User's Manual

The Plastic Pathfinder

The Plastic Pathfinder: A Macroplastic Transport and Fate Model for Terrestrial Environments

Y. Mellink[a,b], T. van Emmerik[a], M. Kooi[c], C. Laufkötter[d,e], and H. Niemann[f,b,g]

[a] Hydrology and Quantitative Water Management Group, Wageningen University, Wageningen, the Netherlands

[b] Department of Earth Sciences, Faculty of Geosciences, Utrecht University, Utrecht, the Netherlands

[c] Aquatic Ecology and Water Quality Group, Wageningen University, Wageningen, the Netherlands

[d] Climate and Environmental Physics, Physics Institute, University of Bern, Bern, Switzerland

[e] Oeschger Centre for Climate Change Research, University of Bern, Bern, Switzerland

[f] Department of Marine Microbiology and Biogeochemistry, NIOZ Royal Netherlands Institute for Sea Research, 't Horntje, the Netherlands

[g] CAGE - Centre for Arctic Gas Hydrate, Environment and Climate, Department of Geosciences, UiT The Arctic University of Norway, Tromsø, Norway

Corresponding author: Y. Mellink (mellinkyvette@gmail.com)

The Plastic Pathfinder (former version: The Trash-Tracker) is a numerical code developed by Mellink et al. (2022) to simulate the transport and accumulation of macroplastic waste through terrestrial systems. The two plastic transport agents are wind and surface runoff. Transport of plastics over land is modelled when the wind and/or surface runoff conditions are sufficient to overcome the wind and/or surface runoff thresholds, respectively. These thresholds reflect the resistance to plastic transport and are a function of land use and terrain slope.

Table of Contents

1. Python ... 3

2. NumPy arrays ... 3

3. Model parameters ... 3

4. Model grid and domain ... 4

5. Directions of motion ... 4

6. Time frame ... 4

7. Model input ... 4

7.1 Land Use ... 5

7.2 Population density & mismanaged plastic waste generation ... 5

7.3 Topography ... 5

7.4 Wind speed and direction ... 7

7.5 Surface runoff flux and direction ... 8

8. Plastic mobilisation and transport thresholds ... 9

8.1 Wind speed thresholds ... 9

8.2 Surface runoff thresholds ... 10

9. Plastic mobilisation ... 10

10. Plastic transport and accumulation ... 11

10.1 General framework ... 11

10.2 Modelling the transport of MPW ... 11

11. Potential plastic transport directions ... 13

12. References ... 14

1. Python

The Plastic Pathfinder is a Python 3.8.3 code written in the open-source web application Jupyter Notebook (Version 6.0.3) available by Anaconda Software Distribution (Anaconda Software Distribution, 2016).

2. NumPy arrays

Throughout the whole code NumPy arrays are used. The elements stored in an array can be called for by referring to its indices. In 2D arrays the index of the first and second dimensions are denoted by `i` and `j`, respectively. The data that are stored in arrays are plotted on geographical maps (e.g. topography or land use maps).

If you wish to manually fill arrays and subsequently plot that data on geographical maps, then it must be noted that the indices of a 2D array are not equivalent to geodetic coordinates and that a 2D array cannot simply be overlain over a geographical map. A data value (e.g. topography) that belongs to location ‘x’, which lies in the southwest, i.e. lower left, corner of a geographical map (grey shaded box in Fig. 1a) is stored in the upper left corner of the 2D array (grey shaded box in Fig. 1b). In addition, in a 2D NumPy array the row number (`i`) increases when going down (see Fig. 1b), while normally on a geographical map the latitude decreases when going down (see Fig. 1a). Therefore, in order to project data stored 2D NumPy arrays to a geographical map, the array elements must be flipped over the first dimensional axis (0 axis), i.e. flipped over the rows.

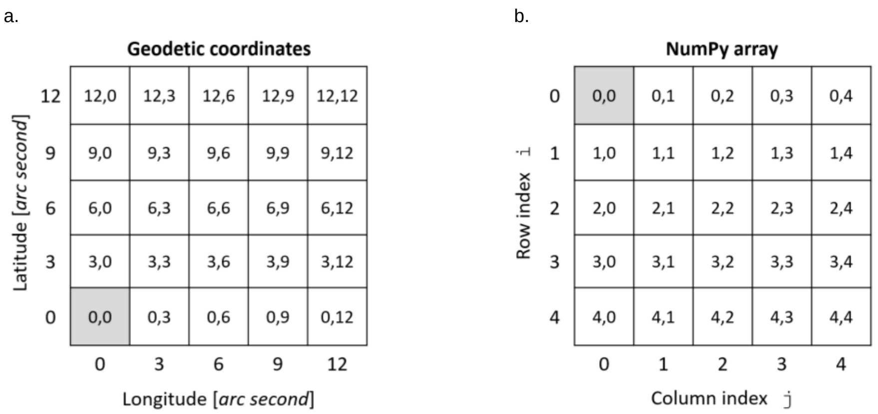

Fig. 1. (a) Map showing the geodetic coordinates of 25 grid cells with a grid resolution of 3 arc seconds. In grey location ‘x’ with 0 latitude and 0 longitude. (b) 5 by 5 2D NumPy array showing the indices `i` and `j` for the 25 elements that can be stored in this array. In grey the location at which the data value (element) that belongs to location ‘x’ is stored in the NumPy array.

