Preprocessor of 2D Wang Tiles with Elliptical Inclusions for T3D

Version 2.1

User Guide

October 26, 2019

Daniel Rypl

Czech Technical University in Prague
Faculty of Civil Engineering, Department of Mechanics

Thákurova 7, 166 29, Prague
Czech Republic

e-mail: drypl@fsv.cvut.cz
http://mech.fsv.cvut.cz/~dr/dr.html

1 Introduction

Wang2d4t3d processes a given geometry of two-dimensional Wang Tiles with elliptical inclusions subjected to generalized periodic conditions and produces input file for finite element mesh generator T3d (version 1.11 or higher). The inclusions are allowed neither to overlap nor to touch. In order to prevent too small elements (induced by the Wang tiles geometry) to appear in the finite element mesh produced by T3d, inclusions close to or intersecting tile boundary may be slightly shifted to satisfy proximity criteria with respect to tile sides and corners. Magnitude of this shifting is controlled by command line parameters and some compilation directives (see Sections 4 and 6).

2 Synopsis

Wang2d4t3d is generally executed as

`wang2d4t3d geometry_file t3d_input_file` [`epsilon` [`subdivisions`]]

where

`geometry_file` – is an obligatory parameter specifying the name of the input file describing geometry of 2D Wang tiles to be processed (see Section 3).

`t3d_input_file` – is an obligatory parameter specifying the name of the output file containing the geometry in the format of T3d finite element mesh generator.

`epsilon` – is an optional parameter specifying tolerance for the evaluation of proximity criteria of inclusion with respect to tile sides and corners (see Section 4). While positive value of epsilon is considered as an absolute tolerance, negative value of epsilon is considered as an relative tolerance with respect to tile size. If zero value of epsilon is specified or if epsilon specification is missing, the default relative tolerance is applied. If the specified or computed absolute tolerance is smaller than problem dependent tolerance, it is replaced by this problem dependent tolerance.

`subdivisions` – is an optional integer parameter (expected only if `epsilon` is specified) defining how many times `epsilon` may be halved to satisfy proximity criteria of inclusion with respect to tile edges, and corners. Missing specification is considered as zero value. Negative value should be used to prevent inclusion shifting (see Section 4).

Basic usage can be obtained by executing `wang2d4t3d` without any parameter. Basic usage together with concise description of the format of the geometry file and with the actual setting of relevant compilation directives can be obtained by executing `wang2d4t3d -h`.

3 Format of Geometry File

Empty lines and comment lines (starting by #) in the geometry file are ignored. Length of each not ignored line in the geometry file may not reach or exceed `BUFFER_SIZE` (see Section 6). The numbers in individual records (except for the problem identification record) must be separated by at least one space or tabulator.

The geometry file is organised into several sections. The first section contains a record (single line) with an arbitrary (but not empty and not starting by #) problem identification string:

`PROBLEM IDENTIFICATION STRING`

In the second section, there is a record containing the number of Wang tiles, the number of reference inclusions, and the number of side codes:

`NWT  NRI  NSC`

The number of Wang tiles `NWT` must be larger than zero and smaller than `MAX_TILES` (see Section 6). The number of reference inclusions `NRI` must be zero or larger. Note that not all inclusions must be used in processed Wang tiles. The number of side codes `NSC`, corresponding to the number of generalized periodic conditions, must be zero or larger. Note that different side codes must be considered for different directions and that not all side codes must be used in processed Wang tiles.

The third section contains a record describing the reference Wang tile:

`XTC YTC  TDX TDY  MMAT  MIMSZ MBMSZ`

The reference tile is placed in xy plane of a global right-handed Cartesian coordinate system xyz with sides aligned with the coordinate system axes. Position of the center of the reference Wang tile is defined by two coordinates `XTC` and `YTC` in the global coordinate system. The (positive) dimensions of the reference Wang tile in the directions of individual axes of the global coordinate system are defined by `TDX` and `TDY`. The identification number of the material of the matrix phase `MMAT` must be larger than zero. Specifications of the internal and boundary mesh size `MIMSZ` and `MBMSZ` of the matrix phase may not be negative. Zero mesh size specification correspond to maximal possible element size. Note that the size of actually generated finite elements may be (even significantly) smaller than the specified non-zero values depending on other mesh density control specifications.

