

Package ‘viadam’

July 13, 2010

Encoding latin1

Title VI-Avalanche-Dams, height and geometry for avalanche dams

Version 1.0

Author Tomas Johannesson, Icelandic Met. Office

Description R-functions for computation of design height and geometry for avalanche dams, run-up height of avalanches on dams and natural obstructions and various other quantities related to the flow of snow avalanches against obstructions. The functions are bundled together in this package to provide a tested software package for evaluating expressions that are described in the report “The design of avalanche protection dams. Recent practical and theoretical developments” edited by Tomas Johannesson, Peter Gauer, Dieter Issler and Karstein Lied (Luxembourg, European Communities, 2008, ISBN 978-92-79-08885-8). Note that this software is only intended to assist an avalanche expert and does not replace the professional judgement of the user. No guaranty is provided that the software is free from errors or ambiguities. The user must verify computational results independently. All design decisions made with the assistance of this software are the responsibility of the user.

Maintainer Tomas Johannesson <tj@vedur.is>

License GPL2

R topics documented:

damangle	2
mound.jump	3
mxphi	4
obliqueshock	5
obliqueshock	6
phi2phi	7
phicontour	8
runuph	9
sdamslope	10
vdamheight	11

Index	13
--------------	-----------

damangle

Angle of a dam side with respect to the surrounding terrain

Description

Computes the angle of a dam side with respect to the sloping, upstream surrounding terrain from deflecting angle, slope of the terrain, and the steepest slope of the dam side.

Usage

```
damangle(phi, psi, alfas)
```

Arguments

phi	deflecting angle with respect to the direction of steepest decent in a sloping coordinate system aligned with the terrain.
psi	the slope of the terrain.
alfas	the steepest slope of the dam side.

Value

damangle returns a vector of angles subtended by the dam side with respect to the terrain with the same number of elements as phi, psi or alfas.

Note

The angle of the dam side with respect to the sloping terrain is measured in the direction normal to the dam axis in a plane normal to the upstream terrain.

The angles phi, psi and alfas are given in radians.

Author(s)

Tomas Johannesson

See Also

[vdamheight](#), [sdamslope](#), [phi2phi](#) and [phicontour](#).

Examples

```
## Not run:  
damangle(25*pi/180, 10*pi/180, 45*pi/180)  
  
## End(Not run)
```

mound.jump

Geometry of a jet formed in the impact with a mound or a dam.

Description

Compute the ballistic trajectory of a jet formed by the flow of an avalanche that hits a braking mound or a dam.

Usage

```
mound.jump(u0, beta, h, psi, k, fph=0.004, dt=0.1)
```

Arguments

u0	upstream velocity in metres per second.
beta	throw angle in the sloping coordinate system aligned with the terrain.
h	height of the mound above the snow cover in the direction normal to the terrain.
psi	slope of the terrain.
k	dimensionless number representing the effect of energy dissipation in the impact of the avalanche with the mound, recommended value is $k = 0.8$, but $k = 0.7$, 0.8 and 0.9 should all be tested. The value of k is the throw velocity of the jet at the top of the dam relative to the velocity that corresponds to no loss of mechanical energy.
fph	a coefficient with dimensions of m^{-1} that represents the effect of air resistance on the jet, default value is 0.004 m^{-1} .
dt	time step in the Runge-Kutta integration, default value is 0.1 s .

Value

The function returns a list of vectors including the elements `res`, which contains the result of the integration: `u1` (throw speed from the top of the mound), `x2` (slope distance to the landing point of the jet), `x2cpsi` (map distance to the landing point of the jet), `u2` (speed of the jet at the landing point) and `betaLnd` (angle of the jet at the landing point with respect of the terrain); `par`, which contains the arguments of the call to the function; and `x`, `y` and `u`, which are arrays of coordinates and speed along the computed trajectory.

Note

The angles `beta` and `psi` are given in radians.

The computation of the ballistic jet is carried out in a sloping coordinate system that is aligned with the terrain in the neighbourhood of the mound or dam. This must be taken into consideration if one wishes to draw the geometry of the jet using the returned arrays `x` and `y` (`x` represents distance from the mound top along the sloping terrain and `y` is height above the terrain/snow cover in the direction normal to the terrain).

The height of the mound is defined as the height above the snow cover. The snow cover is for simplicity assumed to be uniform in thickness all along the terrain down to the landing point of the jet.

