

Package ‘eChem’

July 1, 2018

Type Package

Title Simulations for Electrochemistry Experiments

Version 1.0.0

Author David Harvey

Maintainer David Harvey <harvey@depauw.edu>

Description Simulates cyclic voltammetry, linear-sweep voltammetry (both with and without stirring of the solution), and single-pulse and double-pulse chronoamperometry and chronocoulometry experiments using the implicit finite difference method outlined in Gosser (1993, ISBN: 9781560810261) and in Brown (2015) <doi:10.1021/acs.jchemed.5b00225>. Additional functions provide ways to display and to examine the results of these simulations. The primary purpose of this package is to provide tools for use in courses in analytical chemistry.

URL <https://github.com/dtharvey/eChem>

License GPL-2

Encoding UTF-8

LazyData true

Imports plot3D, animation

Suggests knitr, rmarkdown

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-01 13:30:45 UTC

R topics documented:

| | |
|-------------------------|---|
| eChem-package | 2 |
| animateCA | 3 |

| | |
|-------------------------|-----------|
| animateCC | 3 |
| animateCV | 4 |
| animateLSV | 5 |
| annotateCA | 5 |
| annotateCC | 6 |
| annotateCV | 7 |
| annotateLSV | 8 |
| plotCA | 8 |
| plotCC | 10 |
| plotCV | 11 |
| plotDiffGrid | 12 |
| plotDiffusion | 13 |
| plotGrid | 14 |
| plotLSV | 15 |
| plotPotential | 16 |
| sampleCA | 17 |
| sampleCC | 18 |
| sampleCV | 19 |
| sampleLSV | 20 |
| simulateCA | 21 |
| simulateCC | 23 |
| simulateCV | 25 |
| simulateLSV | 27 |
| Index | 30 |

eChem-package

Simulations for Electrochemistry Experiments

Description

A collection of functions for simulating four different types of electrochemistry experiments: two potential step experiments (chronoamperometry and chronocoulometry) and two potential scan experiments (linear-sweep voltammetry and cyclic voltammetry). Each simulation allows for either an oxidation reaction or for a reduction reaction, and allows for a single following or a single preceding chemical reaction. Also included are a variety of functions for displaying the result of a simulation in either a static or animated format.

Author(s)

David T. Harvey

Maintainer: David T. Harvey <harvey@depauw.edu>

`animateCA`*Animate Chronoamperometry Simulation*

Description

Creates either an HTML or a GIF animation of a chronoamperometry simulation created using `caSim`. The resulting animation displays the diffusion profiles for Ox, Red, and, where appropriate, Z situated above the chronoamperogram. Note: the `animateCA` function requires that the animation package is installed.

Usage

```
animateCA(filename, out_type = c("html", "gif"), out_name = "aniCA")
```

Arguments

| | |
|-----------------------|---|
| <code>filename</code> | Name of the file that contains the results of a chronoamperometry simulation. |
| <code>out_type</code> | Identifies the type of file generated, either an HTML file or a GIF file. |
| <code>out_name</code> | Name used for the file(s) created by this function. |

Value

For an HTML animation, the function saves four items in the working directory: a folder with CSS files, a folder with javascript files, a folder with the image files for the animation, and a .html file; the latter two files are named using the function's `out_name` argument. For a GIF animation, the function saves a single .gif file using the function's `out_name` argument. See the vignettes for examples.

`animateCC`*Animate Chronocoulometry Simulation*

Description

Creates either an HTML or a GIF animation of a chronocoulometry simulation created using `ccSim`. The resulting animation displays the diffusion profiles for Ox, Red, and, where appropriate, Z situated above the chronoamperogram. Note: the `animateCC` function requires that the animation package is installed.

Usage

```
animateCC(filename, out_type = c("html", "gif"), out_name = "aniCC")
```

Arguments

| | |
|----------|--|
| filename | Name of the file that contains the results of a chronocoulometry simulation. |
| out_type | Identifies the type of file generated, either an HTML file or a GIF file. |
| out_name | Name used for the file(s) created by this function. |

Value

For an HTML animation, the function saves four items in the working directory: a folder with CSS files, a folder with javascript files, a folder with the image files for the animation, and a .html file; the latter two files are named using the function's out_name argument. For a GIF animation, the function saves a single .gif file using the function's out_name argument. See the vignettes for examples.

| | |
|-----------|--|
| animateCV | <i>Animate Cyclic Voltammetry Simulation</i> |
|-----------|--|

Description

Creates either an HTML or a GIF animation of a cyclic voltammetry simulation created using cvSim. The resulting animation displays the diffusion profiles for Ox, Red, and, where appropriate, Z situated above the cyclic voltammogram. Note: the animateCV function requires that the animation package is installed.

Usage

```
animateCV(filename, out_type = c("html", "gif"), out_name = "aniCV")
```

Arguments

| | |
|----------|--|
| filename | Name of the file that contains the results of a cyclic voltammetry simulation. |
| out_type | Identifies the type of file generated, either an HTML file or a GIF file. |
| out_name | Name used for the file(s) created by this function. |

Value

For an HTML animation, the function saves four items in the working directory: a folder with CSS files, a folder with javascript files, a folder with the image files for the animation, and a .html file; the latter two files are named using the function's out_name argument. For a GIF animation, the function saves a single .gif file using the function's out_name argument. See the vignettes for examples.

`animateLSV`*Animate Linear Sweep Voltammetry Simulation*

Description

Creates either an HTML or a GIF animation of a linear sweep voltammetry simulation created using `lsvSim`. The resulting animation displays the diffusion profiles for Ox, Red, and, where appropriate, Z situated above the linear sweep voltammogram. Note: the `animateLSV` function requires that the animation package is installed.

Usage

```
animateLSV(filename, out_type = c("html", "gif"), out_name = "aniLSV")
```

Arguments

| | |
|-----------------------|--|
| <code>filename</code> | Name of the file that contains the results of a linear sweep voltammetry simulation. |
| <code>out_type</code> | identifies the type of file generated, either an HTML file or a GIF file. |
| <code>out_name</code> | Name used for the file(s) created by this function. |

Value

For an HTML animation, the function saves four items in the working directory: a folder with CSS files, a folder with javascript files, a folder with the image files for the animation, and a .html file; the latter two files are named using the function's `out_name` argument. For a GIF animation, the function saves a single .gif file using the function's `out_name` argument. See the vignettes for examples.

