

The `sankey` package

Draw Sankey diagrams via TikZ

Paul Gaborit (paul.gaborit@gmail.com)

v3.0.2 – 2025/01/10

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.2), [Examples](#) (p.22) and [Installation & Implementation](#) (p.43).

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

This manual contains three parts: User manual (p.2), Examples (p.22) and Installation & 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	3
2	The <code>sankeydiagram</code> environment	3
3	Sankey diagram options	3
3.1	Keys to choose the scale	4
3.2	Keys to define rotate offset	4
3.3	Keys to define drawing parameters	4
3.4	Keys to choose drawing styles	4
3.5	Keys to define new <i>start</i> and <i>end</i> styles	5
3.6	The <i>debug</i> key	5
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, as well as the `calc`, `decorations.markings` and `dubins` (cf. 5.2 on page 20) TikZ libraries.

2 The `sankeydiagram` environment

`sankeydiagram` (*env.*) 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).

Options can be defined in three ways:

- 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`
- via the `\sankeyset` macro:

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

- 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.

`every diagram` (*Key*) `/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**.

`ratio quantity (Key)` /sankey/**ratio quantity**=⟨number⟩ (initially: 10)

Quantity (in units of flow) to define ratio. The ⟨number⟩ can be any math expression.

`ratio length (Key)` /sankey/**ratio length**=⟨distance⟩ (initially: 1cm)

Distance (a graphical distance) to define scale.

`ratio (Key)` /sankey/**ratio**=⟨distance⟩/⟨number⟩ (initially: 1cm/10)

Fix the ratio to ⟨distance⟩/⟨number⟩.

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

`rotate (Key)` /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

`minimum radius (Key)` /sankey/**minimum radius**=⟨distance⟩ (initially: 5mm)

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

`outin steps (Key)` /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

`fill/.style (Key)` /sankey/**fill/.style**=⟨style⟩ (initially: `line width=0pt,fill=white`)

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

`draw/.style (Key)` /sankey/**draw/.style**=⟨style⟩ (initially: `draw=black,line width=.4pt`)

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

`start style (Key)` /sankey/**start style**=⟨style name⟩ (initially: `none`)

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

`end style (Key)` /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

`new start style (Key) /sankey/new start style={⟨name⟩}{⟨fill path⟩}{⟨draw path⟩}`

Define the new start style named ⟨name⟩ with its ⟨fill path⟩ and its ⟨draw path⟩.

`new end style (Key) /sankey/new end style={⟨name⟩}{⟨fill path⟩}{⟨draw path⟩}`

Define the new end style named ⟨name⟩ with its ⟨fill path⟩ and its ⟨draw path⟩.

The ⟨fill path⟩ and the ⟨draw path⟩ 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

`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⟩}`

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⟩.

`name (Key) /sankey/node parameters/name=⟨name⟩`

The ⟨name⟩ of the new Sankey node (and the associated TikZ node).

`quantity (Key) /sankey/node parameters/quantity=⟨quantity⟩`

The quantity (in flow unit) of the new Sankey node. The ⟨quantity⟩ can be any math expression.

`angle (Key) /sankey/node parameters/angle=⟨angle⟩`

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

`at (Key) /sankey/node parameters/at=⟨at⟩` (initially: `0,0`)

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

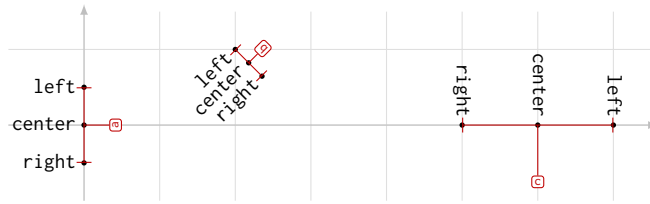
`anchor (Key) /sankey/node parameters/anchor=⟨anchor⟩` (initially: `center`)

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

`as (Key) /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={5,0},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}
```

4.1.1 Choose default parameters

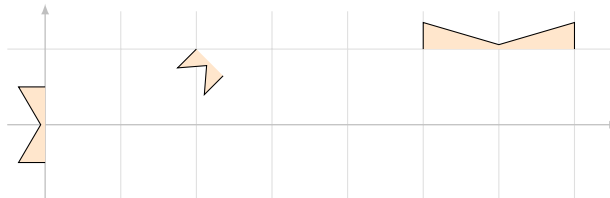
`every node/.style (Key)` `/sankey/every node/.style={⟨node parameters⟩}` (initially: `empty`)

The `⟨node parameters⟩` 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` `\sankeynodestart[⟨options⟩]{⟨node parameters⟩}`

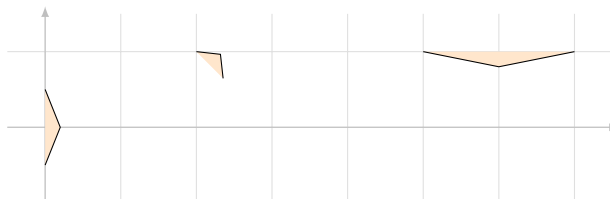
The `\sankeynodestart` creates and fills/draws a starting Sankey node:



```
\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` `\sankeynodeend[⟨options⟩]{⟨node parameters⟩}`

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



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`.

```

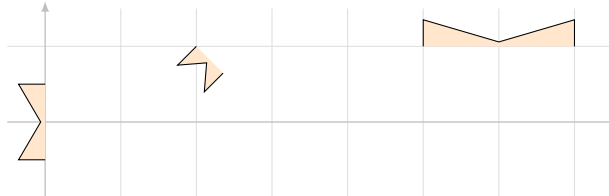
\begin{tikzpicture}
  \begin{sankeydiagram}[end style=simple,fill/.style={fill=orange!20}]
    \sankeynodeend{name=a,quantity=10}
    \sankeynodeend{name=b,quantity=5,angle=45,at={2,1},anchor=left}
    \sankeynodeend{name=c,quantity=20,angle=-90,at={5,1},anchor=right}
  \end{sankeydiagram}
\end{tikzpicture}

```

4.1.3 Create starting and ending nodes via options

`start` (*Key*) /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.



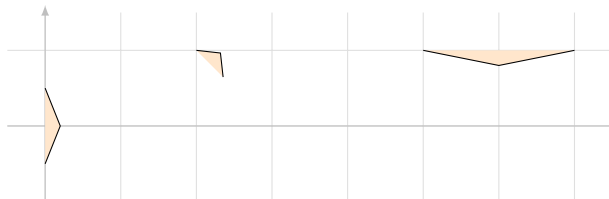
```

\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}

```

`end` (*Key*) /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.

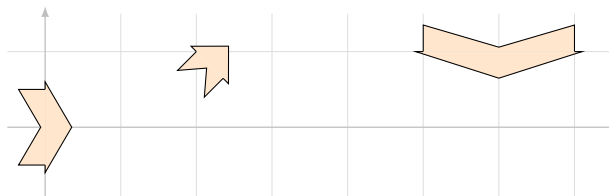


```

\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` `\sankeygetnodeqty{<node name>}`

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

`\sankeyqtytolen` `\sankeyqtytolen{<quantity>}`

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

`\sankeygetnodeorient` `\sankeygetnodeorient{<node name>}`

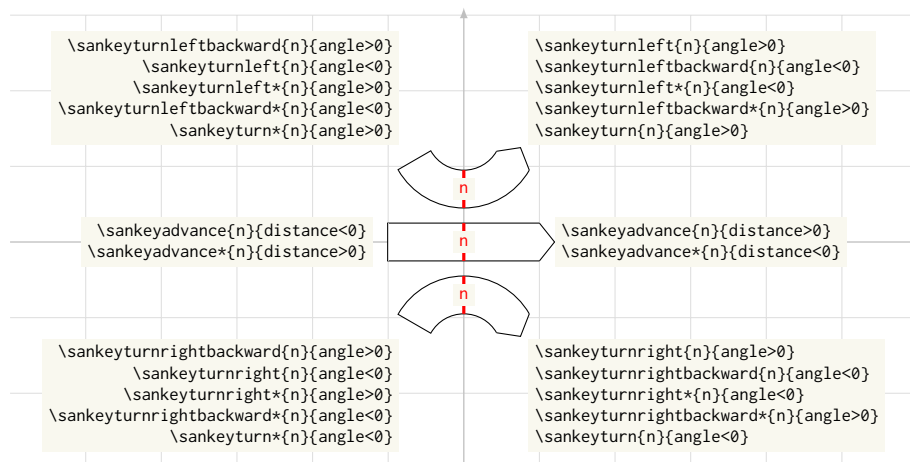
The expandable command `\sankeygetnodeorient` returns the angle (orientation) assigned to the Sankey node named `<node name>`.

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.

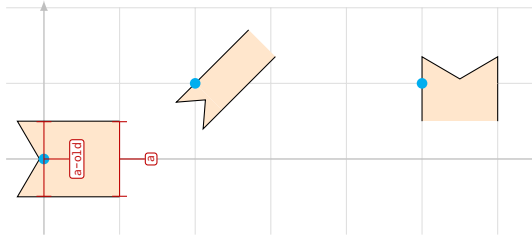
Except for the `\sankeyturn` macro, a negative value (distance or angle) moves in the opposite direction (the `\sankeyturn` macro is an exception: a negative angle turns right while a positive value turns left).



4.3.1 Macro to move straight (forward or backward)

`\sankeyadvance` `\sankeyadvance[(options)]{(node name)}{(distance)}`

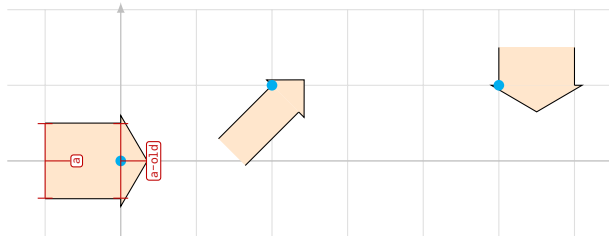
The `\sankeyadvance` moves the sankey node straight ahead and fills/draws this portion of the sankey path. A positive *(distance)* moves forward while a negative *(distance)* moves backward.



```
\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}
\fill[cyan] (0,0) circle(2pt) (2,1) circle(2pt) (5,1) circle(2pt);
\end{sankeydiagram}
\end{tikzpicture}
```

`\sankeyadvance*` `\sankeyadvance*[(options)]{(node name)}{(distance)}`

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.

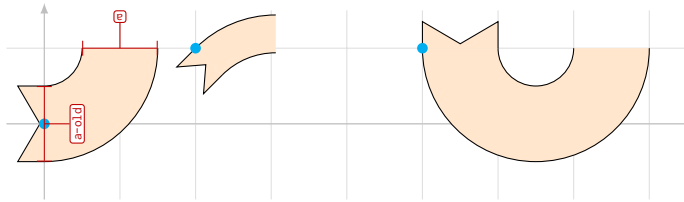


```
\begin{tikzpicture}
\begin{sankeydiagram}[end style=arrow,fill/.style={fill=orange!20}]
\sankeynodeend{name=a,quantity=10}
\sankeyadvance*{a}{1cm}
\sankeynode[debug]{as=a}
\sankeynode[debug]{as=a-old}
\sankeynodeend{name=b,quantity=5,angle=45,at={2,1},anchor=left}
\sankeyadvance*{b}{1cm}
\sankeynodeend{name=c,quantity=10,angle=-90,at={5,1},anchor=right}
\sankeyadvance*{c}{5mm}
\fill[cyan] (0,0) circle(2pt) (2,1) circle(2pt) (5,1) circle(2pt);
\end{sankeydiagram}
\end{tikzpicture}
```

4.3.2 Macro to turn forward or backward

`\sankeyturn` `\sankeyturn[(options)]{(node name)}{(angle)}`

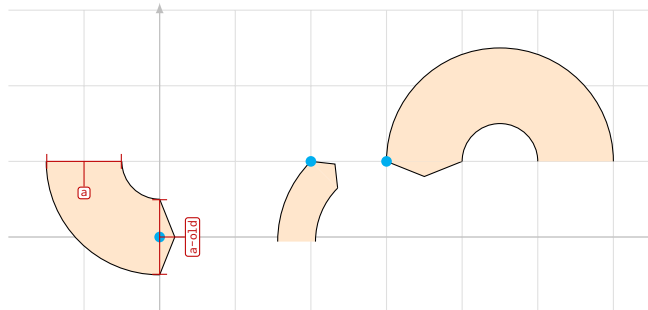
The `\sankeyturn` macro moves the sankey node by turning to one side or the other and fills/draws this portion of the sankey path. A *positive* `(angle)` turns left while a *negative* `(angle)` turns right.



