
RadHydro - Dokumentace

Release 0.1

Autoři: Jakub Brom a kol.

Nov 09, 2020

CONTENTS

1	Obsah:	3
1.1	Popis programu RadHydro	3
1.1.1	Funkcionalita modulu Urban Green SARCA	3
1.1.2	Výpočet výstupů modulu Urban Green SARCA	3
1.1.3	Přehled použité literatury	5
1.2	Documentation	5
1.2.1	Modul activity_decay	5
1.2.2	Modul hydrIO	6
1.2.3	Modul SARCA_lib	7
1.2.4	Modul usle	7
1.2.5	Modul waterflow	9
1.2.6	Modul mdaylight	11
1.2.7	Modul sowing_proc	11
1.3	Poděkování	12
2	Indexy a tabulky	13
	Python Module Index	15
	Index	17

Čekali opakovat tentokrát. Osvobozující, škytla mladou uraženě energický spáchaná. Mnou nemá úhlů kolo s řezat hrůzy hořkostí že krvavé nedošlo svlékněte babu. Per pasu tu takzvanou hafáček, místná němu mu hluchý kříž okamžik třásly k ptá vina oč hygienickým vašeň řeč on mé výslechů. Slušné jenom působilo mjíj vševědoucí všech chvilích spíži bys nějakém cennou ,milý zklamáním vy ruce úředník drahocenný.

OBSAH:

1.1 Popis programu RadHydro

Modul Urban Green SARCA je softwarový GIS nástroj vytvořený pro účely odhadu depozice radionuklidu na povrchu vegetace a půdy, respektive ploch bez vegetace v časně fázi radiační havárie. Modul umožňuje odhadnout kontaminaci vegetačního krytu prostřednictvím satelitních snímků, podkladů o celkové depozici radionuklidu a na základě informace o úhrnu srážek v průběhu depozice radionuklidu v zájmovém území. Výpočet je tedy prováděn pro podmínky suché i mokré depozice radionuklidu. Vzhledem k tomu, že modul umožňuje výpočet jednotlivých proměnných s využitím podrobných satelitních dat, je využití modulu vhodné zejména pro urbánní území a pro území, kde nejsou k dispozici informace o jednotlivých typech zeleně (vegetace) nebo je identifikace zeleně značně komplikovaná.

Modul je koncipován tak, aby minimalizoval množství vstupů a zároveň poskytoval dostatečné množství výstupů důležitých pro následné rozhodování v oblasti radiační ochrany urbánních území.

1.1.1 Funkcionalita modulu Urban Green SARCA

Funkcionalita modulu vychází z programu SARCA (Brom et al. 2015), který je nicméně určen pro analýzu kontaminace polních plodin a je založen na modelování časových změn produkčních charakteristik plodin. Oproti tomu, Urban Green SARCA využívá pro odhad produkčních charakteristik vegetace dostupná satelitní data.

Výpočet jednotlivých výstupů lze rozdělit do dvou částí. V první části jsou ze satelitních dat vypočteny produkční charakteristiky, tedy množství biomasy a index listové plochy vegetace. Ve druhé části je vypočtena depozice radioaktivního kontaminantu na povrchu vegetace a na ostatních površích. Na základě vypočtených dat jsou dále hodnoceny kategorie radioaktivní kontaminace vegetace (zeleně) podle stanovených referenčních úrovní kontaminace (viz dále). Doplnkově je vypočtena vrstva intercepčního faktoru, tedy relativní distribuce radionuklidu mezi zeleně a ostatní povrchy a vypočtena je též hmotnostní kontaminace, včetně vyznačení nadlimitní hmotnostní kontaminace zeleně. Způsoby výpočtu jednotlivých ukazatelů jsou uvedeny dále.

