Package 'hydroroute'

April 4, 2022

Type Package

Title Trace Longitudinal Hydropeaking Waves

Version 0.1.1

Description Implements an empirical approach referred to as PeakTrace which uses multiple hydrographs to detect and follow hydropower plant-specific hydropeaking waves at the subcatchment scale and to describe how hydropeaking flow parameters change along the longitudinal flow path. The method is based on the identification of associated events and uses (linear) regression models to describe translation and retention processes between neighboring hydrographs. Several regression model results are combined to arrive at a power plant-specific model. The approach is proposed and validated in Greimel et al. (2022, accepted with minor revisions). The identification of associated events is based on the event detection implemented in 'hydropeak'.

License GPL-2

Encoding UTF-8

Depends R (>= 4.1.0)

Imports dplyr, ggpmisc, ggplot2, gridExtra, hydropeak, lubridate, parallel, reshape2, stats, utils, scales

RoxygenNote 7.1.2

Suggests rmarkdown, knitr

VignetteBuilder knitr

NeedsCompilation no

Author Bettina Grün [cre, ctb] (<https://orcid.org/0000-0001-7265-4773>), Julia Haider [aut], Franz Greimel [ctb] (<https://orcid.org/0000-0002-8000-1227>)

Maintainer Bettina Grün <Bettina.Gruen@R-project.org>

Repository CRAN

Date/Publication 2022-04-04 06:40:02 UTC

R topics documented:

`estimate_AE` *Estimate Associated Events*

Description

For two neighboring stations, potential associated events (AEs) are determined according to the time lag and metric (amplitude) difference allowed. For all potential AEs, parabolas are fitted to the histogram obtained for the relative difference in amplitude binned into intervals from -1 to 1 of width 0.1 by fixing the vertex at the inner maximum of the histogram and the width is determined by minimizing the average squared distances between the parabola and the histogram data along arbitrary symmetric ranges from the inner maximum. Based on the fitted parabola, cut points with the x-axis are determined such that only those potential AEs are retained where the relative difference is within these cut points. If this automatic scheme does not succeed to determine suitable cut points, e.g., because the estimated cut points are outside -1 and 1, then a strict criterion for the relative difference in amplitude is imposed to identify AEs considering only deviations of at most 10%.

Usage

```
estimate_AE(
  Sx,
  Sy,
  relation,
  timeLag = c(1, 1, 1),
  metricLag = c(1, 1),
  unique = c("time", "metric"),
  TimeFormat = "%Y-%m-%d %H:%M",
  tz = "Etc/GMT-1"
)
```

Arguments

`Sx`	Data frame that consists of flow fluctuation events and computed metrics (see `hydropeak::get_events()`) of an upstream hydrograph S_x.
`Sy`	Data frame that consists of flow fluctuation events and computed metrics (see `hydropeak::get_events()`) of a downstream hydrograph S_y.

`relation` Data frame that contains the relation between upstream and downstream hydrograph. Must only contain two rows (one for each hydrograph) in order of their location in downstream direction. See the appended example data `relation.csv` or the vignette for details on the structure. See `get_lag()` for further information about the relation and the lag between the hydrographs.

`timeLag` Numeric vector specifying factors to alter the interval to capture events from the downstream hydrograph. By default it is `timeLag = c(1,1,1)`, this refers to matches within a time slot $\pm$ the mean translation time from `relation`. For exact time matches, `timeLag = c(0,1,0)` must be specified.

`metricLag` Numeric vector specifying factors to alter the interval of relative metric deviations to capture events from the downstream hydrograph. By default. it is `metricLag = c(1,1)`, such that events are filtered where the amplitude at S_y is at least 0, i.e., amplitude at $S_x - 1\cdot$ amplitude at S_x, and at most two times the amplitude at S_x, i.e., $S_x + 1\cdot$ amplitude at S_x. For exact matches, `metricLag = c(0,0)` must be specified.

`unique` Character string specifying if the potential AEs which meet the `timeLag` and `metricLag` condition should be filtered to contain only unique events using `"time"`, i.e., by selecting those where the time difference is smallest compared to the specified factor of the mean translation time, or using `"metric"`, i.e., by selecting those where the relative difference in amplitude is smallest (default: `"time"`).

`TimeFormat` Character string giving the date-time format of the date-time column in the input data frame (default: "%Y-%m-%d %H:%M").

