# *T-vector manual*

*Author: Arnaud Temme, November 2010*

## Introduction

*The T-vector calculation utility was written as part of a research project of the department of Land Dynamics at Wageningen University – the PhD research of Wim Viveen. Code was provided by Arnaud Temme, as part of the LAPSUS modeling framework. The utility and the code are available through the department's website under the condition that the relevant papers by Viveen are cited. These papers are in preparation at the moment that this text is written, but please consult with the authors about their availability.*

*Calculation of T-vectors was first proposed by Cox (1994), who defined the T-vector as a measure of transverse topographic basin symmetry. The T-vector, originally called drainage-basin asymmetry vector, is scaled between values of 0 (complete symmetry) and 1 (complete asymmetry). The T-vector method has great potential to help decyphering block tilt directions. The method is not widely used however as the T-vectors cannot be automatically calculated in GIS packages. The vectors require step-by-step manual calculation and drawing. For this reason we have developed a method that speeds up the calculation of the T-vectors.*

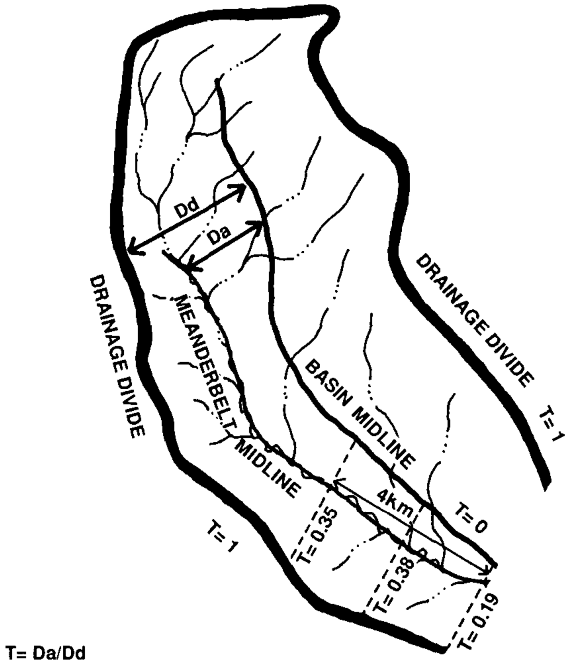
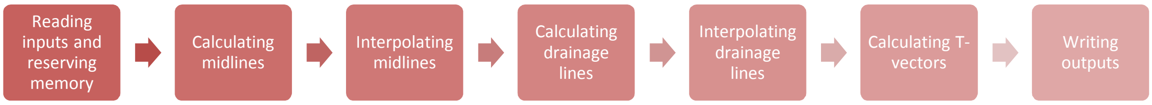


***Fig. 1: A map explaining T-vector calculation, adapted from Cox (1994, GSA Bulletin 106, p. 571-581). The drainage midline is exactly in between the drainage divides. The position of the actual drainage pattern relative to drainage divide and midline is calculated for segments along the actual drainage pattern (in this picture: 2 km segments). For each segment, a T-value and a T-direction (together, the T-vector) are calculated.***

## Overview

*The model performs a number of steps:*



Reading inputs and reserving memory → Calculating midlines → Interpolating midlines → Calculating drainage lines → Interpolating drainage lines → Calculating T-vectors → Writing outputs

***Fig. 2: T-vector utility overview***

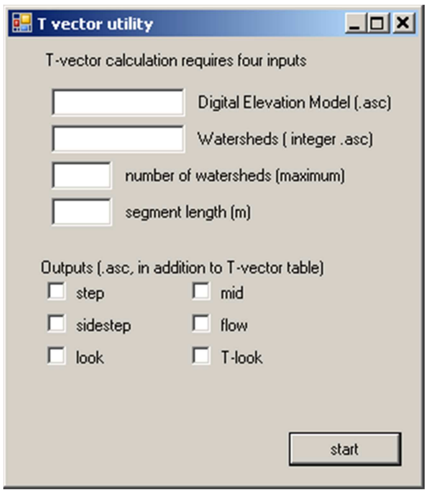
*Each of these steps is explained below.*