```
\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}
    \fill[cyan] (0,0) circle(2pt) (2,1) circle(2pt) (5,1) circle(2pt);
  \end{sankeydiagram}
\end{tikzpicture}
```

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

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.

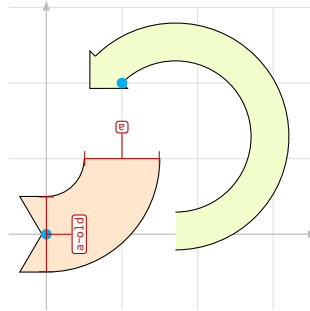


```
\begin{tikzpicture}
  \begin{sankeydiagram}[end style=simple,fill/.style={fill=orange!20}]
    \sankeynodeend{name=a,quantity=10}
    \sankeyturn*{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeynodeend{name=b,quantity=5,angle=45,at={2,1},anchor=left}
    \sankeyturn*[minimum radius=1cm]{b}{-45}
    \sankeynodeend{name=c,quantity=10,angle=-90,at={3,1},anchor=right}
    \sankeyturn*{c}{180}
    \fill[cyan] (0,0) circle(2pt) (2,1) circle(2pt) (3,1) circle(2pt);
  \end{sankeydiagram}
\end{tikzpicture}
```

4.3.3 Macros to turn left (forward or backward)

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

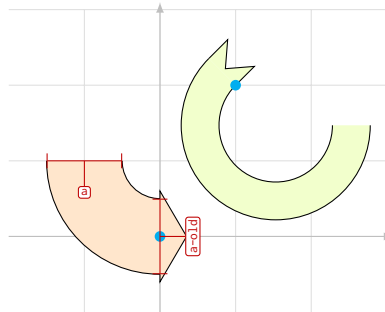
The `\sankeyturnleft` macro moves the sankey node by turning left and fills/draws this portion of the sankey path. A *positive* `angle` turns forward while a *negative* `angle` turns backward.



```
\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}
    \fill[cyan] (0,0) circle(2pt) (1,2) circle(2pt);
  \end{sankeydiagram}
\end{tikzpicture}
```

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

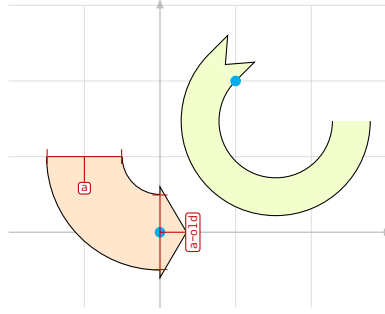
The `\sankeyturnleft*` macro moves the sankey node backward by turning left and fills/draws this portion of the sankey path. A *positive* `angle` turns backward while a *negative* `angle` turns forward.



```
\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}
    \fill[cyan] (0,0) circle(2pt) (1,2) circle(2pt);
  \end{sankeydiagram}
\end{tikzpicture}
```

`\sankeyturnleftbackward` `\sankeyturnleftbackward[(options)]{(node name)}{(angle)}`

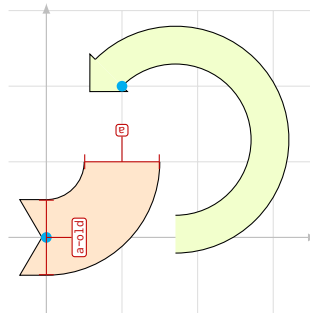
The `\sankeyturnleftbackward` macro moves the sankey node backward by turning left and fills/draws this portion of the sankey path. A *positive* `(angle)` turns backward while a *negative* `(angle)` turns forward.



```
\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}
    \fill[cyan] (0,0) circle(2pt) (1,2) circle(2pt);
  \end{sankeydiagram}
\end{tikzpicture}
```

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

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.

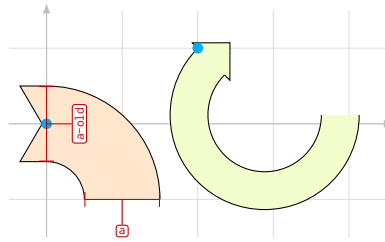


```
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=arrow,end style=arrow]
    \sankeyset{fill/.style={fill=orange!20}}
    \sankeynodestart{name=a,quantity=10}
    \sankeyturnleftbackward*{a}{90}
    \sankeynode[debug]{as=a}
    \sankeynode[debug]{as=a-old}
    \sankeyset{fill/.style={fill=lime!20}}
    \sankeynodeend{name=b,quantity=5,angle=-135,at={1,2},anchor=left}
    \sankeyturnleftbackward*[minimum radius=1cm]{b}{-225}
    \fill[cyan] (0,0) circle(2pt) (1,2) circle(2pt);
  \end{sankeydiagram}
\end{tikzpicture}
```

4.3.4 Macros to turn right (forward or backward)

`\sankeyturnright` `\sankeyturnright[options]{(node name)}{(angle)}`

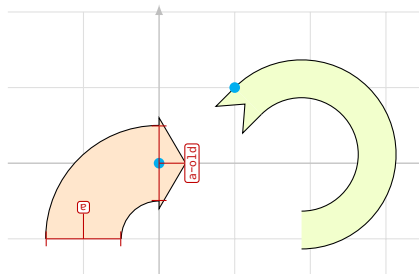
The `\sankeyturnright` macro moves the sankey node by turning right and fills/draws this portion of the sankey path. A *positive* `(angle)` turns forward while a *negative* `(angle)` turns backward.



```
\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}
    \fill[cyan] (0,0) circle(2pt) (2,1) circle(2pt);
  \end{sankeydiagram}
\end{tikzpicture}
```

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

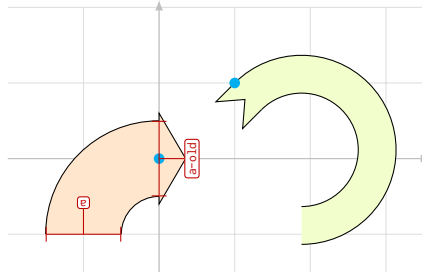
The `\sankeyturnright*` macro moves the sankey node backward by turning right and fills/draws this portion of the sankey path. A *positive* `(angle)` turns backward while a *negative* `(angle)` turns forward.



```
\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}
    \fill[cyan] (0,0) circle(2pt) (1,1) circle(2pt);
  \end{sankeydiagram}
\end{tikzpicture}
```

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

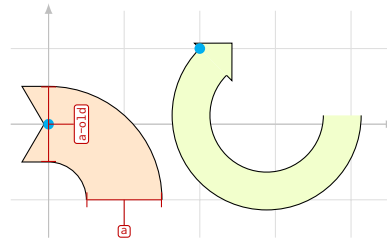
The `\sankeyturnrightbackward` macro moves the sankey node backward by turning right and fills/draws this portion of the sankey path. A *positive* *<angle>* turns backward while a *negative* *<angle>* turns forward.



```
\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}
    \fill[cyan] (0,0) circle(2pt) (1,1) circle(2pt);
  \end{sankeydiagram}
\end{tikzpicture}
```

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

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}
    \fill[cyan] (0,0) circle(2pt) (2,1) circle(2pt);
  \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` `\sankeyoutin[⟨options⟩]{⟨node A⟩}{⟨node B⟩}`

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.

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

`\sankeydubins` `\sankeydubins[⟨options⟩]{⟨node A⟩}{⟨node B⟩}`

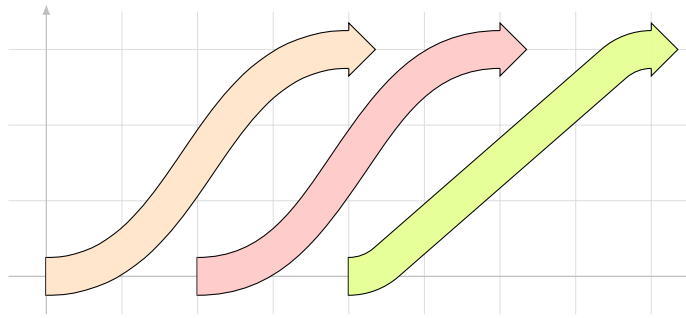
The `\sankeydubins` macro fills/draws a lane between ⟨node A⟩ and ⟨node B⟩ using a Dubins path³.

Note: 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

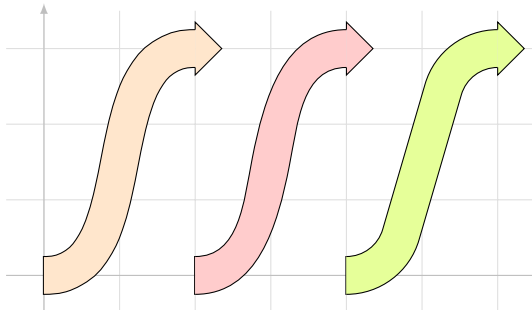
The following diagrams compare `outin` path with 10 steps (orange), `outin` path with 2 steps (red) and `dubins` path (lime) in various positions.



```
\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}

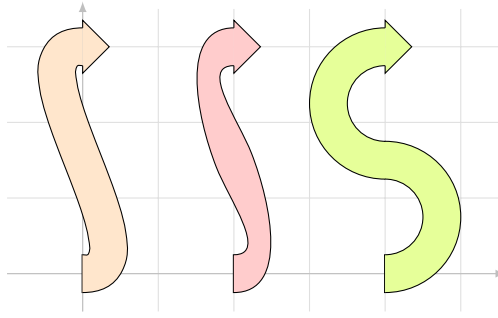
\sankeyset{fill/.style={fill=lime!40}}
\sankeynodestart{name=a,at={4,0},quantity=5}
\sankeynodeend{name=b,at={8,3},quantity=5}
\sankeydubins[minimum radius=5mm]{a}{b}
\end{sankeydiagram}
\end{tikzpicture}
```



```
\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}
```

```

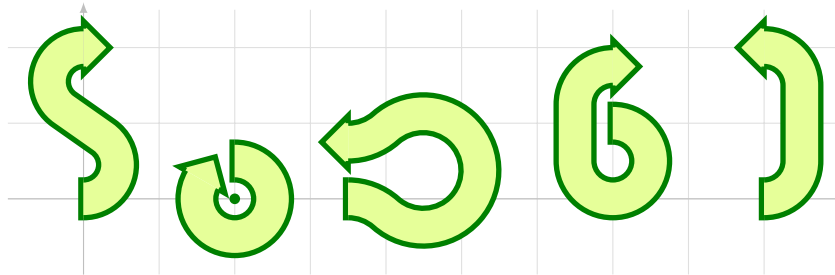
\begin{tikzpicture}
  \begin{sankeydiagram}[start style=simple,end style=arrow]
    \sankeyset{fill/.style={fill=orange!20}}
    \sankeynodestart{name=a,quantity=5}
    \sankeynodeend{name=b,at={0,3},quantity=5}
    \sankeyoutin{a}{b}