`annotateCA`*Annotate Chronoamperogram*

Description

Plots a chronoamperogram and annotates it with either the current for a single pulse experiment, or, for a double pulse experiment, with the currents after both the forward pulse and the reverse pulse, and the current ratio. The currents are displayed for a designated time after a pulse, which defaults to the length of the pulse if a value is not provided.

Usage

```
annotateCA(filename, time.delay, scale.factor = 1, main.title = NULL)
```

Arguments

| | |
|--------------|---|
| filename | Name of the file that contains the results of a simulated chronoamperometry experiment. |
| time.delay | Time after the application of a pulse for which the current is reported. |
| scale.factor | Setting to a value less than 1 adjusts the y-axis limits so that the limits are not defined by the current spike. |
| main_title | An optional main title. |

Value

Returns a plot of the chronoamperogram with annotations.

Examples

```
ex_ca = simulateCA(e.start = 0.25, e.pulse = -0.25, e.form = 0,
  pulses = "double", t.2 = 20, x.units = 100, t.units = 1000)
annotateCA(ex_ca, time.delay = 0.5)
```

 annotateCC

Annotate Chronocoulogram

Description

Plots a chronocoulogram and annotates it with either the charge for a single pulse experiment, or, for a double pulse experiment, with the charge following the forward and the reverse pulse, and the charge ratio. The charges are displayed for a designated time after a pulse, which defaults to the length of the pulse if a value is not provided.

Usage

```
annotateCC(filename, time.delay, scale.factor = 1, main_title = NULL)
```

Arguments

| | |
|--------------|---|
| filename | Name of the file that contains the results of a simulated chronocoulometry experiment. |
| time.delay | Time after the application of a pulse for which the current is reported. |
| scale.factor | Setting to a value less than 1 adjusts the y-axis limits so that the limits are not defined by the current spike. |
| main_title | An optional main title. |

Value

Returns a plot of the chronocoulogram with annotations.

Examples

```
ex_ca = simulateCA(e.start = 0.25, e.pulse = -0.25, e.form = 0,
  pulses = "double", t.2 = 20, x.units = 100, t.units = 1000)
ex_cc = simulateCC(ex_ca)
annotateCC(ex_cc)
```

 annotateCV

Annotate Cyclic Voltammogram

Description

Plots a cyclic voltammogram and annotates it with values for the cathodic peak potential (Epc), the anodic peak potential (Epa), the difference in peak potentials (Delta E), the cathodic peak current (ip,c), the anodic peak current (ip,a), and the peak current ratio (either ip,c/ip,a or ip,a/ip,c). The baseline for determining peak currents is set using a defined percentage of points at the beginning of the forward and the reverse scans. Values are reported as unmeasurable when they fall below a threshold value.

Usage

```
annotateCV(filename, main_title = NULL, forward.per = 5, reverse.per = 5,
  threshold = 0.05)
```

Arguments

| | |
|-------------|--|
| filename | Name of the file that contains the results of a simulated cyclic voltammetry experiment. |
| main_title | An optional main title. |
| forward.per | The percentage of points from the beginning of the forward scan used to set the baseline for measuring the peak current. |
| reverse.per | The percentage of points from the beginning of the reverse scan used to set the baseline for measuring the peak current. |
| threshold | Sets the smallest measurable current. |

Value

Returns a plot of the cyclic voltammogram with annotations.

Examples

```
cv_ex2 = simulateCV(ko = 0.01, e.switch = -0.8, e.form = -0.4,
  x.units = 100, t.units = 1000)
annotateCV(cv_ex2)
```

 annotateLSV

Annotate Linear Sweep Voltammogram

Description

Plots a linear sweep voltammogram and annotates it with values for either the cathodic peak potential (Epc) and cathodic peak current (ip,c), or the anodic peak potential (Epa) and the anodic peak current (ip,a). The baseline for determining the peak currents is set using a defined percentage of points at the beginning of potential scan.

Usage

```
annotateLSV(filename, potential.per = 5, main_title = NULL)
```

Arguments

| | |
|---------------|--|
| filename | Name of the file that contains the results of a simulated linear sweep voltammetry experiment. |
| potential.per | The percentage of points from the beginning of the potential scan used to set the baseline for measuring the peak current. |
| main_title | An optional main title. |

Value

Returns a plot of the linear sweep voltammogram with annotations.

Examples

```
ex_lsv = simulateLSV(e.start = 0.25, e.end = -0.25, e.form = 0,
  stir.rate = "fast", x.units = 100, t.units = 1000)
annotateLSV(ex_lsv)
```

 plotCA

Plot Chronoamperograms

Description

Plots 1–5 chronoamperograms on a single set of axes. The default plot does not include a legend or a title, but providing a vector of character strings to legend_text adds a legend to the final plot, and adding a character string for main_title adds a title to the plot. Line widths, line types, line colors, point symbols, and point colors have default values that can be adjusted. Note: this function accepts both full data files created using caSim or reduced data files created using sampleAmpgram.

Usage

```
plotCA(filenamees = list(file1 = NULL, file2 = NULL), scale = 1,
      legend_text = NULL, legend_position = c("topleft", "topright",
      "bottomleft", "bottomright"), main_title = NULL, line_widths = c(2, 2, 2,
      2, 2), line_types = c(1, 2, 3, 4, 5), point_symbols = c(21, 22, 23, 24,
      25), line_colors = c("blue", "blue", "blue", "blue", "blue"))
```

Arguments

| | |
|-----------------|--|
| filenamees | A list giving the names of 1–5 files that contain the results of a simulated chronomperometry experiment. |
| scale | Setting to a value less than 1 adjusts the y-axis limits so that the limits are not defined by the current spike. |
| legend_text | Optional vector that contains text to include in a legend. Default is NULL, which surpresses the legend. |
| legend_position | One of topleft, topright, bottomleft, or bottomright; defaults to topleft. |
| main_title | An optional main title. |
| line_widths | A vector of line widths for the individual voltammograms; defaults to a common line width of 2, but can be adjusted by supplying a vector with desired values. |
| line_types | A vector of line types for the individual voltammograms; defaults to a set of different line types, but can be adjusted by supplying a vector with desired values. |
| point_symbols | A vector of pch values for plotting points. |
| line_colors | A vector of colors for the individual voltammograms, whether displayed as lines or as points; defaults to a common color, but can be adjusted by supplying a vector with desired values. |

Value

Returns a plot that shows time on the x-axis and current on the y-axis.

