Package ‘rivr’

July 23, 2025

Type Package

Title Steady and Unsteady Open-Channel Flow Computation

Version 1.2-3

Description A tool for undergraduate and graduate courses in open-channel hydraulics. Provides functions for computing normal and critical depths, steady-state water surface profiles (e.g. backwater curves) and unsteady flow computations (e.g. flood wave routing) as described in Koohafkan MC, Younis BA (2015). ``Open-channel computation with R.'' The R Journal, 7(2), 249–262. <doi:10.32614/RJ-2015-034>.

URL https://github.com/mkoohafkan/rivr

BugReports https://github.com/mkoohafkan/rivr/issues

License GPL (>= 3)

Depends R (>= 3.4)

Imports utils, graphics, Rcpp (>= 1.0)

Suggests dplyr, ggplot2, knitr, rmarkdown, shiny

LinkingTo Rcpp

LazyData true

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation yes

Author Michael C Koohafkan [aut, cre]

Maintainer Michael C Koohafkan <michael.koohafkan@gmail.com>

Repository CRAN

Date/Publication 2021-01-21 10:20:02 UTC

Contents

`rivr-package` *Steady and Unsteady Open-Channel Flow Computation*

Description

This package is designed as an educational tool for students and instructors of undergraduate courses in open channel hydraulics. Functions are provided for computing normal and critical depths, steady (e.g. backwater curves) and unsteady (flood wave routing) flow computations for prismatic trapezoidal channels. See the vignettes to get started.

`channel_geom` *Channel geometry*

Description

Compute geometry relations for trapezoidal channels.

Usage

```
channel_geom(y, B, SS)
```

Arguments

`y`	Flow depth $[L]$.
`B`	Channel bottom width $[L]$.
`SS`	Channel sideslope $[LL^{-1}]$. For a rectangular channel, `SS = 0`.

Details

Channel geometry relations are routinely calculated in numerical solutions of steady, gradually-varied and unsteady flows. This function is used extensively by internal procedures and is made accessible to the user for convenience.

Value

Named vector:

A	Flow area $[L^2]$.
P	Wetted perimeter $[L]$.
R	Hydraulic radius $[L]$.
dAdy	Water surface width $[L]$.
dPdy	First derivative of wetted perimeter w.r.t. flow depth.
dRdy	First derivative of hydraulic radius w.r.t. flow depth.
DH	Hydraulic depth $[L]$.
ybar	Vertical distance from water surface to centroid of flow area $[L]$.

Examples

```
channel_geom(1.71, 100, 0) # rectangular channel
channel_geom(5.79, 6.1, 1.5) # trapezoidal channel with sideslope 3H:2V
```

---

compute_profile   *Gradually-varied flow profiles*

Description

Compute the gradually-varied flow profile of a prismatic channel.

Usage

```
compute_profile(
  So,
  n,
  Q,
  y0,
  Cm,
  g,
  B,
  SS,
  z0 = 0,
  x0 = 0,
  stepdist,
  totaldist
)
```

Arguments

So	Channel slope $[LL^{-1}]$.
n	Manning's roughness coefficient.
Q	Flow rate $[L^3T^{-1}]$.
y0	The water depth at the control section $[L]$.
Cm	Unit conversion coefficient for Manning's equation. For SI units, Cm = 1.
g	Gravitational acceleration $[LT^{-2}]$.
B	Channel bottom width $[L]$.
SS	Channel sideslope $[LL^{-1}]$.
z0	Elevation reference datum at control section $[L]$. Default is 0.
x0	Distance reference at control section $[L]$. Default is 0.
stepdist	The spatial interval used in the Standard step method $[L]$.
totaldist	The total distance upstream (or downstream) to compute the profile $[L]$.

Details

Computes the longitudinal water surface profile of a prismatic channel using the standard step method by solving the non-linear ODE

$$\frac{dy}{dx} = \frac{S_0 - S_f}{1 - Fr^2}$$

The standard-step method operates by stepping along the channel by a constant distance interval, starting from a cross-section where the flow depth is known (the control section). The flow depth is computed at the adjacent cross-section (target section). The computed value at the target is then used as the basis for computing flow depth at the next cross-section, i.e. the previous target section becomes the new control section for each step. A Newton-Raphson scheme is used each step to compute the flow depth and friction slope. Technically, the average friction slope of the control and target section is used to compute the flow depth at the target section.