## Reading inputs and reserving memory

*The user interface requires four inputs.*

*First, a Digital Elevation Model (DEM) must be chosen. Although DEMs do not need to be sink-free, we advise to check for spurious elements in input-data to prevent model instability. Ideally, DEMs are supplied in non-integer format (i.e. with decimal values). The utility will give a warning when this is not the case, but will proceed either way. DEMs must be supplied in .asc format.*

*Second, a watershed grid must be supplied. The watershed grid must have integer values (whole numbers), numbered from 0. Missing values are not a problem (for instance when a watershed grid contains watersheds 0-15, 19-28). Although the model will try to calculate T-vectors for incomplete catchments, it only makes sense to include complete catchments surrounded with nodata values (-9999).*



T vector utility

T-vector calculation requires four inputs

Digital Elevation Model (.asc)

Watersheds ( integer .asc)

number of watersheds (maximum)

segment length (m)

Outputs (.asc, in addition to T-vector table)

□ step □ mid

□ sidestep □ flow

□ look □ T-look

start

***Fig. 3: interface of the T-vector utility, showing input and output choices.***

*Third, in order to reserve the proper amount of memory before running, the model requires you to tell it the highest number that a watershed has. It is okay but useless to fill in a higher number, although the model will then use more memory. It is also okay to fill in a lower number, but some of your watersheds will then be ignored. Note that if your watershed grid defines watersheds numbered from 0 to 15 and from 19 to 28, this number should be 28.*

*Fourth, the model calculates T-vectors for segments of the actual drainage pattern in each watershed. The length of these segments must be defined in metres. The model is limited to a maximum of 100 segments per watershed, so very low values for length are not okay.*

## *Calculating midlines*

*The model calculates midlines in a number of steps:*

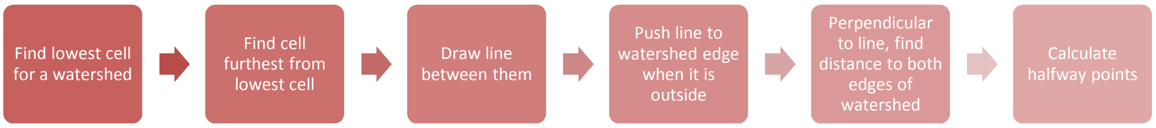

Find lowest cell for a watershed
Find cell furthest from lowest cell
Draw line between them
Push line to watershed edge when it is outside
Perpendicular to line, find distance to both edges of watershed
Calculate halfway points

***Fig. 4: overview of midline calculation***

*First, it finds the lowest cell for a watershed by simply checking all cells in the grid. Then, it finds the cell from the same watershed that is furthest from this lowest cell. This seems the most objective and effective method (because the highest cell in the watershed may well be not a good choice). It is conceivable that the user defines this optimal starting point for the watershed, but the present model version does not incorporate this.*

*Knowing both beginning and end of the watershed, the model draws a straight line between them (called stepline). In cases where this straight line is not inside the watershed (imagine a banana-shaped watershed), the line is pushed sideways to cover the edge of the watershed (the line is then called sidestepline).*

*Walking an imaginary path along this line, the model then looks perpendicular to stepline in both directions from sidestepline to find the watershed boundaries (these lines are called looklines). For banana-shaped watersheds, one of these distances may be zero because sidestepline covers the watershed boundary.*

*Finally, halfway points are calculated from this pair of distances and the halfway points together are called midline.*

## *Interpolating midlines*

*It is entirely possible that the halfway points forming the original midline do not form a continuous line, for instance when, at some point along stepline, a watershed becomes much wider on one side. This may become a problem later in the calculation of T-vectors. To counter this problem, the model starts at the lowest point in a watershed (by definition part of midline), and iteratively tries to find neighbours that are also part of midline. If no direct neighbours are part of midline, it finds the first non-direct neighbour that is part of midline and makes the cells along the way also part of midline.*

## *Calculating drainage lines*

*Starting from the cell furthest from the outlet for a given watershed, the lowest lower neighbour is iteratively selected to become part of the drainage line. This way, the model walks down a watershed in steepest-descent fashion, to arrive at the outlet (the lowest cell).*

*When no lower neighbour is found, the model looks beyond the direct neighbours until it finds a lower neighbour and continues there. Note that it is still a good idea to check your data for spurious pits and sinks to prevent model instability issues.*