Examples

```
ca_ex1 = simulateCA(conc.bulk = 0.01, x.units = 100, t.units = 1000)
ca_ex2 = simulateCA(conc.bulk = 0.001, x.units = 100, t.units = 1000)
ca_ex3 = simulateCA(conc.bulk = 0.0001, x.units = 100, t.units = 1000)
plotCA(filenamees = list(ca_ex1, ca_ex2, ca_ex3),
      legend_text = c("0.01 M", "0.001 M", "0.0001 M"), scale = 0.1)
```

 plotCC

Plot Chronocoulograms

Description

Plots 1–5 chronocoulograms on a single set of axes. The default plot does not include a legend or a title, but providing a vector of character strings to `legend_text` adds a legend to the final plot, and adding a character string for `main_title` adds a title to the plot. Line widths, line types, line colors, point symbols, and point colors have default values that can be adjusted. Note: this function accepts both full data files created using `ccSim` or reduced data files created using `sampleCoulgram`.

Usage

```
plotCC(filenamees = list(file1 = NULL, file2 = NULL), scale = 1,
  legend_text = NULL, legend_position = "topleft", main_title = NULL,
  line_widths = c(2, 2, 2, 2, 2), line_types = c(1, 2, 3, 4, 5),
  point_symbols = c(21, 22, 23, 24, 25), line_colors = c("blue", "blue",
  "blue", "blue", "blue"))
```

Arguments

| | |
|------------------------------|--|
| <code>filenamees</code> | A list giving the names of 1–5 files that contain the results of a simulated chronocoulometry experiment. |
| <code>scale</code> | Setting to value less than 1 adjusts the y-axis limits so that the limits are not defined by the current spike. |
| <code>legend_text</code> | Optional vector that contains text to include in a legend. Default is NULL, which suppresses the legend. |
| <code>legend_position</code> | <code>legend_position</code> One of <code>topleft</code> , <code>topright</code> , <code>bottomleft</code> , or <code>bottomright</code> ; defaults to <code>topleft</code> . |
| <code>main_title</code> | An optional main title. |
| <code>line_widths</code> | A vector of line widths for the individual voltammograms; defaults to a common line width of 2, but can be adjusted by supplying a vector with desired values. |
| <code>line_types</code> | A vector of line types for the individual voltammograms; defaults to a set of different line types, but can be adjusted by supplying a vector with desired values. |
| <code>point_symbols</code> | A vector of <code>pch</code> values for plotting points. |
| <code>line_colors</code> | A vector of colors for the individual voltammograms, whether displayed as lines or as points; defaults to a common color, but can be adjusted by supplying a vector with desired values. |

Value

Returns a plot that shows time on the x-axis and charge on the y-axis.

Examples

```

ca_ex1 = simulateCA(conc.bulk = 0.01, x.units = 100, t.units = 1000)
cc_ex1 = simulateCC(ca_ex1)
ca_ex2 = simulateCA(conc.bulk = 0.001, x.units = 100, t.units = 1000)
cc_ex2 = simulateCC(ca_ex2)
ca_ex3 = simulateCA(conc.bulk = 0.0001, x.units = 100, t.units = 1000)
cc_ex3 = simulateCC(ca_ex3)
plotCC(filenamees = list(cc_ex1, cc_ex2, cc_ex3),
        legend_text = c("0.01 M", "0.001 M", "0.0001 M"), scale = 0.8)

```

plotCV

Plot Cyclic Voltammograms

Description

Plots 1–5 cyclic voltammograms on a single set of axes. The default plot does not include a legend or a title, but providing a vector of character strings to `legend_text` adds a legend to the final plot, and adding a character string for `main_title` adds a title to the plot. Line widths, line types, line colors, point symbols, and point colors have default values that can be adjusted. Note: this function accepts both full data files created using `cvSim` or reduced data files created `sampleCV`.

Usage

```

plotCV(filenamees = list(file1 = NULL, file2 = NULL), legend_text = NULL,
        legend_position = c("topleft", "topright", "bottomleft", "bottomright"),
        main_title = NULL, line_widths = c(2, 2, 2, 2, 2), line_types = c(1, 2,
        3, 4, 5), point_symbols = c(21, 22, 23, 24, 25), line_colors = c("blue",
        "blue", "blue", "blue"))

```

Arguments

| | |
|------------------------------|---|
| <code>filenamees</code> | A list giving the names of 1–5 files that contain the results of a simulated cyclic voltammetry experiment. |
| <code>legend_text</code> | Optional vector that contains text to include in a legend. Default is <code>NULL</code> , which suppresses the legend. |
| <code>legend_position</code> | One of <code>topleft</code> , <code>topright</code> , <code>bottomleft</code> , or <code>bottomright</code> ; defaults to <code>topleft</code> . |
| <code>main_title</code> | An optional main title. |
| <code>line_widths</code> | A vector of line widths for the individual cyclic voltammograms; defaults to a common line width of 2, but can be adjusted by supplying a vector with desired values. |
| <code>line_types</code> | A vector of line types for the individual cyclic voltammograms; defaults to a set of different line types, but can be adjusted by supplying a vector with desired values. |
| <code>point_symbols</code> | A vector of <code>pch</code> values for plotting points. |

`line_colors` A vector of colors for the individual cyclic voltammograms, whether displayed as lines or as points; defaults to a common color, but can be adjusted by supplying a vector with desired values.

Value

Returns a plot that shows the applied potential on the x -axis and current on the y -axis.