`tz` Character string specifying the time zone to be used for the conversion (default: "Etc/GMT-1").

Value

A nested list containing the estimated settings, the histogram obtained for the relative difference data with estimated cut points, and the obtained "real" AEs.

Examples

```
# file paths
Sx <- system.file("testdata", "Events", "100000_2_2014-01-01_2014-02-28.csv",
                  package = "hydroroute")
Sy <- system.file("testdata", "Events", "200000_2_2014-01-01_2014-02-28.csv",
                  package = "hydroroute")
relation <- system.file("testdata", "relation.csv", package = "hydroroute")

# read data
Sx <- utils::read.csv(Sx)
Sy <- utils::read.csv(Sy)
relation <- utils::read.csv(relation)
relation <- relation[1:2, ]

# estimate AE, exact time matches
results <- estimate_AE(Sx, Sy, relation, timeLag = c(0, 1, 0))
results$settings
```

```
results$plot_threshold
results$real_AE
```

`extract_AE` *Extract Associated Events*

Description

For given relation and event data return the associated events which comply with the conditions specified in the settings.

Usage

```
extract_AE(
  relation_path,
  events_path,
  settings_path,
  unique = c("time", "metric"),
  inputdec = ".",
  inputsep = ",",
  saveResults = FALSE,
  outdir = tempdir(),
  TimeFormat = "%Y-%m-%d %H:%M",
  tz = "Etc/GMT-1"
)
```

Arguments

`relation_path`	Character string containing the path of the file where the relation file is to be read from with `utils::read.csv()`. The file must contain a column `ID` that contains the gauging station ID's in the file have to be in order of their location in downstream direction.
`events_path`	Character string containing the path of the directory where the event files corresponding to the 'relation' file are located. Only relevant files in this directory will be used, i.e., files that are related to the 'relation' file.
`settings_path`	Character string containing the path of the file where the settings file is to be read from with `utils::read.csv()`. The file must be in the format of the output of `peaktrace()`.
`unique`	Character string specifying if the potential AEs which meet the `timeLag` and `metricLag` condition should be filtered to contain only unique events using `"time"`, i.e., by selecting those where the time difference is smallest compared to the specified factor of the mean translation time, or using `"metric"`, i.e., by selecting those where the relative difference in amplitude is smallest (default: `"time"`).
`inputdec`	Character string for decimal points in input data.
`inputsep`	Field separator character string for input data.

saveResults	A logical. If FALSE (default), the extracted AEs are not saved. Otherwise the extracted AEs are written to a csv file.
outdir	Character string naming a directory where the extraced AEs should be saved to.
TimeFormat	Character string giving the date-time format of the date-time column in the input data frame (default: "%Y-%m-%d %H:%M").
tz	Character string specifying the time zone to be used for the conversion (default: "Etc/GMT-1").

Value

A data frame containing "real" AEs (i.e., events where the time differences and the relative difference in amplitude is within the limits and cut points provided by the file in settings_path). If no AEs can be found between the first two neighboring stations, NULL is returned. Otherwise the function returns all "real" AEs that could be found along the river section specified in the file from relation_path. A warning message informs when the extraction has stopped early and shows the IDs for which no AEs could be found.

Examples

```
relation_path <- system.file("testdata", "relation.csv", package = "hydroroute")
events_path <- system.file("testdata", "Events", package = "hydroroute")
settings_path <- system.file("testdata", "Q_event_2_AMP-LAG_settings.csv",
                                    package = "hydroroute")
real_AE <- extract_AE(relation_path, events_path, settings_path)
```

---

get_lag *Get Lag*

Description

Given a data frame (time series) of measurements and a vector of gauging station ID's in order of their location in downstream direction, the lag (the amount of passing time between two gauging stations) is estimated based on the cross-correlation function (ccf) of the time series of two adjacent gauging stations (stats::ccf()). To ensure that the same time period is used for every gauging station, intersecting time steps are determined. These time steps are used to estimate the lags. The result of stats::ccf() is rounded to four decimals before selecting the optimal time lag so that minimal differences are neglected. If there are multiple time steps with the highest correlation, the smallest time step is considered. If the highest correlation corresponds to a zero lag or positive lag (note that the result should usually be negative as measurements at the lower gauge are later recorded as measurements at the upper gauge), a time step of length 1 is selected and a warning message is generated.

