *positive* *<angle>* turns backward while a *negative* *<angle>* turns forward.



```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,end style=arrow]
    \sankeyset{fill/.style={fill=orange!20}}
    \sankeynodeend{name=a,quantity=10}
    \sankeyturnleftbackward{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeyset{fill/.style={fill=lime!20}}
    \sankeynodestart{name=b,quantity=5,angle=-135,at={1,2},anchor=left}
    \sankeyturnleftbackward[minimum radius=.75cm]{b}{-225}
  \end{sankeydiagram}
\end{tikzpicture}
```

`\sankeyturnleftbackward*[<options>]{<node name>}{<angle>}`

`\sankeyturnleftbackward*` The `\sankeyturnleftbackward*` macro moves the sankey node by turning left and fills/draws this portion of the sankey path. A *positive* *<angle>* turns forward while a *negative* *<angle>* turns backward.

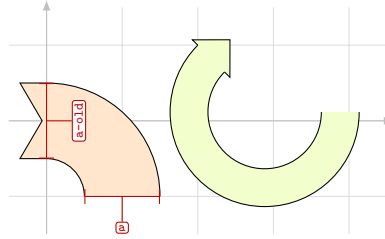


```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,end style=arrow]
    \sankeyset{fill/.style={fill=orange!20}}
    \sankeynodestart{name=a,quantity=10}
    \sankeyturnleftbackward*{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeyset{fill/.style={fill=lime!20}}
    \sankeynodeend{name=b,quantity=5,angle=-135,at={1,2},anchor=left}
    \sankeyturnleftbackward*[minimum radius=1cm]{b}{-225}
  \end{sankeydiagram}
\end{tikzpicture}
```

4.3.4 Macros to turn right (forward or backward)

\sankeyturnright [*options*] {*node name*} {*angle*}

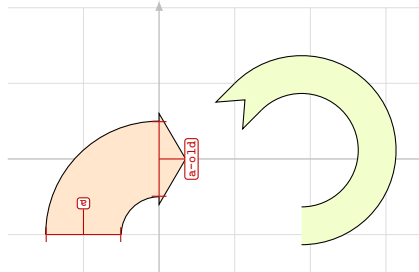
\sankeyturnright The **\sankeyturnright** macro moves the sankey node by turning right and fills/draws this portion of the sankey path. A *positive* *angle* turns forward while a *negative* *angle* turns backward.



```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,end style=arrow]
    \sankeyset{fill/.style={fill=orange!20}}
    \sankeynodestart{name=a,quantity=10}
    \sankeyturnright[fill/.style={fill=orange!20}]{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeyset{fill/.style={fill=lime!20}}
    \sankeynodeend{name=b,quantity=5,angle=45,at={2,1},anchor=left}
    \sankeyturnright[minimum radius=.75cm]{b}{-225}
  \end{sankeydiagram}
\end{tikzpicture}
```

\sankeyturnright* [*options*] {*node name*} {*angle*}

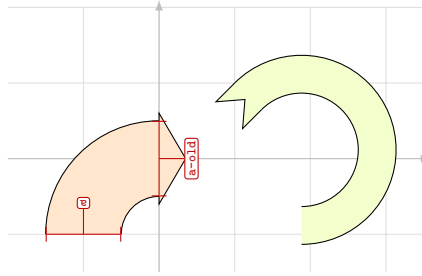
\sankeyturnright* The **\sankeyturnright*** macro moves the sankey node backward by turning right and fills/draws this portion of the sankey path. A *positive* *angle* turns backward while a *negative* *angle* turns forward.



```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,end style=arrow]
    \sankeyset{fill/.style={fill=orange!20}}
    \sankeynodeend{name=a,quantity=10}
    \sankeyturnright*[fill/.style={fill=orange!20}]{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeyset{fill/.style={fill=lime!20}}
    \sankeynodestart{name=b,quantity=5,angle=45,at={1,1},anchor=left}
    \sankeyturnright*[minimum radius=.75cm]{b}{-225}
  \end{sankeydiagram}
\end{tikzpicture}
```

\sankeyturnrightbackward[*(options)*]{*(node name)*}{*(angle)*}

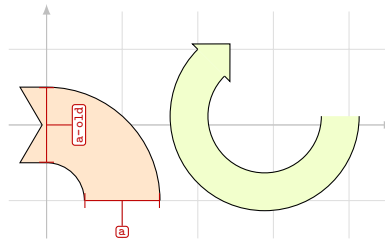
\sankeyturnrightbackward The **\sankeyturnrightbackward** macro moves the sankey node backward by turning right and fills/draws this portion of the sankey path. A *positive* *(angle)* turns backward while a *negative* *(angle)* turns forward.



```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,end style=arrow]
    \sankeyset{fill/.style={fill=orange!20}}
    \sankeynodeend{name=a,quantity=10}
    \sankeyturnrightbackward[fill/.style={fill=orange!20}]{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeyset{fill/.style={fill=lime!20}}
    \sankeynodestart{name=b,quantity=5,angle=45,at={1,1},anchor=left}
    \sankeyturnrightbackward[minimum radius=.75cm]{b}{-225}
  \end{sankeydiagram}
\end{tikzpicture}
```

\sankeyturnrightbackward*[*(options)*]{*(node name)*}{*(angle)*}

\sankeyturnrightbackward* The **\sankeyturnrightbackward*** macro moves the sankey node forward by turning right and fills/draws this portion of the sankey path. A *positive* *(angle)* turns forward while a *negative* *(angle)* turns backward.



```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,end style=arrow]
    \sankeyset{fill/.style={fill=orange!20}}
    \sankeynodestart{name=a,quantity=10}
    \sankeyturnrightbackward*[fill/.style={fill=orange!20}]{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeyset{fill/.style={fill=lime!20}}
    \sankeynodeend{name=b,quantity=5,angle=45,at={2,1},anchor=left}
    \sankeyturnrightbackward*[minimum radius=.75cm]{b}{-225}
  \end{sankeydiagram}
\end{tikzpicture}
```

4.4 Links between nodes

The macros described in this section fill/draw a lane between two Sankey nodes.

Note: since Sankey nodes are oriented, linking **A** node to **B** node does not produce the same result as linking **B** node to **A** node!

`\sankeyoutin[<options>]{<node A>}{<node B>}`

`\sankeyoutin` The `\sankeyoutin` macro fills/draws a lane from *<node A>* to *<node B>* using a Bézier curve with regular steps (10 steps by default) to simulate constant width lane.

The constant width and the minimum curvature are *not* guaranteed!

`\sankeydubins[<options>]{<node A>}{<node B>}`

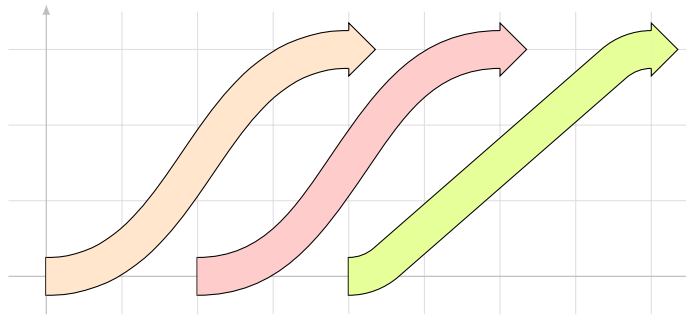
`\sankeydubins` The `\sankeydubins` macro fills/draws a lane between *<node A>* and *<node B>* using a Dubins path³.

The constant width and the minimum curvature are guaranteed.

³https://en.wikipedia.org/wiki/Dubins_path

4.4.1 Comparison between outin and dubins paths

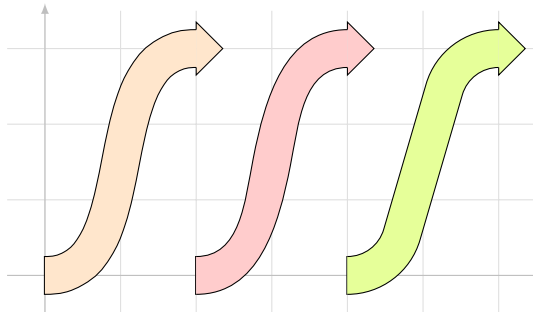
The following diagrams compare outin path with 10 steps (orange), outin path with 2 steps (red) and dubins path (lime) in various positions.



```
\begin{tikzpicture}
\begin{sankeydiagram}[start style=simple,end style=arrow]
\sankeyset{fill/.style={fill=orange!20}}
\sankeynodestart{name=a,quantity=5}
\sankeynodeend{name=b,at={4,3},quantity=5}
\sankeyoutin{a}{b}

\sankeyset{fill/.style={fill=red!20}}
\sankeynodestart{name=a,at={2,0},quantity=5}
\sankeynodeend{name=b,at={6,3},quantity=5}
\sankeyoutin[outin steps=2]{a}{b}

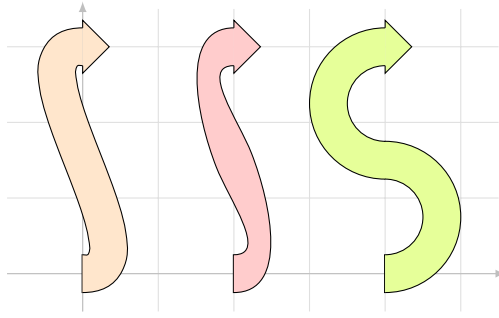
\sankeyset{fill/.style={fill=lime!40}}
\sankeynodestart{name=a,at={4,0},quantity=5}
\sankeynodeend{name=b,at={8,3},quantity=5}
\sankeydubins[minimum radius=5mm]{a}{b}
\end{sankeydiagram}
\end{tikzpicture}
```



```
\begin{tikzpicture}
\begin{sankeydiagram}[start style=simple,end style=arrow]
\sankeyset{fill/.style={fill=orange!20}}
\sankeynodestart{name=a,quantity=5}
\sankeynodeend{name=b,at={2,3},quantity=5}
\sankeyoutin{a}{b}

\sankeyset{fill/.style={fill=red!20}}
\sankeynodestart{name=a,at={2,0},quantity=5}
\sankeynodeend{name=b,at={4,3},quantity=5}
\sankeyoutin[outin steps=2]{a}{b}

\sankeyset{fill/.style={fill=lime!40}}
\sankeynodestart{name=a,at={4,0},quantity=5}
\sankeynodeend{name=b,at={6,3},quantity=5}
\sankeydubins[minimum radius=5mm]{a}{b}
\end{sankeydiagram}
\end{tikzpicture}
```

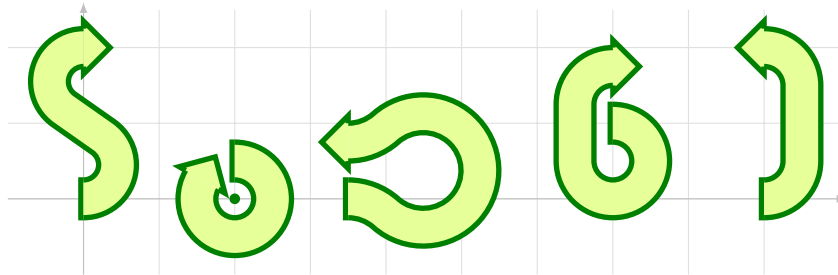



```
\begin{tikzpicture}
\begin{sankeydiagram}[start style=simple,end style=arrow]
\sankeyset{fill/.style={fill=orange!20}}
\sankeynodestart{name=a,quantity=5}
\sankeynodeend{name=b,at={0,3},quantity=5}
\sankeyoutin{a}{b}

\sankeyset{fill/.style={fill=red!20}}
\sankeynodestart{name=a,at={2,0},quantity=5}
\sankeynodeend{name=b,at={2,3},quantity=5}
\sankeyoutin[outin steps=2]{a}{b}

\sankeyset{fill/.style={fill=lime!40}}
\sankeynodestart{name=a,at={4,0},quantity=5}
\sankeynodeend{name=b,at={4,3},quantity=5}
\sankeydubins[minimum radius=5mm]{a}{b}
\end{sankeydiagram}
\end{tikzpicture}
```

4.4.2 Examples of dubins paths



```
\begin{tikzpicture}
\begin{sankeydiagram}[start style=simple,end style=arrow]
\sankeyset{
fill/.style={fill=lime!40},
draw/.style={draw=green!50!black,line width=2pt},
}

\sankeynodestart{name=a,quantity=5}
\sankeynodeend{name=b,at={0,2},quantity=5}
\sankeydubins[minimum radius=2mm]{a}{b}

\fill[green!50!black] (2,0) coordinate (c) circle(2pt);
\sankeynodestart{name=a,at={ [shift={(c)}] 90:5mm },quantity=5}
\sankeynodeend{name=b,at={ [shift={(c)}] 150:5mm },angle=60,quantity=5}
\sankeydubins[minimum radius=2.5mm]{a}{b}

\sankeynodestart{name=a,at={3.5,0},quantity=5}
\sankeynodeend{name=b,at={3.5,.75},angle=-180,quantity=5}
\sankeydubins[minimum radius=5mm]{a}{b}

\sankeynodestart{name=a,at={7,1},quantity=5}
\sankeynodeend{name=b,at={7,1.75},quantity=5}
\sankeydubins[minimum radius=2.5mm]{a}{b}

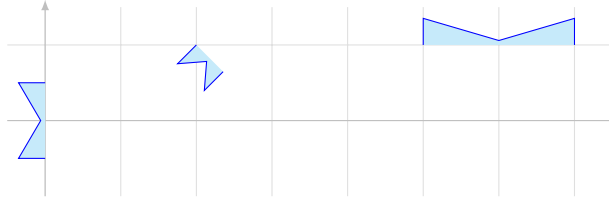
\sankeynodestart{name=a,at={9,0},quantity=5}
\sankeynodeend{name=b,at={9,2},angle=180,quantity=5}
\sankeydubins[minimum radius=2.5mm]{a}{b}
\end{sankeydiagram}
\end{tikzpicture}
```

```
\end{sankeydiagram}
\end{tikzpicture}
```

4.5 Pure filling/drawing macros

\sankeystart [*options*] {*name*}

\sankeystart The **\sankeystart** fills/draws a starting extremity attached to the preexisting Sankey node *name*:

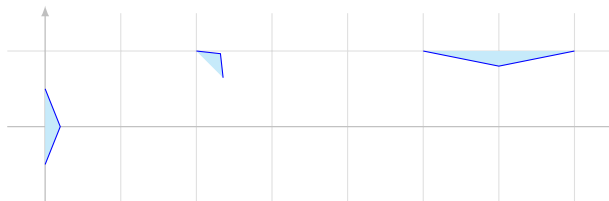


```
\begin{tikzpicture}
\begin{sankeydiagram}
[start style=arrow,fill/.style={fill=cyan!20},draw/.style={draw=blue}]
\sankeynode{name=a,quantity=10}
\sankeystart{a}
\sankeynode{name=b,quantity=5,angle=45,at={2,1},anchor=left}
\sankeystart{b}
\sankeynode{name=c,quantity=20,angle=-90,at={5,1},anchor=right}
\sankeystart{c}
\end{sankeydiagram}
\end{tikzpicture}
```

\sankeyend [*options*] {*name*}

\sankeyend The **\sankeyend** fills/draws an ending extremity attached to the preexisting Sankey node *name*:

```
\begin{tikzpicture}
\begin{sankeydiagram}
[end style=simple,fill/.style={fill=cyan!20},draw/.style={draw=blue}]
\sankeynode{name=a,quantity=10}
\sankeyend{a}
\sankeynode{name=b,quantity=5,angle=45,at={2,1},anchor=left}
\sankeyend{b}
\sankeynode{name=c,quantity=20,angle=-90,at={5,1},anchor=right}
\sankeyend{c}
\end{sankeydiagram}
\end{tikzpicture}
```



4.6 Forked node

4.6.1 Create and fork a Sankey node

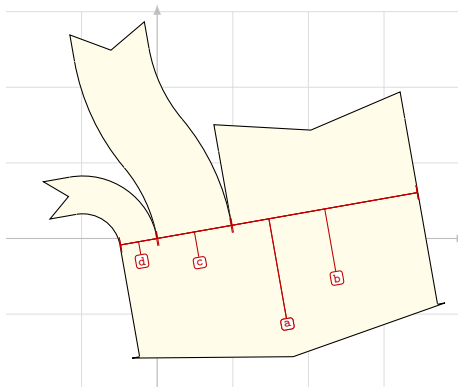
When creating a new Sankey node, the `forked` and `fork anchor` keys allow to fork the node directly *and* to anchor it on an anchor of a forked subnode.

```
/sankey/node parameters/forked={\quantity/name pairs}
```

The `\quantity/name pairs` is a comma separated list of *quantity/name* pairs (one for each subnode, from left to right). The sum of all quantities *must* be equal to the quantity of the new node to fork.

```
/sankey/node parameters/fork anchor=\node.anchor
```

An anchor belonging to the new node *or* belonging to a subnode (the anchor name must be prefixed by the name of the node). *Note:* when a `fork anchor` key is supplied, the `anchor` key is ignored (with a *warning* message).



```
\begin{tikzpicture}
  \begin{sankeydiagram}
    \sankeyset{
      start style=arrow,end style=arrow,
      fill/.style={fill=yellow!10,line width=0pt,draw=yellow!10}
    }

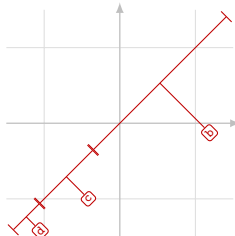
    \sankeynode[debug]{
      name=a,quantity=40,angle=-80,
      forked={25/b,10/c,5/d},
      fork anchor=c.right,
    }

    \sankeyadvance*b}{1cm}
    \sankeyturn*[minimum radius=2cm]{c}{-30}
    \sankeyturn*[minimum radius=2cm]{c}{30}
    \sankeyturn*[minimum radius=5mm]{d}{-90}
    \sankeyadvance*a}{1.5cm}
    \foreach \nodename in {b,c,d}{ \sankeystart{\nodename} }
    \sankeyend{a}
  \end{sankeydiagram}
\end{tikzpicture}
```

4.6.2 Fork a Sankey node

`\sankeyfork[<options>]{<name>}{<quantity/name pairs>}`

`\sankeyfork` The `\sankeyfork` macro splits the preexisting Sankey node named *<name>* in a list of new Sankey subnodes. The *<quantity/name pairs>* is a comma separated list of *quantity/name* pairs, one for each subnode from left to right. The sum of all quantities *must* be equal to the quantity of the node to fork.



```
\begin{tikzpicture}
  \begin{sankeydiagram}
    \sankeynode{name=a,quantity=40,angle=-45}
    \sankeyfork[debug]{a}{25/b,10/c,5/d}
    \path (a.left) rectangle (a.right); % create a bounding box
  \end{sankeydiagram}
\end{tikzpicture}
```

5 Miscellaneous

5.1 The debug layer

The options `debug` key uses the `sankeydebug` layer to draw above the `main` TikZ layer (via `\pgfsetlayers`, the `sankey` package installs four layers: `background`, `main`, `foreground`, `sankeydebug`).

The four following styles define how to display debug information:

```
\sankeyset{
  debug color/.style={/utils/exec={\colorlet{debug color}{#1}}},
  % debug color used by all debug macros
  debug color=red!75!black,
  % debug line between left and right anchors
  debug line/.style={overlay,draw=debug color,|-|},
  % debug line between center and label
  debug normal/.style={overlay,draw=debug color},
  % debug node label
  debug label/.style={
    overlay,
    draw,
    font=\ttfamily\tiny,
    text=debug color,text opacity=1,
    inner sep=.1em,
    fill=white,fill opacity=1,
    rounded corners=.1em,
    node contents={\name},
  },
}
```

5.2 The dubins TikZ library

The `sankey` package uses the `dubins` TikZ library (the `tikzlibrarydubins.code.tex` file) to compute Dubins paths. The documentation for this library does not yet exist.

5.3 How to duplicate a Sankey node

`\sankeynodealias{⟨origname⟩}{⟨clonename⟩}`

`\sankeynodealias` The `\sankeynodealias` macro clones the Sankey node named `⟨origname⟩` into the Sankey node named `⟨clonename⟩`.