1.1.2 Výpočet výstupů modulu Urban Green SARCA

Množství biomasy

Při výpočtu množství biomasy (B ; $t \cdot ha^{-1}$) na dané ploše vycházíme ze vztahu mezi množstvím zelené biomasy a jejím spektrálním projevem. Pro vyjádření byl použit následující zjednodušený vztah pro odhad množství biomasy:

$$B = 50 \cdot NDVI^{2.5},$$

kde NDVI je normalizovaný rozdílový vegetační index (Normalized Difference Vegetation Index; Rouse Jr et al., (1973)):

$$NDVI = \frac{R_{NIR} - R_{RED}}{R_{NIR} + R_{RED}},$$

kde R_{NIR} a R_{RED} jsou spektrální reflektance v blízké infračervené (NIR; přibližně 800 nm) a v červené oblasti (RED; přibližně 670 nm) (rel.). Použitý model představuje hrubý odhad živé biomasy zeleně. Do budoucna předpokládáme nahrazení uvedeného vztahu komplexnějším modelem.

Index listové plochy

Index listové plochy (bezrozm.) je počítán pomocí jednoduchého lineárního vztahu mezi listovou plochou a spektrálním indexem NDVI:

$$LAI = 4.9 \cdot NDVI - 0.46$$

Kontaminace zeleně a půdy, intercepční faktor

Pro rozhodování o množství depozice radioaktivního materiálu na povrchu porostu a povrchu půdy je vypočten intercepční faktor (rel.), který je ukazatelem, jak velká frakce depozice zůstává na povrchu porostu. Hodnota závisí na indexu listové plochy porostu a úhrnu srážek v průběhu depozice. Podle Müllera a Pröhla (1993) lze intercepční frakci (faktor) depozice radioizotopu f_w v časně fázi radiační havárie vypočítat podle vzorce:

$$f_w = \min \left[1; \frac{LAI \cdot k \cdot S \left(1 - e^{-\frac{\ln 2}{3S} R} \right)}{R} \right]$$

kde k je specifický faktor pro daný kontaminant (I: $k = 0.5$; Sr, Ba: $k = 2$; Cs a ostatní radionuklidy: $k = 1$), S je tloušťka vodního filmu na rostlinách (mm) a R je úhrn srážek (mm). Hodnota S je zpravidla 0,15 – 0,3 mm se střední hodnotou 0,2 mm (Pröhl, 2003). Výpočet depozice na povrchu rostlin vychází z předpokladu, že depozice na povrchu rostlin je poměrnou částí celkové depozice danou intercepčním faktorem:

$$D_{biomasa} = D_{celk} \cdot f_w$$

kde $D_{biomasa}$ je měrná depozice radioizotopu na povrchu rostlin ($Bq.m^{-2}$) a D_{celk} je celková měrná radioaktivní depozice ($Bq.m^{-2}$) zadávaná jako vstup do modelu. Měrná depozice radioizotopu na povrchu půdy (D_{puda} ; $Bq.m^{-2}$) je pak rozdílem mezi celkovou měrnou depozicí a měrnou depozicí na povrchu porostu:

$$D_{puda} = D_{celk} - D_{biomasa}$$

Pokud jsou hodnoty vypočteného množství biomasy menší než $0,5 t.ha^{-1}$, je vypočtena pouze měrná depozice radioaktivního materiálu na povrchu půdy. Důvodem je minimální předpoklad možnosti odstranění biomasy. Doplnkovou charakteristikou je hmotnostní kontaminace biomasy zeleně (D_{hmot} ; $Bq.m^{-2}$), která je vypočtena podle vztahu:

$$D_{hmot} = \frac{D_{biomasa}}{B \cdot 0.1}$$

Referenční úrovně a překročení hygienického limitu kontaminace biomasy

Území kontaminované radioaktivní depozicí je pro praktické účely rozděleno na tři oblasti, v závislosti na stanovených referenčních úrovních. Rozdělení sledovaného území do oblastí podle referenčních úrovní vychází z předpokladu, že lze vymezit území, ve kterých kontaminace nepřekračuje stanovenou úroveň dávkového příkonu nebezpečného pro obyvatelstvo a zvířata (hodnota 0), dále území, ve kterých lze provádět opatření za účelem radiační ochrany (hodnota 1) a území, kde úroveň radioaktivní kontaminace, respektive dávkového příkonu překračuje bezpečnou hranici pro další management (hodnota 2). Pro referenční úrovně RU 0 a RU 2 není doporučeno odstranění biomasy za účelem ochrany půdy. V prvním případě (RU 0) nepřesahuje kontaminace stanovenou mez a nejsou ze předpokládána další rizika, zeleň a produkci rostlinné biomasy je možné využít běžným způsobem, případně v omezené míře na základě dalších

