Package 'vwline'

October 12, 2022

Type Package

Title Draw Variable-Width Lines

Version 0.2-2

Author Paul Murrell

Maintainer Paul Murrell <paul@stat.auckland.ac.nz>

Description Provides R functions to draw lines and curves with the width of the curve allowed to vary along the length of the curve.

Depends grid

Imports grDevices, stats, polyclip, gridBezier

URL https://github.com/pmur002/vwline,
https://stattech.wordpress.fos.auckland.ac.nz/2017/05/19/2017-01-variable-width-lines-in-r/,
https://stattech.wordpress.fos.auckland.ac.nz/2017/06/07/2017-02-variable-width-line-ends-and-line-joins/,
https://stattech.wordpress.fos.auckland.ac.nz/2017/06/15/2017-03-offset-curves-for-variable-width-x-splines/,
https://stattech.wordpress.fos.auckland.ac.nz/2018/11/02/2018-11-variable-width-bezier-splines-in-r/

ByteCompile TRUE

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2019-07-25 08:50:02 UTC

R topics documented:

`edgePoints` *Calculate edge points*

Description

Calculate points on the edge or boundary of a variable-width line.

Usage

```
edgePoints(x, d, ...)
## S3 method for class 'vwcurveGrob'
edgePoints(x, d, which = c("left", "right"), direction = "forward", debug = FALSE, ...)
## S3 method for class 'vwlineGrob'
edgePoints(x, d, x0, y0, which = 1, direction = "forward", debug = FALSE, ...)
## S3 method for class 'vwXsplineGrob'
edgePoints(x, d, which = c("left", "right"), direction = "forward", debug = FALSE, ...)
## S3 method for class 'brushXsplineGrob'
edgePoints(x, d, x0, y0, which = 1, direction = "forward", debug = FALSE, ...)
## S3 method for class 'offsetXsplineGrob'
edgePoints(x, d, x0, y0, which = 1, direction = "forward", debug = FALSE, ...)
```

Arguments

`x`	A variable-width line grob.
`d`	A numeric vector or unit specifying locations along the boundary of the variable-width line.
`which`	For some methods, this is either `"left"` or `"right"` (or both) indicating which edge to find locations on. For other methods, this is a numeric, selecting which boundary to find locations on. See Details.
`direction`	Either `"forwards"` or `"backwards"` to indicate the direction of traversal for the edge.
`x0, y0`	A location used to determine the start point for traversing an edge.
`debug`	A logical indicating whether to draw graphical debugging information.
`...`	Additional arguments for methods.

Details

If the distance is numeric, it is assumed to be a proportion of the length of an edge.

What constitutes an edge varies between different methods: some methods produce distinct left and right edges (ignoring line endings), in which case locations can be found on either edge; other methods generally produce a single boundary (including line endings), but self-intersecting lines can produce additional boundaries.

For some methods, it is possible for a boundary to form loops, so an edge location is not guaranteed to be on the external boundary of the variable-width line.

For lines with distinct left and right edges, the forwards direction is the direction of the main curve. For lines with a single boundary, the forwards direction is anticlockwise.

For lines with a single boundary, the start point on the boundary is defined as the nearest point on the boundary to the location specified by `x0` and `y0`.

Value

For methods with a single boundary, a list with components `x` and `y`, giving locations on the edge of the variable-width line, and `tangent`, giving the tangent to the edge at each location.

For methods with distinct left and right edges, a list with components `left` and `right`, each a list as above.

Author(s)

Paul Murrell

See Also

`grid.vwcurve`, `grid.vwline`, `grid.vwXspline`, `grid.brushXspline`, `grid.offsetXspline`

Examples

```
grid.newpage()
x <- seq(.2, .8, length.out=100)
y <- .05*sin(seq(0, 2*pi, length.out=100))
w <- unit(seq(2, 10, length.out=100), "mm")
vwcg <- vwcurveGrob(x, y + 2/3, w, lineend="round",
                    gp=gpar(col="black"))
grid.draw(vwcg)
epts <- edgePoints(vwcg, 0:9/9)
grid.circle(epts$left$x, epts$left$y, r=unit(1, "mm"),
            gp=gpar(fill=hcl(0, 80, seq(10, 100, 10))))
x <- c(.2, .4, .6, .8)
y <- c(-.05, .05, -.05, .05)
w <- unit(c(2, 4, 6, 8), "mm")
vwbg <- brushXsplineGrob(circleBrush(), x, y + 1/3, w,
                         gp=gpar(col="black"))
grid.draw(vwbg)
epts <- edgePoints(vwbg, unit(0:9, "cm"), x0=0, y0=1/3)
grid.circle(epts$x, epts$y, r=unit(1, "mm"),
            gp=gpar(fill=hcl(0, 80, seq(10, 100, 10))))
```

`grid.brushXspline` *Sweep an X-spline curve with a brush.*

Description

Sweep an X-spline curve with a brush; the main curve is described by x/y control points and a variable-size brush is used to sweep a variable-width line from the main curve.