So, you can clone a Sankey node via two methods:

```
\sankeynode{name=a,quantity=10}  
\sankeynode{as=a,name=b}
```

```
\sankeynode{name=a,quantity=10}  
\sankeynodealias{a}{b}
```

5.4 How to define new start and end styles

Here are the definitions of the `arrow` styles:

```
\sankeyset{  
  %% arrow style  
  new start style={arrow}{  
    (\name.left) -- ++(-10pt,0)  
    -- ([xshift=-10pt/6]\name.center)  
    -- ([xshift=-10pt]\name.right)  
    -- (\name.right) -- cycle  
  }{  
    (\name.left) -- ++(-10pt,0)  
    -- ([xshift=-10pt/6]\name.center)  
    -- ([xshift=-10pt]\name.right)  
    -- (\name.right)  
  },  
  new end style={arrow}{  
    (\name.left) -- ([yshift=1mm]\name.left)  
    -- ([xshift=10pt]\name.center)  
    -- ([yshift=-1mm]\name.right) -- (\name.right) -- cycle  
  }{  
    (\name.left) -- ([yshift=1mm]\name.left)  
    -- ([xshift=10pt]\name.center)  
    -- ([yshift=-1mm]\name.right) -- (\name.right)  
  },  
}
```

6 Todo

- Document the `dubins` TikZ library.
- Add a tutorial.
- Add examples with cycle(s).

This manual contains three parts: [User manual](#) (p.1), [Examples](#) (p.22) and [Implementation](#) (p.43).

Part II

Examples

Contents

7	List of Figures	22
8	Simple example	23
9	Energy diagram	25
10	Example from question on TeX.se	28
11	Reproduction of an example from Google Charts documentation	32
11.1	Variation	34
12	Very nice example – Nadieh Bremer creation	38

7 List of Figures

1	Simple example	23
2	Energy diagram	25
3	Example from TeX.se question	28
4	Reproduction of an example from Google Charts documentation . . .	32
5	Reproduction of an example from Google Charts documentation – variation using the <code>rotate</code> key.	35
6	Very nice example – Nadieh Bremer’s creation	39

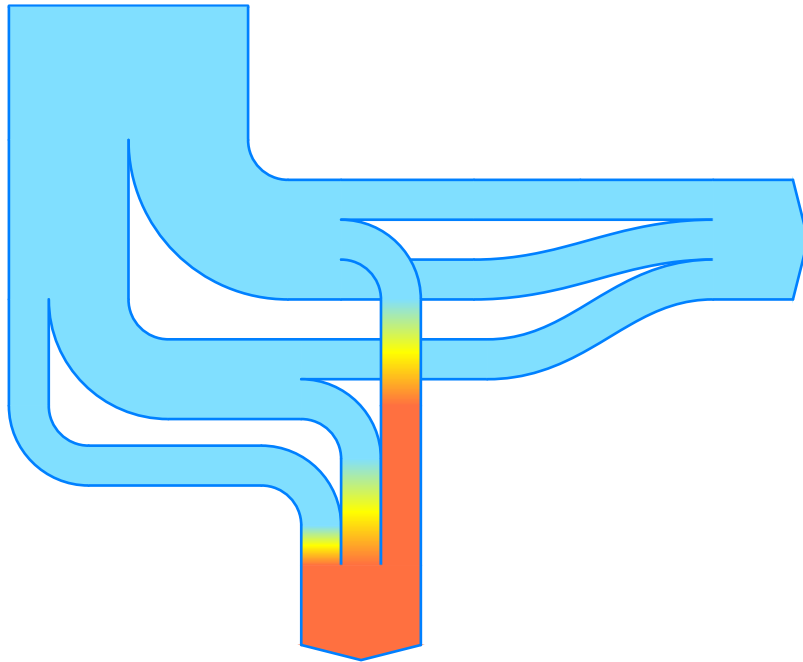


Figure 1: Simple example

8 Simple example

See figure 1. The `sankey-example1.tex` file contains the following code and is an attachment of the current PDF document.

```
\begin{tikzpicture}
\begin{sankeydiagram}%[debug]
\colorlet{cold}[rgb]{cyan!75!blue!50!white}
\colorlet{hot}[rgb]{red!50!orange!75!white}
\sankeyset{
ratio=90pt/6,minimum radius=15pt,
start style=simple,end style=simple,
draw/.style={
draw=blue!50!cyan,
line width=1pt,line cap=round,line join=round,
},
cold/.style={
fill/.style={
draw=cold,line width=0pt,fill=cold,
},
},
cold to hot/.style={
fill/.style={
fill=none,top color=cold,
bottom color=hot,middle color=yellow,
},
},
hot/.style={
fill/.style={
draw=hot,line width=0pt,fill=hot,
},
},
}

\sankeyset{cold}
\sankeynodestart{name=p0,at={100,0},angle=-90,quantity=6}
\sankeyadvance{p0}{50pt}
\sankeyfork{p0}{3/p1,3/p2}
\sankeyturnleft{p1}{90}
\sankeyadvance{p1}{20pt}
\sankeyadvance{p2}{60pt}
\sankeyfork{p2}{2/p3,1/p4}
\sankeyturnleft{p3}{90}
```

```

\sankeyadvance{p3}{50pt}
\sankeyfork{p3}{1/p5,1/p6}
\sankeyadvance{p5}{70pt}
\sankeyfork{p1}{1/p7,1/p8,1/p9}
\sankeyadvance{p7}{50pt}
\sankeyadvance{p9}{50pt}
\sankeyadvance{p4}{40pt}
\sankeyturnleft{p4}{90}
\sankeyadvance{p4}{65pt}
\sankeyadvance{p7}{40pt}
\sankeynode{
  name=p11,at={ [shift={(50pt,-15pt)}] p7},quantity=3,
  forked={1/p7a,1/p9a,1/p5a},
}
\sankeyoutin{p7}{p7a}
\sankeyoutin{p9}{p9a}
\sankeyoutin{p5}{p5a}
\sankeyadvance{p11}{30pt}
\sankeyend{p11}
\sankeyturnright{p8}{90}
\sankeyturnright{p6}{90}
\sankeyturnright{p4}{90}
\sankeyset{hot}
\sankeyadvance[cold to hot]{p8}{40pt}
\sankeynode{
  name=p10,at={ [shift={(-15pt,-60pt)}] p8},angle=-90,quantity=3,
  forked={1/p8a,1/p6a,1/p4a},
}
\sankeyoutin[cold to hot]{p4}{p4a}
\sankeyoutin[cold to hot]{p6}{p6a}
\sankeyoutin{p8}{p8a}
\sankeyadvance{p10}{30pt}
\sankeyend{p10}
\end{sankeydiagram}
\end{tikzpicture}

```

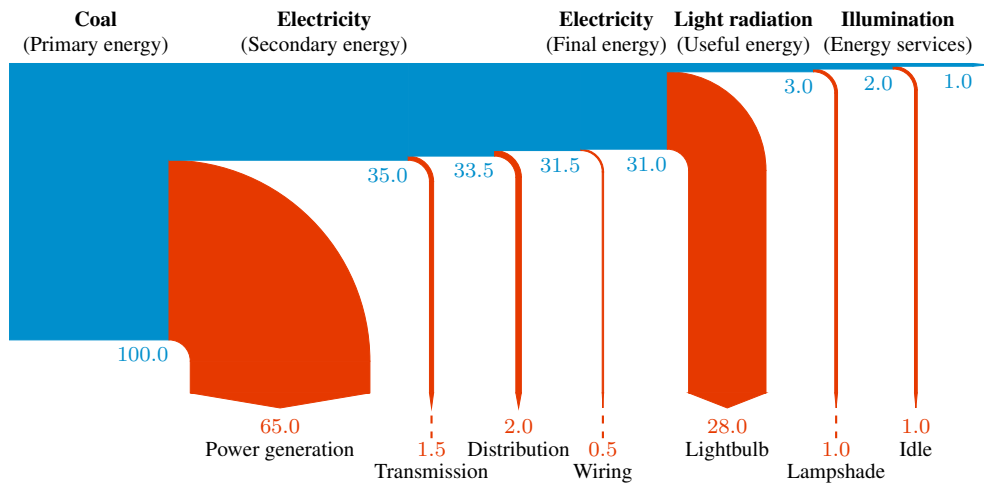



Figure 2: Energy diagram

9 Energy diagram

This example comes from [IB Physics Blog](#) by Kyu Won Shim.

See figure 2. The `sankey-example-energy.tex` file contains the following code and is an attachment of the current PDF document.

```
\begin{tikzpicture}
  % font choice
  \renewcommand{\rmdefault}{txr}\rmfamily\footnotesize
  \sisetup{
    round-mode=places,
    round-precision=1,
    add-decimal-zero,
    round-integer-to-decimal,
  }
  \begin{sankeydiagram}
    \colorlet{energy}{blue!30!cyan!80!black}
    \colorlet{lost energy}{red!50!orange!90!black}
    \sankeyset{
      ratio=13em/100,
      minimum radius=1em,
      start style=simple,end style=simple,
      draw/.style={draw=none,line width=0},
      energy/.style={
        fill/.style={
          draw=energy,
          line width=0,
          fill=energy,
        }
      },
      lost energy/.style={
        fill/.style={
          draw=lost energy,
          line width=0,
          fill=lost energy,
        }
      }
    }
  }

  \newcommand\abovelabel[2]{ % valname, label
    \node[anchor=south east,align=center,inner xsep=0] at (#1.left) {#2};
  }

  \newcommand\energylabel[1]{ % valname
    \node[anchor=north east,text=energy,inner xsep=0] at (#1.right)
    {\num{\sankeygetnodeqty{#1}}};
  }
}
```

```

\newcommand\lostenergylabel[2]{ % valname, label
  \node[anchor=north,text=lost energy] at ([yshift=-2.5mm]#1.center)
    (value)
    {\num{\sankeygetnodeqty{#1}}};
  \node[anchor=north,inner sep=0,align=center] at (value.south) {#2};
}

\newcommand\lostenergylabelbottom[2]{ % valname, label
  \draw[draw=lost energy,dashed,thick]
    ([yshift=-3mm]#1.center) coordinate (#1) -- ([yshift=-3mm]#1.center);
  \lostenergylabel{#1}{#2}
}

\sankeynode{name=Co,quantity=100.0}
\path (Co.right) ++(0,-7mm) coordinate (c);

\newcommand\turnandstop[2]{ % valname, label
  \begingroup
  \sankeyset{lost energy}
  \sankeyturnright{#1}{90}
  \sankeynode{as=#1,name=#1-stop,at={#1 |- c}}
  \sankeyoutin{#1}{#1-stop}
  \sankeynode{as=#1-stop,name=#1}
  \sankeyend{#1}
  \lostenergylabel{#1}{#2}
  \endgroup
}

\newcommand\turnandstopbottom[2]{ % valname, label
  \begingroup
  \sankeyset{lost energy}
  \sankeyturnright{#1}{90}
  \sankeynode{as=#1,name=#1-stop,at={#1 |- c}}
  \sankeyoutin{#1}{#1-stop}
  \sankeynode{as=#1-stop,name=#1}
  \sankeyend{#1}
  \lostenergylabelbottom{#1}{#2}
  \endgroup
}

\def\hshift{6.25em}

\sankeyadvance[energy]{Co}{1.2*\hshift}
\abovelabel{Co}{\textbf{Coal}}\(\text{Primary energy})}
\energylabel{Co}
\sankeyfork{Co}{35/E11,65/Pg}
\turnandstop{Pg}{Power generation}

\sankeyadvance[energy]{E11}{1.8*\hshift}
\abovelabel{E11}{\textbf{Electricity}}\(\text{Secondary energy})}
\energylabel{E11}
\sankeyfork{E11}{33.5/E12,1.5/Tr}
\turnandstopbottom{Tr}{Transmission}

\sankeyadvance[energy]{E12}{.65*\hshift}
\energylabel{E12}
\sankeyfork{E12}{31.5/E13,2.0/Di}
\turnandstop{Di}{Distribution}

\sankeyadvance[energy]{E13}{.65*\hshift}
\energylabel{E13}
\sankeyfork{E13}{31.0/E14,0.5/Wi}
\turnandstopbottom{Wi}{Wiring}

\sankeyadvance[energy]{E14}{.65*\hshift}
\abovelabel{E14}{\textbf{Electricity}}\(\text{Final energy})}
\energylabel{E14}
\sankeyfork{E14}{3.0/Lr1,28.0/Lb}

```

```

\turnandstop{Lb}{Lightbulb}

\sankeyadvance[energy]{Lr1}{1.1*\hshift}
\abovelabel{Lr1}{\textbf{Light radiation}}\|(Useful energy)}
\energylabel{Lr1}
\sankeyfork{Lr1}{2.0/Lr2,1.0/Ls}
\turnandstopbottom{Ls}{Lampshade}

\sankeyadvance[energy]{Lr2}{.6*\hshift}
\energylabel{Lr2}
\sankeyfork{Lr2}{1.0/I1,1.0/Id}
\turnandstop{Id}{Idle}

\sankeyadvance[energy]{I1}{.6*\hshift}
\abovelabel{I1}{\textbf{Illumination}}\|(Energy services)}
\energylabel{I1}
\sankeyend[energy]{I1}
\end{sankeydiagram}
\end{tikzpicture}

```

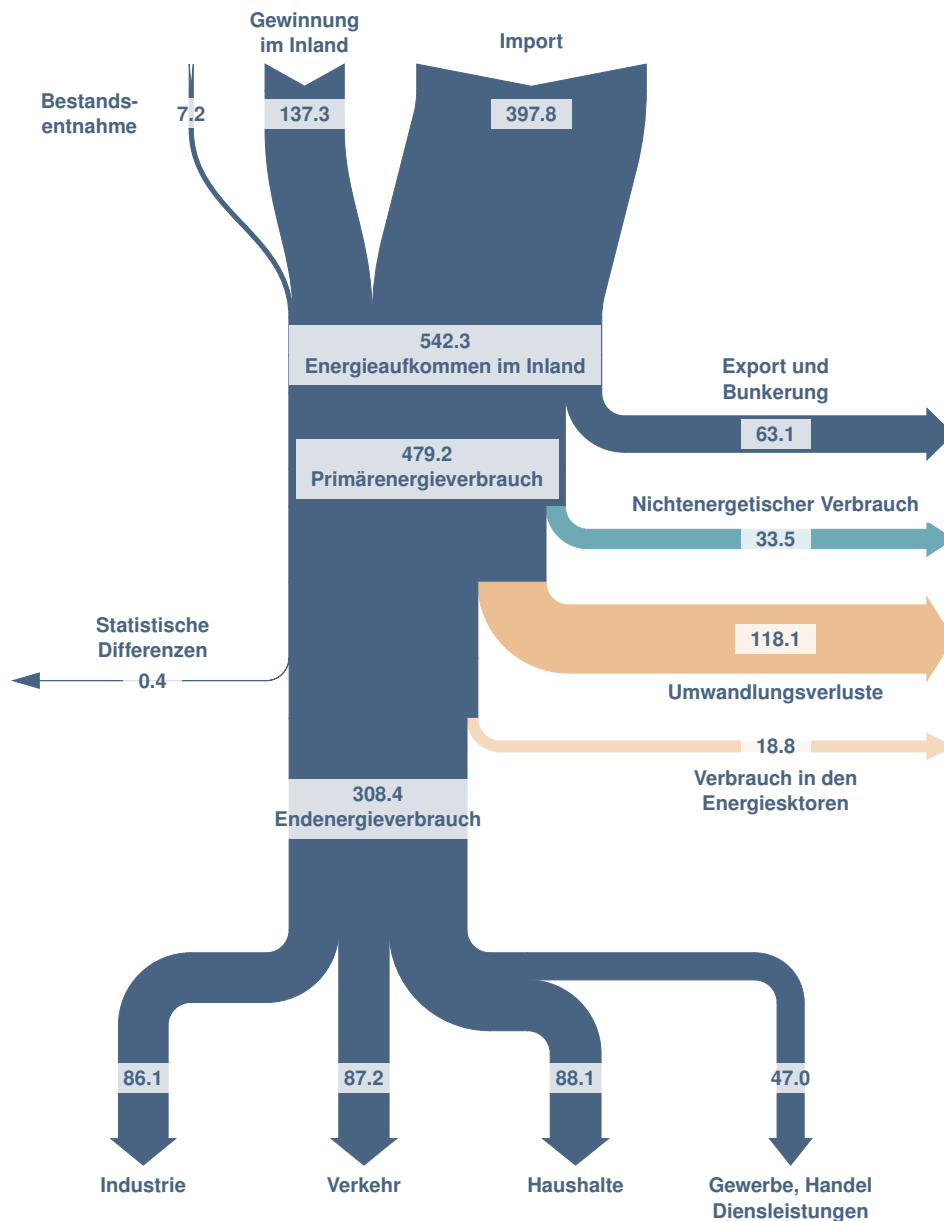


Figure 3: Example from TeX.se question

10 Example from question on TeX.se

This example came from [this question](#) on [TeX.StackExchange](#).

See figure 3. The `sankey-example2.tex` file contains the following code and is an attachment of the current PDF document.

```
\begin{tikzpicture}
  \renewcommand*\sfdefault{txss}

  \begin{sankeydiagram}%[debug]
    \sankeyset{
      ratio=4cm/524.3,
      minimum radius=3mm,
      start style=arrow,
      end style=arrow,
      fill/.style={
        line width=0pt,
        fill=cyan!50!blue!50!black,
        draw=cyan!50!blue!50!black,
      },
      draw/.style={draw=none},
      every node/.style={angle=-90},
    }
  \end{sankeydiagram}
\end{tikzpicture}
```

```

}

\sankeynodestart{name=B,at={-.5,0},quantity=7.2}
\coordinate[below=1mm of B.center] (B label);
\sankeyadvance{B}{5mm}
\sankeynodestart{name=GI,at={1,0},quantity=137.3}
\coordinate[below=1mm of GI.center] (GI label);
\sankeyadvance{GI}{5mm}
\sankeynodestart{name=I,at={4,0},quantity=397.8}
\coordinate[below=1mm of I.center] (I label);
\sankeynode{
  name=EI,at={2.86,-3},quantity=542.3,
  forked={397.8/Ia,137.3/GIa,7.2/Ba}
}
\sankeydubins[minimum radius=1.2cm]{I}{Ia}
\sankeyoutin{GI}{GIa}
\sankeyoutin{B}{Ba}
\sankeyadvance{EI}{5mm}
\coordinate (EI label) at (EI);
\sankeyadvance{EI}{5mm}
\sankeyfork{EI}{63.1/EB,479.2/P}

\sankeyturnleft{EB}{90}
\sankeyadvance{EB}{4cm}
\coordinate (EB label) at ($(EB)!.5!(EB-old)$);
\sankeyend{EB}

\sankeyadvance{P}{10mm}
\coordinate (P label) at (P);
\sankeyadvance{P}{5mm}

\sankeyfork{P}{33.5/NV,445.7/P-NV}

{
  \colorlet{NV color}{cyan!80!lime!50!gray}
  \sankeyset{fill/.append style={fill=NV color,draw=NV color}}
  \sankeyturnleft{NV}{90}
  \sankeynode{as=NV,name=NV2,at=NV -| EB}
  \sankeyoutin{NV}{NV2}
  \coordinate (NV label) at (NV -| EB label);
  \sankeyend{NV2}
}

\sankeyadvance{P-NV}{10mm}
\sankeyfork{P-NV}{118.1/U,327.6/P-NV-U}

{
  \sankeyset{
    fill/.style={fill=orange!70!gray!50,draw=orange!70!gray!50}
  }
  \sankeyturnleft{U}{90}
  \sankeynode{as=U,name=U2,at=U -| EB}
  \sankeyoutin{U}{U2}
  \coordinate (U label) at (U -| EB label);
  \sankeyend{U2}
}

\sankeyadvance{P-NV-U}{10mm}
\sankeyfork{P-NV-U}{327.2/P-NV-U-SD,0.4/SD}

{
  \sankeyturnright{SD}{90}
  \sankeyadvance{SD}{15mm}
  \coordinate (SD label) at (SD);
  \sankeyadvance{SD}{15mm}
  \sankeyend{SD}
}

\sankeyadvance{P-NV-U-SD}{8mm}