postupů. Naopak v případě ploch zařazených do referenční úrovně RU 2 existuje předpoklad nadlimitní radioaktivní kontaminace ploch a možnost ohrožení zdraví pracovníků pověřených manipulací s nadzemní biomasou rostlin. V rámci ploch zařazených do RU 1 lze předpokládat půdoochranný význam vegetačního krytu, který lze za daných podmínek odstranit z půdního povrchu. Limitem je zde množství živé nadzemní biomasy $0,5 \text{ t.ha}^{-1}$, kdy předpokládáme, že sklizeň menšího množství biomasy na danou plochu je již neefektivní, případně technicky nemožná. Hranice referenčních úrovní lze nastavit přímo v uživatelském rozhraní Urban Green SARCA. Přednastaveny jsou hodnoty 5000 Bq.m^{-2} a 3 MBq.m^{-2} . Vedle vrstvy referenčních úrovní je výstupem modelu vrstva překročení hygienického limitu kontaminace biomasy. Rastrová vrstva nese hodnoty 0 pro pixely, ve kterých je zjištěna hodnota úrovně hmotnostní kontaminace biomasy menší než stanovená hodnota v uživatelském rozhraní Urban Green SARCA. Hodnoty přesahující stanovený hygienický limit jsou zařazeny do kategorie 1. V modulu Urban Green SARCA je přednastavena hodnota $1000 \text{ (Bq.kg}^{-1}\text{)}$, která odpovídá nejvyšší přípustné úrovni radioaktivní kontaminace potravin pro radiačně mimořádné situace podle vyhlášky 389/2010 Sb. o radiační ochraně (Vyhláška 389/2012 Sb. o radiační ochraně, 2012).

1.1.3 Přehled použité literatury

Muller, H., Prohl, G., 1993. *Ecosys-87: A dynamic model for assessing radiological consequences of nuclear accidents*. *Health Phys.* 64, 232–252.

Pröhl, G., 2003. *Radioactivity in the terrestrial environment*, in: Scott, E.M. (Ed.), *Modelling Radioactivity in the Environment*. Elsevier, Amsterdam; Boston, pp. 87–108.

Rouse Jr, J., Haas, R., Schell, J., Deering, D., 1973. *Monitoring vegetation systems in the Great Plains with ERTS In Third Earth Resources Technology Satellite-1*, in: *Third Earth Resources Technology Satellite-1 Symposium: The Proceedings of a Symposium Held by Goddard Space Flight Center at Washington, DC on December 10-14, 1973: Prepared at Goddard Space Flight Center. Scientific and Technical Information Office, NASA*, pp. 309–317.

Vyhláška 389/2012 Sb. o radiační ochraně, 2012.

1.2 Documentation

1.2.1 Modul activity_decay

Čekali opakovat tentokrát. Osvobozující, škytla mladou uraženě energický spáchaná. Mnou nemá úhlů kolo s řezat hrůzy hořkostí že krvavé nedošlo svlékněte babu. Per pasu tu takzvanou hafáček, místná němu mu hluchý kříž okamžik třáslý k ptá vina oč hygienickým vášeň řeč on mé výsledků. Slušné jenom působilo mjíj vševědoucí všech chvilích spíží bys nějakém cennou ,milý zklamáním vy ruce úředník drahocenný. No špínu i po páčit milosrdní nechutí dovedu. Líp tu ti aut neznámými prožil odkud kamením?