## *Interpolating drainage lines*

*This occurs in exactly the same manner as for the midlines and ensures that the drainage lines are continuous, even when sinks have been found.*

## Calculating T-vectors

*In the final calculation step, the model finds the first point on drainage line that is at least as far from the outlet of a given watershed as segment length indicates.*

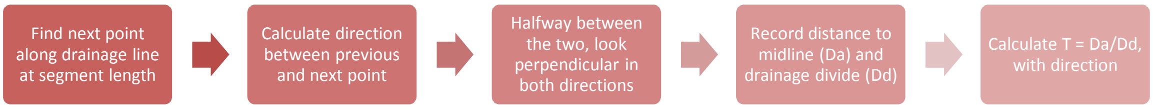

Find next point along drainage line at segment length
Calculate direction between previous and next point
Halfway between the two, look perpendicular in both directions
Record distance to midline (Da) and drainage divide (Dd)
Calculate T = Da/Dd, with direction

***Fig. 5: Overview of T-vector calculation***

*It then calculates the direction between these two points. Halfway between them, it looks sideways (T-lookline), perpendicular to this direction, and records distances to midline (Da) and drainage divide (Dd). The division of these two (T = Da/Dd) then yields the value of T-vector. The direction of T-vector is perpendicular to the line between the two points.*

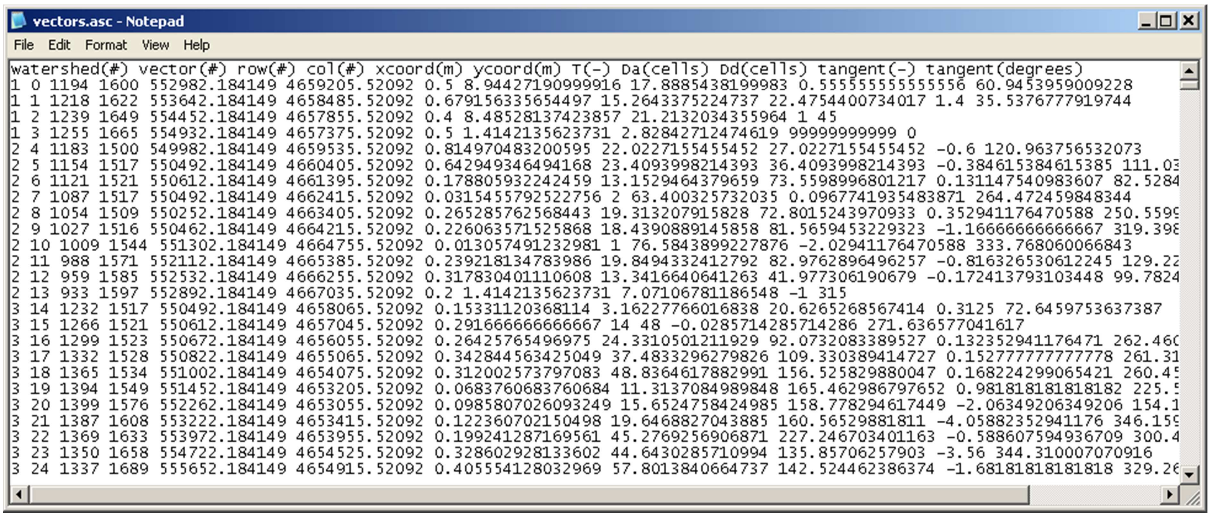
*The model continues iteratively, starting from the previous high point, until it reaches the point furthest from the outlet.*

## Writing outputs

*The model has a standard output containing T-vector information and a number of optional .asc output maps.*

*The T-vector output file is called vectors.asc and can be used as input to GIS software. It contains the following information:*