```

```

\sankeyfork{P-NV-U-SD}-{18.8/VE,308.4/E}

{
  \sankeyset{fill/.append style={orange!70!gray!30}}
  \sankeyturnleft{VE}-{90}
  \sankeynode{as=VE,name=VE2,at=VE -| EB}
  \sankeyoutin{VE}-{VE2}
  \coordinate (VE label) at (VE -| EB label);
  \sankeyend{VE2}
}

\sankeyadvance{E}-{8mm}
\coordinate (E label) at (E);
\sankeyadvance{E}-{20mm}
\sankeyfork{E}-{135.1/H+GHD,87.2/V,86.1/In}

\sankeyturnright{In}-{90}
\sankeyadvance{In}-{10mm}
\sankeyturnleft{In}-{90}
\sankeyadvance{In}-{5mm}
\coordinate (In label) at (In);
\sankeyadvance{In}-{10mm}
\sankeyend{In}

\sankeynode{as=V,name=V2,at=V|-In label}
\sankeyoutin{V}-{V2}
\coordinate (V label) at (V2);
\sankeyadvance{V2}-{10mm}
\sankeyend{V2}

\sankeyturnleft{H+GHD}-{90}
\sankeyadvance{H+GHD}-{5mm}
\sankeyfork{H+GHD}-{47.0/GHD,88.1/H}

\sankeyturnright{H}-{90}
\sankeynode{as=H,name=H2,at=H|-In label}
\sankeyoutin{H}-{H2}
\coordinate (H label) at (H2);
\sankeyadvance{H2}-{10mm}
\sankeyend{H2}

\sankeyadvance{GHD}-{30mm}
\sankeyturnright{GHD}-{90}
\sankeynode{as=GHD,name=GHD2,at=GHD|-In label}
\sankeyoutin{GHD}-{GHD2}
\coordinate (GHD label) at (GHD2);
\sankeyadvance{GHD2}-{10mm}
\sankeyend{GHD2}
\end{sankeydiagram}

% labels
\tikzset{
  label/.style={
    fill=white,fill opacity=.8,text opacity=1,
    inner sep=1mm,
    text=cyan!50!blue!50!black,
    inner xsep=2mm,
    font=\sffamily\bfseries\footnotesize,
    align=center,
  },
}

\node[label,anchor=north] (B label) at (B label) {7.2};
\node[label,left=1mm of B label] {Bestands-\entnahme};
\node[label,anchor=north] at (GI label) {137.3};
\node[label,above=5mm of GI label] {Gewinnung\im Inland};
\node[label,anchor=north] at (I label) {397.8};
\node[label,above=5mm of I label] {Import};

\node[label] at (EI label) {542.3\Energieaufkommen im Inland};

```

```

\node[label,anchor=center] (EB label) at (EB label) {63.1};
\node[label,above=1mm of EB label] {Export und\\Bunkerung};

\node[label] at (P label) {479.2\\Primärenergieverbrauch};

\node[label,anchor=center] (NV label) at (NV label) {33.5};
\node[label,above=0mm of NV label] {Nichtenergetischer Verbrauch};

\node[label,anchor=center] (U label) at (U label) {118.1};
\node[label,below=3mm of U label] {Umwandlungsverluste};

\node[label,anchor=center] (SD label) at (SD label) {0.4};
\node[label,above=0mm of SD label] {Statistische\\Differenzen};

\node[label,anchor=center] (VE label) at (VE label) {18.8};
\node[label,below=0mm of VE label] {Verbrauch in den\\Energiesktoren};

\node[label,anchor=north] (E label) at (E label)
{308.4\\Endenergieverbrauch};

\node[label,anchor=north] (In label) at (In label) {86.1};
\node[label,anchor=north,below=1cm of In label] {Industrie};

\node[label,anchor=north] (V label) at (V label) {87.2};
\node[label,anchor=north,below=1cm of V label] {Verkehr};

\node[label,anchor=north] (H label) at (H label) {88.1};
\node[label,anchor=north,below=1cm of H label] {Haushalte};

\node[label,anchor=north] (GHD label) at (GHD label) {47.0};
\node[label,anchor=north,below=1cm of GHD label]
{Gewerbe, Handel\\Diensleistungen};
\end{tikzpicture}

```

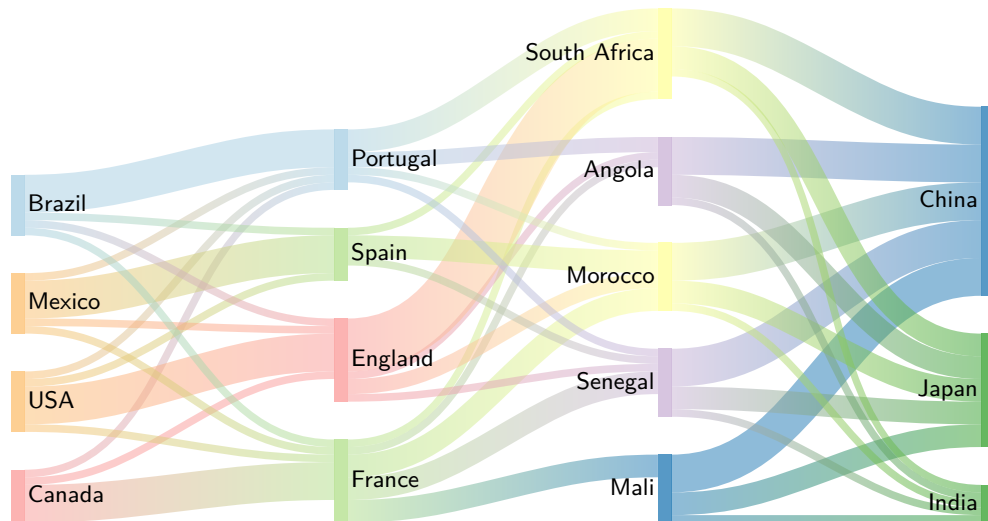


Figure 4: Reproduction of an example from Google Charts documentation

11 Reproduction of an example from Google Charts documentation

This example is a reproduction of an example of Google Charts Documentation⁴.

See figure 4. The `sankey-example3.tex` file contains the following code and is an attachment of the current PDF document.

```
\begin{tikzpicture}
  \begin{sankeydiagram}%[debug]
    \sffamily
    \sankeyset{
      ratio=1cm/10,
      outin steps=2,
      draw/.style={draw=none,line width=0pt},
      color/.style={fill/.style={fill=#1,fill opacity=.75}},
      shade/.style 2 args={fill/.style={left color=#1,
        right color=#2,fill opacity=.5}},
      % colors
      @define HTML color/.code args={#1/#2}{\definecolor{#1}{HTML}{#2}},
      @define HTML color/.list={
        cyan/a6cee3,lime/b2df8a,red/fb9a99,orange/fdbf6f,
        violet/cab2d6,yellow/ffff99,blue/1f78b4,green/33a02c
      },
      % colors of countries
      @let country color/.code args={#1/#2}{\colorlet{#1}[rgb]{#2}},
      @let country color/.list={
        CA/red,US/orange,MX/orange,BR/cyan,FR/lime,GB/red,
        SP/lime,PT/cyan,ML/blue,SN/violet,MA/yellow,
        AO/violet,ZA/yellow,IN/green,JP/green,CN/blue
      },
    },
  \def\vdist{5mm}
  \def\hwidth{.5em}
  \def\hdist{4.1cm}

  \sankeynode{name=CA,quantity=7}
  \sankeynode{name=US,quantity=8,at={[\yshift=\vdist]CA.left},anchor=right}
  \sankeynode{name=MX,quantity=8,at={[\yshift=\vdist]US.left},anchor=right}
  \sankeynode{name=BR,quantity=8,at={[\yshift=\vdist]MX.left},anchor=right}

  \foreach \country in {CA,US,MX,BR}{
    \sankeyadvance[color=\country]{\country}{\hwidth}
  }
}
```

⁴<https://developers.google.com/chart/interactive/docs/gallery/sankey>


```

\sankeyfork{CA}{1/CA-to-PT,1/CA-to-GB,5/CA-to-FR}
\sankeyfork{US}{1/US-to-PT,1/US-to-SP,5/US-to-GB,1/US-to-FR}
\sankeyfork{MX}{1/MX-to-PT,5/MX-to-SP,1/MX-to-GB,1/MX-to-FR}
\sankeyfork{BR}{5/BR-to-PT,1/BR-to-SP,1/BR-to-GB,1/BR-to-FR}

\sankeynode{name=FR,quantity=11,
  at={ [xshift=\hdist] CA.right }, anchor=right }
\sankeynode{name=GB,quantity=11,
  at={ [yshift=\vdist] FR.left }, anchor=right }
\sankeynode{name=SP,quantity=7,
  at={ [yshift=\vdist] GB.left }, anchor=right }
\sankeynode{name=PT,quantity=8,
  at={ [yshift=\vdist] SP.left }, anchor=right }

\sankeyfork{FR}
{1/FR-from-BR,1/FR-from-MX,1/FR-from-US,5/FR-from-CA,3/FR-from-00}
\sankeyfork{GB}
{1/GB-from-BR,1/GB-from-MX,5/GB-from-US,1/GB-from-CA,3/GB-from-00}
\sankeyfork{SP}{1/SP-from-BR,5/SP-from-MX,1/SP-from-US}
\sankeyfork{PT}{5/PT-from-BR,1/PT-from-MX,1/PT-from-US,1/PT-from-CA}

\foreach \country in {FR,GB,SP,PT}{
  \sankeyadvance[color=\country]{\country}{\hwidth}
}

\sankeyfork{FR}{1/FR-to-ZA,1/FR-to-AO,3/FR-to-MA,3/FR-to-SN,3/FR-to-ML}
\sankeyfork{GB}{7/GB-to-ZA,1/GB-to-AO,2/GB-to-MA,1/GB-to-SN}
\sankeyfork{SP}{1/SP-to-ZA,3/SP-to-MA,1/SP-to-SN,2/SP-to-00}
\sankeyfork{PT}{3/PT-to-ZA,2/PT-to-AO,1/PT-to-MA,1/PT-to-SN,1/PT-to-00}

\sankeynode{name=ML,quantity=9,
  at={ [xshift=\hdist] FR.right }, anchor=right }
\sankeynode{name=SN,quantity=9,
  at={ [yshift=\vdist] ML.left }, anchor=right }
\sankeynode{name=MA,quantity=9,
  at={ [yshift=\vdist] SN.left }, anchor=right }
\sankeynode{name=AO,quantity=9,
  at={ [yshift=\vdist] MA.left }, anchor=right }
\sankeynode{name=ZA,quantity=12,
  at={ [yshift=\vdist] AO.left }, anchor=right }

\sankeyfork{ML}{3/ML-from-FR,6/Mail-from-00}
\sankeyfork{SN}
{1/SN-from-PT,1/SN-from-SP,1/SN-from-GB,3/SN-from-FR,3/SN-from-00}
\sankeyfork{MA}{1/MA-from-PT,3/MA-from-SP,2/MA-from-GB,3/MA-from-FR}
\sankeyfork{AO}{2/AO-from-PT,1/AO-from-GB,1/AO-from-FR,5/AO-from-00}
\sankeyfork{ZA}{3/ZA-from-PT,1/ZA-from-SP,7/ZA-from-GB,1/ZA-from-FR}

\foreach \country in {ML,SN,MA,AO,ZA}{
  \sankeyadvance[color=\country]{\country}{\hwidth}
}

\sankeyfork{ML}{5/ML-to-CN,3/ML-to-JP,1/ML-to-IN}
\sankeyfork{SN}{5/SN-to-CN,3/SN-to-JP,1/SN-to-IN}
\sankeyfork{MA}{5/MA-to-CN,3/MA-to-JP,1/MA-to-IN}
\sankeyfork{AO}{5/AO-to-CN,3/AO-to-JP,1/AO-to-IN}
\sankeyfork{ZA}{5/ZA-to-CN,3/ZA-to-JP,1/ZA-to-IN,3/ZA-to-00}

\sankeynode{name=IN,quantity=5,
  at={ [xshift=\hdist] ML.right }, anchor=right }
\sankeynode{name=JP,quantity=15,
  at={ [yshift=\vdist] IN.left }, anchor=right }
\sankeynode{name=CN,quantity=25,
  at={ [yshift=\vdist] JP.left }, anchor=right }

\sankeyfork{IN}
{1/IN-from-ZA,1/IN-from-AO,1/IN-from-MA,1/IN-from-SN,1/IN-from-ML}
\sankeyfork{JP}

```

```

{3/JP-from-ZA,3/JP-from-AO,3/JP-from-MA,3/JP-from-SN,3/JP-from-ML}
\sankeyfork{CN}
{5/CN-from-ZA,5/CN-from-AO,5/CN-from-MA,5/CN-from-SN,5/CN-from-ML}

\foreach \country in {IN,JP,CN}{
  \sankeyadvance[color=\country]{\country}{\hwidth}
}

\foreach \startcountry/\countries in {
  CA/{PT,GB,FR},    US/{PT,SP,GB,FR},    MX/{PT,SP,GB,FR},
  BR/{PT,SP,GB,FR}, FR/{ML,SN,MA,AO,ZA}, GB/{SN,MA,AO,ZA},
  SP/{SN,MA,ZA},    PT/{SN,MA,AO,ZA},    ML/{IN,JP,CN},
  SN/{IN,JP,CN},    MA/{IN,JP,CN},        AO/{IN,JP,CN},
  ZA/{IN,JP,CN}}
{
  \foreach \endcountry in \countries {
    \sankeyoutin[shade=\startcountry]{\endcountry}
    {\startcountry-to-\endcountry}{\endcountry-from-\startcountry}
  }
}

\foreach \country/\countryname in {CA/Canada, US/USA, MX/Mexico,
BR/Brazil, FR/France, GB/England, SP/Spain, PT/Portugal}
{
  \node[anchor=west,inner sep=.1em,font=\small]
  at (\country) {\countryname\phantom{Ag}};
}

\foreach \country/\countryname in {
  ML/Mali, SN/Senegal, MA/Morocco, AO/Angola,
  ZA/South Africa, IN/India, JP/Japan, CN/China}
{
  \node[anchor=east,inner sep=.1em,font=\small]
  at (\country-old) {\countryname\phantom{Ag}};
}
\end{sankeydiagram}
\end{tikzpicture}

```

11.1 Variation

Here is a variation of the previous example using the `rotate` key.

See figure 5 on the next page. The `sankey-example3-variation.tex` file contains the following code and is an attachment of the current PDF document.

```

\begin{tikzpicture}[rotate=-90]
\begin{sankeydiagram}[rotate=-90]
\setfamily
\sankeyset{
  ratio=1.7cm/10,
  outin steps=2,
  start style=arrow,
  end style=simple,
  draw/.style={draw=white,line width=.4pt},
  color/.style={fill/.style={fill=#1,fill opacity=.75}},
  shade/.style 2 args={fill/.style={
    fill=none,line width=0,
    top color=#1,bottom color=#2,
    middle color=#1!50!#2!50!white,
    fill opacity=.75}},
% colors
\definecolor{HTML}{color}{#1/#2}{\definecolor{#1}{HTML}{#2}},
\definecolor{HTML}{color}{.list={
  cyan/a6cee3,lime/b2df8a,red/fb9a99,orange/fdbf6f,
  violet/cab2d6,yellow/ffff99,blue/1f78b4,green/33a02c
}},
% colors of countries
\let country color/.code args={#1/#2}{\colorlet{#1}[rgb]{#2}},
\let country color/.list={

```

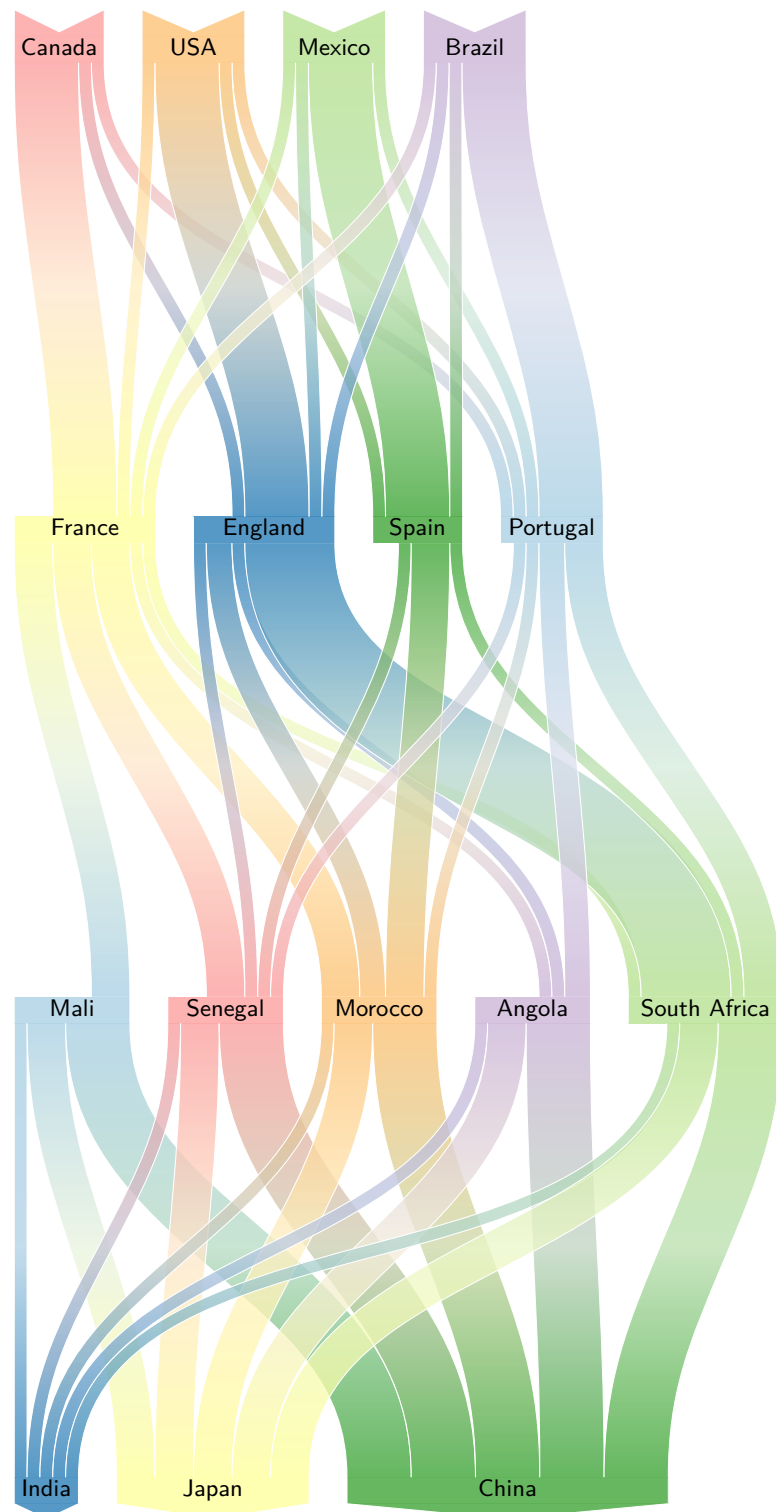


Figure 5: Reproduction of an example from Google Charts documentation – variation using the `rotate` key.

```

CA/red,US/orange,MX/lime,BR/violet,FR/yellow,GB/blue,
SP/green,PT/cyan,ML/cyan,SN/red,MA/orange,
AO/violet,ZA/lime,IN/blue,JP/yellow,CN/green
},
}
\def\vdist{5mm}
\def\hwidth{1em}
\def\hdist{6cm}

\sankeynode{name=CA,quantity=7}
\sankeynode{name=US,quantity=8,at={ [yshift=\vdist]CA.left},anchor=right}
\sankeynode{name=MX,quantity=8,at={ [yshift=\vdist]US.left},anchor=right}
\sankeynode{name=BR,quantity=8,at={ [yshift=\vdist]MX.left},anchor=right}

\foreach \country in {CA,US,MX,BR}{
  \sankeystart[color=\country]{\country}
  \sankeyadvance[color=\country]{\country}{\hwidth}
}

\sankeyfork{CA}{1/CA-to-PT,1/CA-to-GB,5/CA-to-FR}
\sankeyfork{US}{1/US-to-PT,1/US-to-SP,5/US-to-GB,1/US-to-FR}
\sankeyfork{MX}{1/MX-to-PT,5/MX-to-SP,1/MX-to-GB,1/MX-to-FR}
\sankeyfork{BR}{5/BR-to-PT,1/BR-to-SP,1/BR-to-GB,1/BR-to-FR}

\sankeynode{name=FR,quantity=11,
  at={ [xshift=\hdist]CA.right},anchor=right}
\sankeynode{name=GB,quantity=11,
  at={ [yshift=\vdist]FR.left},anchor=right}
\sankeynode{name=SP,quantity=7,
  at={ [yshift=\vdist]GB.left},anchor=right}
\sankeynode{name=PT,quantity=8,
  at={ [yshift=\vdist]SP.left},anchor=right}

\sankeyfork{FR}
{1/FR-from-BR,1/FR-from-MX,1/FR-from-US,5/FR-from-CA,3/FR-from-00}
\sankeyfork{GB}
{1/GB-from-BR,1/GB-from-MX,5/GB-from-US,1/GB-from-CA,3/GB-from-00}
\sankeyfork{SP}{1/SP-from-BR,5/SP-from-MX,1/SP-from-US}
\sankeyfork{PT}{5/PT-from-BR,1/PT-from-MX,1/PT-from-US,1/PT-from-CA}

\foreach \country in {FR,GB,SP,PT}{
  \sankeyadvance[color=\country]{\country}{\hwidth}
}

\sankeyfork{FR}{1/FR-to-ZA,1/FR-to-AO,3/FR-to-MA,3/FR-to-SN,3/FR-to-ML}
\sankeyfork{GB}{7/GB-to-ZA,1/GB-to-AO,2/GB-to-MA,1/GB-to-SN}
\sankeyfork{SP}{1/SP-to-ZA,3/SP-to-MA,1/SP-to-SN,2/SP-to-00}
\sankeyfork{PT}{3/PT-to-ZA,2/PT-to-AO,1/PT-to-MA,1/PT-to-SN,1/PT-to-00}

\sankeynode{name=ML,quantity=9,
  at={ [xshift=\hdist]FR.right},anchor=right}
\sankeynode{name=SN,quantity=9,
  at={ [yshift=\vdist]ML.left},anchor=right}
\sankeynode{name=MA,quantity=9,
  at={ [yshift=\vdist]SN.left},anchor=right}
\sankeynode{name=AO,quantity=9,
  at={ [yshift=\vdist]MA.left},anchor=right}
\sankeynode{name=ZA,quantity=12,
  at={ [yshift=\vdist]AO.left},anchor=right}

\sankeyfork{ML}{3/ML-from-FR,6/ML-from-00}
\sankeyfork{SN}
{1/SN-from-PT,1/SN-from-SP,1/SN-from-GB,3/SN-from-FR,3/SN-from-00}
\sankeyfork{MA}{1/MA-from-PT,3/MA-from-SP,2/MA-from-GB,3/MA-from-FR}
\sankeyfork{AO}{2/AO-from-PT,1/AO-from-GB,1/AO-from-FR,5/AO-from-00}
\sankeyfork{ZA}{3/ZA-from-PT,1/ZA-from-SP,7/ZA-from-GB,1/ZA-from-FR}

\foreach \country in {ML,SN,MA,AO,ZA}{
  \sankeyadvance[color=\country]{\country}{\hwidth}
}