3. Model parameters

All model parameters regarding plastic mobilisation and transport thresholds, runoff coefficients and mismanaged plastic waste generation are stored in the 1D ParVal array at the top of the program. If you wish to use different values for model parameters, simply change the value in the ParVal array.

4. Model grid and domain

The code of the Plastic Pathfinder uses a rectangular domain of which the user sets the boundaries by assigning a minimum and maximum latitude (Lat_min_arc & Lat_max_arc) and longitude (Lon_min_arc & Lon_max_arc). Additionally, the user sets the zonal (Lon_res_arc) and meridional (Lat_res_arc) resolutions, which are used to fit rectangular grid cells within the domain boundaries. At the centre of each grid cell lies a grid point. The model calculates the latitude and longitude of the grid points and stores them in the 1D lat_arc[i] and lon_arc[i] arrays, respectively.

5. Directions of motion

The Plastic Pathfinder models motion in eight directions: north, northeast, east, southeast, south, southwest, west, and northwest. This is due to the rectangular geometry of the model grid whereby a(n) (interior) grid cell is surrounded by a maximum of eight other grid cells (Fig. 2). This approach is based on the encoding of water flow directions described by Jenson and Domingue (1988). The eight directions are each encoded with a label, i.e. a number. These numbers are different for each vector variable (see sections below).

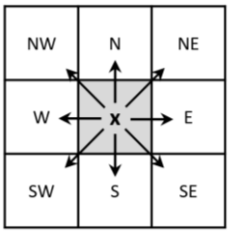

Fig. 2. The eight directions of motion.

6. Time frame

The user can set the time frame by assigning a value to the variable Time_span, specifying the amount of time covered in a single model simulation. The time resolution can be set by assigning a value to the variable Time_step, specifying the amount of time covered in one time step. Dividing the Time_span by the Time_step yields the total number of time steps (Time_range). The points in time at which input data is provided and for which output data is computed, are stored in the 1D array called Time[t].

7. Model input

The model input consists of terrain properties assigned to each grid cell in the domain and the weather conditions that occur within the grid cells. The terrain properties only have a spatial variability. Whereas the weather conditions (might) vary through space and time. For simple (toy) model applications, the input data arrays can be manually filled with values. For real life applications, we recommend to extract the input data from existing data bases.

7.1 Land Use

The land use input data array, $LU[i,j]$, contains for each grid cell the type of land use, whereby the model distinguishes between water (i.e. river), bare land, forest, agricultural land, urban land, and grass/shrubland. Each land use type has a label (Tab. 1). The $LU[i,j]$ array is used to plot the land use map.

Tab. 1. Labels used in the LU array to indicate the type of land use.

Land use	Label
River	0
Bare land	100
Forest	200
Agricultural land	300
Urban land	400
Grass/shrub land	500

7.2 Population density & mismanaged plastic waste generation

The population density input array, $Y[i,j]$, contains for each grid cell the number of inhabitants. This array is used to plot the population density map. The user can decide to assign artificial numbers of inhabitants to grid cells that are in reality uninhabited in order to account for littering associated with recreational activities in those areas, e.g. in forests or on beaches.

The mismanaged plastic waste (MPW) generation for each grid cell and time step is stored in the 3D $p\_conc\_INPUT[i,j,t]$ array. This array is used to plot the MPW generation map. The MPW generation of grid cell $[i,j]$ can be calculated as follows:

$$MPW[i,j] = MPW_{capita} \times MWF \times PP \times Y[i,j] \qquad (1)$$

where MPW_capita equals the amount of municipal solid waste generated per capita per time unit, MWF represents the fraction of waste that is mismanaged, PP is the fraction of waste that consists of plastics and Y is the number of inhabitants of the grid cell in question. The MPW_capita, MWF, and PP values can be achieved from Lebreton and Andrady (2019). It is assumed that all factors in equation (1) are constant through time. Consequently, the MPW generation map is constant through time as well (this means that $p\_conc\_INPUT[i,j,0]$ = $p\_conc\_INPUT[i,j,1]$ = $p\_conc\_INPUT[i,j,2]$ = ... etc.).

7.3 Topography

The topography input array, $Z[i,j]$, contains for each grid cell the elevation above (positive) or below (negative) mean sea level in meters. The $Z[i,j]$ array is used to plot the topography map.

From the topography data the model computes the terrain slopes for each grid cell. The slope in a certain direction is referred to as the distance weight drop (dwd). The model computes for each grid cell in the domain the dwd to each of its neighbouring grid cells, e.g. for interior grid cells surrounded by eight other grid cells, the model calculates eight dwd values (Fig. 3a). The distance weight drop of a grid cell towards a neighbouring grid cell is computed as follows:

$$dwd = \frac{\Delta Z}{\Delta d} \qquad (2)$$

Whereby ΔZ is the change in elevation (m) between the grid cell points, and Δd the horizontal distance between these grid points. The elevations of the grid cells are stored in the Z[i,j] array and the horizontal distances are either the zonal resolution (Lon_res_m), the meridional resolution (Lat_res_m) or the diagonal distance (d_m), which equals $\sqrt{(\mathrm{Lon\_res\_m}^2 + \mathrm{Lat\_res\_m}^2)}$ (Fig. 3b).