1. *Watershed number*
2. *Vector number (cumulative over all watersheds)*
3. *The row of the cell halfway the segment where the vector is based*
4. *The col of the cell halfway the segment where the vector is based*
5. *The size of the vector (T = Da/Dd) – should be always between 0 and 1*
6. *The Da-value of the vector (in grid cells)*
7. *The Dd-value of the vector (in grid cells)*
8. *The direction of the vector in tangent-notation (mathematic).*
9. *The direction of the vector in degrees (geographic)*



```
watershed(#) vector(#) row(#) col(#) xcoord(m) ycoord(m) T(-) Da(cells) Dd(cells) tangent(-) tangent(degrees)
1 0 1194 1600 552982.184149 4659205.52092 0.5 8.94427190999916 17.8885438199983 0.555555555555556 60.9453959009228
1 1 1218 1622 553642.184149 4658485.52092 0.679156335654497 15.2643375224737 22.4754400734017 1.4 35.5376777919744
1 2 1239 1649 554452.184149 4657855.52092 0.4 8.48528137423857 21.2132034355964 1 45
1 3 1255 1665 554932.184149 4657375.52092 0.5 1.4142135623731 2.82842712474619 99999999999 0
2 4 1183 1500 549982.184149 4659535.52092 0.814970483200595 22.0227155455452 27.0227155455452 -0.6 120.963756532073
2 5 1154 1517 550492.184149 4660405.52092 0.642949346494168 23.4093998214393 36.4093998214393 -0.384615384615385 111.03
2 6 1121 1521 550612.184149 4661395.52092 0.178805932242459 13.1529464379659 73.5598996801217 0.131147540983607 82.5284
2 7 1087 1517 550492.184149 4662415.52092 0.0315455792522756 2 63.400325732035 0.0967741935483871 264.472459848344
2 8 1054 1509 550252.184149 4663405.52092 0.265285762568443 19.313207915828 72.8015243970933 0.352941176470588 250.5599
2 9 1027 1516 550462.184149 4664215.52092 0.226063571525868 18.4390889145858 81.5659453229323 -1.16666666666667 319.398
2 10 1009 1544 551302.184149 4664755.52092 0.013057491232981 1 76.5843899227876 -2.02941176470588 333.768060066843
2 11 988 1571 552112.184149 4665385.52092 0.239218134783986 19.8494332412792 82.9762896496257 -0.816326530612245 129.22
2 12 959 1585 552532.184149 4666255.52092 0.317830401110608 13.3416640641263 41.977306190679 -0.172413793103448 99.7824
2 13 933 1597 552892.184149 4667035.52092 0.2 1.4142135623731 7.07106781186548 -1 315
3 14 1232 1517 550492.184149 4658065.52092 0.15331120368114 3.16227766016838 20.6265268567414 0.3125 72.6459753637387
3 15 1266 1521 550612.184149 4657045.52092 0.291666666666667 14 48 -0.0285714285714286 271.636577041617
3 16 1299 1523 550672.184149 4656055.52092 0.26425765496975 24.3310501211929 92.0732083389527 0.132352941176471 262.460
3 17 1332 1528 550822.184149 4655065.52092 0.342844563425049 37.4833296279826 109.330389414727 0.152777777777778 261.31
3 18 1365 1534 551002.184149 4654075.52092 0.312002573797083 48.8364617882991 156.525829880047 0.168224299065421 260.45
3 19 1394 1549 551452.184149 4653205.52092 0.0683760683760684 11.3137084989848 165.462986797652 0.981818181818182 225.5
3 20 1399 1576 552262.184149 4653055.52092 0.0985807026093249 15.6524758424985 158.778294617449 -2.06349206349206 154.1
3 21 1387 1608 553222.184149 4653415.52092 0.122360702150498 19.6468827043885 160.56529881811 -4.05882352941176 346.159
3 22 1369 1633 553972.184149 4653955.52092 0.199241287169561 45.2769256906871 227.246703401163 -0.588607594936709 300.4
3 23 1350 1658 554722.184149 4654525.52092 0.328602928133602 44.6430285710994 135.85706257903 -3.56 344.310007070916
3 24 1337 1689 555652.184149 4654915.52092 0.405554128032969 57.8013840664737 142.524462386374 -1.68181818181818 329.26
```

*Fig. 6: Example T-vector output data*

*The model presents a number of additional output options:*

1. *Stepline.asc contains the steplines : the straight line from outlet to furthest point in the watershed. This output allows you to see whether it makes sense (for your case study) to use the furthest-from-outlet rule that is built in in this utility*
2. *Sidestepline.asc contains the sidesteplines.*
3. *Lookline.asc contains the looklines (spaced five cells apart): the lines perpendicular to stepline (and sidestepline) along which distance to both drainage divides are measured.*
4. *Interpol_midline.asc contains the interpolated midlines.*