```

```

}

\sankeyfork{ML}{5/ML-to-CN,3/ML-to-JP,1/ML-to-IN}
\sankeyfork{SN}{5/SN-to-CN,3/SN-to-JP,1/SN-to-IN}
\sankeyfork{MA}{5/MA-to-CN,3/MA-to-JP,1/MA-to-IN}
\sankeyfork{AO}{5/AO-to-CN,3/AO-to-JP,1/AO-to-IN}
\sankeyfork{ZA}{5/ZA-to-CN,3/ZA-to-JP,1/ZA-to-IN,3/ZA-to-00}

\sankeynode{name=IN,quantity=5,
  at={ [xshift=\hdist] ML.right }, anchor=right }
\sankeynode{name=JP,quantity=15,
  at={ [yshift=\vdist] IN.left }, anchor=right }
\sankeynode{name=CN,quantity=25,
  at={ [yshift=\vdist] JP.left }, anchor=right }

\sankeyfork{IN}
{1/IN-from-ZA,1/IN-from-AO,1/IN-from-MA,1/IN-from-SN,1/IN-from-ML}
\sankeyfork{JP}
{3/JP-from-ZA,3/JP-from-AO,3/JP-from-MA,3/JP-from-SN,3/JP-from-ML}
\sankeyfork{CN}
{5/CN-from-ZA,5/CN-from-AO,5/CN-from-MA,5/CN-from-SN,5/CN-from-ML}

\foreach \country in {IN,JP,CN}{
  \sankeyadvance[color=\country]{\country}{\hwidth}
  \sankeyend[color=\country]{\country}
}

\foreach \startcountry/\countries in {
  CA/{PT,GB,FR},    US/{PT,SP,GB,FR},    MX/{PT,SP,GB,FR},
  BR/{PT,SP,GB,FR}, FR/{ML,SN,MA,AO,ZA}, GB/{SN,MA,AO,ZA},
  SP/{SN,MA,ZA},    PT/{SN,MA,AO,ZA},    ML/{IN,JP,CN},
  SN/{IN,JP,CN},    MA/{IN,JP,CN},        AO/{IN,JP,CN},
  ZA/{IN,JP,CN}}
{
  \foreach \endcountry in \countries {
    \sankeyoutin[shade={\startcountry}{\endcountry}]
    {\startcountry-to-\endcountry}{\endcountry-from-\startcountry}
  }
}

\foreach \country/\countryname in {CA/Canada, US/USA, MX/Mexico,
  BR/Brazil, FR/France, GB/England, SP/Spain, PT/Portugal}
{
  \node[anchor=south,inner sep=.1em,font=\small]
  at (\country) {\countryname\vphantom{Ag}};
}

\foreach \country/\countryname in {
  ML/Mali, SN/Senegal, MA/Morocco, AO/Angola,
  ZA/South Africa, IN/India, JP/Japan, CN/China}
{
  \node[anchor=south,inner sep=.1em,font=\small]
  at (\country) {\countryname\vphantom{Ag}};
}

\end{sankeydiagram}
\end{tikzpicture}

```

12 Very nice example – Nadieh Bremer creation

Graphic designer Nadieh Bremer created this very nice Sankey diagram⁵ for Adyen's second half 2018 report to shareholders. It is coded here with her kind permission.

You can change the data values between lines 28 and 53.

See figure 6 on the following page. The `sankey-example4.tex` file contains the following code and is an attachment of the current PDF document.

```

1 \begin{tikzpicture}
2   \renewcommand*\sfdefault{txss}
3   \sffamily
4   \sisetup{
5     detect-all=true,
6     group-separator={,},
7     group-minimum-digits=4,
8   }
9   % storage of labels
10  \newcommand\LabSet[2]{% node name, label
11    \expandafter\edef\csname#1@Lab\endcsname{#2}}
12  \newcommand\Lab[1]{% node name
13    \csname#1@Lab\endcsname}
14  % storage of quantities
15  \newcommand\QtySet[2]{% node name, quantity
16    \expandafter\edef\csname#1@Qty\endcsname{\fpeval{#2}}}
17  \newcommand\Qty[1]{% node name
18    \csname#1@Qty\endcsname}
19  % all nodes with their name, label and quantity
20  \sankeyset{
21    def data/.code args={#1/#2/#3}{% node name/label/values
22      \LabSet{#1}{#2}
23      \QtySet{#1}{#3}
24      \typeout{#1: \Qty{#1}€ (\Lab{#1})}
25    },
26    def data/.list={
27      {Pf/Processing\fees/71713},
28      {Sog/Sales of\good/4547},
29      {Sf/Settlement\fees/842075},
30      {Os/Other\services/37532},
31      {R/Revenues\Qty{Pf}+\Qty{Sog}+\Qty{Sf}+\Qty{Os}},
32      {Coi/Cost of Inventory/5151},
33      {Ciffi/Cost insecure from financial institutions/758234},
34      {Nr/Net revenue\Qty{R}-\Qty{Coi}-\Qty{Ciffi}},
35      {Aadotaifa/Amortization and\depreciation of tangible and\
36        intangible fixed assets/4688},
37      {Ssapc/Social securities and\pension costs/7860},
38      {Was/Wages and salaries/35627},
39      {Ooe/Other operating expenses/37346},
40      {Nr2/-/\Qty{Nr}-\Qty{Aadotaifa}-\Qty{Ssapc}-\Qty{Was}-\Qty{Ooe}},
41      {Oi/Other income/47},
42      {Ibiieait/Income before interest income,\interest expense and
43        income taxes\Qty{Nr2}+\Qty{Oi}},
44      {Fe/Finance expense/561},
45      {Ofr/Other financial results/2533},
46      {Ibiieait2/-/\Qty{Ibiieait}-\Qty{Fe}-\Qty{Ofr}},
47      {Fi/Finance income/204},
48      {Ibit/Income before income taxes\Qty{Ibiieait2}+\Qty{Fi}},
49      {It/Income taxes/21134},
50      {Niftp/Net income for the period\Qty{Ibit}-\Qty{It}},
51      {Octa/Other currency\translation adjustments/785},
52      {Tci/Total comprehensive income\Qty{Niftp}+\Qty{Octa}}
53    },
54  }
55
56  \definecolor{mygreen}{RGB}{9,192,82}
57  \tikzset{
58    cost node/.style={

```

⁵<https://www.visualcinnamon.com/portfolio/adyen-report-2019/>

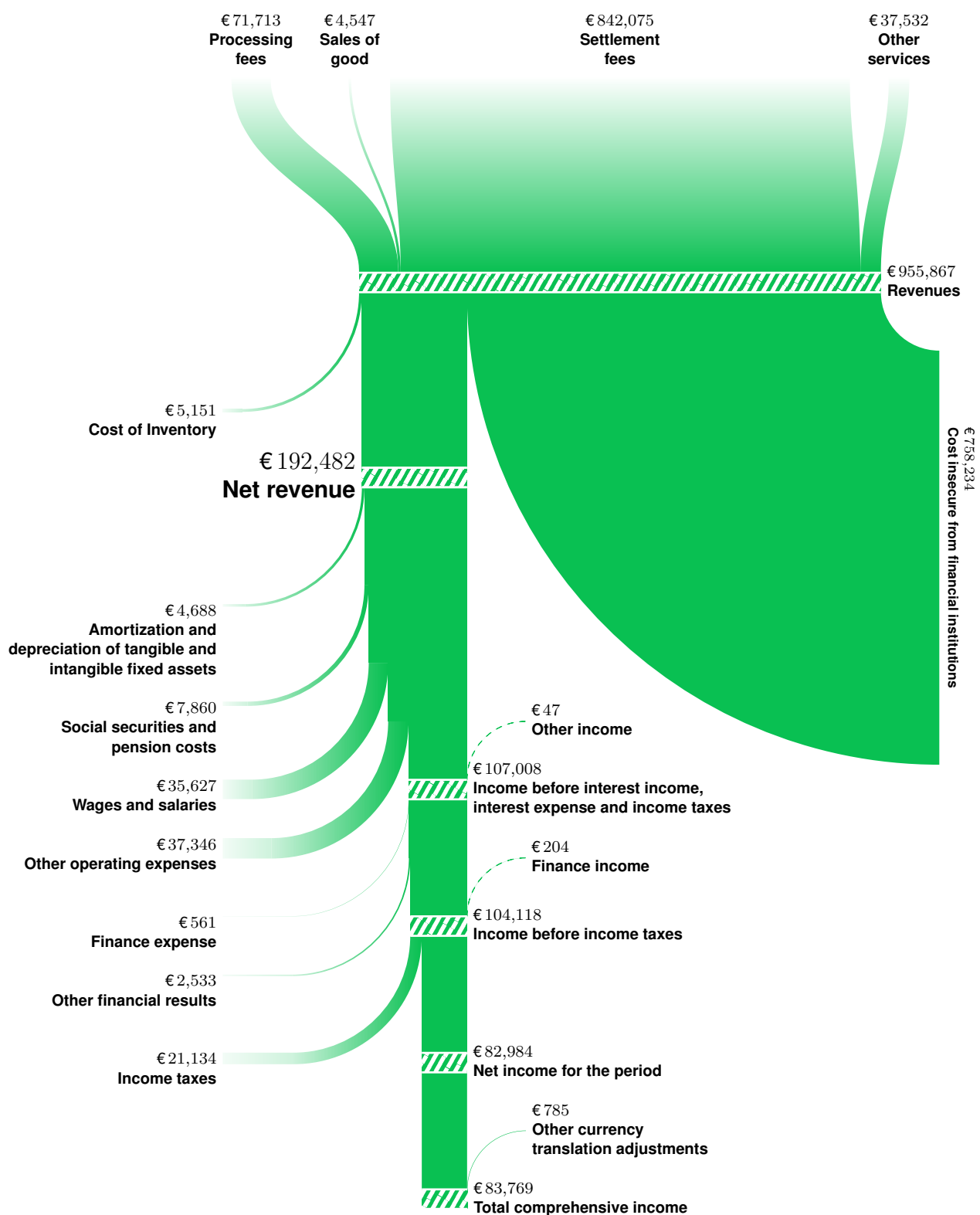


Figure 6: Very nice example – Nadieh Bremer’s creation
(from [Adyen’s Shareholder Report](#))

```

59     overlay,
60     align=flush center,
61     node font=\footnotesize\sffamily\bfseries,
62     inner sep=0,
63     node contents={%
64         {\mdseries\, \num{\Qty{#1}}}\
65         \Lab{#1}\vphantom{g}%
66     },
67 },
68 white hash/.style={
69     draw=none,fill=none,
70     pattern={Lines[angle=60,line width=2pt,distance=4pt]},
71     pattern color=white,
72 },
73 line sep/.style={draw=white,line width=1pt},
74 left label/.style={left=#1,align=flush right,anchor=north east},
75 right label/.style={right=#1,align=flush left},
76 right label hashed/.style={
77     right=1mm of $(#1.left)!.5!(#1-old.left)$,align=flush left,
78 },
79 left label hashed/.style={
80     left=1mm of $(#1.right)!.5!(#1-old.right)$,align=flush right,
81 },
82 }
83
84 \newcommand\turnandstop[1]{
85     \sankeyturn[green to greenwhite]{#1}{-90}
86     \sankeynode{as=#1,name=#1-e,at={#1 -| Coi}}
87     \sankeyoutin[greenwhite to white]{#1}{#1-e}
88     \node[cost node=#1,left label={1mm of #1-e.right}];
89 }
90
91 \begin{sankeydiagram}%[debug]
92     \sankeyset{
93         ratio=28em/1000000,
94         minimum radius=2cm,
95         start style=none,
96         every node/.style={angle=-90},
97         % default fill and draw styles
98         fill/.style={
99             line width=0pt,
100             fill=mygreen,
101         },
102         draw/.style={draw=none},
103         % specific fill and draw styles
104         green to greenwhite/.style={
105             fill/.style={
106                 line width=0pt,
107                 right color=mygreen,
108                 left color=mygreen!20!white,
109             }
110         },
111         greenwhite to white/.style={
112             fill/.style={
113                 line width=0pt,
114                 right color=mygreen!20!white,
115                 left color=mygreen!5!white,
116             }
117         },
118         dashed/.style={draw/.style={draw=mygreen,dashed}},
119     }
120
121     \coordinate (top) at (0,2em);
122
123     \sankeynodestart{name=Pf,quantity=\Qty{Pf}}
124     \node[cost node=Pf,above=.5em of Pf.center];
125
126     \sankeynodestart{name=Sog,quantity=\Qty{Sog},
127         at=[xshift=4em]Pf.left,anchor=right}

```



```

128 \node[cost node=Sog,above=.5em of Sog.center];
129
130 \sankeynodestart{name=Sf,quantity=\Qty{Sf},
131 at={[xshift=2em]Sog.left},anchor=right}
132 \node[cost node=Sf,above=.5em of Sf.center];
133
134 \sankeynodestart{name=Os,quantity=\Qty{Os},
135 at={[xshift=2em]Sf.left},anchor=right}
136 \node[cost node=Os,above=.5em of Os.center];
137
138 \sankeynode{
139 name=R,quantity=\Qty{R},at={[yshift=-10em]Sf.center},
140 forked={\Qty{Os}/Os-a,\Qty{Sf}/Sf-a,\Qty{Sog}/Sog-a,\Qty{Pf}/Pf-a},
141 }
142
143 \foreach \nodename in {Pf,Sog,Sf,Os}{
144 \sankeyoutin[fill/.style={top color=white,bottom color=mygreen}]
145 {\nodename}{\nodename-a}
146 }
147
148 \sankeyadvance{R}{1em}
149 \node[cost node=R,right label hashed=R];
150
151 \sankeyfork{R}{\Qty{Ciffi}/Ciffi,\Qty{Nr}/Nr,\Qty{Coi}/Coi}
152
153 \sankeyturnleft[minimum radius=1cm]{Ciffi}{90}
154 \node[cost node=Ciffi,at={([shift={(1mm,0)}]Ciffi.center)},rotate=-90,
155 anchor=south,align=flush left,node font=\scriptsize\sffamily\bfseries];
156
157 \sankeyturnright[green to greenwhite]{Coi}{90}
158 \sankeyadvance[greenwhite to white]{Coi}{1em}
159 \node[cost node=Coi,left={1mm of [yshift=.75ex]Coi.left},
160 align=flush right,anchor=north east,overlay];
161
162 \sankeyadvance{Nr}{9em}
163 \sankeyadvance{Nr}{1em}
164 \node[cost node=Nr,left label hashed=Nr,
165 node font=\large\sffamily\bfseries];
166
167 \sankeyfork{Nr}{\Qty{Nr2}/Nr2,\Qty{Ooe}/Ooe,
168 \Qty{Was}/Was,\Qty{Ssapc}/Ssapc,\Qty{Aadotaifa}/Aadotaifa}
169
170 \turnandstop{Aadotaifa}
171
172 \sankeyadvance{Ssapc}{5em}
173 \turnandstop{Ssapc}
174
175 \sankeyadvance{Was}{9em}
176 \turnandstop{Was}
177
178 \sankeyadvance{Ooe}{12em}
179 \turnandstop{Ooe}
180
181 \sankeyadvance{Nr2}{15em}
182 \sankeynode{name=Ibiieait,quantity=\Qty{Ibiieait},
183 anchor=right,at={Nr2.right},
184 forked={\Qty{Oi}/Oi,\Qty{Nr2}/Nr2-e}}
185
186 \sankeyturnleftbackward[minimum radius=1cm,dashed]{Oi}{90}
187 \node[cost node=Oi,right label=1mm of Oi.left];
188
189 \sankeyadvance{Ibiieait}{1em}
190 \node[cost node=Ibiieait,right label hashed=Ibiieait];
191
192 \sankeyfork{Ibiieait}
193 {\Qty{Ibiieait2}/Ibiieait2,\Qty{Ofr}/Ofr,\Qty{Fe}/Fe}
194
195 \turnandstop{Fe}
196

```

```

197 \sankeyadvance{0fr}{3em}
198 \turnandstop{0fr}
199
200 \sankeyadvance{Ibiieait2}{6em}
201 \sankeynode{name=Ibit,quantity={\Qty{Ibiieait2}+\Qty{Fi}},
202   anchor=right,at={Ibiieait2.right},
203   forked={\Qty{Fi}/Fi,\Qty{Ibiieait2}/Ibiieait2-e}}
204
205 \sankeyturnleftbackward[minimum radius=1cm,dashed]{Fi}{90}
206 \node[cost node=Fi,right label=1mm of Fi.left];
207
208 \sankeyadvance{Ibit}{1em}
209 \node[cost node=Ibit,right label hashed=Ibit];
210
211 \sankeyfork{Ibit}{\Qty{Niftp}/Niftp,\Qty{It}/It}
212
213 \turnandstop{It}
214
215 \sankeyadvance{Niftp}{6em}
216
217 \sankeyadvance{Niftp}{1em}
218 \node[cost node=Niftp,right label hashed=Niftp];
219
220 \sankeynode{name=Tci,quantity={\Qty{Niftp}+\Qty{Octa}},
221   anchor=right,at={\yshift=-6em}Niftp.right},
222   forked={\Qty{Octa}/Octa,\Qty{Niftp}/Niftp-e}}
223 \sankeyoutin{Niftp}{Niftp-e}
224
225 \sankeyturnleftbackward[minimum radius=1cm]{Octa}{90}
226 \node[cost node=Octa,right label=1mm of Octa.left];
227
228 \sankeyadvance{Tci}{1em}
229 \node[cost node=Tci,right label hashed=Tci];
230
231 \newcommand\hashband[1]{
232   \draw[line sep] (#1-old.right) -- (#1-old.left);
233   \draw[line sep] (#1.right) -- (#1.left);
234   \path[white hash] (#1-old.right) rectangle (#1.left);
235 }
236
237 \foreach \nodename in {R,Nr,Nr,Ibiieait,Ibit,Niftp,Tci}{
238   \hashband{\nodename}
239 }
240
241 \end{sankeydiagram}
242 \end{tikzpicture}

```

This manual contains three parts: [User manual](#) (p.1), [Examples](#) (p.22) and [Implementation](#) (p.43).