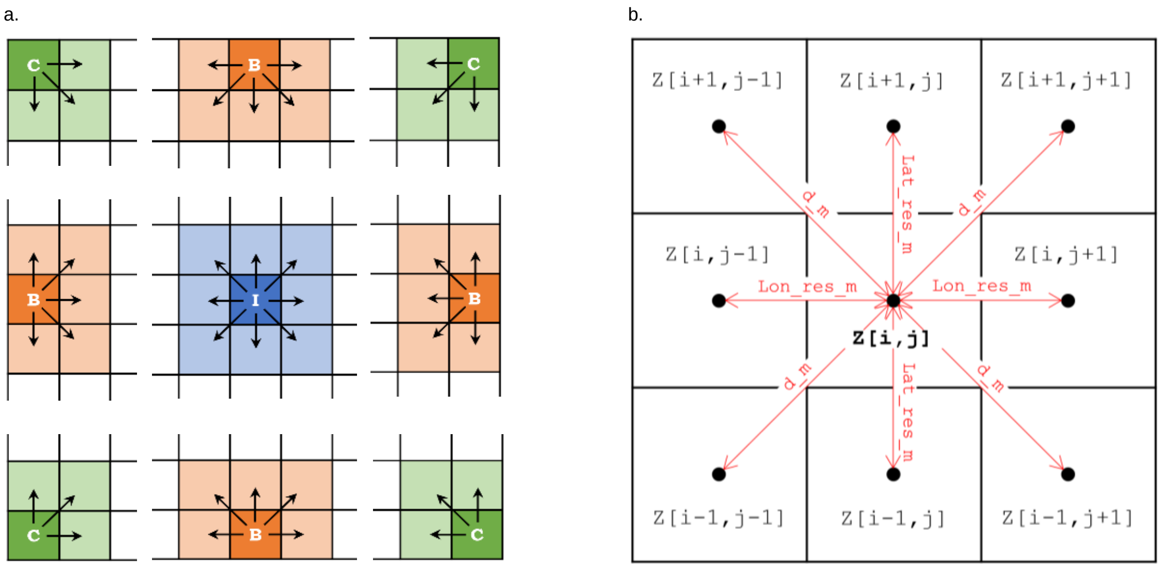

Fig. 3. (a) Schematic drawing of a model grid with rectangular grid cells. The arrows indicate the directions in which the distance weight drop (dwd) is calculated. 'I' presents an interior grid cell, 'C' a corner grid cell, and 'B' a grid cell at the border of the model domain. (b) Geographical map representation of an interior grid cell and its eight neighbouring grid cells. The elevation of the grid cells are stored in the Z[i,j] array (needed to compute ΔZ in equation (2)). The red arrows indicate the horizontal distance between grid points (Δd in equation (2)). Lat_res_m is the meridional grid resolution (m), Lon_res_m is the zonal grid resolution (m), and d_m is the diagonal distance (m) between grid points (= $\sqrt{(\mathrm{Lon\_res\_m}^2 + \mathrm{Lat\_res\_m}^2)}$).

The minimum value of the distance weight drop indicates the steepest *downhill* slope for that grid cell. The magnitude of the steepest downhill slope for each grid cell is stored in the 2D Q[i,j] array. The direction of that steepest downhill slope is stored in the 2D q[i,j] array using the direction labels shown in Fig. 4. The Q[i,j] and q[i,j] arrays are used to plot the steepest downhill slope magnitude and direction maps. In case a grid cell is only surrounding by grid cells that have a higher elevation, the minimum distance weight drop value and its corresponding direction indicate the gentlest *uphill* slope.

1000	1001	1002
1007	**x**	1003
1006	1005	1004

Fig. 4. Map view of an interior grid cell 'x' and its eight neighbouring grid cells. Numbers correspond to the labels used to indicate towards which neighbouring grid cell the steepest downhill slope of grid cell 'x' occurs. For example, if the steepest downhill terrain slope is towards the north, then grid cell 'x' will have label 1001 in the q[i,j] array.

7.4 Wind speed and direction

The wind speed input array, W_speed[i,j,t], indicates for each time step for each grid cell the speed of the air flow (m/s). The wind speed values can be directly extracted from a wind speed data set or be generated based on wind speed frequency tables. In the current version of the Plastic Pathfinder it is assumed that each grid cell has the same wind speed during one time step. The wind speed values in the current version of the model are derived from a wind speed frequency table, which resulted from actual wind speeds that were measured (in the period 1981 – 2000) at the De Bilt weather station, the Netherlands (publicly available on the website from the Royal Netherlands Meteorological Institute (2020a)). The wind speed frequency table was split and assigned to 12 frequency arrays, one for each month (e.g. W_speed_freq_Jan[i], W_speed_freq_Feb[i], etc.). The values in these arrays indicate the frequencies with which the wind speeds measured at ‘De Bilt’ weather station (52.1015441 latitude and 5.1779992 longitude, the Netherlands) in the period 1981-2000, fell within 20 pre-defined wind speed classes. The 1D W_speed_classes[i] array contains the numbers 1 to 20, because there are 20 wind speed classes. To pick a wind speed class for each time step, the model uses the NumPy function numpy.random.choice([A], p = [B]), whereby A = W_speed_classes[i] and B = W_speed_freq_*month*[i]. The selected wind speed classes are stored in the 1D w_speed_class[t] array. The final wind speed value (m/s) for a time step results from a random selection of a wind speed value that falls within the limits of the selected wind speed class. The final wind speed values (m/s) are stored in the 1D w_speed_value[t] array. The 3D W_speed[i,j,t] array is created from the 1D w_speed_value[t] array by simply addressing the wind speed value for each time step to all the grid points.