    \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}

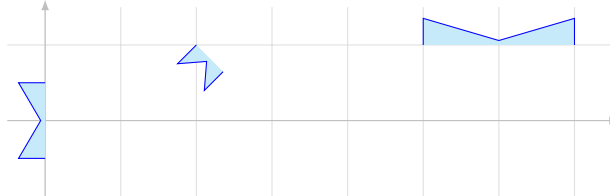
```

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

4.5 Pure filling/drawing macros

`\sankeystart` `\sankeystart[⟨options⟩]{⟨name⟩}`

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

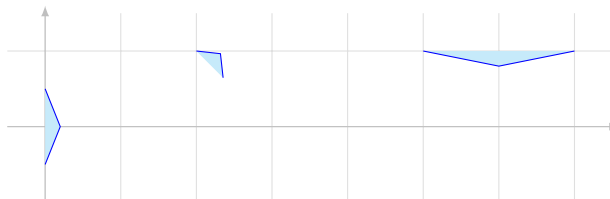


```
\begin{tikzpicture}
\begin{sankeydiagram}
[start style=arrow,fill/.style={fill=cyan!20},draw/.style={draw=blue}]
\sankeynode{name=a,quantity=10}
\sankeystart{a}
\sankeynode{name=b,quantity=5,angle=45,at={2,1},anchor=left}
\sankeystart{b}
\sankeynode{name=c,quantity=20,angle=-90,at={5,1},anchor=right}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

`\sankeyend` `\sankeyend[⟨options⟩]{⟨name⟩}`

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

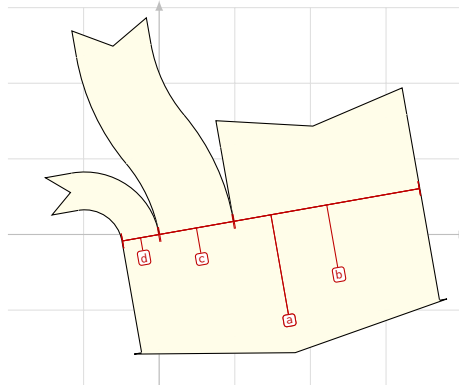
When creating a new Sankey node, the `forked` and `fork anchor` keys allow to fork the node directly *and* to anchor it on an anchor of a forked subnode.

`forked (Key)` `/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.

`fork anchor (Key)` `/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}
    }

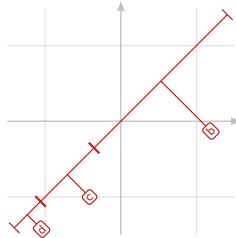
    \sankeynode[debug]{
      name=a,quantity=40,angle=-80,
      forked={25/b,10/c,5/d},
      fork anchor=c.right,
    }

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

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 \sankeynodealias{<origname>}{<clonename>}
```

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.2), Examples (p.22) and Installation & 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

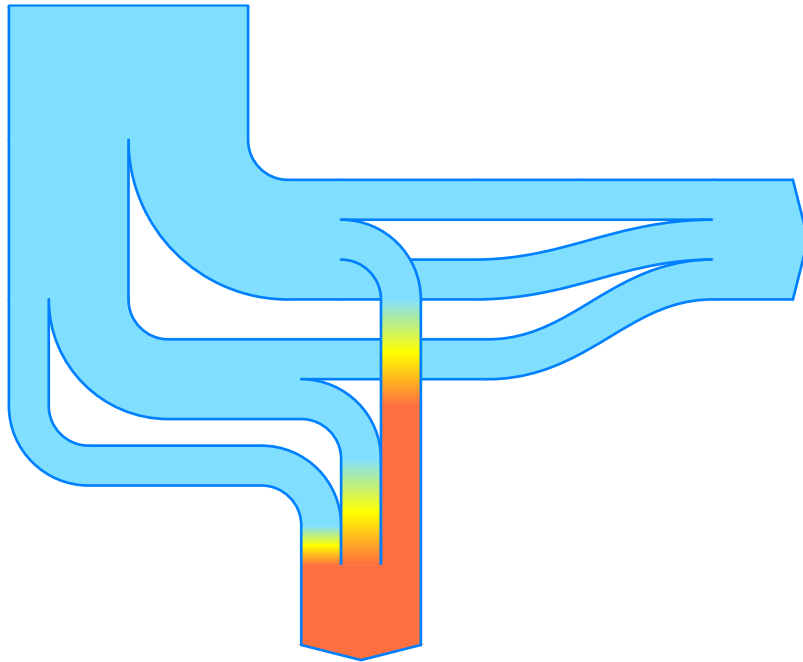


Figure 1: Simple example

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}

```

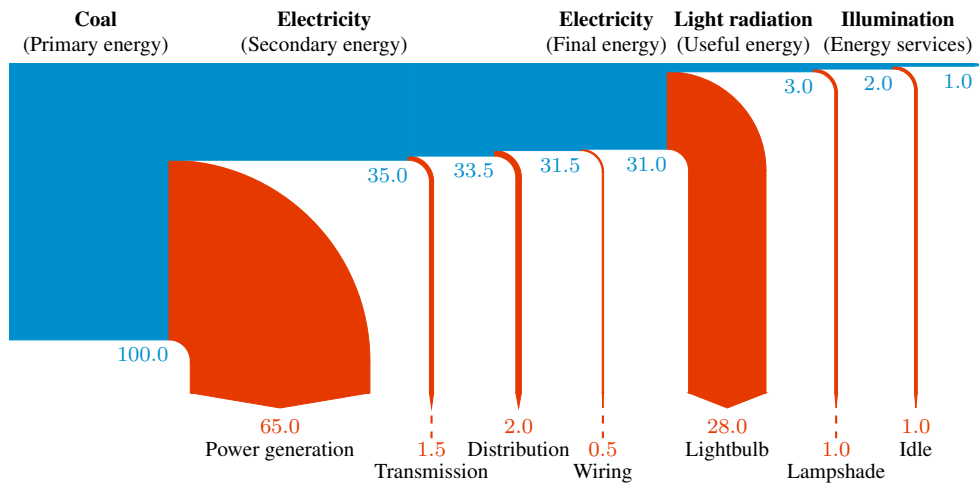



Figure 2: Energy diagram

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-pad=true,
  }
  \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}

```

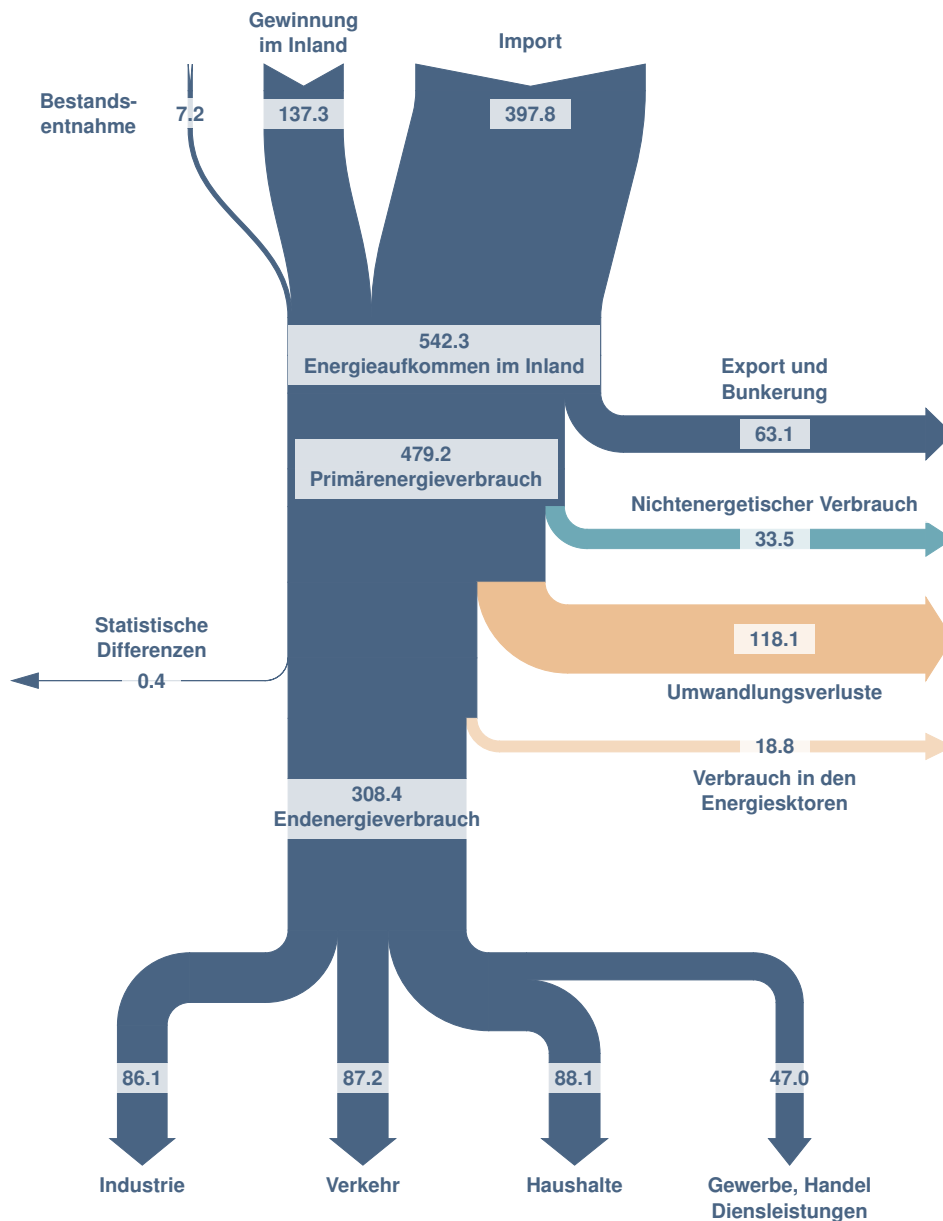


Figure 3: Example from TeX.se question

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}