Value

data.frame with columns:

x	Along-channel distance.
z	Elevation.
y	Flow depth.
v	Flow velocity.
A	Flow area.
Sf	Friction slope.
E	Total energy.
Fr	Froude Number.

Examples

```
# example M1 profile
compute_profile(0.001, 0.045, 250, 2.7, 1.486, 32.2, 100, 0, stepdist = 10, totaldist = 3000)
# example M2 profile
compute_profile(0.001, 0.045, 250, 0.64, 1.486, 32.2, 100, 0, stepdist = 10, totaldist = 3000)
# example S2 profile
compute_profile(0.005, 0.01, 250, 2.65, 1.486, 32.2, 10, 0, stepdist = 10, totaldist = 2000)
# example S3 profile
compute_profile(0.005, 0.01, 250, 0.5, 1.486, 32.2, 10, 0, stepdist = 10, totaldist = 2000)
```

---

conveyance — *Channel conveyance*

Description

Calculate the channel conveyance.

Usage

```
conveyance(n, A, R, Cm)
```

Arguments

n	Manning's roughness coefficient (dimensionless).
A	Flow area $[L^2]$.
R	Hydraulic radius $[L]$.
Cm	Unit conversion coefficient for Manning's equation. For SI units, Cm = 1.

Details

Channel conveyance is routinely calculated in numerical solutions of steady, gradually-varied and unsteady flows. This function is used extensively by internal procedures and is made accessible to the user for convenience.

Value

The channel conveyance.

critical_depth *Critical depth*

Description

Calculate the critical depth.

Usage

```
critical_depth(Q, yopt, g, B, SS)
```

Arguments

Q	Flow rate $[L^3T^{-1}]$.
yopt	Initial guess for normal depth $[L]$.
g	Gravitational acceleration $[LT^{-2}]$.
B	Channel bottom width $[L]$.
SS	Channel sideslope $[LL^{-1}]$.

Details

The critical depth is the water depth at which a channel flow regime will transition from supercritical to subcritical (or vice versa). Calculation of the critical depth is based on a specific energy formulation, i.e.

$$E = y + z + \frac{Q^2}{2gB^2y^2}$$

where y is the flow depth, z is the elevation relative to some datum (assumed to be 0), and the last term represents kinetic energy. More specifically, the function operates by finding the point where the derivative of specific energy w.r.t. y is zero, i.e. $y = y_c$ when

$$\frac{dE}{dy} = 1 - \frac{Q^2}{gA^3}\frac{dA}{dy} = 0$$

.

Value

The critical depth y_c $[L]$.

Examples

```
critical_depth(250, 2, 32.2, 100, 0) # rectangular channel
critical_depth(126, 1, 9.81, 6.1, 1.5) # trapezoidal channel with sideslope 3H:2V
```

`demo_shiny` *Shiny Demonstrations*

Description

Demonstrate package functionality via Shiny apps

Usage

```
demo_shiny(ex)
```

Arguments

`ex`	Example to run.

Details

Demonstrations available: `"gvf"` Gradually-varied flow.

Examples

```
## Not run:
# get list of available demos
demo_shiny()
# run the gradually-varied flow demo
demo_shiny("gvf")

## End(Not run)
```

`froude` *Froude Number*

Description

Calculate the Froude Number.

Usage

```
froude(Q, g, A, DH)
```

Arguments

`Q`	Flow rate $[L^3T^{-1}]$.
`g`	Gravitational acceleration $[LT^{-2}]$.
`A`	Flow area $[L^2]$.
`DH`	Hydraulic depth $[L]$.

Details

The Froude number is a dimensionless measure of bulk flow characteristics that represents the relative importance of inertial forces and gravitational forces. For open channel flow, the Froude number of open channel flow is defined as

$$Fr = \frac{v}{\sqrt{gD_H}}$$

where $v = \frac{Q}{A}$ is the flow velocity, g is the gravitational acceleration and D_H is the hydraulic depth. The Froude number is related to the energy state of the flow and can be used to identify flows as either supercritical ($Fr < 1$) or subcritical ($Fr > 1$).

Value

The Froude Number (dimensionless).