The wind direction input array, W_dir[i,j,t], indicates for each time step for each grid cell the direction *towards* which the air flows. The wind directions can be directly extracted from a wind direction data set or be generated based on wind direction frequency tables. In the current version of the Plastic Pathfinder it is assumed that each grid cell has the same wind direction during one time step. The wind direction values in the current version of the model are derived from a wind direction frequency table, which resulted from actual wind speeds that were measured (in the period 1981 – 2000) at the De Bilt weather station, the Netherlands (publicly available on the website from the Royal Netherlands Meteorological Institute (2020b)). The wind direction frequency table was split into 12 separate arrays, i.e. one for each month (e.g. W_dir_freq_Jan[i], W_dir_freq_Feb[i], etc.), because in some months high wind speeds are more common than in other months. The values in these arrays indicate the frequencies with which the wind directions measured at ‘De Bilt’ weather station (52.1015441 latitude and 5.1779992 longitude, the Netherlands) in the period 1981-2000, fell within 37 wind speed classes. The 1D W_dir_classes[i] array contains the numbers 1 to 37, because there are 37 wind direction classes. To pick a wind direction class for each time step, the model uses the NumPy function numpy.random.choice([A], p = [B]), whereby A = W_dir_classes[i] and B = W_dir_freq_*month*[i]. The selected wind direction classes are stored in the 1D w_dir_class[t] array. The wind direction for a time step is generated by randomly selecting a wind direction that falls within the limits of the selected wind direction class. The wind direction values (in degrees) are stored in the 1D w_dir_value[t] array. It must be noted that by default the wind direction is defined as the direction from which the wind originates. However, (plastic) particles carried by the wind are transported in the opposite direction. Therefore, 180° must be added to the ‘official’ wind direction in order to obtain the wind transport direction. Finally, the wind transport directions, in degrees (0° - 360°), are converted and labelled to the eight possible directions used in the model (Fig. 2) using Tab. 2. The 1D Wind-displacement_dir[t] array contains for each time step the wind transport directions, i.e. the directions in which the wind transports (plastic) particles, using the direction labels shown in Fig. 5. The 3D W_dir[i,j,t] array is created from the 1D Wind-displacement_dir[t] array by simply addressing the wind direction labels for each time step to all the grid points.

Tab. 2. Wind direction conversion table. From degrees to the wind direction labels (Fig. 5).

Wind direction* [*degrees*]	**Label**
292.5 - 337.5	2000
337.5 - 22.5	2001
22.5 - 67.5	2002
67.5 - 112.5	2003
112.5 - 157.5	2004
157.5 - 202.5	2005
202.5 - 247.5	2006
247.5 - 292.5	2007

***** *Direction towards which the wind flows*

2000	2001	2002
2007	**x**	2003
2006	2005	2004

Fig. 5. Map view of an interior grid cell 'x' and its eight neighbouring grid cells. Numbers correspond to the labels used to indicate towards which neighbouring grid cell the wind in grid cell 'x' flows. For example, if the wind flows towards the north, then grid cell 'x' will have label 2001 in the W_dir array.