The fourth section contains `NRI` records describing individual inclusions of (exclusively) elliptical shape:

`IID  XIC YIC  X1 Y1  MAX MIN  IIMSZ IBMSZ  IMAT IFMAT`

Absolute value of the identification number of the inclusion must be integer number in the range ⟨1;`NRI`⟩. The position and orientation of the inclusion is defined by coordinates `XIC` and `YIC` of its center and by coordinates `X1` and `Y1` of a point P_1 on its major positive half-axis. While coordinates of inclusion center are specified in the global coordinate system, coordinates of point P_1 are given either in the global coordinate system if the identification number of the inclusion is positive or in the local coordinate system, aligned with the global coordinate system and with origin at inclusion center, if the number is negative. Note that the center of the inclusion is expected to be inside or on the boundary of the reference tile. If directive `ALLOW_OUTSIDE` (see Section 6) is defined, then the center of the inclusion can be also outside of the reference tile but not by more than `epsilon`. In this case the inclusion is shifted in appropriate direction(s) by corresponding dimension of the reference tile so that the center falls inside the tile. Simultaneously, specifications of periodic repetition of all instances of this inclusion in all tiles corresponding to directions in which the shifts were performed are swapped to opposite (see the fifth section of the geometry file). The shape of the inclusion is defined by sizes of the major and minor half-axes `MAX` and `MIN`, respectively. Note that size of the major half-axis `MAX` may not be larger than minimum dimension of the tile reduced by epsilon divided by `MAX_RADIUS_RATE` (see Section 6) and that the size of the minor half-axis must be larger than 10 times `epsilon`. Specifications of the internal and boundary mesh size IIMSZ and IBMSZ of the inclusion may not be negative. Zero mesh size specification corresponds to the maximal possible element size. Note that the size of actually generated finite elements may be (even significantly) smaller than the specified non-zero values depending on other mesh density control specifications. The identification number of the material of the inclusion IMAT must be larger than zero or zero if the inclusion is a hole. Similarly, the identification number of the material of the interface between the matrix and (non-hole) inclusion IFMAT must be larger than zero or zero if there is no interface. The ordering of the inclusions is irrelevant.

If the directive ALLOW_OVERLAP is defined, the defined inclusions are allowed to touch or even overlap in the reference tile (which normally serves only as a container of individual inclusions and is not subjected to the disrectization). If the directive ALLOW_OVERLAP is not defined, while the directive CHECK_OVERLAP (see Section 6) is defined, the individual inclusions including all their periodic repetitions (see the fifth section of the geometry file) are checked against their overlapping. Note that the overlapping is identified whenever points on two inclusions are closer to each other than epsilon. Currently, there is no space search optimization adopted. This makes the check computationally demanding if large number of inclusions is involved.

The fifth section contains description of individual Wang tiles. The ordering of Wang tiles is irrelevant. For each of the NWT Wang tiles, three sets of records need to be defined.

The first set contains just a single record containing side codes of the tile, number of inclusions related to the tile, and number of control points related to the tile:

```
TID  CXN CXP CYN CYP  NTI  NCP
```

If the directive ARBITRARY_TILE_ID (see Section 6) is defined, then the identification number of the tile TID can be arbitrary non-negative integer, otherwise positive integer number in range ⟨1;NWT⟩ is expected. The codes corresponding to tile sides with outer normal in the direction of negative and positive x- and y-axes CXN, CXP, CYN, and CYP, are integer numbers in range ⟨0;NSC⟩. Non-zero code is used for matching sides of different tiles or for matching (opposite) sides of the same tile on which identical finite element mesh is required. Zero code is used for those sides that are not matching any other side of all tiles or for sides that are matching other side(s) but identical finite element mesh is not required. Two sides are considered matching if their normals are aligned with the same global coordinate system axis and if they are geometrically identical (except for a shift along that axis). If the directive CHECK_COMPATIBILITY (see Section 6) is defined, sides marked as matching are subjected to a check to verify their compatibility. Note that maching sides are not checked againts each other. Instead, each matching side is checked against the side (reference side) on which the particular code appeared in the geometry file for the first time. If the directive REPORT_MISSING_REDUNDANT (see Section 6) is defined, the list of missing and redundant inclusions with respect to the reference side is reported for each tile side violating the compatibility with the reference side. If the directive CHECK_COMPATIBILITY is not defined, only a very limited inexhaustive (and generally unsufficient) check is applied to verify the compatibility of matching sides. Note that this check is very likely to not detect compatibility violation. The number of tile inclusions NTI related to the tile can be larger than the number of reference inclusions NRI because up to four instances of the same inclusion may be related to a single tile due to the periodic repetition of the inclusion. Keep on the mind also that an inclusion touching the tile from outside (unless touching it at the tile corner while directive CHECK_COMPATIBILITY is not defined) must be included in NTI and present later in the enumeration of inclusions. Since the inclusions may be shifted to comply with proximity criteria (see Section 4), it is desirable to account also for inclusions that are outside of the tile up to a reasonable distance (typically an appropriate multiple of `epsilon`). The density of the mesh over the tile can be controlled by additional control points number of which is specified as `NCP`.