Examples

```
froude(250, 32.2, 171, 1.71) # subcritical flow
froude(250, 32.2, 57.9, 0.579) # critical flow
froude(250, 32.2, 45, 0.45) # supercritical flow
```

`normal_depth` *Normal depth*

Description

Calculate the normal (equilibrium) depth using Manning's equation.

Usage

```
normal_depth(So, n, Q, yopt, Cm, B, SS)
```

Arguments

`So`	Channel slope $[LL^{-1}]$.
`n`	Manning's roughness coefficient.
`Q`	Flow rate $[L^3T^{-1}]$.
`yopt`	Initial guess for normal depth $[L]$.
`Cm`	Unit conversion coefficient for Manning's equation. For SI units, Cm = 1.
`B`	Channel bottom width $[L]$.
`SS`	Channel sideslope $[LL^{-1}]$.

Details

The normal depth is the equilibrium depth of a channel for a given flow rate, channel slope, geometry and roughness. Manning's equation is used to calculate the equilibrium depth. Manning's equation for normal flow is defined as

$$Q = \frac{C_m}{n} A R^{2/3} S_0^{1/2}$$

where Q is the channel flow, S_0 is the channel slope, A is the cross-sectional flow area, R is the hydraulic depth and C_m is a conversion factor based on the unit system used. This function uses a Newton-Raphson root-finding approach to calculate the normal depth, i.e. $y = y_n$ when

$$f(y) = \frac{A^{5/3}}{P^{2/3}} - \frac{nQ}{C_m S_0^{1/2}} = 0$$

.

Value

The normal depth y_n $[L]$.

Examples

```
normal_depth(0.001, 0.045, 250, 3, 1.486, 100, 0) # rectangular channel
normal_depth(0.0008, 0.013, 126, 5, 1, 6.1, 1.5) # trapezoidal channel with sideslope 3H:2V
```

`route_wave` *Flood wave routing*

Description

Route a flood wave down a prismatic channel.

Usage

```
route_wave(
  So,
  n,
  Cm,
  g,
  B,
  SS,
  initial.condition,
  boundary.condition,
  downstream.condition,
  timestep,
  spacestep,
  numnodes,
  monitor.nodes,
```

```
  monitor.times,
  engine = c("Dynamic", "Kinematic"),
  scheme = c("MacCormack", "Lax"),
  boundary.type = c("QQ", "Qy", "yQ", "yy")
)
```

Arguments

- `So` Channel slope $[LL^{-1}]$.
- `n` Manning's roughness coefficient.
- `Cm` Unit conversion coefficient for Manning's equation. For SI units, Cm = 1.
- `g` Gravitational acceleration $[LT^{-2}]$.
- `B` Channel bottom width $[L]$.
- `SS` Channel sideslope $[LL^{-1}]$.
- `initial.condition` The initial flow rate $[L^3T^{-1}]$, assumed constant throughout the channel.
- `boundary.condition` Vector specifying the upstream boundary condition for the full duration of the model. If `engine = "Kinematic"`, values are assumed to be flow $[L^3T^{-1}]$. If `engine = "Dynamic"`, the form of the boundary condition is determined by the argument `boundary.type`.
- `downstream.condition` Only used if `engine = "Dynamic"`. Vector specifying the upstream boundary condition for the full duration of the model. Must be the same length as `boundary.condition`.
- `timestep` Temporal resolution of the model. Also the assumed time interval $[T]$ between elements of `boundary.condition` and `downstream.condition`. The user is responsible for ensuring numerical stability.
- `spacestep` the spatial resolution of the model, interpreted as the distance $[L]$ between nodes in the model domain. The user is responsible for ensuring numerical stability.
- `numnodes` The number of nodes used to discretize the channel. The total channel extent is computed as `spacestep*(numnodes - 1)`.
- `monitor.nodes` the nodes to be monitored every time step. Specified as a vector of node indices, with 1 being the upstream boundary and `numnodes` being the downstream boundary.
- `monitor.times` the time steps at which to monitor every node. Specified as a vector of indices of `boundary.condition`. Defaults to five equally-spaced time steps including the first and last time steps.
- `engine` The engine to be used for routing the flood wave. May be either "Kinematic" or "Dynamic".
- `scheme` Only used if `engine = "Dynamic"`. Specifies whether to use the Lax Diffusive scheme or the MacCormack predictor-corrector scheme.
- `boundary.type` Only used if `engine = "Dynamic"`. Specifies what boundary data is supplied. Possible characters are If `boundary.type = "QQ"`, both `boundary.condition` and `downstream.condition` are assumed to be flows $[L^3T^{-1}]$. If `boundary.type = "Qy"` the upstream boundary is assumed to be flow while the downstream boundary is assumed to be depth $[L]$. Other possibilities are `"yQ"` and `"yy"`.