7.5 Surface runoff flux and direction

The surface runoff input data array, $\mathrm{R\_speed}[i,j,t]$, indicates for each time step for each land grid cell the amount of surface runoff (mm/d). The surface runoff values can be directly extracted from a surface runoff (or rainfall) data set or be generated based on rainfall frequency tables. In the current version of the Plastic Pathfinder the surface runoff is computed from rainfall values and it is assumed that each grid cell receives the same amount of rainfall during one time step. . The rainfall values in the current version of the model are derived from a rainfall frequency table, which resulted from actual rainfall fluxes that were measured (in the period 1981 – 2000) at the De Bilt weather station, the Netherlands (publicly available on the website from the Royal Netherlands Meteorological Institute (2020c)). The rainfall data set contains for each day (from 1 January 1981 up and until 31 December 2000) the total amount of rainfall (mm) recorded by the Royal Netherlands Meteorological Institute at 'De Bilt' weather station (52.1015441 latitude and 5.1779992 longitude, the Netherlands). Each day was assigned to one of the 23 rainfall classes based on the total amount of rainfall that fell during that day. Subsequently, the frequencies for each rainfall class were computed and stored in the 1D $\mathrm{Rainfall\_freq}[i]$ array. Note that, unlike the wind data, there are no monthly variations in the rainfall frequencies. Therefore, the frequencies in the $\mathrm{Rainfall\_freq}[i]$ array hold for all months. The $\mathrm{Rainfall\_classes}[i]$ array contains the numbers 1 to 23, because there are 23 rainfall classes. The model uses the NumPy function $\mathrm{numpy.random.choice}([A], p = [B])$, with $A = \mathrm{Rainfall\_classes}[i]$ and $B = \mathrm{Rainfall\_freq}[i]$ in order to pick a rainfall class for each time step. The selected rainfall classes are stored in the 1D $\mathrm{rainfall\_class}[t]$ array. The final rainfall value (mm/d) for a time step is generated by randomly selecting a rainfall value that falls within the limits of the selected rainfall class. The final rainfall values (mm/d) are stored in the 1D $\mathrm{rainfall\_value}[t]$ array. Next, the rainfall values are converted to surface runoff values by using runoff coefficients. The runoff coefficient is the fraction of the rainwater that does not infiltrate in the soil and consequently becomes surface runoff. The surface runoff coefficients for river, bare, forest, agricultural, urban and grass\shrub lands are referred to as $\mathrm{RC\_river\_value}$, $\mathrm{RC\_bare\_value}$, $\mathrm{RC\_forest\_value}$, $\mathrm{RC\_agricul\_value}$, $\mathrm{RC\_uban\_value}$ and $\mathrm{RC\_grass\_value}$, respectively. The values for these runoff coefficients (Tab. 3) are based on typical reported runoff coefficients for these types of land uses (Goel, 2011; Karamage et al, 2017). The rainfall value (mm/d) multiplied with a runoff coefficient yields the surface runoff (mm/d) for each grid cell and is stored for each time step in the 3D $\mathrm{R\_speed}[i,j,t]$ array. For river grid cells the surface runoff value is set at 0 mm/d, as surface runoff is only relevant on land.

Tab. 3. Runoff Coefficients (= runoff / rainfall) used by the Plastic Pathfinder to convert the amount of rainfall in millimetres per day to the amount of surface runoff in millimetres per day. The values are based on typical runoff coefficients reported by Goel (2011) and Karamage et al. (2017).

Type of land use	Runoff Coefficient
River	1.00
Urban land	0.70
Bare land	0.50
Agricultural land	0.30
Grass/Shrubland	0.20
Forest	0.10

The surface runoff direction input array, R_dir[i,j], indicates for each time step for each grid cell the direction towards which the surface runoff flows. For land grid cells, the surface runoff directions are equal to the directions of the steepest terrain slope, i.e. equal to the q[i,j] array. The direction labels used for the surface runoff directions are shown in Fig. 6. For river grid cells, the surface runoff direction label is set to 0, as surface runoff is only relevant on land.

3000	3001	3002
3007	**x**	3003
3006	3005	3004

Fig. 6. Map view of an interior grid cell 'x' and its eight neighbouring grid cells. Numbers correspond to the labels used to indicate towards which neighbouring grid cell the surface runoff in grid cell 'x' flows. For example, if the surface runoff flows towards the north, then grid cell 'x' will have label 3001 in the R_dir array.

8. Plastic mobilisation and transport thresholds

8.1 Wind speed thresholds

The wind speed threshold can be calculated as a function of only the type of land use – option 1 – or as a function of the type of land use and the (combination of) terrain slope and wind direction – option 2. In the ParVal array at the top of the program, the user can chose for option 1 or 2 by filling in a '1' or '2' at ParVal[6], respectively.

Option 1

The wind speed threshold array for option 1, W_T_lu[i,j] indicates for each grid cell the critical wind speed (m/s) required to mobilise and transport plastic over at least a distance of d_m meters (i.e. the horizontal distance between two diagonal grid points – see Fig. 3b). The wind speed threshold values only depend on the type of land use and are therefore independent of time. The wind speed thresholds are referred to as W_T_R, W_T_B, W_T_F, W_T_A, W_T_U and W_T_G, for river, bare, forest, agricultural, urban and grass\shrub lands, respectively. The values for these parameters can be filled in by the user at ParVal[0] to ParVal[5] at the top of the program. An explanation for the wind speed threshold values that are currently used in the Plastic Pathfinder can be found in the Supplementary Materials from Mellink et al. (2021).

Option 2

This approach assumes that if the wind flows uphill/downhill, the ability of the wind to mobilise and transport plastics (in the direction of the wind) proportionally decreases/increases with the steepness of the slope, respectively. As the wind directions (can) vary through time, the wind speed thresholds calculated via option 2 are time dependent. The wind speed threshold array for option 2, W_T_lus[i,j,t,], indicates for each time step for each grid cell the critical wind speed (m/s) required to mobilise and transport plastic over at least a distance of d_m meters (i.e. the horizontal distance between two diagonal grid points – see Fig. 3b). The wind speed thresholds are computed by calculating for each time step and grid cell the terrain slope angle (rad) <u>in the direction of the wind</u> at that time step. If the slope angle in the direction of the wind equals 0°, then the wind speed thresholds from option 1 (i.e. W_T_R, W_T_B, W_T_F, W_T_A, W_T_U or W_T_G) are used. For non-zero terrain slope angles, a value of 4.2 (m/s), referred to as deltaTW (stored at ParVal[8]), is added (for uphill winds) or subtracted (for downhill winds) from the W_T_R, W_T_B, W_T_F, W_T_A, W_T_U or W_T_G value for each radian of terrain slope angle. An explanation for the 4.2 (m/s) deltaTW value can be found in Mellink et al. (2021).