Examples

```
cv_ex1 = simulateCV(ko = 1, e.switch = -0.8, e.form = -0.4,
  x.units = 100, t.units = 1000)
cv_ex2 = simulateCV(ko = 0.01, e.switch = -0.8, e.form = -0.4,
  x.units = 100, t.units = 1000)
cv_ex3 = simulateCV(ko = 0.001, e.switch = -0.8, e.form = -0.4,
  x.units = 100, t.units = 1000)
cv_ex4 = simulateCV(ko = 0.0001, e.switch = -0.8, e.form = -0.4,
  x.units = 100, t.units = 1000)
plotCV(filenamees = list(cv_ex1, cv_ex2, cv_ex3, cv_ex4),
  legend_text = c("ko = 1 cm/s", "ko = 0.01 cm/s",
  "ko = 0.001 cm/s", "ko = 0.0001 cm/s"))

cv_ex1sample = sampleCV(cv_ex1, data.reduction = 5)
plotCV(filenamees = list(cv_ex1, cv_ex1sample),
  line_colors = c("blue", "red"))
```

plotDiffGrid

Plot Diffusion Grids

Description

Plots the diffusion grids used in cyclic voltammetry, linear sweep voltammetry, chronoamperometry, and chronocoulometry simulation. The diffusion grids are presented as heat maps giving the concentration of Ox, Red, or Z as a function of distance from the electrode surface on the x -axis and time on the y - axis. The scale on the x -axis may be adjusted to highlight better the diffusion grid near the electrode's surface. Note: the plotDiffGrid function requires that the plot3D package is installed.

Usage

```
plotDiffGrid(filename, species = c(TRUE, TRUE, FALSE), scale.factor = 1)
```

Arguments

`filename` Name of the file that contains the results of a cyclic voltammetry, linear sweep voltammetry, chronoamperometry, or chronocoulometry simulation created using cvSim, lsvSim, caSim, or ccSim.

| | |
|--------------|--|
| species | A vector of three logical values (T or F) indicating the species for which diffusion grids are displayed. The order in which species are identified is Ox, Red, and Z. |
| scale.factor | A factor for adjusting the scale on the x -axis. Setting the scale.factor to a value of less than 1 decreases the range of distances displayed in the diffusion grids. |

Value

Returns a single plot showing 1–3 diffusion grids.

Examples

```
ex_cv = simulateCV(e.start = 0.25, e.switch = -0.25, e.form = 0,  
  x.units = 100, t.units = 1000)  
plotDiffGrid(ex_cv, species = c(TRUE, TRUE, FALSE),  
  scale.factor = 0.5)
```

plotDiffusion

Plot Diffusion Profiles

Description

Plots a set of diffusion profiles (concentration as a function of distance from the electrode's surface) for an object created using an object created with one of the package's simulation functions: cvSim for cyclic voltammetry, lsvSim for linear sweep voltammetry, caSim for chronoamperometry, or ccSim for chronocoulometry. The plot includes a default title that gives the time and the potential for the diffusion profile. Note: this function will not work with the reduced data file created using sampleAmpgram, sampleCoulgram, or sampleVoltgram.

Usage

```
plotDiffusion(filename, t = 1)
```

Arguments

| | |
|----------|--|
| filename | Name of the file that contains the results of a simulated electrochemistry experiment. |
| t | The time for which the diffusion profile is desired. |

Value

Returns a line plot that shows distance from the electrode's surface on the x -axis and the concentrations of Ox, Red, and, for an EC or a CE mechanism, Z on the y -axis.

Examples

```
ex_cv = simulateCV(e.start = 0.25, e.switch = -0.25, e.form = 0,  
  x.units = 100, t.units = 1000)  
plotDiffusion(ex_cv, t = 0.5)
```

```
ex_ca = simulateCA(e.start = 0.25, e.pulse = -0.25, e.form = 0,  
  pulses = "double", t.2 = 20, x.units = 100, t.units = 1000)  
plotDiffusion(ex_ca, t = 21)
```

plotGrid

Grid of Diffusion Profiles

Description

Plots eight diffusion profiles—at times that are 10 30 experiment—around a central plot that shows the corresponding voltammogram, chronoamperogram, or chronocoulogram using an object created with one of the package’s simulation functions: `cvSim` for cyclic voltammetry, `lsvSim` for linear sweep voltammetry, `caSim` for chronoamperometry, or `ccSim` for chronocoulometry. Note: this function will not work with the reduced data file created using `sampleAmpgram`, `sampleCoulgram`, or `sampleVoltgram`.

Usage

```
plotGrid(filename)
```

Arguments

| | |
|----------|--|
| filename | Name of the file that contains the results of a simulated electrochemistry experiment. |
|----------|--|

Value

Returns a 3 by 3 grid of individual plots.

Examples

```
ex_cv = simulateCV(e.start = 0.25, e.switch = -0.25, e.form = 0,  
  x.units = 100, t.units = 1000)  
plotGrid(ex_cv)
```

```
ex_ca = simulateCA(e.start = 0.25, e.pulse = -0.25, e.form = 0,  
  pulses = "double", t.2 = 20, x.units = 100, t.units = 1000)  
plotGrid(ex_ca)
```

`plotLSV`*Plot Linear Sweep Voltammograms*

Description

Plots 1–5 linear sweep voltammograms on a single set of axes. The default plot does not include a legend or a title, but providing a vector of character strings to `legend_text` adds a legend to the final plot, and adding a character string for `main_title` adds a title to the plot. Line widths, line types, line colors, point symbols, and point colors have default values that can be adjusted. Note: this function accepts both full data files created using `lsvSim`, or reduced data `sampleLSV`.

Usage

```
plotLSV(filenamees = list(file1 = NULL, file2 = NULL), legend_text = NULL,  
  legend_position = c("topleft", "topright", "bottomleft", "bottomright"),  
  main_title = NULL, line_widths = c(2, 2, 2, 2, 2), line_types = c(1, 2,  
  3, 4, 5), point_symbols = c(21, 22, 23, 24, 25), line_colors = c("blue",  
  "blue", "blue", "blue", "blue"))
```

Arguments

| | |
|------------------------------|---|
| <code>filenamees</code> | A list giving the names of 1–5 files that contain the results of a simulated linear sweep voltammetry experiment. |
| <code>legend_text</code> | Optional vector that contains text to include in a legend. Default is <code>NULL</code> , which suppresses the legend. |
| <code>legend_position</code> | One of <code>topleft</code> , <code>topright</code> , <code>bottomleft</code> , or <code>bottomright</code> ; defaults to <code>topleft</code> . |
| <code>main_title</code> | An optional main title. |
| <code>line_widths</code> | A vector of line widths for the individual linear sweep voltammograms; defaults to a common line width of 2, but can be adjusted by supplying a vector with desired values. |
| <code>line_types</code> | A vector of line types for the individual linear sweep voltammograms; defaults to a set of different line types, but can be adjusted by supplying a vector with desired values. |
| <code>point_symbols</code> | A vector of <code>pch</code> values for plotting points. |
| <code>line_colors</code> | A vector of colors for the individual linear sweep voltammograms, whether displayed as lines or as points; defaults to a common color, but can be adjusted by supplying a vector with desired values. |

Value

Returns a plot that shows the applied potential on the *x*-axis and current on the *y*-axis.