Usage

```
get_lag(
  Q,
  relation,
  steplength = 15,
  lag.max = 20,
  na.action = na.pass,
  mc.cores = getOption("mc.cores", 2L),
  tz = "Etc/GMT-1",
  format = "%Y.%m.%d %H:%M",
  cols = c(1, 2, 3)
)
```

Arguments

Q	Data frame (time series) of measurements which contains at least a column with the gauging station ID's (default: column index 1), a column with date-time values in character representation (default: column index 2) and a column with flow measurements (default: column index 3). If the column indices differ from c(1,2,3), they have to be specified in the cols argument in the format c(i,j,k).
relation	A character vector containing the gauging station ID's in order of their location in downstream direction.
steplength	Numeric value that specifies the length between time steps in minutes (default: 15 minutes). As time steps have to be equispaced, this is used by hydropeak::flow() to get a compatible format and fill missing time steps with NA.
lag.max	Numeric value that specifies the maximum lag at which to calculate the ccf in stats::ccf() (default: 20).
na.action	Function to be called to handle missing values in stats::ccf() (default: na.pass).
mc.cores	Number of cores to use with parallel::mclapply(). On Windows, this is set to 1.
tz	Character string specifying the time zone to be used for internal conversion (default: Etc/GMT-1).
format	Character string giving the date-time format of the date-time column in the input data frame Q. This is passed to hydropeak::flow(), to get a compatible format (default: YYYY.mm.dd HH:MM).
cols	Integer vector specifying column indices in Q. The default indices are 1 (ID), 2 (date-time) and 3 (flow rate, Q). This is passed to hydropeak::flow().

Value

A character vector which contains the estimated cumulative lag between neighboring gauging stations in the format HH:MM.

Examples

```
Q_path <- system.file("testdata", "Q.csv", package = "hydroroute")
Q <- utils::read.csv(Q_path)

relation_path <- system.file("testdata", "relation.csv",
                             package = "hydroroute")
relation <- utils::read.csv(relation_path)
# from relation data frame
get_lag(Q, relation$ID, format = "%Y-%m-%d %H:%M", tz = "Etc/GMT-1")

# station ID's in downstream direction as vector
relation <- c("100000", "200000", "300000", "400000")
get_lag(Q, relation, format = "%Y-%m-%d %H:%M", tz = "Etc/GMT-1")
```

---

get_lag_dir *Get Lag from Input Directory*

Description

Given a file path it reads a data frame (time series) of measurements. For each `relation` file in the provided directory path it calls `get_lag_file()`. Make sure that the file with Q data and the relation files have the same separator (`inputsep`) and character for decimal points (`inputdec`). Gauging station ID's in the `relation` files have to be in order of their location in downstream direction. The resulting lags are appended to the relation files. The resulting list of relation files can be returned and each relation file can be saved to its input path.

Usage

```
get_lag_dir(
  Q,
  relation,
  steplength = 15,
  lag.max = 20,
  na.action = na.pass,
  tz = "Etc/GMT-1",
  format = "%Y.%m.%d %H:%M",
  cols = c(1, 2, 3),
  inputsep = ",",
  inputdec = ".",
  relation_pattern = "relation",
  save = FALSE,
  mc.cores = getOption("mc.cores", 2L),
  overwrite = FALSE
)
```