```

```

\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}

```

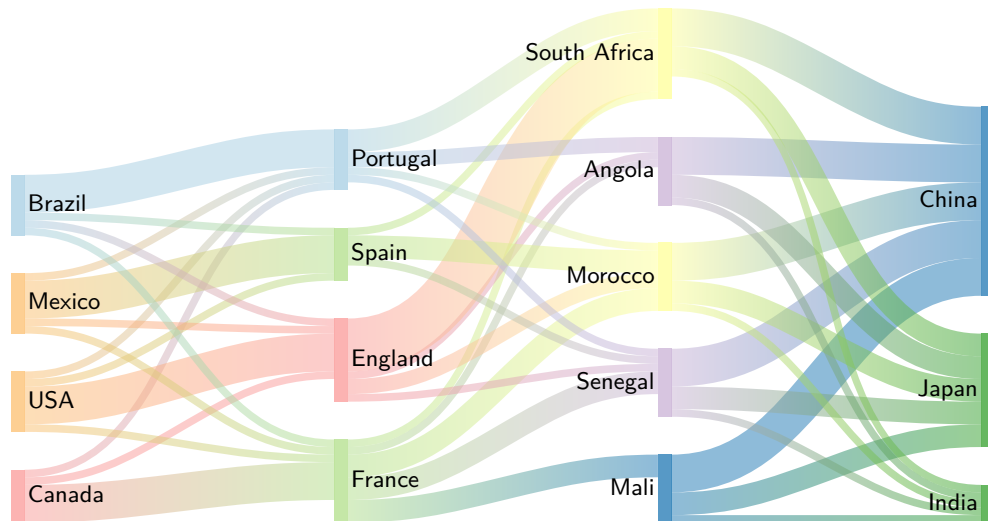


Figure 4: Reproduction of an example from Google Charts documentation

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\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=east,inner sep=.1em,font=\small]
  at (\country-old) {\countryname\vphantom{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]
    \sffamily
    \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
      @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/lime,BR/violet,FR/yellow,GB/blue,

```

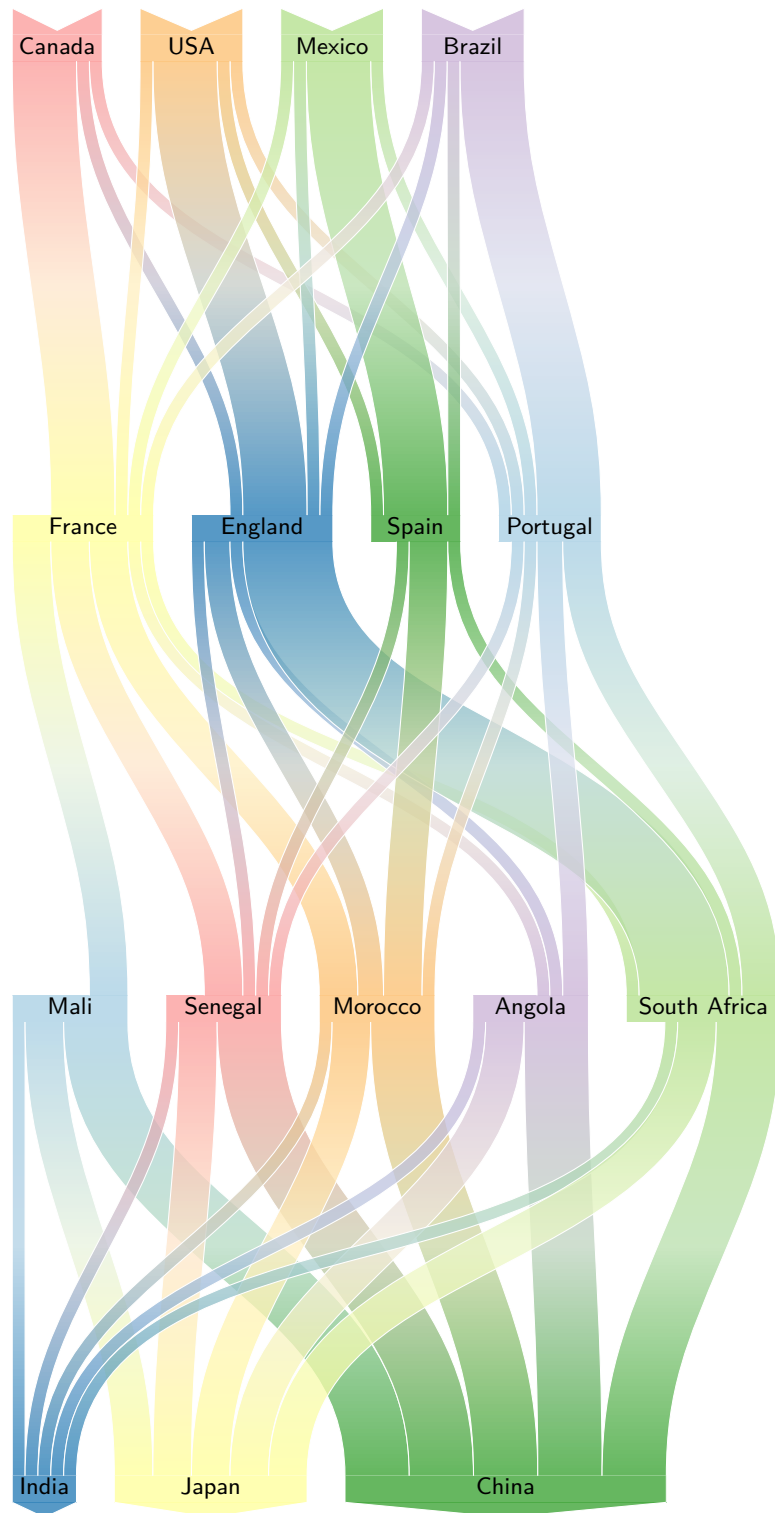


Figure 5: Reproduction of an example from Google Charts documentation – variation using the rotate key.

```

    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 27 and 52.

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      {Ibiiieait/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      {Ibiiieait2/-/\Qty{Ibiiieait}-\Qty{Fe}-\Qty{Ofr}},
47      {Fi/Finance income/204},
48      {Ibit/Income before income taxes/\Qty{Ibiiieait2}+\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/>

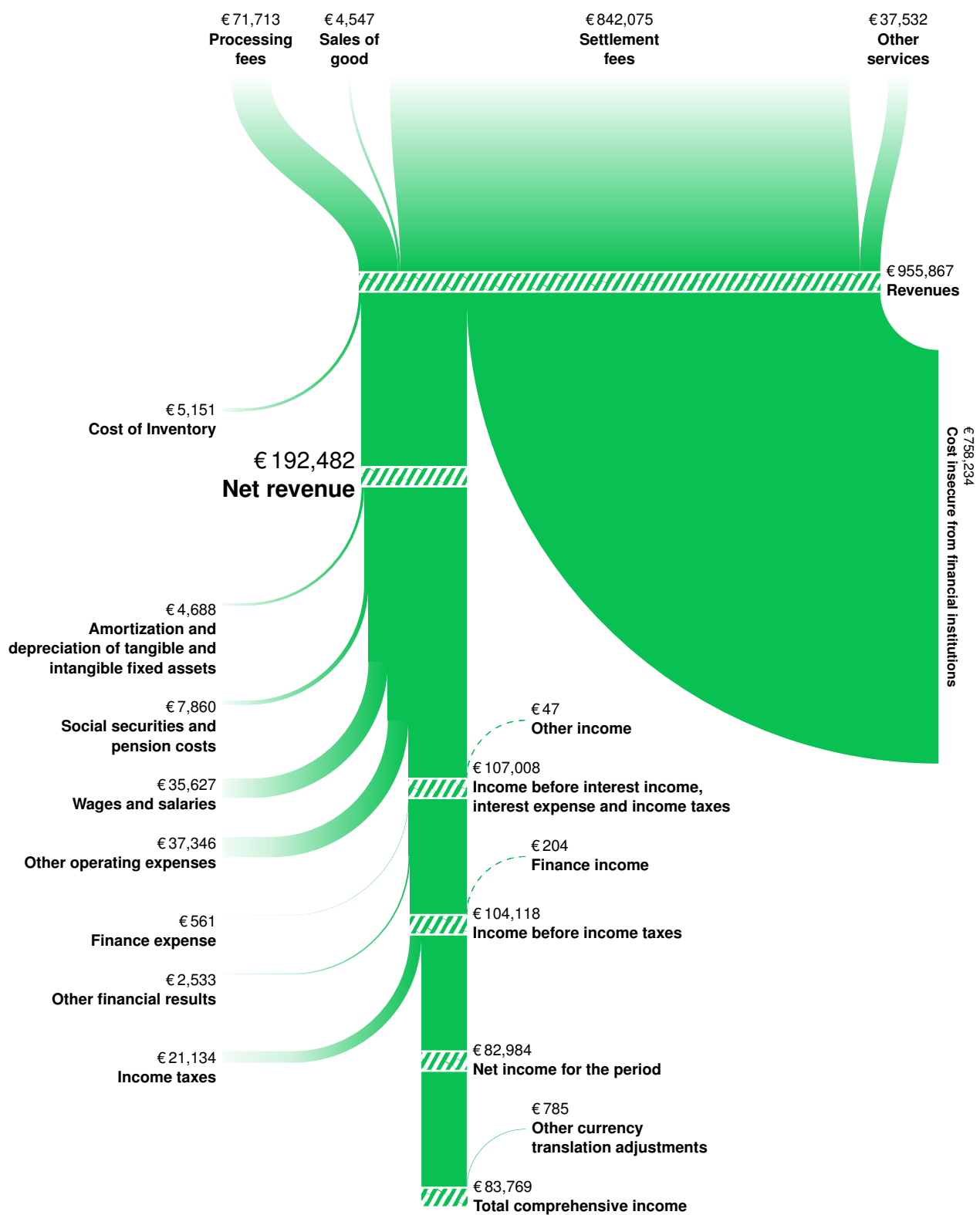


Figure 6: Very nice example – Nadieh Bremer’s creation (from [Adyen’s Shareholder Report](#))

```

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=1.cm]{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{Aadotaiifa}/Aadotaiifa}
169
170 \turnandstop{Aadotaiifa}
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=Ibiiieait,quantity=\Qty{Ibiiieait},
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{Ibiiieait}{1em}
190 \node[cost node=Ibiiieait,right label hashed=Ibiiieait];
191
192 \sankeyfork{Ibiiieait}
193 {\Qty{Ibiiieait2}/Ibiiieait2,\Qty{Ofr}/Ofr,\Qty{Fe}/Fe}
194
195 \turnandstop{Fe}
196

```

```

197 \sankeyadvance{Ofr}{3em}
198 \turnandstop{Ofr}
199
200 \sankeyadvance{Ibiiieait2}{6em}
201 \sankeynode{name=Ibit,quantity={\Qty{Ibiiieait2}+\Qty{Fi}},
202   anchor=right,at={Ibiiieait2.right},
203   forked={\Qty{Fi}/Fi,\Qty{Ibiiieait2}/Ibiiieait2-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,Ibiiieait,Ibit,Niftp,Tci}{
238   \hashband{\nodename}
239 }
240
241 \end{sankeydiagram}
242 \end{tikzpicture}

```

This manual contains three parts: User manual (p.2), Examples (p.22) and Installation & Implementation (p.43).

Part III

Installation & Implementation

Contents

13 Compiling <code>sankey</code>	43
14 The <code>sankey.sty</code> file	44
14.1 Fields	44
14.1.1 Definition of <i>global</i> and <i>expanded</i> fields (using <code>\xdef</code>)	44
14.1.2 Definitions of <i>local</i> fields (using <code>\def</code>)	45
14.1.3 Check if a <code>sankey</code> node is defined	45
14.2 The <code>sankey</code> node shape	45
14.3 Keys	46
14.3.1 Keys to define the scale	46
14.3.2 Rotate offset	46
14.3.3 Minimum radius	46
14.3.4 Outin step	46
14.3.5 Sankey debug	47
14.3.6 Start and end styles	47
14.3.7 Initial parameters	48
14.3.8 Sankey node parameters	48
14.3.9 Internal Tikz style	50
14.4 Internal macros	50
14.5 User macros	52
14.6 The <code>sankeydiagram</code> environment	59
14.7 Predefined start and end styles	60
15 <code>tikzlibrarydubins.code.tex</code>	61
16 Change History	66
17 Index	67

13 Compiling `sankey`

To produce the `sankey` package:

```
pdflatex sankey.ins # or 'latex sankey.ins'
```

To finish the installation you have to move the `tikzlibrarydubins.code.tex` and `sankey.sty` files into a directory searched by L^AT_EX.

To compile the `sankey` documentation (the `sankey.pdf` file):

```
pdflatex sankey.dtx
makeindex -s gind.ist -o sankey.ind sankey.idx
pdflatex sankey.dtx
makeindex -s gind.ist -o sankey.ind sankey.idx
pdflatex sankey.dtx
pdflatex sankey.dtx
```

14 The `sankey.sty` file

Poorly commented source code...