Details

Provides implementations of a Kinematic Wave Model (KWM) and a Dynamic Wave Model (DWM) with the choice of two numerical schemes. The MacCormack scheme is a second-order accurate predictor-corrector scheme that provides efficient flood wave routing. The Lax diffusive scheme can be used to obtain smooth solutions for problems with discontinuities in the boundary conditions, e.g. sudden gate closures. The DWM implementation uses the Method of Characteristics (MOC) to compute the flow regime at the model boundaries, and allows the user to specify boundaries in terms of depths and/or flows. the KWM implementation assumes the normal depth at the upstream boundary and is only first-order accurate.

Value

data.frame with columns:

`step`	Time step.
`node`	Node index.
`time`	Time since start.
`distance`	Downstream distance.
`flow`	Flow rate.
`depth`	Flow depth.
`velocity`	Flow velocity.
`area`	Flow area.
`monitor.type`	Row refers to a monitored node ("node") or timestep ("timestep").

Examples

```
## Not run:
# kinematic wave routing
times = seq(0, 30000, by = 25)
floodwave = ifelse(times >= 9000, 250,
  250 + (750/pi)*(1 - cos(pi*times/(60*75))))
route_wave(0.001, 0.045, 1.486, 32.2, 100, 0, initial.condition = 250,
  boundary.condition = floodwave, timestep = 25, spacestep = 50,
  numnodes=301, monitor.nodes = c(1, 101, 201, 301),
  monitor.times = seq(1, length(times), by = 10), engine = "Kinematic")
# dynamic wave routing with zero-gradient downstream condition using MacCormack scheme
route_wave(0.001, 0.045, 1.486, 32.2, 100, 0, initial.condition = 250,
  boundary.condition = floodwave, downstream.condition = rep(-1, length(times)),
  timestep = 25, spacestep = 500, numnodes = 31, engine = "Dynamic",
  scheme = "MacCormack", monitor.nodes = c(1, 11, 21, 31),
  monitor.times = seq(1, length(times), by = 10))
# mixed boundary conditions (sudden gate closure) using Lax scheme
lax = route_wave(0.00008, 0.013, 1, 9.81, 6.1, 1.5,
  initial.condition = 126, boundary.condition = rep(5.79, 2001),
  downstream.condition = rep(0, 2001), timestep = 1, spacestep = 10,
  numnodes = 501, monitor.nodes = c(1, 151, 251, 301, 501),
  monitor.times = c(1, 501, 1001, 1501, 2001),
  engine="Dynamic", scheme="Lax", boundary.type="yQ")
```

```
# extract data for a monitored point
require(dplyr)
filter(lax, monitor.type == "node", node == 151)

## End(Not run)
```

`waterolympics` *California Water Olympics*

Description

Digitized results from the California Water Olympics. The variables are as follows:

- t The time (in seconds) since the start of the model run.
- Q The flow rate $[ft^3 s^{-1}]$.
- x The distance downstream $[ft]$ at which the hydrograph was recorded.

The data can be used to validate numerical solutions to flood wave routing for a channel under the following conditions:

- Channel width is 100 feet.
- Channel slope is 0.001.
- Channel extent is 150,000 feet.
- Channel roughness (Manning's n) is 0.045.
- Channel sideslope is 0 (rectangular channel).
- Initial flow rate is 250 cfs.
- Upstream boundary condition is defined as

$$Q(t < 9000) = 250 + \frac{750}{\pi}(1 - \cos\frac{\pi t}{4500})$$

$$Q(t >= 9000) = 250$$

Usage

```
data(waterolympics)
```

Format

A data frame with 40 rows and 3 variables

References

Sobey, Rodney. "H11: Hydrograph Routing." Review of One-Dimensional Hydrodynamic and Transport Models. Bay-Delta Modeling Forum, 15 June 2001. Web. 13 Mar. 2015. <http://www.cwemf.org/1-DReview/>.

Index