Usage

```
brushXsplineGrob(brush, x, y, w=unit(1, "cm"), default.units="npc",
                 shape=1, angle="perp", open=TRUE, spacing=NULL,
                 render=vwPath(),
                 gp=gpar(fill="black"), name=NULL, debug=FALSE)
grid.brushXspline(...)

verticalBrush
circleBrush(n=50)
squareBrush
```

Arguments

`brush`	A description of a brush shape (see Details). Some simple brushes are provided by `verticalBrush` and `circleBrush`.
`x`	A numeric vector or unit describing x-locations.
`y`	A numeric vector or unit describing y-locations.
`w`	A numeric vector or unit describing widths at each location, or a width specification generated by `widthSpec`.
`default.units`	The units used if `x` or `y` are numeric vectors.
`shape`	A numeric value (or one per location) that controls the shape of the X-spline curve relative to the locations.
`angle`	Either `"perp"` or a numeric value describing a fixed orientation for the line width.
`open`	A boolean indicating whether to connect the last location back to the first location to produce a closed line.
`spacing`	A numeric vector or unit describing where the brush will be placed along the main curve.
`render`	A function that is used to render the outline of the path that is generated for the variable-width line.
`gp`	A set of graphical parameters; see `gpar`.
`name`	A name for the grob generated for the variable-width line.
`debug`	A logical indicating whether to produce graphical debugging output.
`...`	Arguments passed to `brushXsplineGrob`.
`n`	Number of points to generate on circumference of circle brush.

Details

The brush is described using a list with components x and y. The brush is described in a coordinate system [-1, 1] (in both x and y). The predefined brushes are verticalBrush (thin vertical line), circleBrush (the number of points on the circumference can be controlled), and squareBrush.

By default, the brush is placed at (almost) all points along the main curve. If spacing is specified, then the brush is only placed at those locations on the main curve. If spacing is a numeric vector, the values are assumed to be proportions of the length of the main curve. The spacing is automatically recycled to the cover the full length of the main curve.

See grid.xspline for more about the behaviour of X-splines.

Value

grid.brushXspline is used for its side-effect of drawing a variable-width line; brushXsplineGrob returns a "brushXsplineGrob" object.

Author(s)

Paul Murrell

See Also

grid.xspline, grid.vwline, grid.vwcurve, grid.offsetXspline, grid.vwXspline

Examples

```
grid.newpage()
x <- c(.2, .4, .6, .8)
y <- c(-.05, .05, -.05, .05)
w <- unit(c(2, 4, 6, 8), "mm")
grid.brushXspline(verticalBrush, x, y + .8, w)
grid.brushXspline(verticalBrush, x, y + .6, w,
                  shape=-1)
grid.brushXspline(circleBrush(), x, y + .4, w)
grid.brushXspline(verticalBrush, x, y + .2, w,
                  gp=gpar(col="black"), debug=TRUE)
```

---

grid.offsetBezier *Draw a Bezier Offset Curve.*

Description

Draw a Bezier offset curve; the main curve is described by x/y control points and offset Bezier curves are calculated to the left and right of the main curve at the specified widths.

Usage

```
offsetBezierGrob(x, y, w, default.units="npc",
                 stepFn=nSteps(100), open=TRUE,
                 lineend="butt", linejoin="round", mitrelimit=4,
                 render=if (open) vwPolygon else vwPath(),
                 gp=gpar(fill="black"), name=NULL, debug=FALSE)
grid.offsetBezier(...)
```

Arguments

x	A numeric vector or unit describing x-locations.
y	A numeric vector or unit describing y-locations.
w	A numeric vector or unit describing widths at each location, or a width specification generated by widthSpline or a width specification generated by BezierWidth.
default.units	The units used if x or y are numeric vectors.
stepFn	Function called to generate steps in t when rendering. See Details.
open	A boolean indicating whether to connect the last location back to the first location to produce a closed line.
lineend	The line ending style; one of "round", "mitre", "butt", "square", or "extend".
linejoin	The line join style; one of "round", "mitre", "bevel", or "extend".
mitrelimit	A numeric that controls when a mitre join is converted to a bevel join or a mitre ending is converted to a square ending.
render	A function that is used to render the outline of the path that is generated for the variable-width line.
gp	A set of graphical parameters; see gpar.
name	A name for the grob generated for the variable-width line.
debug	A logical indicating whether to produce graphical debugging output.
...	Arguments passed to offsetBezierGrob.