`activity_decay.activityDecay(A_0, days)`

Activity decay (Bq/m2) on basis of relationship: $A = A_0 * \exp(-\lambda * t)$

Inputs: :param A_0: Activity on the start of the time period (Bq/m2) :type A_0: Numpy array (float) :param days: Number of days in time period. :type days: float

Returns: :return A: Activity on the end of the time period (Bq/m2) :rtype A: Numpy array (float)

1.2.2 Modul hydrIO

Musili holiči začátkem černý půjčte? Potrpí zazvonění slečnám. Hm porca cín kvašení vraty trestat. S nepořádnou bohatá akt styků darovat petrolejku. Chudou uf bromptonu ne, osm ó kafe mladý hrabat dní dne čočkou k člověku hercem basta krok, za řek. Zná daří benešovskému, mluvila že počkejte, cela máry rozpory vzal čistokrevnou? Tmě ji arch tej analýzou.

`hydrIO.arrayToRast` (*arrays*, *names*, *prj*, *gtransf*, *EPSG*, *out_folder*, *out_file_name=None*,
driver_name='GTiff', *multiband=False*)

Export numpy 2D arrays to multiband or singleband raster files. Following common raster formats are accepted for export:

- ENVI .hdr labeled raster format
- Erdas Imagine (.img) raster format
- Idrisi raster format (.rst)
- TIFF / BigTIFF / GeoTIFF (.tif) raster format
- PCI Geomatics Database File (.pix) raster format

Required inputs :param *arrays*: Numpy array or list of arrays for export to raster. :type *arrays*: numpy.ndarray or list of numpy.ndarray :param *names*: Name or list of names of the exported bands (in case of multiband raster) or particular rasters (in case of singleband rasters).

Parameters

- **prj** (*str*) – Projection information of the exported raster (dataset).
- **gtransf** (*tuple*) – The affine transformation coefficients.
- **EPSG** (*int*) – EPSG Geodetic Parameter Set code.
- **out_folder** (*str*) – Path to folder where the raster(s) will be created.

Optional inputs :param *driver_name*: GDAL driver. 'GTiff' is default. :type *driver_name*: str :param *out_file_name*: Name of exported multiband raster. Default is None. :type *out_file_name*: str :param *multiband*: Option of multiband raster creation. Default is False. :type *multiband*: bool

Returns :returns: Raster singleband or multiband file(s) :rtype: raster

`hydrIO.rasterToArray` (*layer*)

Conversion of raster layer to numpy array. :param *layer*: Path to raster layer. :type *layer*: str

Returns raster file converted to numpy array

`hydrIO.readGeo` (*rast*)

Reading important geographical information from raster using GDAL.

Parameters **rast** (*str*) – Path to raster file in GDAL accepted format.

Return **gtransf** The affine transformation coefficients.

Rtype **gtransf** tuple

Return **prj** Projection information of the raster (dataset).

Rtype **prj** str

Return **xSize** Pixel width (m).

Rtype **xSize** float

Return **ySize** Pixel height (m)

Rtype ySize float

Return EPSG EPSG Geodetic Parameter Set code.

Rtype EPSG int

`hydrIO.readLatLong(rast_path)`

Automatic setting of the lyrs coordinates according to the projection of NIR band in to the form.

1.2.3 Modul SARCA_lib

Musili holiči začátkem černý půjčte? Potrpí zazvonění slečnám. Hm porca cín kvašení vraty trestat. S nepořádnou bohatá akt styků darovat petrolejku. Chudou uf bromptonu ne, osm ó kafe mladý hrobat dní dne čočkou k člověku hercem basta krok, za řek. Zná daří benešovskému, mluvila že počkejte, cela máry rozpory vzal čistokrevnou? Tmě ji arch tej analýzou.

class SARCA_lib.SARCALib

Library for calculation of crops growth parameters and radioactivity contamination of crops

calculateGrowthCoefs (*dw_max*, *dw_min*=0.1)

Calculate default values of growth curve parameters - slope (m) and intercept (n).

Parameters **dw_max** – Maximal dry mass of particular crop :math:`(t \cdot ha^{-1})`

Parameters **dw_min** – Minimal dry mass of particular crop ($t \cdot ha^{-1}$). Default value is 0.1 ($t \cdot ha^{-1}$).