Examples

```

lsv_ex1 = simulateLSV(e.start = 0.25, e.end = -0.25, e.form = 0,
  stir.rate = "fast", x.units = 100, t.units = 1000)
lsv_ex2 = simulateLSV(e.start = 0.25, e.end = -0.25, e.form = 0,
  stir.rate = "medium", x.units = 100, t.units = 1000)
lsv_ex3 = simulateLSV(e.start = 0.25, e.end = -0.25, e.form = 0,
  stir.rate = "slow", x.units = 100, t.units = 1000)
lsv_ex4 = simulateLSV(e.start = 0.25, e.end = -0.25, e.form = 0,
  stir.rate = "off", x.units = 100, t.units = 1000)
plotLSV(filename = list(lsv_ex1, lsv_ex2, lsv_ex3, lsv_ex4),
  legend_text = c("stir rate: fast", "stir rate: medium",
    "stir rate: slow", "stir rate: off"))

```

plotPotential

Plot Applied Potential

Description

Plots the applied potential as a function of time for an object created using an object created with one of the package's simulation functions: `cvSim` for cyclic voltammetry, `lsvSim` for linear sweep voltammetry, `caSim` for chronoamperometry, or `ccSim` for chronocoulometry. Note: this function will not work with the reduced data file created using `sampleAmpgram`, `sampleCoulgram`, or `sampleVoltgram`.

Usage

```
plotPotential(filename, main_title = NULL)
```

Arguments

| | |
|------------|--|
| filename | Name of the file that contains the results of a simulated electrochemistry experiment. |
| main_title | An optional main title. |

Value

Returns a line plot that shows time on the *x*-axis and the applied potential on the *y*-axis.

Examples

```

ex_cv = simulateCV(e.start = 0.25, e.switch = -0.25, e.form = 0,
  x.units = 100, t.units = 1000)
plotPotential(ex_cv,
  main_title = "Applied Potential for a Cyclic Voltammetry Simulation")

ex_ca = simulateCA(e.start = 0.25, e.pulse = -0.25, e.form = 0,

```



```
pulses = "double", t.2 = 20, x.units = 100, t.units = 1000)
plotPotential(ex_ca,
  main_title = "Applied Potential for a Chronoamperometry Simulation")
```

sampleCA

Create Subsample of a Simulated Chronoamperogram

Description

Used to create a reduced data file of times and currents for a chronoamperogram. When passed to plotCA, the resulting plot shows the data as discrete points instead of as a line.

Usage

```
sampleCA(filename, data.reduction = 1)
```

Arguments

| | |
|----------------|---|
| filename | The filename that contains the result of a simulated chronoamperometry experiment (created using the caSim function). |
| data.reduction | A value that gives the percentage of the original data to keep, which then is spaced evenly across the full data set |

Value

Returns a list with the following components

| | |
|-----------|---|
| expt | type of experiment; CA for a chronoamperometry experiment |
| file_type | value that indicates whether the output includes all data (full) or a subset of data (reduced); defaults to reduced |
| current | vector giving the current as a function of time |
| potential | vector giving the potential as a function of time |

Examples

```
ex_ca = simulateCA(e.start = 0.25, e.pulse = -0.25, e.form = 0,
  pulses = "double", t.2 = 20, x.units = 100, t.units = 1000)
ex_casample = sampleCA(ex_ca, data.reduction = 5)
str(ex_casample)
```

| | |
|----------|--|
| sampleCC | <i>Create Subsample of a Simulated Chronocoulogram</i> |
|----------|--|

Description

Used to create a reduced data file of times and currents for a chronocoulogram. When passed to plotCC, the resulting plot shows the data as discrete points instead of as a line.

Usage

```
sampleCC(filename, data.reduction = 1)
```

Arguments

| | |
|----------------|---|
| filename | The filename that contains the result of a simulated chronocoulometry experiment (created using the ccSim function). |
| data.reduction | A value that gives the percentage of the original data to keep, which then is spaced evenly across the full data set. |

Value

Returns a list with the following components

| | |
|-----------|---|
| expt | type of experiment; CC for a chronocoulometry experiment |
| file_type | value that indicates whether the output includes all data (full) or a subset of data (reduced); defaults to reduced |
| charge | vector giving the charge as a function of time |
| time | vector giving the time |

Examples

```
ex_ca = simulateCA(e.start = 0.25, e.pulse = -0.25, e.form = 0,  
  pulses = "double", t.2 = 20, x.units = 100, t.units = 1000)  
ex_cc = simulateCC(ex_ca)  
ex_ccsample = sampleCC(ex_cc, data.reduction = 5)  
str(ex_ccsample)
```

`sampleCV`*Create Subsample of a Simulated Cyclic Voltammogram*

Description

Used to create a reduced data file of potentials and currents for a cyclic voltammogram or a linear sweep voltammogram. When passed to `plotCV`, the resulting plot shows the data as discrete points instead of as a line.

Usage

```
sampleCV(filename, data.reduction = 1)
```

Arguments

| | |
|-----------------------------|---|
| <code>filename</code> | The filename that contains the result of a simulated cyclic voltammetry experiment (created using the <code>cvSim</code> function). |
| <code>data.reduction</code> | A value that gives the percentage of the original data to keep, which then is spaced evenly across the full data set. |

Value

Returns a list with the following components

| | |
|------------------------|---|
| <code>expt</code> | type of experiment; CV for a cyclic voltammetry simulation |
| <code>file_type</code> | value that indicates whether the output includes all data (full) or a subset of data (reduced); defaults to reduced |
| <code>current</code> | vector giving the current as a function of time |
| <code>potential</code> | vector giving the potential as a function of time |

Examples

```
ex_cv = simulateCV(e.start = 0.25, e.switch = -0.25, e.form = 0,  
  x.units = 100, t.units = 1000)  
ex_cvsample = sampleCV(ex_cv, data.reduction = 5)  
str(ex_cvsample)
```

`sampleLSV`*Create Subsample of a Simulated Linear Sweep Voltammogram*

Description

Used to create a reduced data file of potentials and currents for a linear sweep voltammogram. When passed to `plotLSV`, the resulting plot shows the data as discrete points instead of as a line.