Version information:

```
1 \NeedsTeXFormat{LaTeX2e}[2015/10/01]
2 \ProvidesPackage{sankey}[2025/01/10 v3.0.2 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}
```

14.1 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 }
```

14.1.1 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'}
```

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

`\snk@setstartfill` The setter and getter macros to store and retrieve the starting fill/draw paths (the `\snk@getstartfill` 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` The setter and getter macros to store and retrieve the ending fill/draw paths (the key `\snk@getendfill` 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}
```

14.1.3 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 }
```

14.2 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   \inheritsavedanchors[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 }
```

14.3 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 `/sankey` as default path.

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

14.3.1 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@totalen` 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@totalen{#1}
73     \edef\snk@totalen{\snk@totalen 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 }
```

14.3.2 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 }
```

14.3.3 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 }
```

14.3.4 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   outin steps/.value required,
96 }
```

14.3.5 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 }
```

14.3.6 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 }
```

14.3.7 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 }
```

14.3.8 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 `\qty` to store the `name`, the `quantity`, the `orientation` (or `angle`) and the `position` of a Sankey `\orient` 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` `\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 }
```

14.3.9 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 }
```

14.4 Internal macros

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

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

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

```
232 \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`).

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

`\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.

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

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

```

`\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.

```

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

```

`\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`.

```

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

```

`\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`.

```

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

```

`\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 `\snk@qtyii` verification).

```

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

```

14.5 User macros

The user macros are globally defined as internal (with `snk@` prefix) and locally defined in the `sankeydiagram` environment as user macros (without the `snk@` prefix).

`\snk@sankeydubins` The `\snk@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).

```

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

```

`\snk@sankeyoutin` The `\snk@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.

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

493 \NewDocumentCommand\snk@sankeynodeend{0{m}}{% options, node parameters
494 \sankeynode[#1]{end,#2}
495 }

```

`\snk@init@move` The `\snk@init@move` macro applies options et clones the current node before its moving.

```

496 \def\snk@init@move#1#2{% params: options, name
497 \sankeyset{#1}
498 \edef\name{#2}
499 \edef\snk@oldname{#2-old}
500 \sankeynodealias{\name}{\snk@oldname}
501 \edef\qty{\sankeygetnodeqty{\name}}
502 }

```

`\snk@sankeyadvance` The `\snk@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`.

```

503 \NewDocumentCommand\snk@sankeyadvance{s0{mm}}{%
504 % params: *(reverse), options, name, distance
505 \snk@errorifnotdefined{#3}
506 \begingroup
507 \snk@init@move{#2}{#3}
508 \IfBooleanTF{#1}{
509 % move backward
510 \sankeynode{
511 at={\snk@oldname.center}!#4!90:(\snk@oldname.left)$,
512 angle=\sankeygetnodeorient{\snk@oldname},
513 quantity=\sankeygetnodeqty{\snk@oldname},
514 name=\name,
515 }
516 \path[/sankey/fill]
517 (\name.left) -- (\snk@oldname.left)
518 -- (\snk@oldname.right) -- (\name.right) -- cycle;
519 \path[/sankey/draw]
520 (\name.left) -- (\snk@oldname.left)
521 (\snk@oldname.right) -- (\name.right);
522 }{
523 % move forward
524 \sankeynode{
525 at={\snk@oldname.center}!#4!-90:(\snk@oldname.left)$,
526 angle=\sankeygetnodeorient{\snk@oldname},
527 quantity=\sankeygetnodeqty{\snk@oldname},
528 name=\name,
529 }
530 \path[/sankey/fill]
531 (\snk@oldname.left) -- (\name.left)
532 -- (\name.right) -- (\snk@oldname.right) -- cycle;
533 \path[/sankey/draw]
534 (\snk@oldname.left) -- (\name.left)
535 (\name.right) -- (\snk@oldname.right);
536 }
537 \snk@show@debug{\name}
538 \endgroup
539 }

```

`\snk@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`.

```

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

```

`\snk@sankeyturnrightbackward` The `\snk@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`.

```

580 \NewDocumentCommand\snk@sankeyturnrightbackward{sO{ }mm}{
581 % *(reverse), options, name, angle
582 \snk@errorifnotdefined{#3}
583 \beginngroup
584 \IfBooleanTF{#1}
585 {\edef\snk@angle{\fpeval{-1*#4}}}
586 {\edef\snk@angle{\fpeval{1*#4}}}
587 \edef\snk@anglesign{\fpeval{sign(\snk@angle)}}
588 \ifnumgreater{\snk@anglesign}{-1}{
589 \snk@init@move{#2}{#3}
590 \edef\orient{\snk@normalize@angle{\sankeygetnodeorient{\name}+\snk@angle}}
591 \snk@getpos\snk@c{$(\name.right)!-\snk@minradius!(\name.left)$}
592 \snk@getpos\pos{$(\snk@c)!!\snk@angle:(\name.center)$}
593 \snk@makenode{ }
594 % fill the region
595 \path[/sankey/fill] let
596 \p1=(\name.left),\p2=(\name.right),
597 \p3=(\snk@oldname.left),\p4=(\snk@oldname.right),
598 \n1={\sankeyqytytolen{\qty}},
599 \n{maxr}={\snk@minradius+\n1}, \n{minr}={\snk@minradius}
600 in
601 (\p1) arc(\orient+90:\orient-\snk@angle+90:\n{maxr}) -- (\p3) --
602 (\p4) arc(\orient-\snk@angle+90:\orient+90:\n{minr}) -- (\p2) -- cycle;

```



```

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

```

`\snk@sankeyturnleft` The `\snk@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`.

```

618 \NewDocumentCommand\snk@sankeyturnleft{sO{mm}}{
619 % *(reverse), options, name, angle
620 \snk@errorifnotdefined{#3}
621 \begingroup
622 \IfBooleanTF{#1}
623 {\edef\snk@angle{\fpeval{-1*#4}}}
624 {\edef\snk@angle{\fpeval{1*#4}}}
625 \edef\snk@anglesign{\fpeval{sign(\snk@angle)}}
626 \ifnumgreater{\snk@anglesign}{-1}{
627 \snk@init@move[#2]{#3}
628 \edef\orient{\snk@normalize@angle{\sankeygetnodeorient{\name}+\snk@angle}}
629 \snk@getpos\snk@c{$(\name.left)!-\snk@minradius!(\name.right)$}
630 \snk@getpos\pos{$(\snk@c)!!\snk@angle:(\name.center)$}
631 \snk@makenode{}
632 % fill the region
633 \path[/sankey/fill] let
634 \p1=(\snk@oldname.left),\p2=(\snk@oldname.right),
635 \p3=(\name.left),\p4=(\name.right),
636 \n1={\sankeyqtytolen{\qty}},
637 \n{maxr}={\snk@minradius+\n1}, \n{minr}={\snk@minradius}
638 in
639 (\p1) arc(\orient-\snk@angle-90:\orient-90:\n{minr}) -- (\p3) --
640 (\p4) arc(\orient-90:\orient-\snk@angle-90:\n{maxr}) -- (\p2) -- cycle;
641 % draw left and right borders
642 \path[/sankey/draw] 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})
649 (\p4) arc(\orient-90:\orient-\snk@angle-90:\n{maxr});
650 \snk@show@debug{\name}
651 }{
652 \sankeyturnleftbackward[#2]{#3}{-1*\snk@angle}
653 }
654 \endgroup
655 }

```

`\snk@sankeyturnleftbackward` The `\snk@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`.

```

656 \NewDocumentCommand\snk@sankeyturnleftbackward{sO{mm}}{
657 % *(reverse), options, name, angle
658 \snk@errorifnotdefined{#3}
659 \begingroup
660 \IfBooleanTF{#1}
661 {\edef\snk@angle{\fpeval{-1*#4}}}
662 {\edef\snk@angle{\fpeval{1*#4}}}
663 \edef\snk@anglesign{\fpeval{sign(\snk@angle)}}

```

```

664 \ifnumgreater{\snk@anglesign}{-1}{
665   \snk@init@move{#2}{#3}
666   \edef\orient{\snk@normalize@angle{\sankeygetnodeorient{\name}-\snk@angle}}
667   \snk@getpos\snk@c{$(\name.left)!-\snk@minradius!(\name.right)$}
668   \snk@getpos\pos{$(\snk@c)!!-\snk@angle:(\name.center)$}
669   \snk@makenode{}
670   % fill the region
671   \path[/sankey/fill] let
672     \p1=(\name.left), \p2=(\name.right),
673     \p3=(\snk@oldname.left), \p4=(\snk@oldname.right),
674     \n1={\sankeyqtytolen{\qty}},
675     \n{maxr}={\snk@minradius+\n1}, \n{minr}={\snk@minradius}
676   in
677     (\p1) arc(\orient-90:\orient+\snk@angle-90:\n{minr}) -- (\p3) --
678     (\p4) arc(\orient+\snk@angle-90:\orient-90:\n{maxr}) -- (\p2) -- cycle;
679   % draw left and right borders
680   \path[/sankey/draw] let
681     \p1=(\name.left), \p2=(\name.right),
682     \p3=(\snk@oldname.left), \p4=(\snk@oldname.right),
683     \n1={\sankeyqtytolen{\qty}},
684     \n{maxr}={\snk@minradius+\n1}, \n{minr}={\snk@minradius}
685   in
686     (\p1) arc(\orient-90:\orient+\snk@angle-90:\n{minr})
687     (\p4) arc(\orient+\snk@angle-90:\orient-90:\n{maxr});
688   \snk@show@debug{\name}
689 }{
690   \sankeyturnleft[#2]{#3}{-1*\snk@angle}
691 }
692 \endgroup
693 }

```

`\snk@sankeyturn` The `\snk@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`.

```

694 \NewDocumentCommand\snk@sankeyturn{s0{mm}}{%
695   % *(reverse), options, name, angle
696   \snk@errorifnotdefined{#3}
697   \begingroup
698   \edef\snk@anglesign{\fpeval{sign{#4}}}
699   \IfBooleanTF{#1}{
700     \ifnumgreater{\snk@anglesign}{-1}
701       {\sankeyturnleftbackward[#2]{#3}{#4}}
702       {\sankeyturnrightbackward[#2]{#3}{-1*#4}}
703   }{
704     \ifnumgreater{\snk@anglesign}{-1}
705       {\sankeyturnleft[#2]{#3}{#4}}
706       {\sankeyturnright[#2]{#3}{-1*#4}}
707   }
708   \endgroup
709 }

```

`\snk@sankeyfork` The `\snk@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.