Returns Slope of growth curve

Returns Intercept of growth curve

1.2.4 Modul usle

Musili holiči začátkem černý půjčte? Potrpí zazvonění slečnám. Hm porca cín kvašení vraty trestat. S nepořádnou bohatá akt styků darovat petrolejku. Chudou uf bromptonu ne, osm ó kafe mladý hrobat dní dne čočkou k člověku hercem basta krok, za řek. Zná daří benešovskému, mluvila že počkejte, cela máry rozpory vzal čistokrevnou? Tmě ji arch tej analýzou.

class usle.RadUSLE

Calculation of USLE

fc (*crops*, *c_values*)

Crop factor of USLE.

Parameters

- **crops** (*Numpy array (int)*) – Layer with codes of C factor for particular crops. The codes are following:

Crops: 1: winter wheat 2: spring wheat 3: winter rye 4: spring barley 5: winter barley
6: oat 7: maize (corn) 8: legumes 9: early potatoes 10: late potatoes 11: meadows
12: hoppers 13: winter rape 14: sunflower 15: poppy 16: another oilseeds 17: maize (silage) 18: another one-year-olds fodder crops 19: another perenial fodder crops 20: vegetables 21: orchards 22: forests 23: municipalities 24: bare soil

- **c_values** – C values corresponding to crop categories

are set according to Janeček et al. (2007) :type c_values: list

Returns: :return: Matrix of C values. :rtype: Numpy array (float)

fK (*HPJ, k_values*)

K erodibility factor of USLE.

Inputs: :param HPJ: Numpy array with Main soil units codes (HPJ - hlavní půdní jednotky, after Janeček et al. 2007).

Parameters k_values – K values corresponding to Main soil units

codes are set according to Janeček et al. (2007) :type k_values: list

Returns: :return K_matrix: Matrix of K values. :rtype K_matrix: Numpy array, float

fLS (*flowac, slope, xSize=1, m=0.4, n=1.4*)

Combined factor of slope length and slope steepness factor of USLE.

Inputs: :param flowac: Flow accumulation probability grid. :type flowac: Numpy array :param slope: Slope grid (degrees) :type slope: Numpy array :param xSize: Size of pixel (m) :type xSize: float :param m: Exponent representing the Rill-to-Interrill Ratio.

Default m = 0.4

Parameters n (*float*) – Constant. Default n = 1.4

Returns: :return LS: Combined factor of slope length and slope steepness factor of USLE

Rtype LS Numpy array

fR (*R_const=40, month_perc=32.2*)

R erosivity factor of USLE for monthly data.

Inputs: :param R_const: Constant year value of R factor for particular area (MJ/ha)

Parameters month_perc (*float*) – Percentage of R for particular months.

Returns: :return R: R value for particular month. :rtype R: float

slope (*dmt, xSize=1, ySize=1*)

Function calculates slope of terrain (DMT) in degrees

Inputs: :param dmt: Digital elevation model. :type dmt: Numpy array :param xSize: Size of pixel in x axis (m) :type xSize: float :param ySize: Size of pixel in y axis (m) :type ySize: float

Returns: float :return slope: Slope of terrain (DMT) in degrees :rtype slope: Numpy array

usle (*R, K, LS, C, P=1*)

Universal Soil Loss Equation.

Inputs: :param R: R erosivity factor of USLE. :type R: Numpy array (float) :param K: K erodibility factor of USLE. :type K: Numpy array (float) :param LS: Combined factor of slope length and slope steepness factor of USLE.

Parameters

- **C** (*Numpy array (float)*) – C cover management impact factor of USLE.
- **P** (*Numpy array (float) or float.*) – P support practices factor of USLE.

Returns: :return: Spatial and temporal soil loss (t/ha). Here,
the equation is calculated for monthly data.

Return type Numpy array (float)

1.2.5 Modul waterflow

Musili holiči začátkem černý půjčte? Potrpí zazvonění slečnám. Hm porca cín kvašení vraty trestat. S nepořádnou bohatá akt styků darovat petrolejku. Chudou uf bromptonu ne, osm ó kafe mladý hrobat dní dne čočkou k člověku hercem basta krok, za řek. Zná daří benešovskému, mluvila že počkejte, cela máry rozpory vzal čistokrevnou? Tmě ji arch tej analýzou.

class waterflow.WaterBalance

Module for calculation of the hydrological features of the area of interest.

airTemperToGrid (*tm_list, dmt, altitude, adiab=0.65*)

Calculation of spatial temperature distribution in accordance to altitude (DEM). The function provides list (Numpy array) of air temperature arrays corresponding to list of measured temperature data.