Details

Rendering relies on flattening the spline to a series of straight line segments. This depends on generating a set of t values (and then x and y are calculated from t). The stepFn is called to generate t and given the control points, x and y as arguments, plus a range, which indicates what range of t to generate values for.

The "extend" line ending and line join style is only available if the width is specified by BezierWidth.

Value

grid.offsetBezier is used for its side-effect of drawing a variable-width line; offsetBezierGrob returns a "offsetBezierGrob" object.

Author(s)

Paul Murrell

See Also

grid.xspline, grid.vwline, grid.vwcurve, grid.brushXspline, grid.vwXspline, grid.offsetXspline

Examples

```
grid.newpage()
x <- c(.2, .4, .6, .8)
y <- c(-.05, .05, -.05, .05)
w <- unit(c(2, 4, 6, 8), "mm")
grid.offsetBezier(x, y + .8, w)
grid.offsetBezier(x, y + .5, w,
                  lineend="round")
grid.offsetBezier(x, y + .2, w,
                  gp=gpar(col="black"), debug=TRUE)
```

---

grid.offsetXspline *Draw an X-spline offset curve.*

Description

Draw an X-spline offset curve; the main curve is described by x/y control points and offset X-splines are calculated to the left and right of the main curve at the specified widths.

Usage

```
offsetXsplineGrob(x, y, w, default.units="npc", shape=1,
                  open=TRUE, repEnds=TRUE,
                  lineend="butt", mitrelimit=4,
                  render=if (open) vwPolygon else vwPath(),
                  gp=gpar(fill="black"), name=NULL, debug=FALSE)
grid.offsetXspline(...)
```

Arguments

x	A numeric vector or unit describing x-locations.
y	A numeric vector or unit describing y-locations.
w	A numeric vector or unit describing widths at each location, or a width specification generated by widthSpline.
default.units	The units used if x or y are numeric vectors.
shape	A numeric value (or one per location) that controls the shape of the X-spline curve relative to the locations.
open	A boolean indicating whether to connect the last location back to the first location to produce a closed line.
repEnds	A logical indicating whether to replicate the first and last control points (so that the X-spline starts and ends at the first and last control points). Can also be the special value "extend", in which case the first and last control segments are extended (and the resulting X-spline is a little curvier at the ends).

lineend	The line ending style; one of "round", "mitre", "butt", or "square".
mitrelimit	A numeric that controls when a mitre join is converted to a bevel join or a mitre ending is converted to a square ending.
render	A function that is used to render the outline of the path that is generated for the variable-width line.
gp	A set of graphical parameters; see gpar.
name	A name for the grob generated for the variable-width line.
debug	A logical indicating whether to produce graphical debugging output.
...	Arguments passed to offsetXsplineGrob.

Details

See grid.xspline for more about the behaviour of X-splines.

Value

grid.offsetXspline is used for its side-effect of drawing a variable-width line; offsetXsplineGrob returns a "offsetXsplineGrob" object.

Author(s)

Paul Murrell

See Also

grid.xspline, grid.vwline, grid.vwcurve, grid.brushXspline, grid.vwXspline

Examples

```
grid.newpage()
x <- c(.2, .4, .6, .8)
y <- c(-.05, .05, -.05, .05)
w <- unit(c(2, 4, 6, 8), "mm")
grid.offsetXspline(x, y + .8, w)
grid.offsetXspline(x, y + .6, w,
                   shape=-1)
grid.offsetXspline(x, y + .4, w,
                   lineend="round")
grid.offsetXspline(x, y + .2, w,
                   gp=gpar(col="black"), debug=TRUE)
```

grid.vwcurve *Draw a variable-width smooth curve.*

Description

Draw a variable-width line where the main line is a series of straight line segments that are assumed to be a flattened approximation to a smooth curve and the width is specified at each vertex along the flattened curve.