8.2 Surface runoff thresholds

The surface runoff threshold array, R_T_lus[i,j], indicates for each grid cell the critical amount of surface runoff (mm/d) required to mobilise and transport plastic over at least a distance of d_m meters (i.e. the horizontal distance between two diagonal grid points – see Fig. 3b). The surface runoff thresholds depend on the type of land use and terrain slope, which are both constant through time. Consequently, the surface runoff thresholds are time independent. The surface runoff threshold is determined by the terrain slope angle of the steepest downhill slope, which is calculated from the steepest slope values (m/m) (stored in Q[i,j]). The slope angle category in which the terrain slope angel falls, combined with the type of land use, determines the final surface runoff threshold. There are 10 slope angle categories of which category 0 corresponds to uphill slopes (for grid cells that are only surrounded by higher topographies). The remaining 9 categories cover the downhill slope angles between 0^0 and 90^0 with intervals of 10^0. The surface runoff threshold values (mm/d) for each combination of land use and slope angle category are set by the user and stored at ParVal[9] to ParVal[58] at the top of the program. For river grid cells the surface runoff threshold is set to 0 (mm/d). An explanation for the surface runoff threshold values that are currently used by the Plastic Pathfinder can be found in Mellink et al. (2021).

9. Plastic mobilisation

At each time step the wind speed (W_speed[i,j,t]) and surface runoff (R_speed[i,j,t]) values are compared to the wind speed and surface runoff thresholds, respectively. The outcome of this comparison is labelled with the numbers 10, 20, 30, and 40 (see Tab. 4). The 3D Check[i,j,t] array contains for each time step for each grid cell the outcome label. The check value for river grid cells is set to 0. The Check[i,j] array is used to plot the plastic mobilisation maps.

Tab. 4. Labels used for the outcome of the comparison between wind speeds and wind speed thresholds, and surface runoff fluxes and surface runoff thresholds.

Outcome of the comparison between weather conditions and thresholds	Label
Only the wind speed threshold is surpassed	10
Only the surface runoff threshold is surpassed	20
Both the wind speed and the surface runoff threshold is surpassed	30
Neither the wind speed or surface runoff threshold is surpassed	40

10. Plastic transport and accumulation

10.1 General framework

Each time step begins with a start MPW distribution, which is stored in the 3D p_conc_start[i,j,t] array. For each time step, the model relocates, or not, the MPW that is present within the river basin, depending on the wind and surface runoff conditions. By the end of each time step, when all MPW has, or not, been relocated, an end MPW distribution has emerged. The end MPW distribution, together with the MPW input, forms the start MPW distribution for the next time step (Fig. 7). Except for the first time step ($t = 0$), then the start MPW distribution is simply the MPW generation distribution (stored in the p_conc_INPUT[i,j,t] array – see section 7.2). The next section provides a more detailed explanation of how the MPW relocation is modelled.

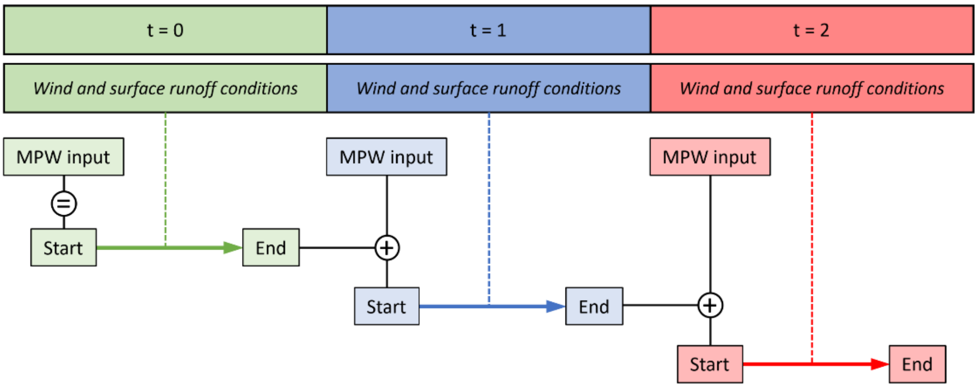

Fig. 7. Schematic representation of the calculation of the (re)distribution of MPW in the model. The first 3 time steps are shown here as an example, but the same structure holds for the remaining time steps. In general, the start MPW distribution of a time step is the sum of the MPW input for that time step and the end MPW distribution of the previous time step. Except for t = 0, where the start MPW distribution simply equals the MWP input for t = 0. Note that the MPW input map is the same for each time step (for explanation see section 7.2). The change from the start MPW to the end MPW distribution depends on the wind and surface runoff conditions that occur during the specific time step (dashed lines).