Part III

Implementation

Contents

13 The <code>sankey.sty</code> file	43
13.1 Fields	44
13.1.1 Definition of <i>global</i> and <i>expanded</i> fields (using <code>\xdef</code>)	44
13.1.2 Definitions of <i>local</i> fields (using <code>\def</code>)	44
13.1.3 Check if a sankey node is defined	45
13.2 The <code>sankey</code> node shape	45
13.3 Keys	45
13.3.1 Keys to define the scale	46
13.3.2 Rotate offset	46
13.3.3 Minimum radius	46
13.3.4 Outin step	46
13.3.5 Sankey debug	46
13.3.6 Start and end styles	47
13.3.7 Initial parameters	47
13.3.8 Sankey node parameters	48
13.3.9 Internal Tikz style	49
13.4 The <code>sankeydiagram</code> environment	49
13.5 Internal macros	50
13.6 User macros	52
13.7 The <code>sankeydiagram</code> environment (the end)	59
13.8 Predefined start and end styles	60
14 <code>tikzlibrarydubins.code.tex</code>	61
15 Change History	65
16 Index	65

13 The `sankey.sty` file

Poorly commented source code...

Version information:

```
1 \NeedsTeXFormat{LaTeX2e}[2015/10/01]
2 \ProvidesPackage{sankey}[2021/03/14 v3.0 to draw Sankey diagrams]
```

All required packages and TikZ libraries:

```
3 \RequirePackage{xparse}
4 \RequirePackage{etoolbox}
5 \RequirePackage{xfp}
6 \RequirePackage{tikz}
7 \usetikzlibrary{
8   calc,
9   decorations.markings,
10  dubins
11 }
```

Declarations of PGF layers (to debug Sankey diagrams):

```
12 %% add a new layer to debug sankey diagrams
13 \pgfdeclarelayer{background}
14 \pgfdeclarelayer{foreground}
15 \pgfdeclarelayer{sankeydebug}
16 \pgfsetlayers{background,main,foreground,sankeydebug}
```

13.1 Fields

`\snk@newfield` The `sankeynewfield` macro defines setter and getter macros for *key/value* pairs. It requires five parameters: the *def* macro used to store a new value, the *setter* macro name, the *getter* macro name, the *cs name* used by the new field (including a #1 parameter – the *key*) and the *error message* (used by the getter macro if the key is not defined).

The *setter* macro requires two parameters: the key and the value. The *getter* macro requires one parameter: the key.

```

17 \def\snk@newfield#1#2#3#4#5{
18   % setter
19   \def#2##1##2{\expandafter#1\csname #4\endcsname{##2}}
20   % getter
21   \def#3##1{%
22     \ifcsdef{#4}{%
23       \csname#4\endcsname%
24     }{%
25       \PackageError{sankey}{#5}{unknown key with \string#3}%
26     }%
27   }
28 }
```

13.1.1 Definition of *global* and *expanded* fields (using `\xdef`)

`\snk@setnodeqty` `\sankeygetnodeqty` The setter and getter macros to store and retrieve the *quantity* field associated with each Sankey node (the key is the name of the Sankey node).

```

29 \snk@newfield\xdef\snk@setnodeqty\sankeygetnodeqty%
30 {\snk@node@qty@#1}{Unknown sankey node '#1'}
```

`\snk@setnodeorient` `\sankeygetnodeorient` The setter and getter macros to store and retrieve the *angle* (or orientation) field associated with each Sankey node (the key is the name of the Sankey node).

```

31 \snk@newfield\xdef\snk@setnodeorient\sankeygetnodeorient%
32 {\snk@node@orient@#1}{Unknown sankey node '#1'}
```

13.1.2 Definitions of *local* fields (using `\def`)

`\snk@setstartfill` `\snk@getstartfill` The setter and getter macros to store and retrieve the starting fill/draw paths (the key is the style name).

```

\snk@setstartdraw
\snk@getstartdraw
33 \snk@newfield\def\snk@setstartfill\snk@getstartfill%
34 {\snk@start@fill@#1}{Unknown sankey start fill path #1}
35
36 \snk@newfield\def\snk@setstartdraw\snk@getstartdraw%
37 {\snk@start@draw@#1}{Unknown sankey start draw path #1}
```

`\snk@setendfill` `\snk@getendfill` The setter and getter macros to store and retrieve the ending fill/draw paths (the key is the style name).

```

\snk@setenddraw
\snk@getenddraw
38 \snk@newfield\def\snk@setendfill\snk@getendfill%
39 {\snk@end@fill@#1}{Unknown sankey end fill path #1}
40
41 \snk@newfield\def\snk@setenddraw\snk@getenddraw%
42 {\snk@end@draw@#1}{Unknown sankey end draw path #1}
```

13.1.3 Check if a sankey node is defined

`\snk@ifnodedefined` The `\snk@ifnodedefined` macro checks if a Sankey node is defined by checking if its name is associated to a *quantity*.

```
43 \newcommand\snk@ifnodedefined[3]{%
44   \ifcsdef{@snk@node@qty@#1}{#2}{#3}%
45 }
```

`\snk@errorifnotdefined` The `\snk@errorifnotdefined` macro generates an error message if the Sankey node is not defined.

```
46 \newcommand\snk@errorifnotdefined[1]{%
47   \snk@ifnodedefined{#1}{}%
48   {\PackageError{sankey}{Unknown sankey node '#1'}{}}%
49 }
```

13.2 The sankey node shape

A **sankey node** is defined as a TikZ node with a particular *shape*: its width is null and its height matches the associated *quantity*. This shape requires only three anchors: **center**, **left** and **right**. These three anchors are sufficient to use the **sankey** package. But the **fit** library needs anchors defined by rectangular node.

```
50 \pgfdeclareshape{sankey node}{
51   \inheritavedanchors[from=rectangle]
52   \inheritanchor[from=rectangle]{center}
53   \inheritanchorborder[from=rectangle]
54   \anchor{left}{\pgf@process{\northeast}}
55   \anchor{right}{\pgf@process{\southwest}}
56   % compatibility with 'fit' library
57   \inheritanchor[from=rectangle]{west}
58   \inheritanchor[from=rectangle]{east}
59   \inheritanchor[from=rectangle]{north}
60   \inheritanchor[from=rectangle]{south}
61   \inheritanchor[from=rectangle]{north west}
62   \inheritanchor[from=rectangle]{south east}
63   \inheritanchor[from=rectangle]{north east}
64   \inheritanchor[from=rectangle]{south west}
65 }
```

13.3 Keys

`\sankeyset` The **sankey** package uses **pgfkeys** to set options via *key=value* pairs using the `/sankey` path (for Sankey diagram options) and using the `/sankey/node parameters` path (for Sankey node parameters).

The `\sankeyset` macro processes its parameter as a list of comma separated pairs of the form *key=value* with the default path set to `/sankey`.

```
66 \pgfkeys{/sankey/.is family}
67 \NewDocumentCommand\sankeyset{m}{\pgfkeys{sankey,#1}}
```

13.3.1 Keys to define the scale

`\snk@totalqty` The **ratio quantity**, **ratio length** and **ratio** keys define the ratio between *flow quantity* and *graphic length* (the scale). The `\snk@totalqty` and `\snk@totallen` macros store the values. All *quantities* are processed by `\fpeval` and all *lengths* are processed by `pgfmath`.

```
68 \sankeyset{
69   ratio quantity/.code={\edef\snk@totalqty{\fpeval{#1}}},
70   ratio quantity/.value required,
71   ratio length/.code={
72     \pgfmathsetmacro\snk@totallen{#1}
73     \edef\snk@totallen{\snk@totallen pt}
74   },
75   ratio length/.value required,
76   ratio/.style args={#1/#2}{
77     ratio length=#1,
78     ratio quantity=#2,
79   },
80   ratio/.value required,
81 }
```

13.3.2 Rotate offset

`\snk@rotate` The **rotate** key stores an offset angle applied to all Sankey nodes. This is useful when using the **rotate** option within a **tikzpicture** or a **scope**. This TikZ option is only applied to coordinates and not to TikZ nodes (remember that Sankey nodes are TikZ nodes). It's up to the author to keep the **rotate** option of the **tikzpicture** and that of the **sankeydiagram** synchronous.

```
82 \sankeyset{
83   rotate/.code={\edef\snk@rotate{\fpeval{#1}}},
84   rotate/.value required,
85 }
```

13.3.3 Minimum radius

`\snk@minradius` The **minimum radius** key processes the minimum radius of curvature by `pgfmath` then stores it in the `\snk@minradius` macro.

```
86 \sankeyset{
87   minimum radius/.code={
88     \pgfmathsetmacro\snk@minradius{#1}
89     \edef\snk@minradius{\snk@minradius pt}
90   },
91   minimum radius/.value required,
92 }
```

13.3.4 Outin step

`\snk@stepoutin` The **outin** key stores its value in the `\snk@stepoutin` macro.

```
93 \sankeyset{
94   outin steps/.estore in=\snk@stepoutin,
95   outon steps/.value required,
96 }
```

13.3.5 Sankey debug

The **debug** key drives the **sankey debug** toggle.

```
97 \newtoggle{sankey debug}
98 \sankeyset{
99   debug/.is choice,
100   debug/true/.code={\toggletrue{sankey debug}},
101   debug/false/.code={\togglefalse{sankey debug}},
102   debug/.default=true,
103 }
```

13.3.6 Start and end styles

`\snk@startstyle` The **start style** and **end style** keys are choices.
`\snk@endstyle`

The **new start style** and **new end style** keys add new option to these choices. They use the **startfill**, **startdraw** fields or the **endfill** and **enddraw** fields (using the style *name* as key) to store the fill and draw paths then create a new option to install the new start (`\snk@startstyle`) or end (`\snk@endstyle`) style.

```
104 \sankeyset{
105   start style/.is choice,
106   end style/.is choice,
107   % to define new start and end styles
108   new start style/.code n args={3}{% name, fill path, draw path
109     \snk@setstartfill{#1}{#2}
110     \snk@setstartdraw{#1}{#3}
111     \sankeyset{start style/#1/.code={\def\snk@startstyle{#1}}}
112   },
113   new end style/.code n args={3}{% name, fill path, draw path
114     \snk@setendfill{#1}{#2}
115     \snk@setenddraw{#1}{#3}
116     \sankeyset{end style/#1/.code={\def\snk@endstyle{#1}}}
117   },
118 }
```

13.3.7 Initial parameters

The **@initial options** style defines default values for options of Sankey diagram. The **every diagram** style (initially empty) allows the user to choose its own default values.

The **@initial options** and the **every diagram** styles are applied (in this order) at the beginning of each Sankey diagram.

```
119 \sankeyset{
120   debug color/.style={/utils/exec={\colorlet{debug color}{#1}}},
121   @initial options/.style={
122     ratio=1cm/10,
123     minimum radius=5mm,%
124     outin steps=10,
125     debug=false,
126     start style=none,
127     end style=none,
128     rotate=0,
129     % default fill/draw styles,
130     fill/.style={line width=0pt,fill=white},
131     draw/.style={draw=black,line width=.4pt},
132     % debug color used by all debug macros
133     debug color=red!75!black,
134     % debug line between left and right anchors
135     debug line/.style={overlay,draw=debug color,|-|},
136     % debug line between center and label
137     debug normal/.style={overlay,draw=debug color},
138     % debug node label
139     debug label/.style={
140       overlay,
141       draw,
142       font=\ttfamily\tiny,
143       text=debug color,text opacity=1,
144       inner sep=.1em,
145       fill=white,fill opacity=1,
146       rounded corners=.1em,
147       node contents={\name},
148     },
149     every node/.style={},
150   },
151   every diagram/.style={},
152 }
153
```

13.3.8 Sankey node parameters

The `/sankey/node parameters` family defines all parameters during creation of Sankey node.

```
154 \sankeyset{node parameters/.is family}
```

`\name` The `name`, `quantity`, `angle` and `at` keys use the `\name`, `\qty`, `\orient` and `\pos` macros to store the *name*, the *quantity*, the *orientation* (or *angle*) and the *position* of a Sankey node during its creation.

`\pos` The *quantity* is processed via `\fpeval`. The *orientation* is normalized.

```
155 \sankeyset{node parameters,
156   name/.estore in=\name,
157   name/.value required,
158   quantity/.code={\edef\qty{\fpeval{#1}}},
159   quantity/.value required,
160   angle/.code={\edef\orient{\snk@normalize@angle{#1}}},
161   angle/.value required,
162   at/.code={\snk@getpos\pos{#1}},
163   at/.value required,
164 }
```

The `as` key is just a shortcut to define the four current node parameters by copying them from an existing Sankey node.

```
165 \sankeyset{node parameters,
166   as/.style={
167     name=#1,
168     quantity=\sankeygetnodeqty{#1},
169     angle=\sankeygetnodeorient{#1},
170     at={#1.center},
171   },
172   as/.value required,
173 }
```

`\snk@anchor` The `anchor` key stores in the `\snk@anchor` macro the anchor name to use to create the new current Sankey node

```
174 \sankeyset{node parameters,
175   anchor/.is choice,
176   anchor/left/.code={\def\snk@anchor{left}},
177   anchor/right/.code={\def\snk@anchor{right}},
178   anchor/center/.code={\def\snk@anchor{center}},
179   anchor/.value required,
180 }
```

The `start` and `end` keys drive the `sankey node start` and `sankey node end` toggles.

```
181 \newtoggle{sankey node start}
182 \newtoggle{sankey node end}
183 \sankeyset{node parameters,
184   start/.is choice,
185   start/true/.code={\toggletrue{sankey node start}},
186   start/false/.code={\togglefalse{sankey node start}},
187   start/.default=true,
188   %
189   end/.is choice,
190   end/true/.code={\toggletrue{sankey node end}},
191   end/false/.code={\togglefalse{sankey node end}},
192   end/.default=true,
193 }
```


`\snk@listofforks` The `forked` and `fork anchor` keys store their value in the `\snk@listofforks` and `\snk@forkanchor` macros.

```

194 \sankeyset{node parameters,
195   forked/.estore in=\snk@listofforks,
196   forked/.value required,
197   fork anchor/.estore in=\snk@forkanchor,
198   fork anchor/.value required,
199 }

```

The `@initial parameters` style initialises all Sankey node parameters at the start of the creation of a new Sankey node.

```

200 \sankeyset{node parameters,
201   @initial parameters/.style={
202     start=false,
203     end=false,
204     forked=,
205     fork anchor=,
206     anchor=center,
207     at={0,0},
208     angle=0,
209   },
210 }

```

13.3.9 Internal Tikz style

To apply this style with Tikz, use absolute key name (`/sankey/@sankey node`). This style is used to create the Tikz node associated to a Sankey node.

```

211 \sankeyset{
212   % sankey node TikZ style
213   @sankey node/.style n args={3}{% name, pos, anchor
214     shape=sankey node,
215     inner sep=0,
216     minimum height={\sankeyqtytolen{\sankeygetnodeqty{#1}}},
217     minimum width=0,
218     draw=none,
219     line width=0pt,
220     fill=none,
221     node contents={},
222     rotate=\sankeygetnodeorient{#1}+\snk@rotate,
223     at={(#2)},
224     name=#1,
225     anchor=#3,
226   },
227 }

```

13.4 The `sankeydiagram` environment

`sankeydiagram` The `sankeydiagram` environment defines locally all the macros used by a Sankey diagram. Then it applies the `@initial options` and `every diagram` styles (in this order) and applies all the keys provided in its optional argument.

```

228 \NewDocumentEnvironment{sankeydiagram}{0}{}{

```

13.5 Internal macros

`\snk@getpos` The `\snk@getpos` macro extracts the position of a TikZ node⁶.

```
229 \def\snk@getpos##1##2{%
230   \tikz@scan@one@point\pgfutil@firstofone(##2)\relax%
231   \edef##1{\the\pgf@x,\the\pgf@y}%
232 }
```

`\snk@modulo` The `\snk@modulo` macro evaluates #1 modulo #2 using `\fpeval`.

```
233 \def\snk@modulo##1##2{\fpeval{##1-(floor((##1)/(##2),0)*##2)}}}
```

`\snk@normalize@angle` The `\snk@normalize@angle` macro normalizes #1 (an angle) between -180 and 180 (using `\fpeval`).

```
234 \def\snk@normalize@angle##1{%
235   \fpeval{\snk@modulo{(##1)+180}{360}-180}%
236 }
```

`\snk@show@debug` The `snk@show@debug` macro draws debug information of the Sankey node named #1 but only if the `sankey debug` toggle is true. Everything is drawn on the `sankeydebug` layer. It uses the `/sankey/debug line`, `/sankey/debug normal` and `/sankey/debug label` Tikz styles.

```
237 \def\snk@show@debug##1{% node name
238   \iftoggle{sankey debug}{
239     \begin{group}
240     \edef\name{##1}
241     \edef\qty{\sankeygetnodeqty{\name}}
242     \edef\orient{\sankeygetnodeorient{\name}}
243     \begin{pgfonlayer}{sankeydebug}
244       \path[/sankey/debug line] (\name.left) -- (\name.right);
245       \pgfmathsetmacro{\snk@len}{\sankeyqtytolen{\qty}/3}
246       \path[/sankey/debug normal] (\name.center)
247       -- ($(\name.center)!\snk@len pt!90:(\name.right)$)
248       node[/sankey/debug label,rotate=\orient+90+\snk@rotate,anchor=north];
249     \end{pgfonlayer}
250   \end{group}
251   }{}
252 }
```

⁶Thanks to Andrew Stacey <https://tex.stackexchange.com/a/33765/14500>

`\snk@makeforkednode` The `\snk@makeforkednode` forks a Sankey node.

```

\snk@tot
\snk@subnodeqty
\snk@subnodename
\snk@added@values
253 \def\snk@makeforkednode{
254   \begin{group}
255   \ifdefempty{\snk@listofforks}{
256     \ifdefempty{\snk@forkanchor}{
257       \PackageWarning{sankey}%
258       {Can't use 'fork anchor' key without 'forked' key}
259     }
260   }{
261     \def\snk@tot{0}
262     \def\snk@added@values{}
263     \sankeyset{
264       @add forked node/.code args={####1/####2}{
265         \coordinate (####2) at ($(\name.left)%
266         !\fpeval{(\snk@tot+.5*(####1))/\qty}%
267         !(\name.right)$);
268         \edef\snk@orient{\orient}
269         \sankeynode[debug=false]
270         {name=####2,quantity=####1,at=####2,angle=\snk@orient}
271         \edef\snk@tot{\fpeval{\snk@tot+####1}}
272         \edef\snk@added@values{\snk@added@values+####1}
273       },
274       @add forked node/.list/.expand once=\snk@listofforks,
275     }
276     \edef\snk@diff{\fpeval{abs(\qty-\snk@tot)}}
277     \ifnumequal{\snk@diff}{0}{
278       \PackageWarning{sankey}%
279       {^^J*** Warning: bad sankey fork: %
280       \qty\space!=\space\snk@added@values(=\snk@tot)%
281       ^^J\snk@listofforks}
282     }
283     \ifdefempty{\snk@forkanchor}{
284       \edef\snk@forkanchor{\name.\snk@anchor}
285     }{
286       \snk@getpos\snk@c{$(\snk@forkanchor) - (\pos)$}
287       \sankeynode{as=\name,at={$(\name) - (\snk@c)$}}
288       \foreach \snk@subnodeqty/\snk@subnodename in \snk@listofforks {
289         \sankeynode{as=\snk@subnodename,at={$(\snk@subnodename) - (\snk@c)$}}
290       }
291     }
292   \end{group}
293 }

```

`\snk@makenode` The `\snk@makenode` macro creates a new Sankey node named `\name` with `\qty` quantity, oriented at `\orient` degrees (but modified by the `\snk@rotate` angle offset), anchored by its `\anchor` (or its *center* by default) at `\pos` position.

```

294 \def\snk@makenode{
295   \begin{group}
296   \snk@setnodeqty{\name}{\qty}
297   \edef\orient{\snk@normalize@angle{\orient}}
298   \snk@setnodeorient{\name}{\orient}
299   \ifundef{\snk@anchor}{\def\snk@anchor{center}}{}
300   \node[/sankey/@sankey node={\name}{\pos}{\snk@anchor}];
301   \end{group}
302 }