Arguments

Q	Data frame or character string. If it is a data frame, it corresponds to the Q data frame in get_lag(). It contains at least a column with the gauging station ID's (default: column index 1), a column with date-time values in character representation (default: column index 2) and a column with flow measurements (default: column index 3). If the column indices differ from c(1,2,3), they have to be specified as cols argument in the format c(i,j,k). If it is a character string, it contains the path to the corresponding file which is then read within the function with utils::read.csv().
relation	A character string containing the path to the directory where the relation files are located. They are read within the function with utils::read.csv().
steplength	Numeric value that specifies the length between time steps in minutes (default: 15 minutes). As time steps have to be equispaced, this is used by hydropeak::flow() to get a compatible format and fill missing time steps with NA.
lag.max	Maximum lag at which to calculate the ccf in stats::ccf() (default: 20).
na.action	Function to be called to handle missing values in stats::ccf() (default: na.pass).
tz	Character string specifying the time zone to be used for internal conversion (default: Etc/GMT-1).
format	Character string giving the date-time format of the date-time column in the input data frame Q. This is passed to hydropeak::flow(), to get a compatible format (default: YYYY.mm.dd HH:MM).
cols	Integer vector specifying column indices in the input data frame which contain gauging station ID, date-time and flow rate to be renamed. The default indices are 1 (ID), 2 (date-time) and 3 (flow rate, Q).
inputsep	Field separator character string for input data.
inputdec	Character string for decimal points in input data.
relation_pattern	Character string containing a regular expression to filter relation files (default: relation, to filter files that contain relation with no restriction) (see base::grep()).
save	A logical. If FALSE (default) the lag, appended to the relation file, overwrites the original relation input file.
mc.cores	Number of cores to use with parallel::mclapply(). On Windows, this is set to 1.
overwrite	A logical. If FALSE (default), it produces an error if a LAG column already exists in the relation file. Otherwise, it overwrites an existing column.

Value

Returns invisibly a list of data frames where each list element represents a relation file from the input directory. Optionally, the data frames are used to overwrite the existing relation files with the appended LAG column.

Examples

```
Q_file <- system.file("testdata", "Q.csv", package = "hydroroute")
relations_path <- system.file("testdata", package = "hydroroute")
lag_list <- get_lag_dir(Q_file, relations_path, inputsep = ",",
                        inputdec = ".", format = "%Y-%m-%d %H:%M",
                        overwrite = TRUE)
lag_list
```

---

get_lag_file *Get Lag from Input File*

Description

Given a file path it reads a data frame (time series) of measurements which combines several gauging station ID's and calls `get_lag()`. The relation (ID's) of gauging stations is read from a file (provided through the file path). The file with `Q` data and the relation file need to have the same separator (`inputsep`) and character for decimal points (`inputdec`). Gauging station ID's have to be in order of their location in downstream direction. The resulting lag is appended to the relation file. This can be saved to a file.

Usage

```
get_lag_file(
  Q_file,
  relation_file,
  steplength = 15,
  lag.max = 20,
  na.action = na.pass,
  tz = "Etc/GMT-1",
  format = "%Y.%m.%d %H:%M",
  cols = c(1, 2, 3),
  inputsep = ";",
  inputdec = ".",
  save = FALSE,
  outfile = file.path(tempdir(), "relation.csv"),
  mc.cores = getOption("mc.cores", 2L),
  overwrite = FALSE
)
```

Arguments

`Q_file`	Data frame or character string. If it is a data frame, it corresponds to the `Q` data frame in `get_lag()`. It contains at least a column with the gauging station ID's (default: column index 1), a column with date-time values in character representation (default: column index 2) and a column with flow measurements (default: column index 3). If the column indices differ from `c(1,2,3)`, they have to be specified as `cols` argument in the format `c(i,j,k)`. If it is a character

	string, it contains the path to the corresponding file which is then read within the function with `utils::read.csv()`.
`relation_file`	A character string containing the path to the relation file. It is read within the function with `utils::read.csv()`. The file must contain a column `ID` that contains the gauging station ID's in order of their location in downstream direction. The lag will then be appended as column to the data frame. For more details on the relation file, see the vignette.
`steplength`	Numeric value that specifies the length between time steps in minutes (default: `15` minutes). As time steps have to be equispaced, this is used by `hydropeak::flow()` to get a compatible format and fill missing time steps with `NA`.
`lag.max`	Maximum lag at which to calculate the ccf in `stats::ccf()` (default: `20`).
`na.action`	Function to be called to handle missing values in `stats::ccf()` (default: `na.pass`).
`tz`	Character string specifying the time zone to be used for internal conversion (default: `Etc/GMT-1`).
`format`	Character string giving the date-time format of the date-time column in the input data frame `Q`. This is passed to `hydropeak::flow()`, to get a compatible format (default: `YYYY.mm.dd HH:MM`).
`cols`	Integer vector specifying column indices in the input data frame which contain gauging station ID, date-time and flow rate to be renamed. The default indices are 1 (ID), 2 (date-time) and 3 (flow rate, Q).
`inputsep`	Character string for the field separator in input data.
`inputdec`	Character string for decimal points in input data.
`save`	A logical. If `FALSE` (default) the lag, appended to the relation file, is not written to a file, otherwise it is written to `outfile`.
`outfile`	A character string naming a file path and name where the output file should be written to.
`mc.cores`	Number of cores to use with `parallel::mclapply()`. On Windows, this is set to 1.
`overwrite`	A logical. If `FALSE` (default), it produces an error if a `LAG` column already exists in the `relation` file. Otherwise, it overwrites an existing column.