Usage

```
sampleLSV(filename, data.reduction = 1)
```

Arguments

| | |
|-----------------------------|--|
| <code>filename</code> | The filename that contains the result of a simulated linear sweep voltammetry experiment (created using the <code>lsvSim</code> function). |
| <code>data.reduction</code> | A value that gives the percentage of the original data to keep, which then is spaced evenly across the full data set. |

Value

Returns a list with the following components

| | |
|------------------------|---|
| <code>expt</code> | type of experiment; LSV for a linear sweep voltammetry experiment |
| <code>file_type</code> | value that indicates whether the output includes all data (full) or a subset of data (reduced); defaults to reduced |
| <code>current</code> | vector giving the current as a function of time |
| <code>potential</code> | vector giving the potential as a function of time |

Examples

```
ex_lsv = simulateLSV(e.start = 0.25, e.end = -0.25, e.form = 0,  
  stir.rate = "fast", x.units = 100, t.units = 1000)  
ex_lsvsample = sampleLSV(ex_lsv, data.reduction = 5)  
str(ex_lsvsample)
```

simulateCA

*Simulate a Chronoamperometry Experiment***Description**

Simulates either a single pulse or a double pulse chronoamperometry experiment using either an E, EC, or CE mechanism, where E is a redox reaction and where C is a chemical reaction that either precedes or follows the redox reaction.

Usage

```
simulateCA(e.start = 0, e.pulse = -0.5, e.form = -0.25,
  mechanism = c("E", "EC", "CE"), ko = 1, kcf = 0, kcr = 0,
  pulses = c("single", "double"), t.1 = 10, t.2 = 0, t.end = 30,
  n = 1, alpha = 0.5, d = 1e-05, area = 0.01, temp = 298.15,
  conc.bulk = 0.001, t.units = 2000, x.units = 180, sd.noise = 0)
```

Arguments

| | |
|-----------|---|
| e.start | Initial potential (in volts). |
| e.pulse | Potential after applying the initial pulse (in volts). |
| e.form | Formal potential for the redox reaction (in volts). |
| mechanism | Mechanism for the electrochemical system; one of E for redox reaction only, EC for redox reaction with a following chemical reaction, or CE for redox reaction with a preceding chemical reaction. Default is E. |
| ko | Standard heterogeneous electron transfer rate constant for the redox reaction (in cm/s). |
| kcf | Homogeneous first-order rate constant for the forward chemical reaction (in s ⁻¹). |
| kcr | Homogeneous first-order rate constant for the reverse chemical reaction (in s ⁻¹). |
| pulses | Either single or double. For a single pulse experiment, the initial potential is e.start and the final potential is e.pulse, and for a double pulse potential, the initial potential is e.start, the intermediate potential is e.pulse, and the final potential is e.start; the default is a single pulse experiment. |
| t.1 | The time at which the first pulse is applied (in s). |
| t.2 | The time at which the second pulse is applied (in s). |
| t.end | The time at which the experiment ends (in s). |
| n | Number of electrons in the redox reaction. |
| alpha | Transfer coefficient. |
| d | Diffusion coefficient for Ox and Red (in cm ² s ⁻¹). |
| area | Surface area of the electrode (in cm ²). |

| | |
|-----------|---|
| temp | Temperature (in K). |
| conc.bulk | Initial bulk concentration of Ox or Red for an E or an EC mechanism, or the combined initial concentrations of Ox and Z, or of Red and Z for a CE mechanism (in mol/L). |
| t.units | The number of increments in time for the diffusion grids. |
| x.units | The number of increments in distance for the diffusion grids. |
| sd.noise | The standard deviation for noise as a percent of maximum current (in μA). |

Value

Returns a list with the following components

| | |
|-------------|---|
| expt | type of experiment; defaults to CA for a chronoamperometry simulation |
| mechanism | type of mechanism used for the simulation |
| file_type | value that indicates whether the output includes all data (full) or a subset of data (reduced); defaults to full for caSim |
| current | vector giving the current as a function of time |
| potential | vector giving the potential as a function of time |
| time | vector giving the times used for the diffusion grids |
| distance | vector giving the distances from electrode surface used for the diffusion grids |
| oxdata | diffusion grid, as a matrix, giving the concentration of Ox |
| reddata | diffusion grid, as a matrix, giving the concentrations of Red |
| chemdata | diffusion grid, as a matrix, giving the concentrations of Z |
| formalE | formal potential for the redox reaction |
| initialE | initial potential |
| pulseE | potential after apply the initial pulse |
| electrons | number of electrons, n, in the redox reaction |
| ko | standard heterogeneous electron transfer rate constant |
| kcf | homogeneous first-order rate constant for forward chemical reaction |
| kcr | homogeneous first-order rate constant for reverse chemical reaction |
| alpha | transfer coefficient |
| diffcoef | diffusion coefficient for Ox and Red |
| area | surface area for electrode |
| temperature | temperature |
| conc.bulk | initial concentration of Ox or Red for an E or EC mechanism, or the combined initial concentrations of Ox and Z, or of Red and Z for a CE mechanism |
| tunits | the number of increments in time for the diffusion grids |
| xunits | the number of increments in distance for the diffusion grids |
| sdnoise | standard deviation, as percent of maximum current, used to add noise to simulated data |

| | |
|-------------|--|
| direction | -1 for an initial reduction reaction of Ox to Red; +1 for an initial oxidation reaction of Red to Ox |
| pulses | number of pulses: either single or double |
| time_pulse1 | time when first pulse is applied |
| time_pulse2 | time when second pulse is applied |
| time_end | time when experiment ends |
| k_f | vector of forward electron transfer rate constant as a function of potential |
| k_b | vector of reverse electron transfer rate constant as a function of potential |
| jox | vector giving the flux of Ox to the electrode surface as a function of potential |
| jred | vector giving the flux of Red to the electrode surface as a function of potential |

Examples

```
ex_ca = simulateCA(e.start = 0.25, e.pulse = -0.25, e.form = 0,
  pulses = "double", t.2 = 20, x.units = 100, t.units = 1000)
str(ex_ca)
```

simulateCC

Simulate a Chronocoulometry Experiment

Description

Simulates either a single pulse or a double pulse chronocoulometry experiment as either an E, EC, or CE mechanism, where E is a redox reaction and where C is a chemical reaction that either precedes or follows the redox reaction. The function operates on an object created using caSim, which simulates the corresponding chronoamperometry experiment, integrating current over time using the trapezoidal integration rule.