```

`\snk@filldrawstart` The `\snk@filldrawstart` macro fills (with the `/sankey/fill` TikZ style) then draws (with the `/sankey/draw` TikZ style) a start of flow using paths from style `\snk@startstyle` on the Sankey node named `\name`.

```

303 \def\snk@filldrawstart{
304   \begin{scope}[shift={(\name)},rotate=\orient]
305     \path[/sankey/fill] \snk@getstartfill{\snk@startstyle};
306     \path[/sankey/draw] \snk@getstartdraw{\snk@startstyle};
307   \end{scope}
308 }

```

`\snk@filldrawend` The `\snk@filldrawend` macro fills (with the `/sankey/fill` TikZ style) then draws (with the `/sankey/draw` TikZ style) a end of flow using paths from style `\snk@endstyle` on the Sankey node named `\name`.

```

309 \def\snk@filldrawend{
310   \begin{scope}[shift={(\name)},rotate=\sankeygetnodeorient{\name}]
311     \path[/sankey/fill] \snk@endfill{\snk@endstyle};
312     \path[/sankey/draw] \snk@enddraw{\snk@endstyle};
313   \end{scope}
314 }

```

`\snk@checkquantities` The `\snk@checkquantities` compares quantities from Sankey nodes `#1` and `#2` and `\snk@qtyi` emits an error message if they differ (`#3` is the name of the macro which requested the verification).

```

315 \def\snk@checkquantities##1##2##3{
316   \begin{group}
317     \edef\snk@qtyi{\sankeygetnodeqty{##1}}
318     \edef\snk@qtyii{\sankeygetnodeqty{##2}}
319     \ifdefstrequal{\snk@qtyi}{\snk@qtyii}{%
320       \PackageError{sankey}%
321       {~J*** \string##3: quantities differ between %
322         ##1 (\snk@qtyi) and ##2 (\snk@qtyii)%
323         ~J}%
324       {The quantities of the two Sankey nodes must be equal.}
325     }
326   \end{group}
327 }

```

13.6 User macros

`\sankeydubins` The `\sankeydubins` macro links two Sankey nodes via a Dubins path. First, it computes the Dubins path between centers (left and right radii are the same) and stores the result in `sankey`. Then it uses the stored result to fill and draw the lane (left border and right borders use Dubins paths with asymmetric radii).

```

328 \NewDocumentCommand\sankeydubins{0{}}mm}{% options, sn, en
329   \snk@errorifnotdefined{##2}
330   \snk@errorifnotdefined{##3}
331   \snk@checkquantities{##2}{##3}{\sankeydubins}
332   \begin{group}
333     \sankeyset{##1}
334     \pgfmathsetmacro\qty{\sankeygetnodeqty{##2}}
335     \dubinspathset{
336       sankey/.style={
337         start point=##2.center,start angle=\sankeygetnodeorient{##2},
338         end point=##3.center,end angle=\sankeygetnodeorient{##3},
339         minimum radius=\snk@minradius + .5 * \sankeyqtytolen{\qty} pt,
340       },
341     }
342     \dubinspathcalc{sankey,store=sankey}
343     \dubinspathset{
344       left border/.style={
345         sankey, use store=sankey,
346         left and right minimum radii={\snk@minradius}
347         and {\snk@minradius + \sankeyqtytolen{\qty} pt},
348       },
349       right border/.style={
350         sankey, use store=sankey,
351         left and right minimum radii=
352         {\snk@minradius + \sankeyqtytolen{\qty} pt}
353         and {\snk@minradius},
354       },
355     }
356     % fill the region
357     \path[/sankey/fill] (##2.left) \dubinspath{left border}
358     -- (##3.left) -- (##3.right) \dubinspath{right border,reverse}
359     -- (##2.right) -- cycle;
360     % draw left and right borders
361     \path[/sankey/draw] (##2.left) \dubinspath{left border}
362     (##2.right) \dubinspath{right border};
363   \end{group}
364 }

```

`\sankeyoutin` The `\sankeyoutin` macro links two Sankey nodes via a Bézier curve. First, to simulate constant width, it creates `\snk@stepoutin` intermediate Sankey nodes along the Bézier curve. Then, the lane is filled and drawn linking all the intermediate Sankey node via smaller Bézier curves.

```

365 \NewDocumentCommand\sankeyoutin{0{mm}}{% options, sn, en
366   \snk@errorifnotdefined{##2}
367   \snk@errorifnotdefined{##3}
368   \snk@checkquantities{##2}{##3}{\sankeyoutin}
369   \begingroup
370   \sankeyset{##1}
371   \edef\qty{\sankeygetnodeqty{##2}}
372   \pgfmathsetmacro\snk@len{\sankeyqtytolen{\qty}/2}
373   \edef\snk@step{\fpeval{1/\snk@stepoutin}}
374   \edef\snk@laststep{\inteval{\snk@stepoutin-1}}
375   \path[overlay,decorate,decoration={
376     markings,
377     mark=between positions \snk@step and {\fpeval{1-.5*\snk@step}}
378     step \snk@step with {
379       \edef\snk@outinmidptname{%
380         snk@outinmidpt-~%
381         \pgfkeysvalueof{/pgf/decoration/mark info/sequence number}}%
382       }
383       \path
384         (0,0) coordinate(\snk@outinmidptname)
385         (0,-\snk@len pt) coordinate (\snk@outinmidptname-r)
386         (0,\snk@len pt) coordinate (\snk@outinmidptname-l)
387       ;
388     }
389   }]
390   (##2.center)
391   to[out=\sankeygetnodeorient{##2},in=\sankeygetnodeorient{##3}+180]
392   (##3.center);
393   \foreach \snk@ptnum in {1,...,\snk@laststep}{
394     \edef\snk@outinmidptname{snk@outinmidpt-\snk@ptnum}
395     \dbp@anglebetween\snk@outinmidptangle%
396     {\snk@outinmidptname-r}{\snk@outinmidptname-l}
397     \sankeynode[debug=false]{%
398       name=\snk@outinmidptname,
399       quantity=\qty,
400       angle=\snk@outinmidptangle-90,%
401       at=\snk@outinmidptname%
402     }
403   }
404   \sankeynode[debug=false]{%
405     name={snk@outinmidpt-0},
406     quantity=\sankeygetnodeqty{##2},
407     angle=\sankeygetnodeorient{##2},%
408     at={##2}%
409   }
410   \sankeynode[debug=false]{%
411     name={snk@outinmidpt-\snk@stepoutin},
412     quantity={\sankeygetnodeqty{##3}},
413     angle={\sankeygetnodeorient{##3}},%
414     at={##3}%
415   }
416   \path[/sankey/fill,looseness=1]
417   (snk@outinmidpt-0.left)
418   \foreach \snk@curpt
419   [remember=\snk@curpt as \snk@prevpt (initially 0)]
420   in {1,...,\snk@stepoutin}{
421     to[out=\sankeygetnodeorient{snk@outinmidpt-\snk@prevpt},
422     in=\sankeygetnodeorient{snk@outinmidpt-\snk@curpt}+180]
423     (snk@outinmidpt-\snk@curpt.left)
424   }
425   --
426   (snk@outinmidpt-\snk@stepoutin.right)
427   \foreach \snk@curpt
428   [remember=\snk@curpt as \snk@prevpt (initially \snk@stepoutin)]
429   in {\snk@laststep,...,0}{
430     to[out=\sankeygetnodeorient{snk@outinmidpt-\snk@prevpt}+180,
431     in=\sankeygetnodeorient{snk@outinmidpt-\snk@curpt}]
432     (snk@outinmidpt-\snk@curpt.right)
433   }
434   -- cycle;
435   \path[/sankey/draw,looseness=1]
436   (snk@outinmidpt-0.left)
437   \foreach \snk@curpt

```

```

438 [remember=\snk@curpt as \snk@prevpt (initially 0)]
439 in {1,...,\snk@stepoutin}{
440   to[out=\sankeygetnodeorient{snk@outinmidpt-\snk@prevpt},
441     in=\sankeygetnodeorient{snk@outinmidpt-\snk@curpt}+180]
442     (snk@outinmidpt-\snk@curpt.left)
443   }
444   (snk@outinmidpt-\snk@stepoutin.right)
445   \foreach \snk@curpt
446   [remember=\snk@curpt as \snk@prevpt (initially \snk@stepoutin)]
447   in {\snk@laststep,...,0}{
448     to[out=\sankeygetnodeorient{snk@outinmidpt-\snk@prevpt}+180,
449       in=\sankeygetnodeorient{snk@outinmidpt-\snk@curpt}]
450       (snk@outinmidpt-\snk@curpt.right)
451   };
452   \endgroup
453 }

```

`\sankeynodealias` The `\sankeynodealias` macro clones the Sankey node named **#1** into a Sankey node named **#2**.

```

454 \NewDocumentCommand\sankeynodealias{mm}{%name, alias
455   \snk@ifnotdefined{##1}
456   \path[late options={name=##1,alias=##2}];
457   \snk@setnodeqty{##2}{\sankeygetnodeqty{##1}}
458   \snk@setnodeorient{##2}{\sankeygetnodeorient{##1}}
459 }

```

`\sankeynode` The `\sankeynode` macro creates the new Sankey node named **#2**.

```

460 \NewDocumentCommand\sankeynode{0{}m}{% options, node parameters
461   \beginpgroup
462   \sankeyset{##1}
463   \sankeyset{node parameters,@initial parameters,/sankey/every node,##2}
464   \snk@makenode{}
465   \snk@makeforkednode{}
466   \iftoggle{sankey node start}{\snk@filldrawstart}{}
467   \iftoggle{sankey node end}{\snk@filldrawend}{}
468   \snk@show@debug{name}
469   \endpgroup
470 }

```

`\sankeystart` The `\sankeystart` macro fills and draws a starting lane attached to the Sankey node named **#2**.

```

471 \NewDocumentCommand\sankeystart{0{}m}{% options, name
472   \snk@ifnotdefined{##2}
473   \beginpgroup
474   \sankeyset{##1}
475   \edef\name{##2}
476   \edef\orient{\sankeygetnodeorient{##2}}
477   \edef\qty{\sankeygetnodeqty{##2}}
478   \snk@filldrawstart
479   \endpgroup
480 }

```

`\sankeynodestart` The `\sankeynodestart` macro creates the new Sankey node named **#2** then fills and draws a starting lane attached to this new Sankey node.

```

481 \NewDocumentCommand\sankeynodestart{0{}m}{% option, node parameters
482   \sankeynode[##1]{start,##2}
483 }

```

`\sankeyend` The `\sankeyend` macro fills and draws an ending lane attached to the Sankey node named `#2`.

```

484 \NewDocumentCommand\sankeyend{0}{m}{%options, name
485   \snk@ifnotdefined{##2}
486   \beginpgroup
487   \sankeyset{##1}
488   \edef\name{##2}
489   \edef\orient{\sankeygetnodeorient{##2}}
490   \edef\qty{\sankeygetnodeqty{##2}}
491   \snk@filldrawend
492   \endpgroup
493 }

```

`\sankeynodeend` The `\sankeynodeend` macro creates the new Sankey node named `#2` then fills and draws an ending lane attached to this new Sankey node.

```

494 \NewDocumentCommand\sankeynodeend{0}{m}{% options, node parameters
495   \sankeynode{##1}{end,##2}
496 }

```

`\sankeyadvance` The `\sankeyadvance` macro moves toward (or backward if *starred* calls `-#1`) the Sankey node named `#3`. `#4` is a distance. The previous position is kept by a Sankey node named `#3-old`.

```

497 \NewDocumentCommand\sankeyadvance{s0}{mm}{%
498   %params: *(reverse), options, name, distance
499   \snk@ifnotdefined{##3}
500   \beginpgroup
501   \sankeyset{##2}
502   \edef\name{##3}
503   \edef\snk@oldname{##3-old}
504   \sankeynodealias{\name}{\snk@oldname}
505   \IfBooleanTF{##1}{
506     % move backward
507     \sankeynode{
508       at={(\snk@oldname.center)!##4!90:(\snk@oldname.left)$},
509       angle=\sankeygetnodeorient{\snk@oldname},
510       quantity=\sankeygetnodeqty{\snk@oldname},
511       name=\name,
512     }
513     \path[/sankey/fill]
514       (\name.left) -- (\snk@oldname.left)
515       -- (\snk@oldname.right) -- (\name.right) -- cycle;
516     \path[/sankey/draw]
517       (\name.left) -- (\snk@oldname.left)
518       (\snk@oldname.right) -- (\name.right);
519   }{
520     % move forward
521     \sankeynode{
522       at={(\snk@oldname.center)!##4!-90:(\snk@oldname.left)$},
523       angle=\sankeygetnodeorient{\snk@oldname},
524       quantity=\sankeygetnodeqty{\snk@oldname},
525       name=\name,
526     }
527     \path[/sankey/fill]
528       (\snk@oldname.left) -- (\name.left)
529       -- (\name.right) -- (\snk@oldname.right) -- cycle;
530     \path[/sankey/draw]
531       (\snk@oldname.left) -- (\name.left)
532       (\name.right) -- (\snk@oldname.right);
533   }
534   \snk@show@debug{\name}
535   \endpgroup
536 }

```

`\sankeyturnright` The **sankeyturnright** macro moves forward the Sankey node named **#3** by turning right. The angle is **#4** (the starred version uses the opposite of **#4**). If the angle is *negative*, the macro calls the **\sankeyturnrightbackward** macro to move backward else the macro fills/draws the lane between the previous position and the new position. The previous position is kept by a Sankey node named **#3-old**.

```

537 \NewDocumentCommand\sankeyturnright{s0{}}mm){
538   % *(reverse), options, name, angle
539   \snk@errorifnotdefined{##3}
540   \beginpgroup
541   \IfBooleanTF{##1}
542   {\edef\snk@angle{\fpeval{-1*##4}}}
543   {\edef\snk@angle{\fpeval{1*##4}}}
544   \edef\snk@anglesign{\fpeval{sign(\snk@angle)}}
545   \ifnumgreater{\snk@anglesign}{-1}{
546     \sankeyset{##2}
547     \edef\name{##3}
548     \edef\snk@oldname{##3-old}
549     \sankeynodealias{\name}{\snk@oldname}
550     \edef\qty{\sankeygetnodeqty{\name}}
551     \edef\orient{\snk@normalize@angle{\sankeygetnodeorient{\name}-\snk@angle}}
552     \snk@getpos\snk@c{$(\name.right)!-\snk@minradius!(\name.left)}$}
553     \snk@getpos\pos{$(\snk@c)!1!\snk@angle:(\name.center)}$}
554     \snk@makenode{}
555     % fill the region
556     \path[/sankey/fill] let
557     \p1=(\snk@oldname.left),\p2=(\snk@oldname.right),
558     \p3=(\name.left),\p4=(\name.right),
559     \n1={\sankeyqtytolen{\qty}},
560     \n{maxr}={\snk@minradius+\n1},
561     \n{minr}={\snk@minradius}
562     in
563     (\p1) arc(\orient+\snk@angle+90:\orient+90:\n{maxr}) -- (\p3) --
564     (\p4) arc(\orient+90:\orient+\snk@angle+90:\n{minr}) -- (\p2) -- cycle;
565     % draw left and right borders
566     \path[/sankey/draw] let
567     \p1=(\snk@oldname.left),\p2=(\snk@oldname.right),
568     \p3=(\name.left),\p4=(\name.right),
569     \n1={\sankeyqtytolen{\qty}},
570     \n{maxr}={\snk@minradius+\n1},
571     \n{minr}={\snk@minradius}
572     in
573     (\p1) arc(\orient+\snk@angle+90:\orient+90:\n{maxr})
574     (\p4) arc(\orient+90:\orient+\snk@angle+90:\n{minr});
575     \snk@show@debug{\name}
576   }{
577     \sankeyturnrightbackward[##2]{##3}{-1*\snk@angle}
578   }
579   \endpgroup
580 }

```

`\sankeyturnrightbackward` The **sankeyturnrightbackward** macro moves backward the Sankey node named **#3** by turning right. The angle is **#4** (the starred version uses the opposite of **#4**). If the angle is *negative*, the macro calls the **\sankeyturnright** macro to move forward else the macro fills/draws the lane between the previous position and the new position. The previous position is kept by a Sankey node named **#3-old**.

```

581 \NewDocumentCommand\sankeyturnrightbackward{s0{}}mm){
582   % *(reverse), options, name, angle
583   \snk@errorifnotdefined{##3}
584   \beginpgroup
585   \IfBooleanTF{##1}
586   {\edef\snk@angle{\fpeval{-1*##4}}}
587   {\edef\snk@angle{\fpeval{1*##4}}}
588   \edef\snk@anglesign{\fpeval{sign(\snk@angle)}}
589   \ifnumgreater{\snk@anglesign}{-1}{
590     \sankeyset{##2}
591     \edef\name{##3}
592     \edef\snk@oldname{##3-old}
593     \sankeynodealias{\name}{\snk@oldname}
594     \edef\qty{\sankeygetnodeqty{\name}}
595     \edef\orient{\snk@normalize@angle{\sankeygetnodeorient{\name}+\snk@angle}}
596     \snk@getpos\snk@c{$(\name.right)!-\snk@minradius!(\name.left)}$}
597     \snk@getpos\pos{$(\snk@c)!1!\snk@angle:(\name.center)}$}
598     \snk@makenode{}
599     % fill the region

```



```

600 \path[/sankey/fill] let
601 \p1=(\name.left),\p2=(\name.right),
602 \p3=(\snk@oldname.left),\p4=(\snk@oldname.right),
603 \n1={\sankeyqtytolen{\qty}},
604 \n{maxr}={\snk@minradius+\n1}, \n{minr}={\snk@minradius}
605 in
606 (\p1) arc(\orient+90:\orient-\snk@angle+90:\n{maxr}) -- (\p3) --
607 (\p4) arc(\orient-\snk@angle+90:\orient+90:\n{minr}) -- (\p2) -- cycle;
608 % draw left and right borders
609 \path[/sankey/draw] let
610 \p1=(\name.left),\p2=(\name.right),
611 \p3=(\snk@oldname.left),\p4=(\snk@oldname.right),
612 \n1={\sankeyqtytolen{\qty}},
613 \n{maxr}={\snk@minradius+\n1}, \n{minr}={\snk@minradius}
614 in
615 (\p1) arc(\orient+90:\orient-\snk@angle+90:\n{maxr})
616 (\p4) arc(\orient-\snk@angle+90:\orient+90:\n{minr});
617 \snk@show@debug{\name}
618 }{
619 \sankeyturnright[##2]{##3}{-1*\snk@angle}
620 }
621 \endgroup
622 }

```

`\sankeyturnleft` The `sankeyturnleft` macro moves forward the Sankey node named `#3` by turning left. The angle is `#4` (the starred version uses the opposite of `#4`). If the angle is *negative*, the macro calls the `\sankeyturnleftbackward` macro to move backward else the macro fills/draws the lane between the previous position and the new position. The previous position is kept by a Sankey node named `#3-old`.

```

623 \NewDocumentCommand\sankeyturnleft{s0{}}mm){
624 % *(reverse), options, name, angle
625 \snk@errorifnotdefined{##3}
626 \begingroup
627 \IfBooleanTF{##1}
628 {\edef\snk@angle{\fpeval{-1*##4}}}
629 {\edef\snk@angle{\fpeval{1*##4}}}
630 \edef\snk@anglesign{\fpeval{sign(\snk@angle)}}
631 \ifnumgreater{\snk@anglesign}{-1}{
632 \sankeyset{##2}
633 \edef\name{##3}
634 \edef\snk@oldname{##3-old}
635 \sankeynodealias{\name}{\snk@oldname}
636 \edef\qty{\sankeygetnodeqty{\name}}
637 \edef\orient{\snk@normalize@angle{\sankeygetnodeorient{\name}+\snk@angle}}
638 \snk@getpos\snk@c{$(\name.left)!-\snk@minradius!(\name.right)$}
639 \snk@getpos\pos{$(\snk@c)!1!\snk@angle:(\name.center)$}
640 \snk@makenode{}
641 % fill the region
642 \path[/sankey/fill] let
643 \p1=(\snk@oldname.left),\p2=(\snk@oldname.right),
644 \p3=(\name.left),\p4=(\name.right),
645 \n1={\sankeyqtytolen{\qty}},
646 \n{maxr}={\snk@minradius+\n1}, \n{minr}={\snk@minradius}
647 in
648 (\p1) arc(\orient-\snk@angle-90:\orient-90:\n{minr}) -- (\p3) --
649 (\p4) arc(\orient-90:\orient-\snk@angle-90:\n{maxr}) -- (\p2) -- cycle;
650 % draw left and right borders
651 \path[/sankey/draw] let
652 \p1=(\snk@oldname.left),\p2=(\snk@oldname.right),
653 \p3=(\name.left),\p4=(\name.right),
654 \n1={\sankeyqtytolen{\qty}},
655 \n{maxr}={\snk@minradius+\n1}, \n{minr}={\snk@minradius}
656 in
657 (\p1) arc(\orient-\snk@angle-90:\orient-90:\n{minr})
658 (\p4) arc(\orient-90:\orient-\snk@angle-90:\n{maxr});
659 \snk@show@debug{\name}
660 }{
661 \sankeyturnleftbackward[##2]{##3}{-1*\snk@angle}
662 }
663 \endgroup
664 }