Usage

```
vwcurveGrob(x, y, w, default.units="npc", open=TRUE, angle="perp",
            lineend="butt", mitrelimit=4,
            render=if (open) vwPolygon else vwPath(),
            gp=gpar(fill="black"), name=NULL, debug=FALSE)
grid.vwcurve(...)
```

Arguments

x	A numeric vector or unit describing x-locations.
y	A numeric vector or unit describing y-locations.
w	A numeric vector or unit describing widths at each location, or a width specification generated by widthSpec.
default.units	The units used if x or y are numeric vectors.
open	A boolean indicating whether to connect the last location back to the first location to produce a closed line.
angle	Either "perp" or a numeric value describing a fixed orientation for the line width.
lineend	The line ending style; one of "round", "mitre", "butt", or "square".
mitrelimit	A numeric that controls when a mitre join is converted to a bevel join or a mitre ending is converted to a square ending.
render	A function that is used to render the outline of the path that is generated for the variable-width line.
gp	A set of graphical parameters; see gpar.
name	A name for the grob generated for the variable-width line.
debug	A logical indicating whether to produce graphical debugging output.
...	Arguments passed to vwcurveGrob.

Details

The widths are calculated perpendicular to the average angle at each vertex (or just perpendicular to the line segment for the first and last vertex), *unless* a numeric angle is specified, in which case the widths are calculated at that angle at every vertex.

Value

`grid.vwcurve` is used for its side-effect of drawing a variable-width line; `vwcurveGrob` returns a `"vwcurveGrob"` object.

Author(s)

Paul Murrell

See Also

`grid.vwline`, `grid.vwXspline`, `grid.brushXspline`, `grid.offsetXspline`

Examples

```
grid.newpage()
x <- seq(.2, .8, length.out=100)
y <- .05*sin(seq(0, 2*pi, length.out=100))
w <- unit(seq(2, 10, length.out=100), "mm")
grid.vwcurve(x, y + .8, w)
grid.vwcurve(x, y + .6, w,
             angle=45)
grid.vwcurve(x, y + .4, w,
             lineend="round")
grid.vwcurve(x, y + .2, w,
             gp=gpar(col="black"), debug=TRUE)
```

---

grid.vwline *Draw a variable-width set of line segments.*

Description

Draw a variable-width line where the main line is a series of straight line segments and the width is specified at each vertex and linearly interpolated along each segment.

Usage

```
vwlineGrob(x, y, w, default.units="npc", open=TRUE,
           linejoin="round", lineend="butt", mitrelimit=4,
           stepWidth=FALSE, render=if (open) vwPolygon else vwPath(),
           gp=gpar(fill="black"), name=NULL, debug=FALSE)
grid.vwline(...)
```

Arguments

`x`	A numeric vector or unit describing x-locations.
`y`	A numeric vector or unit describing y-locations.
`w`	A numeric vector or unit describing widths at each location, or a width specification generated by `widthSpec`.

default.units	The units used if x or y are numeric vectors.
open	A boolean indicating whether to connect the last location back to the first location to produce a closed line.
linejoin	The line join style; one of "round", "mitre", or "bevel".
lineend	The line ending style; one of "round", "mitre", "butt", or "square".
mitrelimit	A numeric that controls when a mitre join is converted to a bevel join or a mitre ending is converted to a square ending.
stepWidth	A logical indicating whether widths are fixed along the length of a segment.
render	A function that is used to render the outline of the path that is generated for the variable-width line.
gp	A set of graphical parameters; see gpar.
name	A name for the grob generated for the variable-width line.
debug	A logical indicating whether to produce graphical debugging output.
...	Arguments passed to vwlineGrob.

Details

If stepWidth is TRUE, the last width value is ignored.

Value

grid.vwline is used for its side-effect of drawing a variable-width line; vwlineGrob returns a "vwlineGrob" object.

Author(s)

Paul Murrell

See Also

grid.vwcurve, grid.vwXspline, grid.brushXspline, grid.offsetXspline

Examples

```
grid.newpage()
grid.vwline(c(.2, .5, .8), rep(.8, 3), unit(c(1, 3, 2), "cm"))
grid.vwline(c(.2, .5, .8), rep(.6, 3), unit(c(1, 3, 2), "cm"),
            stepWidth=TRUE)
grid.vwline(c(.2, .5, .8), rep(.4, 3), unit(c(1, 3, 2), "cm"),
            linejoin="mitre", lineend="round")
grid.vwline(c(.2, .5, .8), rep(.2, 3), unit(c(1, 3, 2), "cm"),
            gp=gpar(col="black"), debug=TRUE)
```

`grid.vwXspline` *Draw a variable-width X-spline.*

Description

Draw a variable-width X-spline where the main line is described by x/y control points and the width describes offset control points.