10.2 Modelling the transport of MPW

For each time step, the model loops through all the grid cells in the model domain, if no MPW is present in a grid cell, it goes to the next. If a grid cell contains MPW, the value of the move_to variable will be determined, which is an indication of the displacement direction. Fig. 8 shows to which directions the values of move_to correspond (e.g. move_to = 3 means that the MPW is displaced to the neighbouring grid cell in the east). The value of move_to depends on whether and which thresholds are surpassed (the outcome of this comparison is stored in the Check[i,j] array – see section 9). If the Check[i,j] array indicates that only the wind threshold is surpassed, then the value of move_to is equal to the wind direction (stored in W_dir[i,j,t]). If the Check[i,j] array indicates that only the surface runoff threshold is surpassed, then the move_to value is equal to the surface runoff direction (stored in R_dir[i,j,t]). If the Check[i,j] array indicates that both thresholds are surpassed, then the model randomly picks either the wind or the surface runoff direction to assign a value to move_to. If the Check[i,j] array indicates that none of the thresholds are surpassed, then the MPW does not move, i.e. stays in its current grid cell, and the value of move_to equals 8.

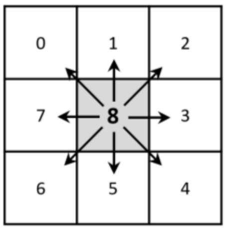

Fig. 8. The value of the move_to variable indicates towards which neighbouring grid cell the MPW is displaced. If no MPW displacement occurs: move_to = 8.

The zonal and meridional components of the MPW transport vectors for each grid cell and for each time step are stored in the 3D TA_u[i,j,t] and TA_v[i,j,t] arrays, respectively. Each move_to value has a corresponding MPW transport vector. Tab. 5 contains the zonal and meridional components of these MPW transport vectors. The MPW vectors can be plotted in the MPW distribution maps with thicknesses that are linearly proportional with the MPW flux (mass/day) that they represent.

Tab. 5. The zonal and meridional components of the MPW transport vector for all the possible values of the move_to variable.

move_to	TA_u	TA_v
0	-1	1
1	0	1
2	1	1
3	1	0
4	1	-1
5	0	-1
6	-1	-1
7	-1	0
8	0	0

Based on the value of move_to the model computes the indices of the grid cell towards which the MPW has been relocated. These new array indices (i_new and j_new) are computed using the indices (i and j) of the current grid cell (Tab. 6). In case the new location of the MPW lies inside the model domain, this new location indicates its end location for this time step. If the MPW has not moved the start location is equal to the end location, i.e. i_new = i and j_new = j. In case the new location is a river grid cell, the MPW has left the terrestrial environment and will from that point in time onwards not move anymore (as modelling riverine plastic transport lies outside of the scope of the (current version of) Plastic Pathfinder). In other words, once the MPW entered a river grid cell, it cannot be relocated anymore and consequently will (fictively) accumulate there. The model keeps track of the total amount of MPW in the terrestrial environment and in the freshwater environment: the 1D Terrestrial_plastic[t] and Riverine_plastic[t] arrays contain the total mass of MPW that is located in land and river grid cells, respectively, by the end of time step t. In case the new location of the MPW lies <u>outside</u> the model domain, it means that these plastics are leaving the river basin. The model records for each time step the amount of MPW that has left the river basin, in which it distinguishes between two types of 'leakage':

1. direct coastal leakage; from a land grid cell that in reality presents a coastline to the sea
2. direct leakage to adjacent land; from a land grid cell that in reality lies inland

For each time step the amount (mass) of MPW that is lost via direct coastal leakage is stored in the 1D OUT_coast[t] array and the amount (mass) of MPW that is lost via direct leakage to adjacent land is stored in the 1D OUT_land[t] array. The 1D OUT_land[t] array is simply the sum of the OUT_coast[t] and OUT_land[t] arrays and indicates for each time step the amount (mass) of MPW that has left the model domain during that time step. The 1D OUT_land_cum[t] array contains the cumulative amount (mass) of MPW that has left the domain by the end of each time step. The 1D Present_in_river_basin[t] array contains the <u>total</u> amount (mass) of MPW that is present in the entire model domain by the end of time step t. This amount is computed by subtracting the cumulative amount of MPW that has left the river basin since that time step (OUT_land_cum[t]), from the cumulative amount of MPW that has been generated since that time step (IN_cum[t]).

Tab. 6. Computation of the i_new and j_new indices from the i and j indices, respectively, for all the possible values of the move_to variable.

move_to	i_new	j_new
0	i + 1	j – 1
1	i + 1	j
2	i + 1	j + 1
3	i	j + 1
4	i – 1	j + 1
5	i – 1	j
6	i – 1	j – 1
7	i	j – 1
8	i	j

To be able to study the MPW stock evolution at the level of a single grid cell the model stores the MPW in- and output fluxes for each time step for each grid cell. The 3D IN_total[i,j,t] array contains the total amount (mass) of MPW that has entered the grid cell [i,j] during time step t. The 3D OUT_total[i,j,t] array contains the total amount (mass) of MPW that has left the grid cell [i,j] during time step t. The IN_total[i,j,t] array is equal to the sum of the two possible MPW input fluxes into a grid cell: (1) the MPW generation within the grid cell (e.g. direct littering from inhabitants) (IN_MPW_generation[i,j,t]), and (2) the MPW that comes from neighbouring grid cells by wind or surface runoff driven transport (IN_from_adj_cell[i,j,t]) (Fig. 9). MPW can leave the grid cell when wind or surface runoff relocates MPW and thus removes MPW from the grid cell (OUT_to_adj_cell[i,j,t]) (Fig. 9).