Author(s)

Tomas Johannesson

See Also[viadam.runge.kutta](#).**Examples**

```
## Not run:
mound.jump(32, 55*pi/180, 10, 11*pi/180, k=0.8)$res

## End(Not run)
```

mxphi

The maximum deflecting angle of an oblique shock

Description

Computes the deflecting and shock angles corresponding to the maximum deflecting angle that separates the weak and strong oblique shocks for a particular value of the Froude number.

Usage

```
mxphi(Fr, method=c("exact", "approximate"))
```

Arguments

Fr	Froude number.
method	method for the computation. For <code>method = "exact"</code> , a numerical solution of the exact oblique shock relations is found. For <code>method = "approximate"</code> , an approximate explicit solution valid for large Froude numbers is used.

Value

mxphi returns returns dataframe with two column: the shock angle `teta`) and the deflecting angle `phi`. They have the same number of elements as `Fr`.

Note

The returned angles `teta` and `phi` are given/returned in radians.

The approximate expression for the maximum deflecting angle, used when `method = "approximate"`, is accurate to $O(Fr^{-5/2})$.

Author(s)

Tomas Johannesson

See Also[obliqueshock](#) and [obliquenshock](#).

Examples

```
## Not run:
mxphi(7,method="exact")

## End(Not run)
```

obliquenshock

Approximate characteristics of an oblique shock

Description

Computes the shock angle, shock thickness, downstream velocity and downstream momentum flux for an oblique shock as functions of deflecting angle and Froude number using an approximation based on the flow against a normal shock with the speed of the normal component of the oblique flow.

Usage

```
obliquenshock(phi, Fr)
```

Arguments

phi	deflecting angle with respect to the direction of steepest decent in a sloping coordinate system aligned with the terrain.
Fr	Froude number.

Value

obliquenshock returns a dataframe containing the variables phi (deflecting angle), delta (difference between teta and phi, *i.e.* width of the stream along the deflecting wall), teta (shock angle), h2ph1 (relative change in flow depth across the shock), u2pu1 (relative change in flow speed across the shock) and h2u2ph1u1 (relative change in momentum flux across the shock). These variables have the same number of elements as teta or Fr. The Froude number and the Froude number normal to the wall are returned as the attributes Fr and Fp of the list.

Note

The angle phi is given in radians and so are all angles that are returned by the function.

This function computes the characteristics of a weak oblique shock. The corresponding strong oblique shock with a (much) larger value of the shock angle teta is eliminated by the approximations used in the derivations of the formulas used. The function [obliquenshock](#) may be used to obtain both types of the oblique shock.

The approximations used in this function provide good accuracy for $Fr > 2.5$ and deflecting angles phi somewhat below the maximum deflecting angle which separates the weak and strong shocks for a particular value of the Froude number Fr.

Author(s)

Tomas Johannesson

See Also

[obliqueshock](#) and [mxphi](#).

Examples

```
## Not run:
obliqueshock(15*pi/180, 7)

## End(Not run)
```

obliqueshock	<i>Characteristics of an oblique shock</i>
--------------	--

Description

Computes the deflecting angle, shock thickness, downstream velocity and downstream momentum flux for an oblique shock as functions of shock angle and Froude number.

Usage

```
obliqueshock(teta, Fr)
```

Arguments

teta	shock angle.
Fr	Froude number.

Value

`obliqueshock` returns a dataframe containing the variables `teta` (shock angle), `phi` (deflecting angle), `delta` (difference between `teta` and `phi`, *i.e.* width of the stream along the deflecting wall), `h2ph1` (relative change in flow depth across the shock), `u2pu1` (relative change in flow speed across the shock) and `h2u2ph1u1` (relative change in momentum flux across the shock). These variables have the same number of elements as `teta` or `Fr`. The Froude number is returned as the attribute `Fr` of the list.

Note

The angle `teta` is given in radians and so are all angles that are returned by the function.

This function returns the theoretically exact oblique shock solution, which represents both the weak and strong shocks, so that the shock angle `teta` is a double valued function of the deflecting angle `phi`.

Author(s)

Tomas Johannesson

See Also

[obliqueshock](#) and [mxphi](#).

Examples

```
## Not run:  
obliqueshock(22*pi/180,7)  
  
## End(Not run)
```

`phi2phi`*Deflecting angles in different coordinate systems*

Description

Convert from a deflecting angle in a sloping coordinate system aligned with the terrain to deflecting angle in a horizontal projection.