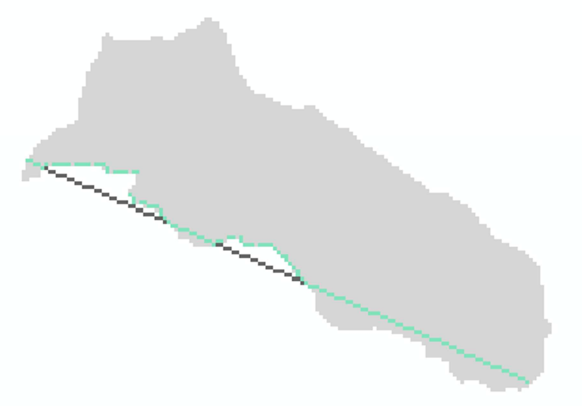


*Fig. 7: Stepline in dark gray, sidestepline in green*

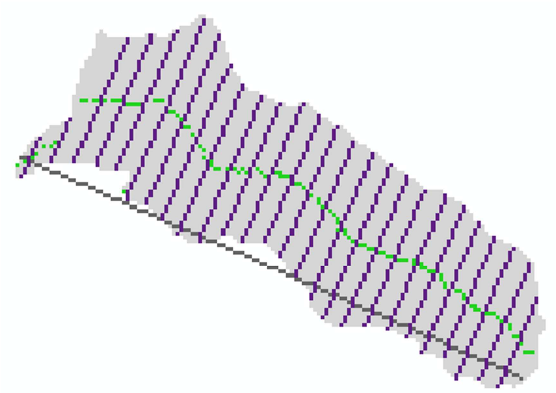


*Fig. 8: Stepline in dark gray, looklines (displayed five cells apart) in purple and midline in green*

5. *Flowline.asc contains the actual drainage line, along which segments are measured for T-vector calculation*
6. *T-lookline.asc contains the T-looklines for every segment. Perpendicular to segments, used for measuring Da and Dd.*

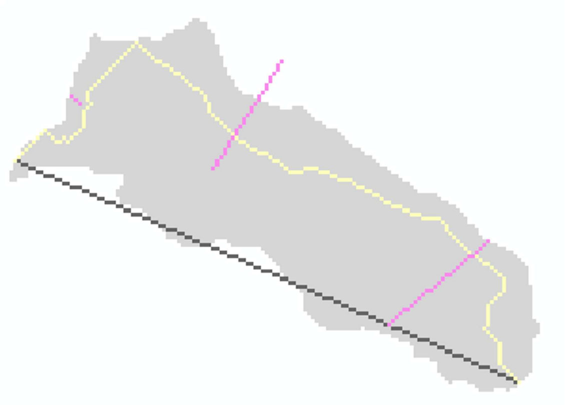


*Fig. 9: Stepline in dark gray, flowline in yellow, T-looklines in pink. In this example, the watershed contains three segments (segment lines not shown and not available). This output seems to indicate that there may be a problem with the DEM for this watershed, because flowline seems to hit the edge of the watershed before flowing to the outlet. Smaller segments may be of value in this watershed.*

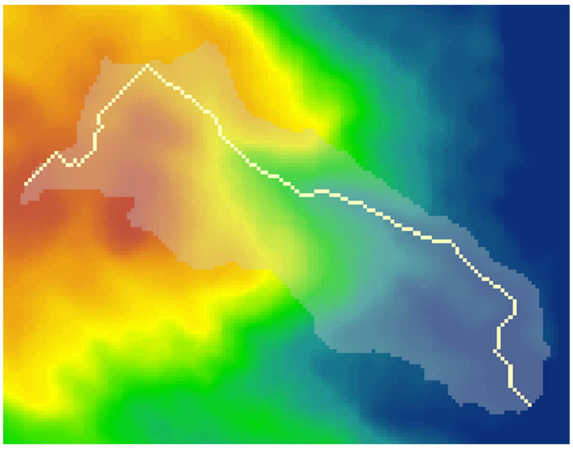


*Fig. 10: Flowline, watershed and DEM, confirming that flowline almost hits the edge of the watershed in the upper part before draining in the southeast.*