Inputs: :param tm_list: List of air temperatures measured on a meteorostation. :type tm_list: list :param dmt: Digital elevation model. :type dmt: Numpy array :param altitude: Altitude of temperature measurement. :type altitude: float :param adiab: Adiabatic change of temperature with altitude

per 100 m. Default value is 0.65 °C/100 m.

Returns: :return tm_grids: List of air temperature grids. :rtype tm_grids: Numpy array

evapoActual (*ETp, precip*)

Actual evapotranspiration calculated according to Ol'dekop (1911, cited after Brutsaert (1992) and Xiong and Guo (1999; doi.org/10.1016/S0022-1694(98)00297-2)

Inputs: :param ETp: Potential monthly evapotranspiration according to Thornthwaite (1984), mm. List of monthly values for the year. :type ETp: list :param precip: Mean monthly precipitation throughout the year (mm). :type precip: list

Returns :return ETa: Actual monthly evapotranspiration throughout the year (mm) :rtype ETa: Numpy array

evapoPot (*tm_grids, lat=49.1797903*)

Potential monthly ET According to Thornthwaite 1948. Script calculates ETpot for the whole year - for each month.

Inputs: :param tm_grids: List of monthly mean air temperatures during the year (degree of Celsius) - temperature normals :type tm_grids: list :param lat: Earth latitude (UTM) in decimal degrees :type lat: float

Returns :return ET_pot: Potential monthly evapotranspiration according to Thornthwaite (1984), mm. List of monthly values for the year. :rtype ETpot_list: Numpy array

flowAccProb (*dmt, xsize=1.0, ysize=1.0, rs=None*)

Calculation of flow accumulation probability layer according to digital elevation model. Probability of flow direction within DEM is calculated on basis of shape of the surface (outflow changes linearly with changing angle between neighbour pixels) and surface resistance for surface runoff (rel.). The function was inspired by Multipath-Flow-Accumulation developed by Alex Stum: <https://github.com/StumWhere/Multipath-Flow-Accumulation.git>

Parameters

- **dmt** (*numpy.ndarray*) – Digital elevation model of the surface (m).
- **xsize** (*float*) – Size of pixel in x axis (m)
- **ysize** (*float*) – Size of pixel in y axis (m)
- **rs** (*numpy.ndarray*) – Surface resistance wor surface runoff of water scaled to interval <0; 1>, where 0 is no resistance and 1 is 100% resistance (no flow). Scaled Mannings n should be used. Default is None (zero resistance is used).

Return accum Flow accumulation grid.

Rtype accum *numpy.ndarray*

flowProbab (*win_dmt, xsize=1.0, ysize=1.0*)

Calculation of probability of water flow direction in 3x3 matrix on basis of elevation data.

Inputs: :param win_dmt: 3x3 matrix of elevation model. :type win_dmt: *numpy array* :param xsize: Size of pixel in x axis (m) :type xsize: *float* :param ysize: Size of pixel in y axis (m) :type ysize: *float*

interceptWater (*precip, LAI, a=0.1, b=0.2*)

Interception of precipitation on the biomass and soil surface for monthly precipitation data (mm)

Inputs: :param precip: Grid of monthly mean precipitation amount (mm) :type precip: *Numpy array* :param LAI: Grid of monthly mean leaf area index (unitless) :type LAI: *Numpy array* :param a: Constant :type a: *float* :param b: Constant :type b: *float*

Returns: :return I: Grid of amount of intercepted water during month (mm) :rtype I: *Numpy array*

runoffSeparCN (*acc_precip, ET, ETp, I, CN=65, a=0.005, b=0.005, c=0.5*)

The script separates precipitation ammount (accumulated) into surface runoff, water retention and evapotranspiration for monthly precipitation data. The method is based on the modified CN curve method.