Usage

```
phi2phi(phi, phih, psi)
```

Arguments

<code>phi</code>	deflecting angle with respect to the direction of steepest decent in a sloping coordinate system aligned with the terrain.
<code>phih</code>	deflecting angle with respect to the direction of steepest decent in the map plane, that is the angle between the horizontal projections of the avalanche direction and the dam axis.
<code>psi</code>	the slope of the terrain.

Value

`phi2phi` returns a vector of angles with the same number of elements as `phi`, `phih` or `psi`.

Note

The arguments must be named so that the routine can differentiate between `phi` and `phih`.

The angles `phi`, `phih` and `psi` are given in radians.

In general, $\phi \leq \phi_{ih}$ so that the deflecting angle in the map plane may be used as a conservative estimate of the deflecting angle for dam design.

Author(s)

Tomas Johannesson

See Also

[vdamheight](#), [sdamslope](#), [damangle](#) and [phicontour](#).

Examples

```
## Not run:  
phi2phi(phi=25*pi/180,psi=10*pi/180)  
  
## End(Not run)
```

phicontour	<i>"Deflecting" angle of contour lines on a dam side</i>
------------	--

Description

Computes the “deflecting angle” of contour lines on a dam side from deflecting angle, slope of the terrain, and the angle of the dam side with respect to the terrain.

Usage

```
phicontour(phi, psi, alfa)
```

Arguments

phi	deflecting angle with respect to the direction of steepest decent in a sloping coordinate system aligned with the terrain.
psi	the slope of the terrain.
alfa	the angle of the dam side with respect to the sloping terrain in the direction normal to the dam axis in a plane normal to the upstream terrain.

Value

phicontour returns a vector of angles with the same number of elements as phi, psi or alfa.

Note

The returned “deflecting angle” of the contour lines on the dam side is defined in the map plane.

This function is useful in an analysis of the flow of avalanches against natural obstructions where a “dam axis” is in many cases not easily defined. The returned “deflecting angle” of the contour lines may be compared with angles measured from maps and used to compute the deflecting angles phi and phih (see [phi2phi](#)) corresponding to a dam that is dynamically equivalent to the terrain obstruction.

The angles phi, psi and alfa are given in radians.

Author(s)

Tomas Johannesson

See Also

[vdamheight](#), [sdamslope](#), [damangle](#) and [phi2phi](#).

Examples

```
## Not run:
phicontour(25*pi/180, 10*pi/180, 40*pi/180)

## End(Not run)
```


runuph

*Run-up height of an avalanche on a dam or an obstacle.***Description**

Run-up height of an avalanche on a dam or an obstacle corresponding to both supercritical overflow and downstream thickness of a shock.

Usage

```
runuph(phi_h, u1, h1, psi, psip, alfa, k, kp, xi, hs)
```

Arguments

phi_h	deflecting angle with respect to the direction of steepest decent in the map plane, that is the angle between the horizontal projections of the avalanche direction and the dam axis.
u1	upstream velocity in metres per second.
h1	upstream flow depth in metres.
psi	the slope of the terrain.
psip	the slope of the terrain in the direction normal to the dam.
alfa	the angle of the dam side with respect to the sloping terrain in the direction normal to the dam axis in a plane normal to the upstream terrain.
k	a dimensionless coefficient representing momentum loss normal to the dam axis in the impact, computed from <code>alfa</code> if not specified.
kp	curvature of the dam axis for a deflecting dam, equal to one over the radius of curvature, set to zero if not specified.
xi	distance along the dam from its upstream end or the upstream end of the impact with the avalanche, set to zero if not specified.
hs	snow depth and thickness of previous avalanche deposits on the terrain, set to zero if not specified.

Value

The function returns a dataframe with variables/columns describing supercritical overflow and shock that may be formed by the upstream dam face. The variables of the returned dataframe are:

phi	deflecting angle with respect to the direction of steepest decent in a sloping coordinate system aligned with the terrain.
Fr1	upstream Froude number.
u1n	velocity normal to the dam axis.
Fr1n	“Froude number” in the direction normal to the dam axis.
Hcr	critical dam height.
hcr	critical flow depth.
hrp	supercritical run-up, <i>i.e.</i> $H_{cr} + h_{cr}$.

teta	shock angle.
delta	shock widening, <i>i.e.</i> \ the angle subtended by the shock with respect to the dam.
h2	flow depth downstream of a shock.
u2	flow velocity downstream of a shock.
dHslp	extra run-up due to slope towards the dam for a deflecting dam.
dHcrv	extra run-up due to dam curvature for a deflecting dam.
hrmx	maximum of supercritical run-up and flow depth downstream of a shock plus extra run-up due to slope towards the dam and dam curvature for a deflecting dam and plus snow depth on the terrain (in the sloping coordinate system aligned with the terrain).
hd	vertical runoff corresponding to <code>hrmx</code> .