```

710 \NewDocumentCommand\snk@sankeyfork{0{mm}}{%options, name, list of forks
711   \snk@errorifnotdefined{#2}
712   \begingroup
713   \sankeyset{#1}
714   \sankeynode[debug=false]{as={#2}, forked={#3}}
715   \foreach \qty/\snk@subnodename in {#3}{\sankeynode{as={\snk@subnodename}}}
716   \endgroup
717 }

```

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

```

718 \NewExpandableDocumentCommand\snk@sankeyqtytolen{m}{%
719   \fpeval{(#1)/\snk@totalqty*\snk@totallen}%
720 }

```

14.6 The `sankeydiagram` environment

`sankeydiagram` (*env.*) The `sankeydiagram` environment allows the creation of Sankey diagrams.

`\sankeyadvance` It defines locally all the macros used by a Sankey diagram.

`\sankeydubins` Then it applies the `@initial options` and `every diagram` styles (in this order) and
`\sankeyend` applies all the keys provided in its optional argument.

`\sankeyfork`

`\sankeynodealias` 721 \NewDocumentEnvironment{sankeydiagram}{0}{}

722 \NewCommandCopy{\sankeyadvance}{\snk@sankeyadvance}

`\sankeynodeend` 723 \NewCommandCopy{\sankeydubins}{\snk@sankeydubins}

724 \NewCommandCopy{\sankeyend}{\snk@sankeyend}

`\sankeynodestart` 725 \NewCommandCopy{\sankeyfork}{\snk@sankeyfork}

`\sankeynode` 726 \NewCommandCopy{\sankeynodealias}{\snk@sankeynodealias}

727 \NewCommandCopy{\sankeynodeend}{\snk@sankeynodeend}

`\sankeyoutin` 728 \NewCommandCopy{\sankeynodestart}{\snk@sankeynodestart}

729 \NewCommandCopy{\sankeynode}{\snk@sankeynode}

`\sankeyqtytolen` 730 \NewCommandCopy{\sankeyoutin}{\snk@sankeyoutin}

`\sankeystart` 731 \NewCommandCopy{\sankeyqtytolen}{\snk@sankeyqtytolen}

732 \NewCommandCopy{\sankeystart}{\snk@sankeystart}

`\sankeyturnleftbackward` 733 \NewCommandCopy{\sankeyturnleftbackward}{\snk@sankeyturnleftbackward}

`\sankeyturnleft` 734 \NewCommandCopy{\sankeyturnleft}{\snk@sankeyturnleft}

735 \NewCommandCopy{\sankeyturnrightbackward}{\snk@sankeyturnrightbackward}

`\sankeyturnrightbackward` 736 \NewCommandCopy{\sankeyturnright}{\snk@sankeyturnright}

737 \NewCommandCopy{\sankeyturn}{\snk@sankeyturn}

`\sankeyturnright`

738 \sankeyset{

`\sankeyturn` 739 @initial options,

740 every diagram,

741 % user values

742 #1}

743 }

744 {} % empty but mandatory ! :-)

14.7 Predefined start and end styles

The `none` style.

```
745 \sankeyset{
746   new start style={none}{}{ },
747   new end style={none}{}{ },
748 }
```

The `simple` style.

```
749 \sankeyset{
750   new start style={simple}{
751     (\name.left) -- ([xshift=-.5\pgflinewidth]\name.left)
752     -- ([xshift=-.5\pgflinewidth]\name.right) -- (\name.right) -- cycle
753   }{
754     (\name.left) -- ([xshift=-.5\pgflinewidth]\name.left)
755     -- ([xshift=-.5\pgflinewidth]\name.right) -- (\name.right)
756   },
757   new end style={simple}{
758     (\name.left) -- ([xshift=2mm]\name.center)
759     -- (\name.right) -- cycle
760   }{
761     (\name.left) -- ([xshift=2mm]\name.center) -- (\name.right)
762   },
763 }
```

The `arrow` style.

```
764 \sankeyset{
765   new start style={arrow}{
766     (\name.left) -- ++(-10pt,0) -- ([xshift=-10pt/6]\name.center)
767     -- ([xshift=-10pt]\name.right) -- (\name.right) -- cycle
768   }{
769     (\name.left) -- ++(-10pt,0) -- ([xshift=-10pt/6]\name.center)
770     -- ([xshift=-10pt]\name.right) -- (\name.right)
771   },
772   new end style={arrow}{
773     (\name.left) -- ([yshift=1mm]\name.left)
774     -- ([xshift=10pt]\name.center) -- ([yshift=-1mm]\name.right)
775     -- (\name.right) -- cycle
776   }{
777     (\name.left) -- ([yshift=1mm]\name.left)
778     -- ([xshift=10pt]\name.center) -- ([yshift=-1mm]\name.right)
779     -- (\name.right)
780   },
781 }
```

15 tikzlibrarydubins.code.tex

Not yet documented nor commented...

```
\tikzlibrarydubins@version
\tikzlibrarydubins@date
782 \def\tikzlibrarydubins@version{v3.0.2}
783 \def\tikzlibrarydubins@date{2025/01/10}

784 \usetikzlibrary{calc}
785 \RequirePackage{etoolbox}
786 \RequirePackage{xfp}
787
788 \newbool{dubinspathreverse}

\ifpgfmathcond

789 \def\ifpgfmathcond#1{%
790   \pgfmathparse{(#1)?1:0}%
791   \ifnumequal{\pgfmathresult}{1}%
792 }

\dbp@getxy

793 \def\dbp@getxy#1#2#3{%
794   \tikz@scan@one@point\pgfutil@firstofone(#3)\relax%
795   \edef#1{\the\pgf@x}%
796   \edef#2{\the\pgf@y}%
797 }

\dbp@anglebetween

798 \def\dbp@anglebetween#1#2#3{%macro, s, t
799   \dbp@getxy\dbp@ax\dbp@ay{#2}
800   \dbp@getxy\dbp@bx\dbp@by{#3}
801   \pgfmathsetmacro#1{atan2(\dbp@by-\dbp@ay,\dbp@bx-\dbp@ax)}
802 }

\dbp@distancebetween

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{\fppeval{\sqrt{
807     (\dbp@bx-\dbp@ax)*(\dbp@bx-\dbp@ax)%
808     +(\dbp@by-\dbp@ay)*(\dbp@by-\dbp@ay)%
809     }}}%
810 }

\dbp@rsr

811 \newcommand\dbp@rsr{% s, sa, t, ta, as, len, at, r
812   let
813     \p{tr}=(\shift=(\dbp@angb-90:\dbp@radius pt)]\dbp@b),
814     \n1=(\dbp@anga+90},
815     \n2=(\dbp@angb+90},
816     \n3={\n2+\dbp@lastangle}
817   in
818   arc(\n1:\n1-\dbp@firstangle:\dbp@radius pt)
819   -- (\shift=(\p{tr}]]\n3:\dbp@radius pt)
820   arc(\n3:\n2:\dbp@radius pt)
821 }
```

\dbp@lsl

```
822 \newcommand\dbp@lsl{% s, sa, t, ta, as, len, at, r
823   let
824   \p{t1}=(\p{t1}=\p{t1}+\p{t1})\dbp@b),
825   \n1={\dbp@anga-90}, \n2={\n1+\dbp@firstangle},
826   \n3={\dbp@angb-90}, \n4={\n3-\dbp@lastangle}
827   in
828   arc(\n1:\n2:\dbp@lradius pt)
829   -- ([shift={(\p{t1})}]\n4:\dbp@lradius pt)
830   arc(\n4:\n3:\dbp@lradius pt)
831 }
```

\dbp@rsl

```
832 \newcommand\dbp@rsl{% s, sa, t, ta, as, len, at, r
833   let
834   \p{t1}=(\p{t1}=\p{t1}+\p{t1})\dbp@b),
835   \n1={\dbp@anga+90}, \n2={\n1-\dbp@firstangle},
836   \n3={\dbp@angb-90}, \n4={\n3-\dbp@lastangle}
837   in
838   arc(\n1:\n2:\dbp@rradius pt)
839   -- ([shift={(\p{t1})}]\n4:\dbp@lradius pt)
840   arc(\n4:\n3:\dbp@lradius pt)
841 }
```

\dbp@lsr

```
842 \newcommand\dbp@lsr{% s, sa, t, ta, as, len, at, r
843   let
844   \p{tr}=(\p{tr}=\p{tr}+\p{tr})\dbp@b),
845   \n1={\dbp@anga-90}, \n2={\n1+\dbp@firstangle},
846   \n3={\dbp@angb+90}, \n4={\n3+\dbp@lastangle}
847   in
848   arc(\n1:\n2:\dbp@lradius pt)
849   -- ([shift={(\p{tr})}]\n4:\dbp@rradius pt)
850   arc(\n4:\n3:\dbp@rradius pt)
851 }
```

\dbp@lr1

```
852 \newcommand\dbp@lr1{% s, sa, t, ta, as, ai, at, r
853   let
854   \n1={\dbp@anga-90}, \n2={\n1+\dbp@firstangle},
855   \n3={\dbp@angb-90}, \n4={\n3-\dbp@lastangle}
856   in
857   arc(\n1:\n2:\dbp@lradius pt)
858   arc(\n2+180:\n2+180-\dbp@midparam:\dbp@rradius pt)
859   arc(\n4:\n3:\dbp@lradius pt)
860 }
```

\dbp@r1r

```
861 \newcommand\dbp@r1r{% s, sa, t, ta, as, ai, at, r
862   let
863   \n1={\dbp@anga+90}, \n2={\n1-\dbp@firstangle},
864   \n3={\dbp@angb+90}, \n4={\n3+\dbp@lastangle}
865   in
866   arc(\n1:\n2:\dbp@rradius pt)
867   arc(\n2+180:\n2+180+\dbp@midparam:\dbp@lradius pt)
868   arc(\n4:\n3:\dbp@rradius pt)
869 }
```

\dbp@rev@lsl

```
870 \newcommand\dbp@rev@lsl{\dbp@rsr}
```

\dbp@rev@rsr

```
871 \newcommand\dbp@rev@rsr{\dbp@lsl}
```

```

\dbp@rev@lsl
872 \newcommand\dbp@rev@lsl{\dbp@lsl}

\dbp@rev@rsl
873 \newcommand\dbp@rev@rsl{\dbp@rsl}

\dbp@rev@lrl
874 \newcommand\dbp@rev@lrl{\dbp@r1r}

\dbp@rev@r1r
875 \newcommand\dbp@rev@r1r{\dbp@lrl}

\dubinspath
876 \newcommand\dubinspath[1]{%
877   \pgfextra{
878     \dubinspathset{#1}
879     \ifbool{dubinspathreverse}{
880       \edef\dbp@newa{\dbp@b}
881       \edef\dbp@newb{\dbp@a}
882       \pgfmathsetmacro\dbp@newanga{180+\dbp@angb}
883       \pgfmathsetmacro\dbp@newangb{180+\dbp@anga}
884       \edef\dbp@newfirstangle{\dbp@lastangle}
885       \edef\dbp@newlastangle{\dbp@firstangle}
886       \edef\dbp@newmethod{rev@\dbp@method}
887       \edef\dbp@newlradius{\dbp@rradius}
888       \edef\dbp@newrradius{\dbp@lradius}
889       \dubinspathset{
890         start point=\dbp@newa,
891         end point=\dbp@newb,
892         start angle=\dbp@newanga,
893         end angle=\dbp@newangb,
894         first angle=\dbp@newfirstangle,
895         last angle=\dbp@newlastangle,
896         left and right minimum radii=\dbp@newlradius pt and \dbp@newrradius pt,
897         method=\dbp@newmethod,
898       }
899     }{}
900   }
901   \csname dbp@\dbp@method\endcsname%
902 }

\dbp@store
\dbp@get
903 \def\dbp@store#1#2{%
904   \expandafter\xdef\csname dbp@store@#1@#2\endcsname%
905   {\csname dbp@#2\endcsname}%
906 }
907 \def\dbp@get#1#2{%
908   \csname dbp@store@#1@#2\endcsname%
909 }

\dbp@setparams
910 \def\dbp@setparams#1#2#3#4#5{%
911   % method, length, first angle, middle param, last angle
912   \edef\dbp@method{#1}
913   \edef\dbp@length{#2}
914   \edef\dbp@firstangle{#3}
915   \edef\dbp@middleparam{#4}
916   \edef\dbp@lastangle{#5}
917   \ifdef{\dbp@storename}{
918     \foreach \p in {method,length,firstangle,middleparam,lastangle}{
919       \dbp@store{\dbp@storename}{\p}
920     }
921   }{}
922 }

```

`\dbp@updateparams`

```
923 \def\dbp@updateparams#1#2#3#4#5{
924   \ifpgfmathcond{#2<\dbp@length}{
925     \dbp@setparams{#1}{#2}{#3}{#4}{#5}
926   }{}
927 }