```

`\sankeyturnleftbackward` The `\sankeyturnleftbackward` macro moves backward the Sankey node named `#3` by turning left. The angle is `#4` (the starred version uses the opposite of `#4`). If the angle is *negative*, the macro calls the `\sankeyturnleft` macro to move forward else the macro fills/draws the lane between the previous position and the new position. The previous position is kept by a Sankey node named `#3-old`.

```

665 \NewDocumentCommand\sankeyturnleftbackward{s0{}}mm){
666   % *(reverse), options, name, angle
667   \snk@errorifnotdefined{##3}
668   \beginngroup
669   \IfBooleanTF{##1}{
670     {\edef\snk@angle{\fpeval{-1*##4}}}
671     {\edef\snk@angle{\fpeval{1*##4}}}
672     \edef\snk@anglesign{\fpeval{sign(\snk@angle)}}
673     \ifnumgreater{\snk@anglesign}{-1}{
674       \sankeyset{##2}
675       \edef\name{##3}
676       \edef\snk@oldname{##3-old}
677       \sankeynodealias{\name}{\snk@oldname}
678       \edef\qty{\sankeygetnodeqty{\name}}
679       \edef\orient{\snk@normalize@angle{\sankeygetnodeorient{\name}-\snk@angle}}
680       \snk@getpos\snk@c{$(\name.left)!-\snk@minradius!(\name.right)$}
681       \snk@getpos\pos{$(\snk@c)!1!-\snk@angle:(\name.center)$}
682       \snk@makenode{}
683       % fill the region
684       \path[/sankey/fill] let
685         \p1=(\name.left),\p2=(\name.right),
686         \p3=(\snk@oldname.left),\p4=(\snk@oldname.right),
687         \n1={\sankeyqtytolen{\qty}},
688         \n{maxr}={\snk@minradius+\n1}, \n{minr}={\snk@minradius}
689       in
690         (\p1) arc(\orient-90:\orient+\snk@angle-90:\n{minr}) -- (\p3) --
691         (\p4) arc(\orient+\snk@angle-90:\orient-90:\n{maxr}) -- (\p2) -- cycle;
692       % draw left and right borders
693       \path[/sankey/draw] let
694         \p1=(\name.left),\p2=(\name.right),
695         \p3=(\snk@oldname.left),\p4=(\snk@oldname.right),
696         \n1={\sankeyqtytolen{\qty}},
697         \n{maxr}={\snk@minradius+\n1}, \n{minr}={\snk@minradius}
698       in
699         (\p1) arc(\orient-90:\orient+\snk@angle-90:\n{minr})
700         (\p4) arc(\orient+\snk@angle-90:\orient-90:\n{maxr});
701       \snk@showdebug{\name}
702     }{
703       \sankeyturnleft[##2]{##3}{-1*\snk@angle}
704     }
705   \endgroup
706 }

```

`\sankeyturn` The `\sankeyturn` macro moves toward (or backward if *starred* calls `- #1`) the Sankey node named `#3` by turning left (angle `#4` is positive) or right (angle `#4` is negative). The previous position is kept by a Sankey node named `#3-old`.

```

707 \NewDocumentCommand\sankeyturn{s0{}}mm){%
708   % *(reverse), options, name, angle
709   \snk@errorifnotdefined{##3}
710   \beginngroup
711   \edef\snk@anglesign{\fpeval{sign(##4)}}
712   \IfBooleanTF{##1}{
713     \ifnumgreater{\snk@anglesign}{-1}
714       {\sankeyturnleftbackward[##2]{##3}{##4}}
715       {\sankeyturnrightbackward[##2]{##3}{-1*##4}}
716     }{
717       \ifnumgreater{\snk@anglesign}{-1}
718       {\sankeyturnleft[##2]{##3}{##4}}
719       {\sankeyturnright[##2]{##3}{-1*##4}}
720     }
721   \endgroup
722 }

```

`\sankeyfork` The `\sankeyfork` macro forks the Sankey node named `#2` to the list of subnodes given by `#3`. The subnodes are cloned to take into account the `debug` option.

```

723 \NewDocumentCommand\sankeyfork{0{}mm}{%options, name, list of forks
724   \snk@errorifnotdefined{##2}
725   \beginngroup
726   \sankeyset{##1}
727   \sankeynode[debug=false]{as=##2,forked=##3}
728   \foreach \qty/\snk@subnodename in {##3}{\sankeynode{as={\snk@subnodename}}}
729   \endgroup
730 }
```

`\sankeyqtytolen` The `\sankeyqtytolen` macro converts quantity to length using `\fpeval` and the ratio determined by `\snk@totalqty` and `sankeytotalen`.

```

731 \def\sankeyqtytolen##1{\fpeval{##1)/\snk@totalqty*\snk@totalen}}
```

13.7 The `sankeydiagram` environment (the end)

Here is the end of the definition of the `sankeydiagram` environment. The `@initial options` style and the `every diagram` style are applied before options provided by user.

```

732 \sankeyset{
733   @initial options,
734   every diagram,
735   % user values
736   #1}
737 }
738 {} % empty but mandatory ! :-)
```

13.8 Predefined start and end styles

The `none` style.

```
739 \sankeyset{
740   new start style={none}{-}{-},
741   new end style={none}{-}{-},
742 }
```

The `simple` style.

```
743 \sankeyset{
744   new start style={simple}{
745     (\name.left) -- ([xshift=-.5\pgflinewidth]\name.left)
746     -- ([xshift=-.5\pgflinewidth]\name.right) -- (\name.right) -- cycle
747   },
748   (\name.left) -- ([xshift=-.5\pgflinewidth]\name.left)
749   -- ([xshift=-.5\pgflinewidth]\name.right) -- (\name.right)
750 },
751   new end style={simple}{
752     (\name.left) -- ([xshift=2mm]\name.center)
753     -- (\name.right) -- cycle
754   },
755   (\name.left) -- ([xshift=2mm]\name.center) -- (\name.right)
756 },
757 }
```

The `arrow` style.

```
758 \sankeyset{
759   new start style={arrow}{
760     (\name.left) -- ++(-10pt,0) -- ([xshift=-10pt/6]\name.center)
761     -- ([xshift=-10pt]\name.right) -- (\name.right) -- cycle
762   },
763   (\name.left) -- ++(-10pt,0) -- ([xshift=-10pt/6]\name.center)
764   -- ([xshift=-10pt]\name.right) -- (\name.right)
765 },
766   new end style={arrow}{
767     (\name.left) -- ([yshift=1mm]\name.left)
768     -- ([xshift=10pt]\name.center) -- ([yshift=-1mm]\name.right)
769     -- (\name.right) -- cycle
770   },
771   (\name.left) -- ([yshift=1mm]\name.left)
772   -- ([xshift=10pt]\name.center) -- ([yshift=-1mm]\name.right)
773   -- (\name.right)
774 },
775 }
```

14 tikzlibrarydubins.code.tex

Not yet documented nor commented...

```

776 \def\tikzlibrarydubins@version{v3.0}
777 \def\tikzlibrarydubins@date{2021/03/14}
778
779 \usetikzlibrary{calc}
780 \RequirePackage{etoolbox}
781 \RequirePackage{xfp}
782
783 \newbool{dubinspathreverse}
784
785 \def\ifpgfmathcond#1{%
786   \pgfmathparse{(#1)?1:0}%
787   \ifnumequal{\pgfmathresult}{1}%
788 }
789
790
791 \def\dbp@getxy#1#2#3{%
792   \tikz@scan@one@point\pgfutil@firstofone(#3)\relax%
793   \edef#1{\the\pgf@x}%
794   \edef#2{\the\pgf@y}%
795 }
796
797 \def\dbp@anglebetween#1#2#3{%macro, s, t
798   \dbp@getxy\dbp@ax\dbp@ay{#2}
799   \dbp@getxy\dbp@bx\dbp@by{#3}
800   \pgfmathsetmacro#1{atan2(\dbp@by-\dbp@ay,\dbp@bx-\dbp@ax)}
801 }
802
803 \def\dbp@distancebetween#1#2#3{%macro, s, t
804   \dbp@getxy\dbp@ax\dbp@ay{#2}
805   \dbp@getxy\dbp@bx\dbp@by{#3}
806   \edef#1{\fpeval{sqrt(
807     (\dbp@bx-\dbp@ax)*(\dbp@bx-\dbp@ax)%
808     +(\dbp@by-\dbp@ay)*(\dbp@by-\dbp@ay)%
809   )}}%
810 }
811
812 \newcommand\dbp@rsr{% s, sa, t, ta, as, len, at, r
813   let
814     \p{tr}=(\pgfmathanglebetween{\dbp@angb-90:\dbp@radius pt})\dbp@b),
815     \n1={\dbp@anga+90},
816     \n2={\dbp@angb+90},
817     \n3={\n2+\dbp@lastangle}
818   in
819     arc(\n1:\n1-\dbp@firstangle:\dbp@rradius pt)
820     -- ([shift={(\p{tr})}]\n3:\dbp@rradius pt)
821     arc(\n3:\n2:\dbp@rradius pt)
822 }
823
824 \newcommand\dbp@lsl{% s, sa, t, ta, as, len, at, r
825   let
826     \p{tl}=(\pgfmathanglebetween{\dbp@angb+90:\dbp@radius pt})\dbp@b),
827     \n1={\dbp@anga-90},\n2={\n1+\dbp@firstangle},
828     \n3={\dbp@angb-90},\n4={\n3-\dbp@lastangle}
829   in
830     arc(\n1:\n2:\dbp@lradius pt)
831     -- ([shift={(\p{tl})}]\n4:\dbp@lradius pt)
832     arc(\n4:\n3:\dbp@lradius pt)
833 }
834
835 \newcommand\dbp@rs1{% s, sa, t, ta, as, len, at, r
836   let
837     \p{tl}=(\pgfmathanglebetween{\dbp@angb+90:\dbp@radius pt})\dbp@b),
838     \n1={\dbp@anga+90},\n2={\n1-\dbp@firstangle},
839     \n3={\dbp@angb-90},\n4={\n3-\dbp@lastangle}
840   in
841     arc(\n1:\n2:\dbp@rradius pt)
842     -- ([shift={(\p{tl})}]\n4:\dbp@lradius pt)
843     arc(\n4:\n3:\dbp@lradius pt)
844 }
845
846 \newcommand\dbp@lsr{% s, sa, t, ta, as, len, at, r
847   let
848     \p{tr}=(\pgfmathanglebetween{\dbp@angb-90:\dbp@radius pt})\dbp@b),
849     \n1={\dbp@anga-90},\n2={\n1+\dbp@firstangle},
850     \n3={\dbp@angb+90},\n4={\n3+\dbp@lastangle}

```

```

851 in
852 arc(\n1:\n2:\dbp@lradius pt)
853 -- ([shift={(\p{tr})}]\n4:\dbp@rradius pt)
854 arc(\n4:\n3:\dbp@rradius pt)
855 }
856
857 \newcommand\dbp@lrl{% s, sa, t, ta, as, ai, at, r
858 let
859 \n1={\dbp@anga-90},\n2={\n1+\dbp@firstangle},
860 \n3={\dbp@angb-90},\n4={\n3-\dbp@lastangle}
861 in
862 arc(\n1:\n2:\dbp@lradius pt)
863 arc(\n2+180:\n2+180-\dbp@midparam:\dbp@rradius pt)
864 arc(\n4:\n3:\dbp@lradius pt)
865 }
866
867 \newcommand\dbp@rlr{% s, sa, t, ta, as, ai, at, r
868 let
869 \n1={\dbp@anga+90},\n2={\n1-\dbp@firstangle},
870 \n3={\dbp@angb+90},\n4={\n3+\dbp@lastangle}
871 in
872 arc(\n1:\n2:\dbp@rradius pt)
873 arc(\n2+180:\n2+180+\dbp@midparam:\dbp@lradius pt)
874 arc(\n4:\n3:\dbp@rradius pt)
875 }
876 \newcommand\dbp@rev@lsl{\dbp@rsr}
877 \newcommand\dbp@rev@rsr{\dbp@lsl}
878 \newcommand\dbp@rev@lsr{\dbp@lsr}
879 \newcommand\dbp@rev@rs1{\dbp@rs1}
880 \newcommand\dbp@rev@lrl{\dbp@rlr}
881 \newcommand\dbp@rev@rlr{\dbp@lrl}
882
883
884 \newcommand\dubinspath[1]{%
885 \pgfextra{
886 \dubinspathset{#1}
887 \ifbool{dubinspathreverse}{
888 \edef\dbp@newa{\dbp@b}
889 \edef\dbp@newb{\dbp@a}
890 \pgfmathsetmacro\dbp@newanga{180+\dbp@angb}
891 \pgfmathsetmacro\dbp@newangb{180+\dbp@anga}
892 \edef\dbp@newfirstangle{\dbp@lastangle}
893 \edef\dbp@newlastangle{\dbp@firstangle}
894 \edef\dbp@newmethod{rev@\dbp@method}
895 \edef\dbp@newlradius{\dbp@rradius}
896 \edef\dbp@newrradius{\dbp@lradius}
897 \dubinspathset{
898 start point=\dbp@newa,
899 end point=\dbp@newb,
900 start angle=\dbp@newanga,
901 end angle=\dbp@newangb,
902 first angle=\dbp@newfirstangle,
903 last angle=\dbp@newlastangle,
904 left and right minimum radii=\dbp@newlradius pt and \dbp@newrradius pt,
905 method=\dbp@newmethod,
906 }
907 }{}
908 }
909 \csname dbp@\dbp@method\endcsname%
910 }
911
912
913 %% solution
914
915 \def\dbp@store#1#2{%
916 \expandafter\xdef\csname dbp@store@#1@#2\endcsname%
917 {\csname dbp@#2\endcsname}%
918 }
919 \def\dbp@get#1#2{%
920 \csname dbp@store@#1@#2\endcsname%
921 }
922
923 \def\dbp@setparams#1#2#3#4#5{%
924 % method, length, first angle, middle param, last angle
925 \edef\dbp@method{#1}
926 \edef\dbp@length{#2}
927 \edef\dbp@firstangle{#3}
928 \edef\dbp@middleparam{#4}
929 \edef\dbp@lastangle{#5}

```

```

930 \ifdef{\dbp@storename}{
931   \foreach \p in {method,length,firstangle,middleparam,lastangle}{
932     \dbp@store{\dbp@storename}{\p}
933   }
934 }{}
935 }
936
937 \def\dbp@updateparams#1#2#3#4#5{
938   \ifpgfmathcond{#2<\dbp@length}{
939     \dbp@setparams{#1}{#2}{#3}{#4}{#5}
940   }{}
941 }
942
943 \tikzset{
944   dubins path/.is family,
945   dubins path,
946   start point/.store in=\dbp@a,
947   start angle/.store in=\dbp@anga,
948   end point/.store in=\dbp@b,
949   end angle/.store in=\dbp@angb,
950   store/.store in=\dbp@storename,
951   use store/.style={
952     method=\dbp@get{#1}{method},
953     first angle=\dbp@get{#1}{firstangle},
954     last angle=\dbp@get{#1}{lastangle},
955     middle param=\dbp@get{#1}{middleparam},
956   },
957   minimum radius/.code={
958     \pgfmathsetmacro\dbp@radius{#1}
959     \pgfmathsetmacro\dbp@rradius{#1}
960     \pgfmathsetmacro\dbp@lradius{#1}
961   },
962   left and right minimum radii/.code args={#1 and #2}{
963     \pgfmathsetmacro\dbp@lradius{#1}
964     \pgfmathsetmacro\dbp@rradius{#2}
965     \pgfmathsetmacro\dbp@radius{(\dbp@lradius + \dbp@rradius)/2}
966   },
967   method/.store in=\dbp@method,
968   first angle/.store in=\dbp@firstangle,
969   last angle/.store in=\dbp@lastangle,
970   middle param/.store in=\dbp@midparam,
971   reverse/.is if=dubinspathreverse,
972 }
973
974 \newcommand\dubinspathset[1]{\tikzset{dubins path,#1}}
975
976 \newcommand\dubinspathcalc[1]{%
977   \begingroup
978   \dubinspathset{#1}
979   \tikzset{
980     declare function={
981       angtodist(\dbp@a,\dbp@r)={abs(\dbp@a)*.01745329*\dbp@r};
982       modangr(\dbp@a,\dbp@b)={
983         (
984           Mod(\dbp@a,360)<Mod(\dbp@b,360)
985           ?
986           Mod(\dbp@a,360)
987           :
988           Mod(\dbp@a,360)-360+\dbp@b-Mod(\dbp@b,360)
989         )
990       };
991       modangl(\dbp@a,\dbp@b)={
992         (
993           Mod(\dbp@a,360)<Mod(\dbp@b,360)
994           ?
995           Mod(\dbp@a,360)+360
996           :
997           Mod(\dbp@a,360)+(\dbp@b)-Mod(\dbp@b,360)
998         );
999       },
1000   }
1001
1002   \pgfmathsetmacro\dbp@radius{\dbp@rradius}
1003   \pgfmathsetmacro\dbp@anga{mod((\dbp@anga)+180,360)-180}
1004   \pgfmathsetmacro\dbp@angb{mod((\dbp@angb)+180,360)-180}
1005   \path
1006   let
1007   \p{a}=(\dbp@a),
1008   \p{b}=(\dbp@b),