Value

Returns invisibly the data frame of the relation data with the estimated cumulative lag between neighboring gauging stations in the format `HH:MM` appended.

Examples

```
Q_file <- system.file("testdata", "Q.csv", package = "hydroroute")
relation_file <- system.file("testdata", "relation.csv",
                             package = "hydroroute")
get_lag_file(Q_file, relation_file, inputsep = ",", inputdec = ".",
             format = "%Y-%m-%d %H:%M", save = FALSE, overwrite = TRUE)

Q_file <- read.csv(Q_file)
get_lag_file(Q_file, relation_file, inputsep = ",", inputdec = ".",
             format = "%Y-%m-%d %H:%M", save = FALSE, overwrite = TRUE)
```

merge_time *Merge Events*

Description

Given two event data frames of neighboring stations S_x and S_y that consist of flow fluctuation events and computed metrics (see `hydropeak::get_events()`), the translation time indicated by the relation file as well as `timeLag` between these two stations is subtracted from S_y and events are merged where matches according to differences allowed to `timeLag` can be found.

Usage

```
merge_time(
  Sx,
  Sy,
  relation,
  timeLag = c(1, 1, 1),
  TimeFormat = "%Y-%m-%d %H:%M",
  tz = "Etc/GMT-1"
)
```

Arguments

`Sx`	Data frame that consists of flow fluctuation events and computed metrics (see `hydropeak::get_events()`) of an upstream hydrograph S_x.
`Sy`	Data frame that consists of flow fluctuation events and computed metrics (see `hydropeak::get_events()`) of a downstream hydrograph S_y.
`relation`	Data frame that contains the relation between upstream and downstream hydrograph. Must only contain two rows (one for each hydrograph) in order of their location in downstream direction. See the appended example data `relation.csv` or vignette for details on the structure. See `get_lag()` for further information about the relation and the lag between the hydrographs.
`timeLag`	Numeric vector specifying factors to alter the interval to capture events from the downstream hydrograph. By default it is `timeLag = c(1,1,1)`, this refers to matches within a time slot $\pm$ the mean translation time from `relation`. For exact time matches, `timeLag = c(0,1,0)` must be specified.
`TimeFormat`	Character string giving the date-time format of the date-time column in the input data frame (default: "%Y-%m-%d %H:%M").
`tz`	Character string specifying the time zone to be used for the conversion (default: "Etc/GMT-1").

Value

Data frame that has a matched event at S_x and S_y in each row.

Examples

```
Sx <- system.file("testdata", "Events", "100000_2_2014-01-01_2014-02-28.csv",
                  package = "hydroroute")
Sy <- system.file("testdata", "Events", "200000_2_2014-01-01_2014-02-28.csv",
                  package = "hydroroute")
relation <- system.file("testdata", "relation.csv", package = "hydroroute")
# read data
Sx <- utils::read.csv(Sx)
Sy <- utils::read.csv(Sy)
relation <- utils::read.csv(relation)
relation <- relation[1:2, ]

# exact matches
merged <- merge_time(Sx, Sy, relation, timeLag = c(0, 1, 0))
head(merged)

# matches within +/- mean translation time
merged <- merge_time(Sx, Sy, relation)
head(merged)
```

`peaktrace` *Trace Longitudinal Hydropeaking Waves Along a River Section*

Description

Estimates all settings based on the 'relation' file of a river section. The function uses a single 'relation' file and determines the settings for all neighboring stations with `estimate_AE()` for all event types specified in `event_type`. It fits models to describe translation and retention processes between neighboring hydrographs, and generates plots (see vignette for details). Given a file with initial values (see vignette), predictions are made and visualized in a plot. Optionally, the results can be written to a directory. All files need to have the same separator (`inputsep`) and character for decimal points (`inputdec`).