The attribute `par` of the returned variable contains the arguments of the call to the function, also as a dataframe.

Note

In contrast to most of the other functions in the `viadam` library, this function accepts dimensional arguments (*e.g.*\ flow depth in metres and velocity in metres per second) and returns dimensional quantities (*e.g.*\ dam height in metres). Most of the other functions accept non-dimensional quantities such as Froude numbers and angles and return non-dimensional quantities such as the downstream flow depth h_2 relative to the upstream flow depth h_1 .

The arguments `k`, `kp`, `xi` and `hs` are optional and have default values that are described above. They should be named if they are specified (see example with `hs` specified in the Examples section).

Author(s)

Tomas Johannesson

See Also

[obliquenshock](#), [mxphi](#) and [vdamheight](#).

Examples

```
## Not run:
runuph(18*pi/180, 45, 2, 11*pi/180, 3*pi/180, 35*pi/180, hs=3)

## End(Not run)
```

sdamslope

Steepest slope of a dam side

Description

Computes the steepest slope of a dam side from deflecting angle, slope of the terrain, and the angle of the dam side with respect to the terrain.

Usage

```
sdamslope(phi,psi,alfa)
```

Arguments

phi	deflecting angle with respect to the direction of steepest decent in a sloping coordinate system aligned with the terrain.
psi	the slope of the terrain.
alfa	the angle of the dam side with respect to the sloping terrain in the direction normal to the dam axis in a plane normal to the upstream terrain.

Value

sdamslope returns a vector of slopes with the same number of elements as phi, psi or alfa.

Note

The angles phi, psi and alfa are given in radians.

Author(s)

Tomas Johannesson

See Also

[vdamheight](#), [damangle](#), [phi2phi](#) and [phicontour](#).

Examples

```
## Not run:  
sdamslope(25*pi/180,10*pi/180,40*pi/180)  
  
## End(Not run)
```

vdamheight	<i>Vertical dam height</i>
------------	----------------------------

Description

Computes vertical dam height from dam height normal to the terrain, deflecting angle, slope of the terrain, and the angle of the dam side with respect to the terrain.

Usage

```
vdamheight(phi,psi,alfa,h)
```

Arguments

<code>phi</code>	deflecting angle with respect to the direction of steepest decent in a sloping coordinate system aligned with the terrain.
<code>psi</code>	the slope of the terrain.
<code>alfa</code>	the angle of the dam side with respect to the sloping terrain in the direction normal to the dam axis in a plane normal to the upstream terrain.
<code>h</code>	dam height normal to the terrain.

Value

`vdamheight` returns a vector of vertical dam heights with the same number of elements as `phi`, `psi`, `alfa` or `h`.

Note

The vertical dam height is measured in a vertical cross section normal to the dam or obstacle axis in the map plane.

The angles `phi`, `psi` and `alfa` are given in radians.

Author(s)

Tomas Johannesson

See Also

[sdamslope](#), [damangle](#), [phi2phi](#), [phicontour](#) and [vdamheightngi](#).

Examples

```
## Not run:  
vdamheight(25*pi/180, 10*pi/180, 40*pi/180, 15)  
  
## End(Not run)
```

Index

*Topic **math**

- damangle, 1
- mound.jump, 2
- mxphi, 4
- obliquenshock, 5
- obliqueshock, 6
- phi2phi, 7
- phicontour, 8
- runuph, 9
- sdamslope, 10
- vdamheight, 11

damangle, 1, 7, 8, 11, 12

mound.jump, 2
mxphi, 4, 5, 6, 10

obliquenshock, 4, 5, 6, 10
obliqueshock, 4, 5, 6

phi2phi, 2, 7, 8, 11, 12
phicontour, 2, 7, 8, 11, 12

runuph, 9

sdamslope, 2, 7, 8, 10, 12

vdamheight, 2, 7, 8, 10, 11, 11
vdamheightngi, 12
viadam.runge.kutta, 3