928 \tikzset{
929   dubins path/.is family,
930   dubins path,
931   start point/.store in=\dbp@a,
932   start angle/.store in=\dbp@anga,
933   end point/.store in=\dbp@eb,
934   end angle/.store in=\dbp@angb,
935   store/.store in=\dbp@storename,
936   use store/.style={
937     method=\dbp@get{#1}{method},
938     first angle=\dbp@get{#1}{firstangle},
939     last angle=\dbp@get{#1}{lastangle},
940     middle param=\dbp@get{#1}{middleparam},
941   },
942   minimum radius/.code={
943     \pgfmathsetmacro\dbp@radius{#1}
944     \pgfmathsetmacro\dbp@rradius{#1}
945     \pgfmathsetmacro\dbp@lradius{#1}
946   },
947   left and right minimum radii/.code args={#1 and #2}{
948     \pgfmathsetmacro\dbp@lradius{#1}
949     \pgfmathsetmacro\dbp@rradius{#2}
950     \pgfmathsetmacro\dbp@radius{(\dbp@lradius + \dbp@rradius)/2}
951   },
952   method/.store in=\dbp@method,
953   first angle/.store in=\dbp@firstangle,
954   last angle/.store in=\dbp@lastangle,
955   middle param/.store in=\dbp@midparam,
956   reverse/.is if=dubinspathreverse,
957 }
```

`\dubinspathset`

```
958 \newcommand\dubinspathset[1]{\tikzset{dubins path,#1}}
```

`\dubinspathcalc`

```
959 \newcommand\dubinspathcalc[1]{%
960   \begingroup
961   \dubinspathset{#1}
962   \tikzset{
963     declare function={
964       angtodist(\dbp@a,\dbp@r)={abs(\dbp@a)*.01745329*\dbp@r};
965       modangr(\dbp@a,\dbp@b)={
966         (
967           Mod(\dbp@a,360)<Mod(\dbp@b,360)
968           ?
969           Mod(\dbp@a,360)
970           :
971           Mod(\dbp@a,360)-360+\dbp@b-Mod(\dbp@b,360)
972         )
973       };
974       modangl(\dbp@a,\dbp@b)={
975         (
976           Mod(\dbp@a,360)<Mod(\dbp@b,360)
977           ?
978           Mod(\dbp@a,360)+360
979           :
980           Mod(\dbp@a,360)+(\dbp@b)-Mod(\dbp@b,360)
981         )
982       },
983     }
984   }
985   \pgfmathsetmacro\dbp@radius{\dbp@rradius}
986   \pgfmathsetmacro\dbp@anga{mod((\dbp@anga)+180,360)-180}
987   \pgfmathsetmacro\dbp@angb{mod((\dbp@angb)+180,360)-180}
988   \path
989   let
990   \p{a}=(\dbp@a),
```



```

991 \p{b}=(\dbp@b),
992 \p{ar}=(\p{a}) + (\dbp@anga-90:\dbp@radius pt)$,
993 \p{al}=(\p{a}) + (\dbp@anga+90:\dbp@radius pt)$,
994 \p{br}=(\p{b}) + (\dbp@angb-90:\dbp@radius pt)$,
995 \p{bl}=(\p{b}) + (\dbp@angb+90:\dbp@radius pt)$
996 in \pgfextra{
997   \pgfinterruptpath
998
999   % RSR (ar and br)
1000   \dbp@anglebetween\dbp@rsrarbr{\p{ar}}{\p{br}}
1001   \dbp@distancebetween\dbp@rsrdarbr{\p{ar}}{\p{br}}
1002   \pgfmathsetmacro\dbp@rsrangone{Mod(\dbp@anga-\dbp@rsrarbr,360)}
1003   \pgfmathsetmacro\dbp@rsrangtwo{Mod(\dbp@rsrarbr-\dbp@angb,360)}
1004   \pgfmathsetmacro\dbp@rsrlen{\dbp@rsrdarbr}
1005   \pgfmathsetmacro\dbp@rsrdist{
1006     angtodist(\dbp@rsrangone,\dbp@radius)
1007     +\dbp@rsrlen
1008     +angtodist(\dbp@rsrangtwo,\dbp@radius)
1009   }
1010   \dbp@setparams%
1011   {rsr}{\dbp@rsrdist}{\dbp@rsrangone}{\dbp@rsrlen}{\dbp@rsrangtwo}
1012
1013   % LSL (al and bl)
1014   \dbp@anglebetween\dbp@lslalbl{\p{al}}{\p{bl}}
1015   \dbp@distancebetween\dbp@lsldalbl{\p{al}}{\p{bl}}
1016   \pgfmathsetmacro\dbp@lslangone{mod(\dbp@lslalbl-\dbp@anga+720,360)}
1017   \pgfmathsetmacro\dbp@lslangtwo{mod(\dbp@angb-\dbp@lslalbl+720,360)}
1018   \pgfmathsetmacro\dbp@lsllen{\dbp@lsldalbl}
1019   \pgfmathsetmacro\dbp@lsldist{
1020     angtodist(\dbp@lslangone,\dbp@radius)
1021     +\dbp@lsllen
1022     +angtodist(\dbp@lslangtwo,\dbp@radius)
1023   }
1024   \dbp@updateparams%
1025   {ls1}{\dbp@lsldist}{\dbp@lslangone}{\dbp@lsllen}{\dbp@lslangtwo}
1026
1027   % RSL (ar and bl)
1028   \dbp@distancebetween\dbp@rsldarbl{\p{ar}}{\p{bl}}
1029   \pgfmathtruncatemacro\dbp@rslok{(\dbp@rsldarbl>=2*\dbp@radius)?1:0}
1030   \ifnumequal{\dbp@rslok}{1}{
1031     \dbp@anglebetween\dbp@rslarbl{\p{ar}}{\p{bl}}
1032     \pgfmathsetmacro\dbp@rslanglesup{
1033       asin(\dbp@radius/\dbp@rsldarbl*2)}
1034     \pgfmathsetmacro\dbp@rslangone
1035     {Mod(\dbp@anga-\dbp@rslarbl+\dbp@rslanglesup,360)}
1036     \pgfmathsetmacro\dbp@rslangtwo
1037     {Mod(\dbp@angb-\dbp@rslarbl+\dbp@rslanglesup,360)}
1038     \pgfmathsetmacro\dbp@rsllen{veclen(\dbp@rsldarbl,\dbp@radius)}
1039     \pgfmathsetmacro\dbp@rsldist{
1040       angtodist(\dbp@rslangone,\dbp@radius)
1041       +\dbp@rsllen
1042       +angtodist(\dbp@rslangtwo,\dbp@radius)
1043     }
1044
1045     \dbp@updateparams%
1046     {rs1}{\dbp@rsldist}{\dbp@rslangone}{\dbp@rsllen}{\dbp@rslangtwo}
1047   }{ }
1048
1049   % LSR (al and br)
1050   \dbp@distancebetween\dbp@lsrdalbr{\p{al}}{\p{br}}
1051   \pgfmathtruncatemacro\dbp@lsrok{(\dbp@lsrdalbr>=2*\dbp@radius)?1:0}
1052   \ifnumequal{\dbp@lsrok}{1}{
1053     \dbp@anglebetween\dbp@lsralbr{\p{al}}{\p{br}}
1054     \pgfmathsetmacro\dbp@lsranglesup{
1055       asin(\dbp@radius/\dbp@lsrdalbr*2)}
1056     \pgfmathsetmacro\dbp@lsrangone
1057     {Mod(\dbp@lsralbr+\dbp@lsranglesup-\dbp@anga,360)}
1058     \pgfmathsetmacro\dbp@lsrangtwo
1059     {Mod(\dbp@lsralbr+\dbp@lsranglesup-\dbp@angb,360)}
1060     \pgfmathsetmacro\dbp@lsrlen{veclen(\dbp@lsrdalbr,\dbp@radius)}
1061     \pgfmathsetmacro\dbp@lsrdist{
1062       angtodist(\dbp@lsrangone,\dbp@radius)
1063       +\dbp@lsrlen
1064       +angtodist(\dbp@lsrangtwo,\dbp@radius)
1065     }
1066     \dbp@updateparams%
1067     {lsr}{\dbp@lsrdist}{\dbp@lsrangone}{\dbp@lsrlen}{\dbp@lsrangtwo}
1068   }{ }
1069

```