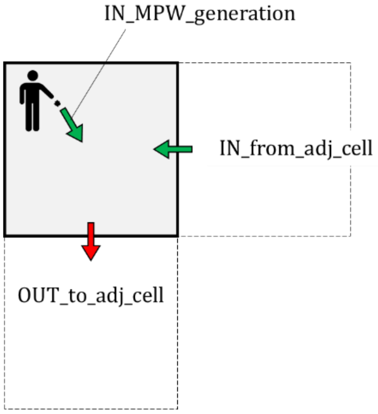

Fig. 9. The in- and output fluxes on the scale of an individual model grid cell. The labels of the arrows indicate the names of the NumPy arrays in which the values of these fluxes are stored.

11. Potential plastic transport directions

The Plastic Pathfinder can be used to predict the (potential) transport directions of plastics without actually providing an MPW input. In order to do so 3D tally arrays, initially filled with zeros, are created for each of the eight directions of motion: tally_dir_0[i,j,t], tally_dir_1[i,j,t], tally_dir_2[i,j,t], tally_dir_3[i,j,t], tally_dir_4[i,j,t], tally_dir_5[i,j,t], tally_dir_6[i,j,t] and tally_dir_7[i,j,t]. For each time step the model determines for each grid cell in which direction MPW (if present) would have been transported under the provided set of wind and surface runoff conditions and thresholds. This transport direction is encoded with a move_to value (Fig. 8). Each transport direction, i.e. each move_to value, has its own tally array (e.g. move_to = 0 belongs to tally_dir_0[i,j,t] and move_to = 1 belongs to tally_dir_1[i,j,t] etc.). Each time wind and/or surface runoff conditions force MPW transport in a specific grid cell, +1 is added in the tally array that belongs to that transport direction. This is done for each grid cell and time step.

In this way the Plastic Pathfinder is able to tally for each grid cell how often plastic transport occurs in all of the eight possible transport directions. When the model has looped through all time steps, the eight tally arrays contain the cumulative amount of events during which plastics would have been transported in each of the eight directions. The model extracts the final time steps from the eight tally arrays and stores these values in the cumulative tally arrays: tally_dir_0_cum[i,j], tally_dir_1_cum[i,j], tally_dir_2_cum[i,j], tally_dir_3_cum[i,j], tally_dir_4_cum[i,j], tally_dir_5_cum[i,j], tally_dir_6_cum[i,j] and tally_dir_7_cum[i,j]. Subsequently, the cumulative tally arrays are used to create the potential plastic routing map. To do so the transport directions are converted to 2D vectors, of which the zoanl and meridional components are stored in the 2D dir_0/1/2/3/4/5/6/7_u[i,j] and dir_0/1/2/3/4/5/6/7_v[i,j] arrays, respectively. The thickness of the transport vectors in the potential plastic routing map are linearly proportional to the frequency with which MPW was forced in that particular direction.

---

12. References

- Anaconda Software Distribution (2016). Computer software. Vers. 2-2.4.0. Anaconda. https://anaconda.com
- Goel, M. K. (2011). Runoff coefficient. In: *Encyclopedia of Snow, Ice and Glaciers*. Dordrecht: Springer. p. 952–3. http://dx.doi.org/10.1007/978-90-481-2642-2_456
- Jenson, S. K., & Domingue, J. O. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric engineering and remote sensing*, *54*(11), 1593-1600.
- Karamage, F., Zhang, C., Fang, X., Liu, T., Ndayisaba, F., Nahayo, L., ... & Nsengiyumva, J. B. (2017). Modeling rainfall-runoff response to land use and land cover change in Rwanda (1990–2016). *Water*, *9*(2), 147. http://dx.doi.org/10.3390/w9020147
- Lebreton, L., & Andrady, A. (2019). Future scenarios of global plastic waste generation and disposal. *Palgrave Communications*, *5*(1), 1-11. http://dx.doi.org/10.1057/s41599-018-0212-7
- Mellink, Y., van Emmerik, T., Kooi, M., Laufkötter, C., Niemann, H. (2022). The Plastic Pathfinder: A Macroplastic Transport and Fate Model for Terrestrial Environments. *Manuscript submitted for publication*.
- Royal Netherlands Meteorological Institute (2020a). Frequentietabellen Windsnelheid in m/s, Distributief in procenten, De Bilt. Accessed 29 May 2020. http://projects.knmi.nl/klimatologie/frequentietabellen/maand.cgi
- Royal Netherlands Meteorological Institute (2020b). Frequentietabellen Windrichting in graden, Distributief in procenten, De Bilt. Accessed 29 May 2020. http://projects.knmi.nl/klimatologie/frequentietabellen/maand.cgi
- Royal Netherlands Meteorological Institute (2020c). Daggegevens van het weer in Nederland, RH, Etmaalsom van de neerslag (in 0.1 mm) (-1 voor <0.05 mm). Accessed 29 May 2020. http://projects.knmi.nl/klimatologie/daggegevens/selectie.cgi