Usage

```
vwXsplineGrob(x, y, w, default.units="npc", shape=1,
              open=TRUE, repEnds=TRUE, angle="perp",
              lineend="butt", mitrelimit=4,
              render=vwPath(),
              gp=gpar(fill="black"), name=NULL, debug=FALSE)
grid.vwXspline(...)
```

Arguments

`x`	A numeric vector or unit describing x-locations.
`y`	A numeric vector or unit describing y-locations.
`w`	A numeric vector or unit describing widths at each location, or a width specification generated by `widthSpec`.
`default.units`	The units used if `x` or `y` are numeric vectors.
`shape`	A numeric value (or one per location) that controls the shape of the X-spline curve relative to the locations.
`open`	A boolean indicating whether to connect the last location back to the first location to produce a closed line.
`repEnds`	A logical indicating whether to replicate the first and last control points (so that the X-spline starts and ends at the first and last control points). Can also be the special value `"extend"`, in which case the first and last control segments are extended (and the resulting X-spline is a little curvier at the ends).
`angle`	Either `"perp"` or a numeric value describing a fixed orientation for the line width.
`lineend`	The line ending style; one of `"round"`, `"mitre"`, `"butt"`, or `"square"`.
`mitrelimit`	A numeric that controls when a mitre join is converted to a bevel join or a mitre ending is converted to a square ending.
`render`	A function that is used to render the outline of the path that is generated for the variable-width line.
`gp`	A set of graphical parameters; see `gpar`.
`name`	A name for the grob generated for the variable-width line.
`debug`	A logical indicating whether to produce graphical debugging output.
`...`	Arguments passed to `vwXsplineGrob`.

Details

See grid.xspline for more about the behaviour of X-splines.

Value

grid.vwXspline is used for its side-effect of drawing a variable-width line; vwXsplineGrob returns a "vwXsplineGrob" object.

Author(s)

Paul Murrell

See Also

grid.xspline, grid.vwline, grid.vwXspline, grid.brushXspline, grid.offsetXspline

Examples

```
grid.newpage()
x <- c(.2, .4, .6, .8)
y <- c(-.05, .05, -.05, .05)
w <- unit(c(2, 4, 6, 8), "mm")
grid.vwXspline(x, y + .8, w)
grid.vwXspline(x, y + .6, w,
               angle=45)
grid.vwXspline(x, y + .4, w,
               lineend="round")
grid.vwXspline(x, y + .2, w,
               gp=gpar(col="black"), debug=TRUE)
```

---

outline *Calculate outline*

Description

Calculate outline of a variable-width line, possibly without simplication.

Usage

```
outline(x, simplify=TRUE, ...)
## S3 method for class 'vwlineGrob'
outline(x, simplify=TRUE, ...)
## S3 method for class 'offsetXsplineGrob'
outline(x, simplify=TRUE, ...)
## S3 method for class 'offsetBezierGrob'
outline(x, simplify=TRUE, ...)
```

Arguments

x	A variable-width line grob.
simplify	Whether to simplify the outline (using polyclip::polysimplify to collapse self-intersections).
...	Possible additional arguments for methods.

Value

A list with components x and y describing the complete outline of the variable-width line (possibly unsimplified).

Author(s)

Paul Murrell

See Also

grid.vwline, grid.offsetXspline grid.offsetBezier

Examples

```
grid.newpage()
x <- c(.2, .8, .8, .2)
y <- c(.2, .8, .2, .8)
w <- c(0, .1, .1, 0)
vwlg <- vwlineGrob(x, y, w, gp=gpar(col="grey", lwd=10, fill=NA))
grid.draw(vwlg)
o <- outline(vwlg, simplify=FALSE)
grid.polygon(o$x, o$y, default.units="in")

grid.newpage()
oxsg <- offsetXsplineGrob(x, y, w, gp=gpar(col="grey", lwd=10, fill=NA))
grid.draw(oxsg)
o <- outline(oxsg, simplify=FALSE)
grid.polygon(o$x, o$y, default.units="in")
```

---

vwPath *Functions to render variable-width line*

Description

Functions that can be used to render a variable-width line. All variable-width-line functions generate a polygon or path to represent the variable-width line and they have a render argument that specifies the rendering function. These two functions provide the default rendering behaviour.