```

1070 % LRL (a1 and b1)
1071 \dbp@distancebetween\dbp@lrlalbl{\p{a1}}{\p{b1}}
1072 \pgfmathtruncatemacro\dbp@lrlrok{(\dbp@lrlalbl<=4*\dbp@radius)?1:0}
1073 \ifnumequal{\dbp@lrlrok}{1}{
1074   \dbp@anglebetween\dbp@lrlalbl{\p{a1}}{\p{b1}}
1075   \pgfmathsetmacro\dbp@lrlangsup{acos(\dbp@lrlalbl/\dbp@radius/4)}
1076   \pgfmathsetmacro\dbp@lrlangone{
1077     modangl(\dbp@lrlalbl+\dbp@lrlangsup,\dbp@anga-90)-(\dbp@anga-90)}
1078   \pgfmathsetmacro\dbp@lrlangtwo{%
1079     (\dbp@angb-90)-modangr(\dbp@lrlalbl+180-\dbp@lrlangsup,\dbp@angb-90)}
1080   \pgfmathsetmacro\dbp@lrlangthree{360-2*(90-\dbp@lrlangsup)}
1081   \pgfmathsetmacro\dbp@lrlldist{
1082     angtodist(\dbp@lrlangone,\dbp@radius)
1083     +angtodist(\dbp@lrlangthree,\dbp@radius)
1084     +angtodist(\dbp@lrlangtwo,\dbp@radius)
1085   }
1086   \dbp@updateparams%
1087   {lrl}{\dbp@lrlldist}{\dbp@lrlangone}{\dbp@lrlangthree}{\dbp@lrlangtwo}
1088 }{ }
1089
1090 % RLR (ar and br)
1091 \dbp@distancebetween\dbp@lrlrarbr{\p{ar}}{\p{br}}
1092 \pgfmathtruncatemacro\dbp@lrlrok{(\dbp@lrlrarbr<=4*\dbp@radius)?1:0}
1093 \ifnumequal{\dbp@lrlrok}{1}{
1094   \dbp@anglebetween\dbp@lrlrarbr{\p{ar}}{\p{br}}
1095   \pgfmathsetmacro\dbp@lrlangsup{acos(\dbp@lrlrarbr/\dbp@radius/4)}
1096   \pgfmathsetmacro\dbp@lrlangone{
1097     (\dbp@anga+90)-modangr(\dbp@lrlrarbr-\dbp@lrlangsup,\dbp@anga+90)}
1098   \pgfmathsetmacro\dbp@lrlangtwo{%
1099     modangl(\dbp@lrlrarbr+180+\dbp@lrlangsup,\dbp@angb+90)-(\dbp@angb+90)}
1100   \pgfmathsetmacro\dbp@lrlangthree{360-2*(90-\dbp@lrlangsup)}
1101   \pgfmathsetmacro\dbp@lrlldist{
1102     angtodist(\dbp@lrlangone,\dbp@radius)
1103     +angtodist(\dbp@lrlangthree,\dbp@radius)
1104     +angtodist(\dbp@lrlangtwo,\dbp@radius)
1105   }
1106   \dbp@updateparams%
1107   {rlr}{\dbp@lrlldist}{\dbp@lrlangone}{\dbp@lrlangthree}{\dbp@lrlangtwo}
1108 }{ }
1109
1110 \endpgfinterruptpath
1111 };
1112 \endgroup
1113 }

```

16 Change History

v1.0 – 2016/03/06	\snk@sankeyturnrightbackward: New macro	56
General: First version		44
v2.0 – 2021/01/27	v3.0.1 – 2022/02/04	
General: First public version (on CTAN)	General: Add instructions for compiling and installing the package	43
v3.0 – 2021/03/14	Fix options used by the new version of siunitx package	43
General: Add keys to fork a Sankey node during its creation		19
Add the reproduction of an example from Google Charts documentation	v3.0.2 – 2025/01/10	
Better naming rule	sankeydiagram : Simplification of the environment by copying internal macros	59
Fix bad names in second example		28
Use .ins and .dtx files	v3.0.2 – 2025/10/10	
\snk@sankeyturn : Simplification by using new <i>turn</i> macros	General: Fixed extra parentheses in some code examples	58
\snk@sankeyturnleft : New macro	Update for new version of ltxdoc class	57
\snk@sankeyturnleftbackward : New macro		57
\snk@sankeyturnright : New macro	\snk@init@move : Factorization of the node move initialization code	56

17 Index

Italic numbers refer page where the corresponding entry is described; Underlined numbers refer to the code line of the definition; Roman numbers refer to the code lines where the entry is used.

A	
anchor (key)	<i>5</i>
angle (key)	<i>5</i>
as (key)	<i>5</i>
at (key)	<i>5</i>
D	
\dbp@a	881, 931, 964, 965, 967, 969, 971, 974, 976, 978, 980, 990
\dbp@anga	814, 825, 835, 845, 854, 863, 883, 932, 986, 992, 993, 1002, 1016, 1035, 1057, 1077, 1097
\dbp@angb	813, 815, 824, 826, 834, 836, 844, 846, 855, 864, 882, 934, 987, 994, 995, 1003, 1017, 1037, 1059, 1079, 1099
\dbp@anglebetween	394, <u>798</u> , 1000, 1014, 1031, 1053, 1074, 1094
\dbp@ax	799, 801, 804, 807
\dbp@ay	799, 801, 804, 808
\dbp@eb	813, 824, 834, 844, 880, 933, 965, 967, 971, 974, 976, 980, 991
\dbp@bx	800, 801, 805, 807
\dbp@by	800, 801, 805, 808
\dbp@distancebetween	<u>803</u> , 1001, 1015, 1028, 1050, 1071, 1091
\dbp@firstangle	818, 825, 835, 845, 854, 863, 885, 914, 953
\dbp@get	<u>903</u> , 937, 938, 939, 940
\dbp@getxy	<u>793</u> , 799, 800, 804, 805
\dbp@lastangle	816, 826, 836, 846, 855, 864, 884, 916, 954
\dbp@length	913, 924
\dbp@lradius	828, 829, 830, 839, 840, 848, 857, 859, 867, 888, 945, 948, 950
\dbp@lrl	<u>852</u> , 875
\dbp@lrlalbl	1074, 1077, 1079
\dbp@lrlangone	1076, 1082, 1087
\dbp@lrlangsup	1075, 1077, 1079, 1080
\dbp@lrlangthree	1080, 1083, 1087
\dbp@lrlangtwo	1078, 1084, 1087
\dbp@lrladalbl	1071, 1072, 1075
\dbp@lrlldist	1081, 1087
\dbp@lrllok	1072, 1073
\dbp@lsl	<u>822</u> , 871
\dbp@lslalbl	1014, 1016, 1017
\dbp@lslangone	1016, 1020, 1025
\dbp@lslangtwo	1017, 1022, 1025
\dbp@lsladalbl	1015, 1018
\dbp@lslldist	1019, 1025
\dbp@lslllen	1018, 1021, 1025
\dbp@lsr	<u>842</u> , 872
\dbp@lsralbr	1053, 1057, 1059
\dbp@lsranglesup	1054, 1057, 1059
\dbp@lsrangone	1056, 1062, 1067
\dbp@lsrangtwo	1058, 1064, 1067
\dbp@lsrdalbr	1050, 1051, 1055, 1060
\dbp@lsrdist	1061, 1067
\dbp@lsrlen	1060, 1063, 1067
\dbp@lsrok	1051, 1052
\dbp@method	886, 901, 912, 952
\dbp@middleparam	915
\dbp@midparam	858, 867, 955
\dbp@newa	880, 890
\dbp@newanga	882, 892
\dbp@newangb	883, 893
\dbp@newb	881, 891
\dbp@newfirstangle	884, 894
\dbp@newlastangle	885, 895
\dbp@newlradius	887, 896
\dbp@newmethod	886, 897
\dbp@newrradius	888, 896
\dbp@r	964
\dbp@radius	813, 824, 834, 844, 943, 950, 985, 992, 993, 994, 995, 1006, 1008, 1020, 1022, 1029, 1033, 1038, 1040, 1042, 1051, 1055, 1060, 1062, 1064, 1072, 1075, 1082, 1083, 1084, 1092, 1095, 1102, 1103, 1104
\dbp@rev@lrl	<u>874</u>
\dbp@rev@lsl	<u>870</u>
\dbp@rev@lsr	<u>872</u>
\dbp@rev@lrl	<u>875</u>
\dbp@rev@rsl	<u>873</u>
\dbp@rev@rsr	<u>871</u>
\dbp@rlr	<u>861</u> , 874
\dbp@rlrangone	1096, 1102, 1107
\dbp@rlrangsup	1095, 1097, 1099, 1100
\dbp@rlrangthree	1100, 1103, 1107
\dbp@rlrangtwo	1098, 1104, 1107
\dbp@rlrarbr	1094, 1097, 1099
\dbp@rlrdarbr	1091, 1092, 1095
\dbp@rlrdist	1101, 1107
\dbp@rlrok	1092, 1093
\dbp@rradius	818, 819, 820, 838, 849, 850, 858, 866, 868, 887, 944, 949, 950, 985
\dbp@rsl	<u>832</u> , 873
\dbp@rslanglesup	1032, 1035, 1037
\dbp@rslangone	1034, 1040, 1046
\dbp@rslangtwo	1036, 1042, 1046
\dbp@rslarbl	1031, 1035, 1037
\dbp@rslarbl	1028, 1029, 1033, 1038
\dbp@rslldist	1039, 1046
\dbp@rslllen	1038, 1041, 1046
\dbp@rsllok	1029, 1030
\dbp@rsr	<u>811</u> , 870
\dbp@rsrangone	1002, 1006, 1011
\dbp@rsrangtwo	1003, 1008, 1011
\dbp@rsrarbr	1000, 1002, 1003
\dbp@rsrdarbr	1001, 1004
\dbp@rsrdist	1005, 1011
\dbp@rsrlen	1004, 1007, 1011
\dbp@setparams	<u>910</u> , 925, 1010
\dbp@store	<u>903</u> , 919
\dbp@storename	917, 919, 935
\dbp@updateparams <u>923</u> , 1024, 1045, 1066, 1086, 1106
debug (key)	<i>5</i>
draw/.style (key)	<i>4</i>
\dubinspath	356, 357, 360, 361, <u>876</u>
\dubinspathcalc	341, <u>959</u>
\dubinspathset	334, 342, 878, 889, <u>958</u> , 961
E	
end (key)	<i>7</i>

end_style (key)	4		
environments:			
sankeydiagram	3, 721		
every_diagram (key)	3		
every_node/.style (key)	6		
			Q
		\qty	155 , 240, 244, 265, 275, 279, 295, 333, 338, 346, 351, 370, 371, 398, 476, 489, 501, 558, 568, 598, 607, 636, 645, 674, 683, 715
		quantity (key)	5
			F
fill/.style (key)	4		
fork_anchor (key)	19		
forked (key)	19		
			I
\ifpgfmathcond	789 , 924		
			K
Keys:			
anchor	5		
angle	5		
as	5		
at	5		
debug	5		
draw/.style	4		
end	7		
end_style	4		
every_diagram	3		
every_node/.style	6		
fill/.style	4		
fork_anchor	19		
forked	19		
minimum_radius	4		
name	5		
new_end_style	5		
new_start_style	5		
outin_steps	4		
quantity	5		
ratio	4		
ratio_length	4		
ratio_quantity	4		
rotate	4		
start	7		
start_style	4		
			M
minimum_radius (key)	4		
			N
\name	147, 155 , 239, 240, 241, 243, 245, 246, 264, 266, 283, 286, 295, 297, 299, 303, 309, 467, 474, 487, 498, 500, 501, 514, 517, 518, 520, 521, 528, 531, 532, 534, 535, 537, 550, 551, 552, 557, 567, 574, 590, 591, 592, 596, 605, 612, 628, 629, 630, 635, 644, 650, 666, 667, 668, 672, 681, 688, 751, 752, 754, 755, 758, 759, 761, 766, 767, 769, 770, 773, 774, 775, 777, 778, 779		
name (key)	5		
new_end_style (key)	5		
new_start_style (key)	5		
			O
\orient 155 , 241, 247, 267, 296, 297, 303, 475, 488, 550, 562, 563, 572, 573, 590, 601, 602, 610, 611, 628, 639, 640, 648, 649, 666, 677, 678, 686, 687			
outin_steps (key)	4		
			P
\pos	155 , 285, 299, 552, 592, 630, 668		
			R
ratio (key)	4		
ratio_length (key)	4		
ratio_quantity (key)	4		
rotate (key)	4		
			S
\sankeyadvance	9, 59, 722		
sankeydiagram (env.)	3, 721		
\sankeydubins	15, 59, 330, 723		
\sankeyend	18, 59, 724		
\sankeyfork	20, 59, 725		
\sankeygetnodeorient	8, 31 , 169, 222, 241, 309, 336, 337, 390, 406, 412, 420, 421, 429, 430, 439, 440, 447, 448, 457, 475, 488, 512, 526, 550, 590, 628, 666		
\sankeygetnodeqty	8, 29 , 168, 216, 240, 316, 317, 333, 370, 405, 411, 456, 476, 489, 501, 513, 527		
\sankeynode 5, 59, 268, 286, 288, 396, 403, 409, 481, 494, 510, 524, 714, 715, 729			
\sankeynodealias	21, 59, 500, 726		
\sankeynodeend	6, 59, 727		
\sankeynodestart	6, 59, 728		
\sankeyoutin	15, 59, 367, 730		
\sankeyqytolen	8, 59, 216, 244, 338, 346, 351, 371, 558, 568, 598, 607, 636, 645, 674, 683, 731		
\sankeyset	3, 66 , 68, 82, 86, 93, 98, 104, 111, 116, 119, 154, 155, 165, 174, 183, 194, 200, 211, 262, 332, 369, 461, 462, 473, 486, 497, 713, 738, 745, 749, 764		
\sankeystart	18, 59, 732		
\sankeyturn	10, 59, 737		
\sankeyturnleft	11, 59, 690, 705, 734		
\sankeyturnleftbackward	12, 59, 652, 701, 733		
\sankeyturnright	13, 59, 614, 706, 736		
\sankeyturnrightbackward	14, 59, 576, 702, 735		
\snk@added@values	252		
\snk@anchor	174 , 283, 298, 299		
\snk@angle	545, 546, 547, 550, 552, 562, 563, 572, 573, 576, 585, 586, 587, 590, 592, 601, 602, 610, 611, 614, 623, 624, 625, 628, 630, 639, 640, 648, 649, 652, 661, 662, 663, 666, 668, 677, 678, 686, 687, 690		
\snk@anglesign	547, 548, 587, 588, 625, 626, 663, 664, 698, 700, 704		
\snk@c	285, 286, 288, 551, 552, 591, 592, 629, 630, 667, 668		
\snk@checkquantities	314 , 330, 367		
\snk@curpt	417, 418, 421, 422, 426, 427, 430, 431, 436, 437, 440, 441, 444, 445, 448, 449		
\snk@diff	275, 276		
\snk@endstyle	104 , 310, 311		