Usage

```
simulateCC(filename)
```

Arguments

| | |
|----------|---|
| filename | The filename that contains the results of a chronamperometry simulation created using the caSim function. |
|----------|---|

Value

Returns a list with the following components

| | |
|-----------|--|
| expt | type of experiment; defaults to CC for a chronocoulometry simulation |
| mechanism | type of mechanism used for the simulation |
| file_type | value that indicates whether the output includes all data (full) or a subset of data (reduced); defaults to full for ccSim |

| | |
|-------------|---|
| charge | vector giving the charge as a function of time |
| potential | vector giving the potential as a function of time |
| time | vector giving the times used for the diffusion grids |
| distance | vector giving the distances from electrode surface used for the diffusion grids |
| oxdata | diffusion grid, as a matrix, giving the concentration of Ox |
| reddata | diffusion grid, as a matrix, giving the concentrations of Red |
| chemdata | diffusion grid, as a matrix, giving the concentrations of Z |
| formalE | formal potential for the redox reaction |
| initialE | initial potential |
| pulseE | potential after apply the initial pulse |
| electrons | number of electrons, n, in the redox reaction |
| ko | standard heterogeneous electron transfer rate constant |
| kcf | homogeneous first-order rate constant for forward chemical reaction |
| kcr | homogeneous first-order rate constant for reverse chemical reaction |
| alpha | transfer coefficient |
| diffcoef | diffusion coefficient for Ox and Red |
| area | surface area for electrode |
| temperature | temperature |
| conc.bulk | initial concentration of Ox or Red for an E or EC mechanism, or the combined initial concentrations of Ox and Z, or of Red and Z for a CE mechanism |
| tunits | the number of increments in time for the diffusion grids |
| xunits | the number of increments in distance for the diffusion grids |
| sdnoise | standard deviation, as percent of maximum current, used to add noise to simulated data |
| direction | -1 for an initial reduction reaction of Ox to Red; +1 for an initial oxidation reaction of Red to Ox |
| pulses | number of pulses: either single or double |
| time_pulse1 | time when first pulse is applied |
| time_pulse2 | time when second pulse is applied |
| time_end | time when experiment ends |
| k_f | vector of forward electron transfer rate constant as a function of potential |
| k_b | vector of reverse electron transfer rate constant as a function of potential |
| jox | vector giving the flux of Ox to the electrode surface as a function of potential |
| jred | vector giving the flux of Red to the electrode surface as a function of potential |

Examples

```
ex_ca = simulateCA(e.start = 0.25, e.pulse = -0.25, e.form = 0,
  pulses = "double", t.2 = 20, x.units = 100, t.units = 1000)
ex_cc = simulateCC(ex_ca)
str(ex_cc)
```


simulateCV

*Simulate a Cyclic Voltammetry Experiment***Description**

Simulates a cyclic voltammetry experiment as either an E, EC, or CE mechanism, where E is a redox reaction and where C is a chemical reaction that either precedes or follows the redox reaction.

Usage

```
simulateCV(e.start = 0, e.switch = -0.5, e.form = -0.25,
  mechanis = c("E", "EC", "CE"), ko = 1, kcf = 0, kcr = 0, n = 1,
  alpha = 0.5, d = 1e-05, area = 0.01, temp = 298.15, scan.rate = 1,
  conc.bulk = 0.001, t.units = 2000, x.units = 180, sd.noise = 0)
```

Arguments

| | |
|-----------|--|
| e.start | Initial potential (in volts). |
| e.switch | Switching potential (in volts). |
| e.form | Formal potential for the redox reaction (in volts). |
| mechanism | Mechanism for the electrochemical system; one of E for redox reaction only, EC for redox reaction with a following chemical reaction, or CE for redox reaction with a preceding chemical reaction. Default is E. |
| ko | Standard heterogeneous electron transfer rate constant for the redox reaction (in cm/s). |
| kcf | Homogeneous first-order rate constant for the forward chemical reaction (in s ⁻¹). |
| kcr | Homogeneous first-order rate constant for the reverse chemical reaction (in s ⁻¹). |
| n | Number of electrons in the redox reaction. |
| alpha | Transfer coefficient. |
| d | Diffusion coefficient for Ox and Red (in cm ² s ⁻¹). |
| area | Surface area of the electrode (in cm ²). |
| temp | Temperature (in K). |
| scan.rate | Rate at which the potential is changed (in V/s). |
| conc.bulk | Initial bulk concentration of Ox or Red for an E or an EC mechanism, or the combined initial concentrations of Ox and Z, or of Red and Z for a CE mechanism (in mol/L). |
| t.units | The number of increments in time for the diffusion grids. |
| x.units | The number of increments in distance for the diffusion grids. |
| sd.noise | The standard deviation for noise as a percent of maximum current (in μ A). |

Value

Returns a list with the following components

| | |
|-------------|---|
| expt | type of experiment; defaults to CV for a cyclic voltammetry simulation |
| mechanism | type of mechanism used for the simulation |
| file_type | value that indicates whether the output includes all data (full) or a subset of data (reduced); defaults to full for cvSim |
| current | vector giving the current as a function of time |
| potential | vector giving the potential as a function of time |
| time | vector giving the times used for the diffusion grids |
| distance | vector giving the distances from electrode surface used for the diffusion grids |
| oxdata | diffusion grid, as a matrix, giving the concentration of Ox |
| reddata | diffusion grid, as a matrix, giving the concentrations of Red |
| chemdata | diffusion grid, as a matrix, giving the concentrations of Z |
| formalE | formal potential for the redox reaction |
| initialE | initial potential |
| switchE | switching potential |
| electrons | number of electrons, n, in the redox reaction |
| ko | standard heterogeneous electron transfer rate constant |
| kcf | homogeneous first-order rate constant for forward chemical reaction |
| kcr | homogeneous first-order rate constant for reverse chemical reaction |
| alpha | transfer coefficient |
| diffcoef | diffusion coefficient for Ox and Red |
| area | surface area for electrode |
| temperature | temperature |
| scanrate | scan rate |
| conc.bulk | initial concentration of Ox or Red for an E or EC mechanism, or the combined initial concentrations of Ox and Z, or of Red and Z for a CE mechanism |
| tunits | the number of increments in time for the diffusion grids |
| xunits | the number of increments in distance for the diffusion grids |
| sdnoise | standard deviation, as percent of maximum current, used to add noise to simulated data |
| direction | -1 for an initial reduction reaction of Ox to Red; +1 for an initial oxidation reaction of Red to Ox |
| k_f | vector of forward electron transfer rate constant as a function of potential |
| k_b | vector of reverse electron transfer rate constant as a function of potential |
| jox | vector giving the flux of Ox to the electrode surface as a function of potential |
| jred | vector giving the flux of Red to the electrode surface as a function of potential |