Usage

```
peaktrace(
  relation_path,
  events_path,
  initial_values_path,
  unique = c("time", "metric"),
  inputdec = ".",
  inputsep = ",",
  event_type = c(2, 4),
  saveResults = FALSE,
  outdir = tempdir(),
  TimeFormat = "%Y-%m-%d %H:%M",
  tz = "Etc/GMT-1",
  formula = y ~ x,
```

```
  model = stats::lm,
  FKM_MAX = 65,
  impute_method = base::max,
  ...
)
```

Arguments

relation_path	Character string containing the path of the file where the relation file is to be read from with utils::read.csv(). The file must contain a column ID that contains the gauging station ID. ID's in the file have to be in order of their location in downstream direction.
events_path	Character string containing the path of the directory where the event files corresponding to the 'relation' file are located. Only relevant files in this directory will be used, i.e., files that are related to the 'relation' file.
initial_values_path	Character string containing the path of the file which contains initial values for predictions (see vignette).
unique	Character string specifying if the potential AEs which meet the timeLag and metricLag condition should be filtered to contain only unique events using "time", i.e., by selecting those where the time difference is smallest compared to the specified factor of the mean translation time, or using "metric", i.e., by selecting those where the relative difference in amplitude is smallest (default: "time").
inputdec	Character string for decimal points in input data.
inputsep	Field separator character string for input data.
event_type	Vector specifying the event type that is used to identify event files by their file names (see hydropeak::get_events()). Default: c(2,4), i.e., increasing and decreasing events.
saveResults	A logical. If FALSE (default), the generated plots and the estimated settings are not saved. Otherwise the settings are written to a csv file and the plots are saved as png and pdf files.
outdir	Character string naming a directory where the estimated settings should be saved to.
TimeFormat	Character string giving the date-time format of the date-time column in the input data frame (default: "%Y-%m-%d %H:%M").
tz	Character string specifying the time zone to be used for the conversion (default: "Etc/GMT-1").
formula	An object of class stats::formula() to fit models.
model	Function which specifies the method used for fitting models (default: stats::lm()). The model class must have a stats::predict() function.
FKM_MAX	Numeric value that specifies the maximum fkm (see 'relation' file) for which predictions seem valid.
impute_method	Function which specifies the method used for imputing missing values in initial values based on potential AEs (default: base::max()).'
...	Additional arguments to be passed to the function specified in argument model.

Value

A nested list containing an element for each event type in order as defined in `event_type`. Each element contains again six elements, namely a data frame of estimated settings, a 'gtable' object that specifies the combined plot of all stations (plot it with `grid::grid.draw()`), a data frame containing "real" AEs (i.e., events where the relative difference in amplitude is within the estimated cut points), a grid of scatterplots ('gtable' object) for neighboring hydrographs with a regression line for each metric, a data frame of results of the model fitting where each row contains the corresponding stations and metric, the model type (default: "lm"), formula, coefficients, number of observations and R^2, and a plot of predicted values based on the "initial values".

`routing` *Estimate Models and Make Predictions*

Description

Performs the "routing" procedure, i.e., based on associated events, it uses (linear) models to describe translation and retention processes between neighboring hydrographs.

Usage

```
routing(
  real_AE,
  initials,
  relation,
  formula = y ~ x,
  model = stats::lm,
  FKM_MAX = 65,
  ...
)
```

Arguments

`real_AE`	Data frame that contains real AEs of two neighboring hydrographs estimated with `estimate_AE()`.
`initials`	Data frame that contains initial values for predictions (see vignette).
`relation`	Data frame that contains the relation between upstream and downstream hydrograph. Must only contain two rows (one for each hydrograph) in order of their location in downstream direction. See the appended example data `relation.csv` or vignette for details on the structure. See `get_lag()` for further information about the relation and the lag between the hydrographs.
`formula`	An object of class `stats::formula()` to fit models.
`model`	Function which specifies the method used for fitting models (default: `stats::lm()`). The model class must have a `stats::predict()` function.
`FKM_MAX`	Numeric value that specifies the maximum fkm (see relation file) for which predictions seem valid.
`...`	Additional arguments to be passed to the function specified in argument `model`.

Value

A nested list containing a grid of scatterplots (‘gtable‘ object) for neighboring hydrographs with a regression line for each metric, a data frame of results of the model fitting where each row contains the corresponding stations and metric, the model type (default: "lm"), formula, coefficients, number of observations and R^2, and a plot of predicted values based on the “initial values”.

Index