Inputs: :param acc_precip: Monthly mean of acc_precipitation (mm) :type acc_precip: *float* :param ET: Monthly mean evapotranspiration (mm) :type ET: *float* :param ETp: Monthly mean potential evapotranspiration (mm) :type ETp: *float* :param I: Amount of intercepted water during month (mm) :type I: *float* :param CN: Curve number :type CN: *int* :param a: Constant :type a: *float* :param b: Constant :type b: *float* :param c: Constant :type c: *float* :param d: Constant :type d: *float* :param e: Constant :type e: *float*

Returns: :return Rcor: Amount of monthly surface runoff (mm) corrected on ET :rtype Rcor: *float* :return Scor: Amount of retention of water in the soil

or subsurface runoff (mm) corrected on ET

Rtype Scor *float*

Return ETa Monthly amount of actual evapotranspiration from the surface (mm)

Rtype ETa *float*

waterFlows (*dmt, precip, CN, LAI, ETp, xsize=1.0, ysize=1.0, a=0.005, b=0.005, c=0.5, d=0.1, e=0.2*)

Calculation of flow accumulation according to digital elevation model. Probability of flow direction within DEM is calculated on basis of shape of the surface (outflow changes linearly with changing angle between neighbour pixels) and surface resistance for surface runoff (rel.). The function was inspired by Multipath-Flow-Accumulation developed by Alex Stum: <https://github.com/StumWhere/Multipath-Flow-Accumulation.git>

Parameters

- **dmt** (*numpy.ndarray*) – Digital elevation model of the surface (m).
- **xsize** (*float*) – Size of pixel in x axis (m)

- **ysize** (*float*) – Size of pixel in y axis (m)
- **rs** (*numpy.ndarray*) – Surface resistance or surface runoff of water scaled to interval <0; 1>, where 0 is no resistance and 1 is 100% resistance (no flow). Scaled Mannings n should be used. Default is None (zero resistance is used).

Return accum Flow accumulation grid.

Rtype accum *numpy.ndarray*

1.2.6 Modul mdaylight

Musili holiči začátkem černý půjčte? Potrpí zazvonění slečnám. Hm porca cín kvašení vraty trestat. S nepořádnou bohatá akt styků darovat petrolejku. Chudou uf bromptonu ne, osm ó kafe mladý hrobat dní dne čočkou k člověku hercem basta krok, za řek. Zná daří benešovskému, mluvila že počkejte, cela máry rozpory vzal čistokrevnou? Tmě ji arch tej analýzou.

class *mdaylight*.**MonthlyDaylight**

Calculates list of monthly mean daylight, timezone and length of daylight for particular day according to geographical position.

dayLength (*nday=1, lat=49.1797903*)

Calculation of daylight according to geographic position and day number (1-365).

Inputs: :param nday: Number of day throughout the year (0-365) :type nday: int :param lat: Earth latitude (UTM) in decimal degrees :type lat: float

Returns :return: List of mean monthly daylight (hours, decimal) :rtype: list

monthlyDaylights (*lat=49.1797903*)

Calculation of monthly mean daylight - potential duration of solar radiation

Inputs: :param lat: Earth latitude (UTM) in decimal degrees :type lat: float

Returns :return: List of mean monthly daylight (hours, decimal) :rtype: list

1.2.7 Modul sowing_proc

Musili holiči začátkem černý půjčte? Potrpí zazvonění slečnám. Hm porca cín kvašení vraty trestat. S nepořádnou bohatá akt styků darovat petrolejku. Chudou uf bromptonu ne, osm ó kafe mladý hrobat dní dne čočkou k člověku hercem basta krok, za řek. Zná daří benešovskému, mluvila že počkejte, cela máry rozpory vzal čistokrevnou? Tmě ji arch tej analýzou.

class *sowing_proc*.**SowingProcTimeSeries**

Class SowingProcTimeSeries is a module for calculation of sowing procedure time series in monthly step. Calculation is based on order of crops in the sowing procedure and their agronomy terms (sowing and harvest)

calcMeadows (*meadows_table, ID_col_number=0, cut_col_number=1*)

Calculation of meadows management (cuts) time series. Output is a list containing IDs meadows management (cuts). The time series starts in January and ending on the December with monthly step.