Examples

```
ex_cv = simulateCV(e.start = 0.25, e.switch = -0.25, e.form = 0,
  x.units = 100, t.units = 1000)
str(ex_cv)
```

simulateLSV

Simulate a Linear Sweep Voltammetry Experiment

Description

Simulates a linear sweep voltammetry experiment as either an E, EC, or CE mechanism, where E is a redox reaction and where C is a chemical reaction that either precedes or follows the redox reaction.

Usage

```
simulateLSV(e.start = 0, e.end = -1, e.form = -0.25, mechanism = c("E",
  "EC", "CE"), ko = 1, kcf = 0, kcr = 0, n = 1, alpha = 0.5,
  d = 1e-05, area = 0.01, temp = 298.15, scan.rate = 1,
  conc.bulk = 0.001, t.units = 2000, x.units = 180, sd.noise = 0,
  stir.rate = c("off", "slow", "medium", "fast"))
```

Arguments

| | |
|-----------|--|
| e.start | Initial potential (in volts). |
| e.end | Final potential (in volts). |
| e.form | Formal potential for the redox reaction (in volts). |
| mechanism | Mechanism for the electrochemical system; one of E for redox reaction only, EC for redox reaction with a following chemical reaction, or CE for redox reaction with a preceding chemical reaction. Default is E. |
| ko | Standard heterogeneous electron transfer rate constant for the redox reaction (in cm/s). |
| kcf | Homogeneous first-order rate constant for the forward chemical reaction (in s ⁻¹). |
| kcr | Homogeneous first-order rate constant for the reverse chemical reaction (in s ⁻¹). |
| n | Number of electrons in the redox reaction. |
| alpha | Transfer coefficient. |
| d | Diffusion coefficient for Ox and Red (in cm ² s ⁻¹). |
| area | Surface area of the electrode (in cm ²). |
| temp | Temperature (in K). |
| scan.rate | Rate at which the potential is changed (in V/s). |

| | |
|-----------|---|
| conc.bulk | Initial bulk concentration of Ox or Red for an E or an EC mechanism, or the combined initial concentrations of Ox and Z, or of Red and Z for a CE mechanism (in mol/L). |
| t.units | The number of increments in time for the diffusion grids. |
| x.units | The number of increments in distance for the diffusion grids. |
| sd.noise | The standard deviation for noise as a percent of maximum current (in μA). |
| stir.rate | The rate at which the solution is stirred with options for off, or no stirring, and for slow, medium, and fast stirring; defaults to off. |

Value

Returns a list with the following components

| | |
|-------------|---|
| expt | type of experiment; defaults to LSV for a linear sweep voltammetry simulation |
| mechanism | type of mechanism used for the simulation |
| file_type | value that indicates whether the output includes all data (full) or a subset of data (reduced); defaults to full for lsvSim |
| current | vector giving the current as a function of time |
| potential | vector giving the potential as a function of time |
| time | vector giving the times used for the diffusion grids |
| distance | vector giving the distances from electrode surface used for the diffusion grids |
| oxdata | diffusion grid, as a matrix, giving the concentration of Ox |
| reddata | diffusion grid, as a matrix, giving the concentrations of Red |
| chemdata | diffusion grid, as a matrix, giving the concentrations of Z |
| formalE | formal potential for the redox reaction |
| initialE | initial potential |
| endE | end potential |
| electrons | number of electrons, n, in the redox reaction |
| ko | standard heterogeneous electron transfer rate constant |
| kcf | homogeneous first-order rate constant for forward chemical reaction |
| kcr | homogeneous first-order rate constant for reverse chemical reaction |
| alpha | transfer coefficient |
| diffcoef | diffusion coefficient for Ox and Red |
| area | surface area for electrode |
| temperature | temperature |
| scanrate | scan rate |
| conc.bulk | initial concentration of Ox or Red for an E or EC mechanism, or the combined initial concentrations of Ox and Z, or of Red and Z for a CE mechanism |
| tunits | the number of increments in time for the diffusion grids |
| xunits | the number of increments in distance for the diffusion grids |

| | |
|-----------|--|
| sdnoise | standard deviation, as percent of maximum current, used to add noise to simulated data |
| direction | -1 for an initial reduction reaction of Ox to Red; +1 for an initial oxidation reaction of Red to Ox |
| stir_rate | rate at which solution is stirred |
| k_f | vector of forward electron transfer rate constant as a function of potential |
| k_b | vector of reverse electron transfer rate constant as a function of potential |
| jox | vector giving the flux of Ox to the electrode surface as a function of potential |
| jred | vector giving the flux of Red to the electrode surface as a function of potential |

Examples

```
ex_lsv = simulateLSV(e.start = 0.25, e.end = -0.25, e.form = 0,  
  stir.rate = "fast", x.units = 100, t.units = 1000)  
str(ex_lsv)
```

Index

[animateCA](#), [3](#)
[animateCC](#), [3](#)
[animateCV](#), [4](#)
[animateLSV](#), [5](#)
[annotateCA](#), [5](#)
[annotateCC](#), [6](#)
[annotateCV](#), [7](#)
[annotateLSV](#), [8](#)

[eChem-package](#), [2](#)

[plotCA](#), [8](#)
[plotCC](#), [10](#)
[plotCV](#), [11](#)
[plotDiffGrid](#), [12](#)
[plotDiffusion](#), [13](#)
[plotGrid](#), [14](#)
[plotLSV](#), [15](#)
[plotPotential](#), [16](#)

[sampleCA](#), [17](#)
[sampleCC](#), [18](#)
[sampleCV](#), [19](#)
[sampleLSV](#), [20](#)
[simulateCA](#), [21](#)
[simulateCC](#), [23](#)
[simulateCV](#), [25](#)
[simulateLSV](#), [27](#)