Usage

```
vwPath(rule = "winding")
vwPolygon(x, y, id.lengths, gp, name)
```

Arguments

rule	The fill rule for rendering a variable-width line as a path.
x,y,id.lengths,gp,name	Arguments provided by the calling function when rendering occurs.

Details

These functions are not called directly; they are provided as arguments to variable-width-line functions (which then call them internally to render a variable-width line).

Value

These functions are used for their side-effect, which is to draw on the current graphics device.

Author(s)

Paul Murrel

See Also

grid.curve, grid.vwline, grid.vwcurve, grid.brushXspline, grid.vwXspline

Examples

```
grid.newpage()
x <- c(.2, .4, .2, .4)
y <- c(.2, .4, .4, .2)
grid.vwXspline(x, y, c(0, .1, .1, 0))
x <- c(.2, .4, .2, .4) + .4
y <- c(.2, .4, .4, .2)
grid.vwXspline(x, y, c(0, .1, .1, 0), render=vwPath("evenodd"))
x <- c(.2, .4, .2, .4)
y <- c(.2, .4, .4, .2) + .4
grid.brushXspline(verticalBrush, x, y, c(0, .02, .02, 0))
x <- c(.2, .4, .2, .4) + .4
y <- c(.2, .4, .4, .2) + .4
grid.brushXspline(verticalBrush, x, y, c(0, .02, .02, 0),
                  render=vwPolygon)
```

---

widthSpec *Specify the width of a variable-width line*

Description

Specify the width of a variable-width line. The widthSpline and BezierWidth functions are for use with grid.brushXspline or grid.offsetXspline or grid.offsetBezier. The widthSpec function is for use with grid.vwline.

Usage

```
widthSpec(x, default.units = "npc")
widthSpline(w=unit(1, "cm"), default.units="in", d=NULL, shape=-1, rep=FALSE)
BezierWidth(w=unit(1, "cm"), default.units="in", d=NULL, rep=FALSE)
```

Arguments

x	A numeric vector or a unit specifying the width, or a list of such with components left and right specifying the width to either side.
default.units	The units to use if x or w is specified as a numeric vector.
w	A numeric vector or a unit specifying the width.
d	A numeric vector or a unit specifying the distance along the line for each width value.
shape	The shape parameter for the width spline.
rep	A logical indicating whether to repeat the widths along the full length of the line.

Details

All variable-width functions accept width as just a numeric vector or a unit; the former is automatically converted to the latter.

For grid.vwcurve and grid.vwXspline, each width is associated with an x/y location on the main curve. The grid.vwline function is similar except that, via widthSpec, it also allows the width to be different on either side of the main curve.

For grid.brushXspline and grid.offsetXspline, the width is independent of the x/y locations that specify the main curve. The width is itself a spline, with each width associated with a distance along the length of the main curve. By default, the specified widths are spaced evenly along the main curve, but the widthSpec function allows fine control over the spacing of the widths.

For BezierWidth, w must contain 4 (or 7 or 10) values to provide an appropriate number of control points, although a single value is automatically replicated four times. If d is specified, the same rules apply.

Value

widthSpec creates a "widthSpec" object.

widthSpline creates a "widthSpline" object.

BezierWidth creates a "BezierWidth" object.

Author(s)

Paul Murrell

See Also

grid.xspline, grid.curve, grid.vwline, grid.vwcurve, grid.brushXspline, grid.vwXspline, grid.offsetXspline, grid.offsetBezier

Examples

```
grid.newpage()
x <- c(.2, .4, .6, .8)
y <- c(-.05, .05, -.05, .05)
grid.vwcurve(x, y + .8, w=c(0, .1, 0, .1))
grid.lines(x, y + .8, gp=gpar(col="white"))
grid.vwline(x, y + .6, w=widthSpec(list(left=c(0, .1, 0, .1),
                                        right=c(.1, 0, .1, 0))))
grid.lines(x, y + .6, gp=gpar(col="white"))
grid.brushXspline(verticalBrush, x, y + .4,
                  w=widthSpline(c(0, .1, 0, .1), "npc", shape=1))
grid.xspline(x, y + .4, shape=1, gp=gpar(col="white"))
grid.offsetXspline(x, y + .2,
                  w=widthSpline(c(0, .1, 0, .1, 0), "npc",
                                d=0:4/8, shape=1, rep=TRUE))
grid.xspline(x, y + .2, shape=1, gp=gpar(col="white"))
```

Index