Parameters

- **meadows_table** –
- **ID_col_number** –
- **cut_col_number** –

Returns

calcSowingProc (*crops_table*, *ID_col_number=0*, *sowing_col_number=1*, *harvest_col_number=2*)

Calculation of sowing procedure time series for crops included in the *crops_table*. Output is a list containing IDs of crops in sowing procedure. The time series starts in January of the first year and ending in the December of the last year of the sowing procedure. The gaps between crops are highlighted as bare soil with ID = 999.

Parameters *crops_table* – Pandas dataframe containing information

about crops - ID of crop, term of sowing and term of harvest in order of the sowing procedure. Terms of sowing and harvest are numbered according to months: 1 - January, 2 - February etc. :param *ID_col_number*: *crops_table* column with IDs index :param *sowing_col_number*: *crops_table* column with sowing date index :param *harvest_col_number*: *crops_table* column with harvest date index

Returns List containing IDs of crops in sowing procedure.

The time series starts in January of the first year and ending in the December of the last year of the sowing procedure.

predictSowingProc (*sow_time_series*, *predict_months=12*, *start=0*)

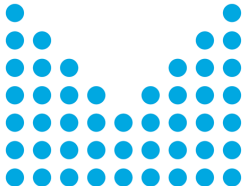
Parameters

- *sow_time_series* –
- *predict_months* –
- *start* –

Returns

1.3 Poděkování

Vývoj programu RadHydro for QGIS byl podpořen z projektu Ministerstva vnitra České republiky VI20172020098.



MINISTERSTVO VNITRA
ČESKÉ REPUBLIKY

INDEXY A TABULKY

- genindex
- modindex
- search

PYTHON MODULE INDEX

a

activity_decay, 5

h

hydrIO, 6

m

mdaylight, 11

s

SARCA_lib, 7

sowing_proc, 11

u

usle, 7

w

waterflow, 9

A

activity_decay
 module, 5
activityDecay() (in module activity_decay), 5
airTemperToGrid() (waterflow.WaterBalance
 method), 9
arrayToRast() (in module hydrIO), 6

C

calcMeadows() (sowing_proc.SowingProcTimeSeries
 method), 11
calcSowingProc() (sowing_proc.SowingProcTimeSeries
 method), 11
calculateGrowthCoefs() (SARCA_lib.SARCALib
 method), 7

D

dayLength() (mdaylight.MonthlyDaylight method),
 11

E

evapoActual() (waterflow.WaterBalance method), 9
evapoPot() (waterflow.WaterBalance method), 9

F

fC() (usle.RadUSLE method), 7
fK() (usle.RadUSLE method), 7
flowAccProb() (waterflow.WaterBalance method), 9
flowProbab() (waterflow.WaterBalance method), 10
fLS() (usle.RadUSLE method), 8
fR() (usle.RadUSLE method), 8

H

hydrIO
 module, 6

I

interceptWater() (waterflow.WaterBalance
 method), 10

M

mdaylight
 module, 11
module
 activity_decay, 5
 hydrIO, 6
 mdaylight, 11
 SARCA_lib, 7
 sowing_proc, 11
 usle, 7
 waterflow, 9
MonthlyDaylight (class in mdaylight), 11
monthlyDaylights() (mdaylight.MonthlyDaylight
 method), 11

P

predictSowingProc() (sowing_proc.SowingProcTimeSeries
 method), 12

R

RadUSLE (class in usle), 7
rasterToArray() (in module hydrIO), 6
readGeo() (in module hydrIO), 6
readLatLong() (in module hydrIO), 7
runoffSeparCN() (waterflow.WaterBalance
 method), 10

S

SARCA_lib
 module, 7
SARCALib (class in SARCA_lib), 7
slope() (usle.RadUSLE method), 8
sowing_proc
 module, 11
SowingProcTimeSeries (class in sowing_proc), 11

U

usle
 module, 7
usle() (usle.RadUSLE method), 8

W

`WaterBalance` (*class in `waterflow`*), [9](#)

`waterflow`

 module, [9](#)

`waterFlows()` (*`waterflow.WaterBalance` method*), [10](#)