```

```

1009 \p{ar}=( $\angle p{a}$ ) + ( $\text{dbp@anga}-90:\text{dbp@radius pt}$ ),
1010 \p{al}=( $\angle p{a}$ ) + ( $\text{dbp@anga}+90:\text{dbp@radius pt}$ ),
1011 \p{br}=( $\angle p{b}$ ) + ( $\text{dbp@angb}-90:\text{dbp@radius pt}$ ),
1012 \p{bl}=( $\angle p{b}$ ) + ( $\text{dbp@angb}+90:\text{dbp@radius pt}$ )
1013 in \pgfextra{
1014   \pgfinterruptpath
1015
1016   % RSR (ar and br)
1017   \dbp@anglebetween\dbp@rsrarbr{\p{ar}}{\p{br}}
1018   \dbp@distancebetween\dbp@rsrdarbr{\p{ar}}{\p{br}}
1019   \pgfmathsetmacro\dbp@rsrangone{Mod(\dbp@anga-\dbp@rsrarbr,360)}
1020   \pgfmathsetmacro\dbp@rsrangtwo{Mod(\dbp@rsrarbr-\dbp@angb,360)}
1021   \pgfmathsetmacro\dbp@rsrlen{\dbp@rsrdarbr}
1022   \pgfmathsetmacro\dbp@rsrdist{
1023     angtodist(\dbp@rsrangone,\dbp@radius)
1024     +\dbp@rsrlen
1025     +angtodist(\dbp@rsrangtwo,\dbp@radius)
1026   }
1027   \dbp@setparams{rsr}{\dbp@rsrdist}{\dbp@rsrangone}{\dbp@rsrlen}{\dbp@rsrangtwo}
1028
1029   % LSL (al and bl)
1030   \dbp@anglebetween\dbp@lslalbl{\p{al}}{\p{bl}}
1031   \dbp@distancebetween\dbp@lsldalbl{\p{al}}{\p{bl}}
1032   \pgfmathsetmacro\dbp@lslangone{mod(\dbp@lslalbl-\dbp@anga+720,360)}
1033   \pgfmathsetmacro\dbp@lslangtwo{mod(\dbp@angb-\dbp@lslalbl+720,360)}
1034   \pgfmathsetmacro\dbp@lsllen{\dbp@lsldalbl}
1035   \pgfmathsetmacro\dbp@lsldist{
1036     angtodist(\dbp@lslangone,\dbp@radius)
1037     +\dbp@lsllen
1038     +angtodist(\dbp@lslangtwo,\dbp@radius)
1039   }
1040   \dbp@updateparams%
1041   {ls1}{\dbp@lsldist}{\dbp@lslangone}{\dbp@lsllen}{\dbp@lslangtwo}
1042
1043   % RSL (ar and bl)
1044   \dbp@distancebetween\dbp@rsldarbl{\p{ar}}{\p{bl}}
1045   \pgfmathtruncatemacro\dbp@rslok{(\dbp@rsldarbl>=2*\dbp@radius)?1:0}
1046   \ifnumequal{\dbp@rslok}{1}{
1047     \dbp@anglebetween\dbp@rsrarbl{\p{ar}}{\p{bl}}
1048     \pgfmathsetmacro\dbp@rsranglesup{
1049       asin(\dbp@radius/\dbp@rsldarbl*2)}
1050     \pgfmathsetmacro\dbp@rsrangone
1051     {Mod(\dbp@anga-\dbp@rsrarbl+\dbp@rsranglesup,360)}
1052     \pgfmathsetmacro\dbp@rsrangtwo
1053     {Mod(\dbp@angb-\dbp@rsrarbl+\dbp@rsranglesup,360)}
1054     \pgfmathsetmacro\dbp@rsllen{vecLen(\dbp@rsldarbl,\dbp@radius)}
1055     \pgfmathsetmacro\dbp@rsldist{
1056       angtodist(\dbp@rsrangone,\dbp@radius)
1057       +\dbp@rsllen
1058       +angtodist(\dbp@rsrangtwo,\dbp@radius)
1059     }
1060
1061     \dbp@updateparams%
1062     {rs1}{\dbp@rsldist}{\dbp@rsrangone}{\dbp@rsllen}{\dbp@rsrangtwo}
1063   }{}
1064
1065   % LSR (al and br)
1066   \dbp@distancebetween\dbp@lsrdalbr{\p{al}}{\p{br}}
1067   \pgfmathtruncatemacro\dbp@lsrok{(\dbp@lsrdalbr>=2*\dbp@radius)?1:0}
1068   \ifnumequal{\dbp@lsrok}{1}{
1069     \dbp@anglebetween\dbp@lsralbr{\p{al}}{\p{br}}
1070     \pgfmathsetmacro\dbp@lsranglesup{
1071       asin(\dbp@radius/\dbp@lsrdalbr*2)}
1072     \pgfmathsetmacro\dbp@lsrangone
1073     {Mod(\dbp@lsralbr+\dbp@lsranglesup-\dbp@anga,360)}
1074     \pgfmathsetmacro\dbp@lsrangtwo
1075     {Mod(\dbp@lsralbr+\dbp@lsranglesup-\dbp@angb,360)}
1076     \pgfmathsetmacro\dbp@lsrlen{vecLen(\dbp@lsrdalbr,\dbp@radius)}
1077     \pgfmathsetmacro\dbp@lsrdist{
1078       angtodist(\dbp@lsrangone,\dbp@radius)
1079       +\dbp@lsrlen
1080       +angtodist(\dbp@lsrangtwo,\dbp@radius)
1081     }
1082     \dbp@updateparams%
1083     {lsr}{\dbp@lsrdist}{\dbp@lsrangone}{\dbp@lsrlen}{\dbp@lsrangtwo}
1084   }{}
1085
1086   % LRL (al and bl)
1087   \dbp@distancebetween\dbp@lrlalbl{\p{al}}{\p{bl}}

```



```

1088 \pgfmathtruncatemacro\dbp@lrlrok{(\dbp@lrlalbl<=4*\dbp@radius)?1:0}
1089 \ifnumequal{\dbp@lrlrok}{1}{
1090   \dbp@anglebetween\dbp@lrlalbl{\p{a}}{\p{b}}
1091   \pgfmathsetmacro\dbp@lrlangsup{acos(\dbp@lrlalbl/\dbp@radius/4)}
1092   \pgfmathsetmacro\dbp@lrlangone{
1093     modangl(\dbp@lrlalbl+\dbp@lrlangsup,\dbp@anga-90)-(\dbp@anga-90)}
1094   \pgfmathsetmacro\dbp@lrlangtwo{
1095     (\dbp@angb-90)-modangr(\dbp@lrlalbl+180-\dbp@lrlangsup,\dbp@angb-90)}
1096   \pgfmathsetmacro\dbp@lrlangthree{360-2*(90-\dbp@lrlangsup)}
1097   \pgfmathsetmacro\dbp@lrlldist{
1098     angtodist(\dbp@lrlangone,\dbp@radius)
1099     +angtodist(\dbp@lrlangthree,\dbp@radius)
1100     +angtodist(\dbp@lrlangtwo,\dbp@radius)
1101   }
1102   \dbp@updateparams%
1103   {lrl}{\dbp@lrlldist}{\dbp@lrlangone}{\dbp@lrlangthree}{\dbp@lrlangtwo}
1104 }{}
1105
1106 % RLR (ar and br)
1107 \dbp@distancebetween\dbp@rlrdarbr{\p{ar}}{\p{br}}
1108 \pgfmathtruncatemacro\dbp@lrlrok{(\dbp@rlrdarbr<=4*\dbp@radius)?1:0}
1109 \ifnumequal{\dbp@lrlrok}{1}{
1110   \dbp@anglebetween\dbp@rlrarbr{\p{ar}}{\p{br}}
1111   \pgfmathsetmacro\dbp@lrlangsup{acos(\dbp@rlrdarbr/\dbp@radius/4)}
1112   \pgfmathsetmacro\dbp@lrlangone{
1113     (\dbp@anga+90)-modangr(\dbp@rlrarbr-\dbp@lrlangsup,\dbp@anga+90)}
1114   \pgfmathsetmacro\dbp@lrlangtwo{
1115     modangl(\dbp@rlrarbr+180+\dbp@lrlangsup,\dbp@angb+90)-(\dbp@angb+90)}
1116   \pgfmathsetmacro\dbp@lrlangthree{360-2*(90-\dbp@lrlangsup)}
1117   \pgfmathsetmacro\dbp@lrlldist{
1118     angtodist(\dbp@lrlangone,\dbp@radius)
1119     +angtodist(\dbp@lrlangthree,\dbp@radius)
1120     +angtodist(\dbp@lrlangtwo,\dbp@radius)
1121   }
1122   \dbp@updateparams%
1123   {rlr}{\dbp@lrlldist}{\dbp@lrlangone}{\dbp@lrlangthree}{\dbp@lrlangtwo}
1124 }{}
1125
1126 \endpgfinterruptpath
1127 };
1128 \endgroup
1129 }

```

15 Change History

v1.0		\sankeyturnright*: new macro ...	56
General: first version	43	\sankeyturnrightbackward*: new macro	56
v2.0		General: add keys to fork a Sankey node during its creation	19
General: first public version (on CTAN)	43	add the reproduction of an example from Google Charts	
v3.0		\sankeyturn*: simplification by using new <i>turn</i> macros	58
\sankeyturnleft*: new macro	57	documentation	32
\sankeyturnleftbackward*: new macro	58	better naming rule	43
		fix bad names in second example	28
		use .ins and .dtx files	43

16 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.

D	870, 890, 949, 1004, 1011, 1012, 1020, 1033, 1053, 1075, 1095, 1115
\dbp@a	889, 946, 981, 982, 984, 986, 988, 991, 993, 995, 997, 1007
\dbp@anga	815, 827, 838, 849, 859, 869, 891, 947, 1003, 1009, 1010, 1019, 1032, 1051, 1073, 1093, 1113
\dbp@angb	814, 816, 826, 828, 837, 839, 848, 850, 860,
\dbp@anglebetween	395, 797, 1017, 1030, 1047, 1069, 1090, 1110
\dbp@ax	798, 800, 804, 807
\dbp@ay	798, 800, 804, 808
\dbp@b	814, 826, 837, 848, 888, 948, 982, 984, 988, 991, 993, 997, 1008

\dbp@bx	799, 800, 805, 807	\dbp@rlrdarbr	1107, 1108, 1111
\dbp@by	799, 800, 805, 808	\dbp@rlrdist	1117, 1123
\dbp@distancebetween	803, 1018, 1031, 1044, 1066, 1087, 1107	\dbp@rlrok	1108, 1109
\dbp@firstangle	819, 827, 838, 849, 859, 869, 893, 927, 968	\dbp@rradius	819, 820, 821, 841, 853, 854, 863, 872, 874, 895, 959, 964, 965, 1002
\dbp@get	919, 952, 953, 954, 955	\dbp@rsl	835, 879
\dbp@getxy	791, 798, 799, 804, 805	\dbp@rslanglesup	1048, 1051, 1053
\dbp@lastangle	817, 828, 839, 850, 860, 870, 892, 929, 969	\dbp@rslangone	1050, 1056, 1062
\dbp@length	926, 938	\dbp@rslangtwo	1052, 1058, 1062
\dbp@lradius	830, 831, 832, 842, 843, 852, 862, 864, 873, 896, 960, 963, 965	\dbp@rslarbl	1047, 1051, 1053
\dbp@lrl	857, 881	\dbp@rslarbl	1044, 1045, 1049, 1054
\dbp@lrlalbl	1090, 1093, 1095	\dbp@rslldist	1055, 1062
\dbp@lrlangone	1092, 1098, 1103	\dbp@rslllen	1054, 1057, 1062
\dbp@lrlangsup	1091, 1093, 1095, 1096	\dbp@rsllok	1045, 1046
\dbp@lrlangthree	1096, 1099, 1103	\dbp@rsr	812, 876
\dbp@lrlangtwo	1094, 1100, 1103	\dbp@rsrangone	1019, 1023, 1027
\dbp@lrlldalbl	1087, 1088, 1091	\dbp@rsrangtwo	1020, 1025, 1027
\dbp@lrlldist	1097, 1103	\dbp@rsrarbr	1017, 1019, 1020
\dbp@lrllok	1088, 1089	\dbp@rsrdarbr	1018, 1021
\dbp@lsl	824, 877	\dbp@rsrdist	1022, 1027
\dbp@lslalbl	1030, 1032, 1033	\dbp@rsrlen	1021, 1024, 1027
\dbp@lslangone	1032, 1036, 1041	\dbp@setparams	923, 939, 1027
\dbp@lslangtwo	1033, 1038, 1041	\dbp@store	915, 932
\dbp@lslldalbl	1031, 1034	\dbp@storename	930, 932, 950
\dbp@lslldist	1035, 1041	\dbp@updateparams	937, 1040, 1061, 1082, 1102, 1122
\dbp@lslllen	1034, 1037, 1041	\dubinspath	357, 358, 361, 362, 884
\dbp@lsr	846, 878	\dubinspathcalc	342, 976
\dbp@lsralbr	1069, 1073, 1075	\dubinspathset	335, 343, 886, 897, 974, 978
\dbp@lsranglesup	1070, 1073, 1075		
\dbp@lsrangone	1072, 1078, 1083		
\dbp@lsrangtwo	1074, 1080, 1083		
\dbp@lsrdalbr	1066, 1067, 1071, 1076		
\dbp@lsrdist	1077, 1083		
\dbp@lsrlen	1076, 1079, 1083		
\dbp@lsrok	1067, 1068		
\dbp@method	894, 909, 925, 967		
\dbp@middleparam	928		
\dbp@midparam	863, 873, 970		
\dbp@newa	888, 898		
\dbp@newanga	890, 900		
\dbp@newangb	891, 901		
\dbp@newb	889, 899		
\dbp@newfirstangle	892, 902		
\dbp@newlastangle	893, 903		
\dbp@newlradius	895, 904		
\dbp@newmethod	894, 905		
\dbp@newrradius	896, 904		
\dbp@r	981		
\dbp@radius	814, 826, 837, 848, 958, 965, 1002, 1009, 1010, 1011, 1012, 1023, 1025, 1036, 1038, 1045, 1049, 1054, 1056, 1058, 1067, 1071, 1076, 1078, 1080, 1088, 1091, 1098, 1099, 1100, 1108, 1111, 1118, 1119, 1120		
\dbp@rev@lrl	880		
\dbp@rev@lsl	876		
\dbp@rev@lsr	878		
\dbp@rev@rlr	881		
\dbp@rev@rsl	879		
\dbp@rev@rsr	877		
\dbp@rlr	867, 880		
\dbp@rlrangone	1112, 1118, 1123		
\dbp@rlrangsup	1111, 1113, 1115, 1116		
\dbp@rlrangthree	1116, 1119, 1123		
\dbp@rlrangtwo	1114, 1120, 1123		
\dbp@rlrarbr	1110, 1113, 1115		
		E	
		environments:	
		sankeydiagram	2, 228
		N	
\name	147, 155, 240, 241, 242, 244, 246, 247, 265, 267, 284, 287, 296, 298, 300, 304, 310, 468, 475, 488, 502, 504, 511, 514, 515, 517, 518, 525, 528, 529, 531, 532, 534, 547, 549, 550, 551, 552, 553, 558, 568, 575, 591, 593, 594, 595, 596, 597, 601, 610, 617, 633, 635, 636, 637, 638, 639, 644, 653, 659, 675, 677, 678, 679, 680, 681, 685, 694, 701, 745, 746, 748, 749, 752, 753, 755, 760, 761, 763, 764, 767, 768, 769, 771, 772, 773		
		O	
\orient	155, 242, 248, 268, 297, 298, 304, 476, 489, 551, 563, 564, 573, 574, 595, 606, 607, 615, 616, 637, 648, 649, 657, 658, 679, 690, 691, 699, 700		
		P	
\pos	155, 286, 300, 553, 597, 639, 681		
		Q	
\qty	155, 241, 245, 266, 276, 280, 296, 334, 339, 347, 352, 371, 372, 399, 477, 490, 550, 559, 569, 594, 603, 612, 636, 645, 654, 678, 687, 696, 728		
		S	
\sankeyadvance	9, 497		
\sankeyadvance*	9, 497		
sankeydiagram (environment)	2, 228		

<code>\sankeydubins</code>	15, 328	<code>\snk@filldrawend</code>	309, 467, 491
<code>\sankeyend</code>	18, 484	<code>\snk@filldrawstart</code>	303, 466, 478
<code>\sankeyfork</code>	20, 723	<code>\snk@forkanchor</code>	194, 256, 283, 284, 286
<code>\sankeygetnodeorient</code>	8, 31, 169, 222, 242, 310, 337, 338, 391, 407, 413, 421, 422, 430, 431, 440, 441, 448, 449, 458, 476, 489, 509, 523, 551, 595, 637, 679	<code>\snk@getenddraw</code>	38, 312
<code>\sankeygetnodeqty</code>	8, 29, 168, 216, 241, 317, 318, 334, 371, 406, 412, 457, 477, 490, 510, 524, 550, 594, 636, 678	<code>\snk@getendfill</code>	38, 311
<code>\sankeynode</code>	5, 269, 287, 289, 397, 404, 410, 460, 482, 495, 507, 521, 727, 728	<code>\snk@getpos</code>	162, 229, 286, 552, 553, 596, 597, 638, 639, 680, 681
<code>\sankeynodealias</code>	21, 454, 504, 549, 593, 635, 677	<code>\snk@getstartdraw</code>	33, 306
<code>\sankeynodeend</code>	6, 494	<code>\snk@getstartfill</code>	33, 305
<code>\sankeynodestart</code>	6, 481	<code>\snk@ifnodedefined</code>	43, 47
<code>\sankeyoutin</code>	15, 365	<code>\snk@laststep</code>	374, 393, 429, 447
<code>\sankeyqtytolen</code>	8, 216, 245, 339, 347, 352, 372, 559, 569, 603, 612, 645, 654, 687, 696, 731	<code>\snk@len</code>	245, 247, 372, 385, 386
<code>\sankeyset</code>	2, 66, 68, 82, 86, 93, 98, 104, 111, 116, 119, 154, 155, 165, 174, 183, 194, 200, 211, 263, 333, 370, 462, 463, 474, 487, 501, 546, 590, 632, 674, 726, 732, 739, 743, 758	<code>\snk@listofforks</code>	194, 255, 274, 281, 288
<code>\sankeystart</code>	18, 471	<code>\snk@makeforkednode</code>	253, 465
<code>\sankeyturn</code>	10, 707	<code>\snk@makenode</code>	294, 464, 554, 598, 640, 682
<code>\sankeyturn*</code>	10, 707	<code>\snk@minradius</code>	86, 339, 346, 347, 352, 353, 552, 560, 561, 570, 571, 596, 604, 613, 638, 646, 655, 680, 688, 697
<code>\sankeyturnleft</code>	11, 623, 703, 718	<code>\snk@modulo</code>	233, 235
<code>\sankeyturnleft*</code>	11, 623	<code>\snk@newfield</code>	17, 29, 31, 33, 36, 38, 41
<code>\sankeyturnleftbackward</code>	12, 661, 665, 714	<code>\snk@normalize@angle</code>	160, 234, 297, 551, 595, 637, 679
<code>\sankeyturnleftbackward*</code>	12, 665	<code>\snk@oldname</code>	503, 504, 508, 509, 510, 514, 515, 517, 518, 522, 523, 524, 528, 529, 531, 532, 548, 549, 557, 567, 592, 593, 602, 611, 634, 635, 643, 652, 676, 677, 686, 695
<code>\sankeyturnright</code>	13, 537, 619, 719	<code>\snk@orient</code>	268, 270
<code>\sankeyturnright*</code>	13, 537	<code>\snk@outinmidptangle</code>	395, 400
<code>\sankeyturnrightbackward</code>	14, 577, 581, 715	<code>\snk@outinmidptname</code>	379, 384, 385, 386, 394, 396, 398, 401
<code>\sankeyturnrightbackward*</code>	14, 581	<code>\snk@prevpt</code>	419, 421, 428, 430, 438, 440, 446, 448
<code>\snk@added@values</code>	253	<code>\snk@ptnum</code>	393, 394
<code>\snk@anchor</code>	174, 284, 299, 300	<code>\snk@qtyi</code>	315
<code>\snk@angle</code>	542, 543, 544, 551, 553, 563, 564, 573, 574, 577, 586, 587, 588, 595, 597, 606, 607, 615, 616, 619, 628, 629, 630, 637, 639, 648, 649, 657, 658, 661, 670, 671, 672, 679, 681, 690, 691, 699, 700, 703	<code>\snk@qtyii</code>	315
<code>\snk@anglesign</code>	544, 545, 588, 589, 630, 631, 672, 673, 711, 713, 717	<code>\snk@rotate</code>	82, 222, 248
<code>\snk@c</code>	286, 287, 289, 552, 553, 596, 597, 638, 639, 680, 681	<code>\snk@setenddraw</code>	38, 115
<code>\snk@checkquantities</code>	315, 331, 368	<code>\snk@setendfill</code>	38, 114
<code>\snk@curpt</code>	418, 419, 422, 423, 427, 428, 431, 432, 437, 438, 441, 442, 445, 446, 449, 450	<code>\snk@setnodeorient</code>	31, 298, 458
<code>\snk@diff</code>	276, 277	<code>\snk@setnodeqty</code>	29, 296, 457
<code>\snk@endstyle</code>	104, 311, 312	<code>\snk@setstartdraw</code>	33, 110
<code>\snk@errorifnotdefined</code>	46, 329, 330, 366, 367, 455, 472, 485, 499, 539, 583, 625, 667, 709, 724	<code>\snk@setstartfill</code>	33, 109
		<code>\snk@show@debug</code>	237, 468, 534, 575, 617, 659, 701
		<code>\snk@startstyle</code>	104, 305, 306
		<code>\snk@step</code>	373, 377, 378
		<code>\snk@stepoutin</code>	93, 373, 374, 411, 420, 426, 428, 439, 444, 446
		<code>\snk@subnodename</code>	253, 728
		<code>\snk@subnodeqty</code>	253
		<code>\snk@tot</code>	253
		<code>\snk@totallen</code>	68, 731
		<code>\snk@totalqty</code>	68, 731
T			
<code>\tikzlibrarydubins@date</code>	777		
<code>\tikzlibrarydubins@version</code>	776		