The second set contains `NTI` records enumerating all inclusions (including their periodic repetitions). The format of the records depends on the directive `SEPARATE_PER_REP` (see Section 6). If the directive is defined, the format of the record is:

```
IID   PRX PRY
```

Inclusion identification number `IID` must be in the range $\langle 1;\texttt{NRI}\rangle$. The flags of periodic repetition in $x-$ and $y-$directions `PRX` and `PRY`, being either zero or one, indicate whether that particular instance of the inclusion is or is not shifted by the dimension of the tile in the corresponding direction with respect to its reference position defined in the fourth section. Note that the shift is always performed towards the tile center. The ordering of the inclusions and their periodic repetitions is irrelevant.

If the directive `SEPARATE_PER_REP` is not defined, the record is expected in a concise format:

```
IID   PRXY
```

Specification `PRXY` is defined as `PRX` $<< 0 +$ `PRY` $<< 1$, where $<<$ stands for binary shift. Thus `PRXY` is the integer number ranging from 0 to 3.

If directive `CHECK_OVERLAP` is defined, then specified inclusions including applied periodic repetitions are checked against their overlapping. The overlapping of two inclusions is identified whenever points on these two inclusions are closer to each other than `epsilon`. Note that the check works with inclusions shifted in order to satisfy the proximity criteria (see Section 4). This is also the reason, why the check is performed even if directive `ALLOW_OVERLAP` is not defined, in which case, however, only those pairs of inclusions, in which at least one is shifted, are investigated. Currently, there is no space search optimization adopted. This makes the check computationally demanding if large number of inclusions is involved.

The third set contains `NCP` specifications of mesh density at individual control points:

```
XCP YCP  MSZ
```

The coordinates of the control point `XCP` and `YCP` in the global coordinate system may not fall outside the tile by more than `epsilon`. The required size of elements `MSZ` at the control point and in its immediate vicinity must be positive.

For an example of the geometry file see Section 7.

4 Proximity Criteria

In order to prevent undesirable mesh refinement induced by the Wang tiles or even a mesh generation failure, the inclusions close to or intersecting tile boundary may be subjected to slight shift to comply with two kinds of proximity criteria. The first proximity criterion is related to tile sides. Generally, if the inclusion is closer[1] to a tile side than `epsilon`, then the inclusion is shifted in appropriate direction (inclusion fully inside the tile is shifted towards the side, otherwise towards tile center) to enforce its touching the tile side. If directive `DETOUCH` (see Section 6) is defined, inclusion fully inside the tile (but closer to the tile side than `epsilon`) or touching the tile side is shifted towards the tile center to make its distance from the tile side exactly `epsilon`, and inclusion intersecting the tile side (but closer to the tile side than `epsilon`) is shifted outwards from the tile center to make its distance from the tile side exactly `epsilon`. If (except directive `DETOUCH`) also directive `OVERDETOUCH` (see Section 6) is defined, then inclusion touching or intersecting the tile side (but closer to the tile side than `epsilon`) is shifted towards the tile center to make its distance from the tile side exactly `epsilon`. The second proximity criterion is related to tile corners. If the intersection of the inclusion with relevant tile edges is closer to tile corner than current (possibly reduced) tolerance, the inclusion is further shifted in appropriate direction, while not exceeding the maximal shift, to pass through the tile corner or to touch tile sides. The shift is however limited in the sense that the overall shift of the inclusion with respect to its reference position may not exceed in either direction maximal shift defined as `MULTIPLE_EPS`×`epsilon` (see Section 6). If such shift does not exist, the tolerance (initially `epsilon`) is halved and a new attempt to compute an appropriate shift is made. Note that halving is applied repeatedly but not more than `subdivisions` times and also until the problem dependent tolerance is reached. If appropriate shift is not found for any of the considered tolerances, an error message is issued and the program is prematurely terminated. If an inclusion intersects a tile side at two points closer to each other than twice the problem dependent tolerance, the inclusions is subjected to shift perpendicularly to the side to either eliminate side intersection or to make the distance of the two intersection points larger than twice the problem dependent tolerance. If such shift does not exist, an error message is issued and the program is prematurely terminated. Note that the reduced tolerance is reset back to `epsilon` before other inclusion is processed. The problem dependent tolerance is influenced by the size of the complete problem, when all processed tiles are distributed in space in order to not interfere with each other when being discretized in T3d. In order to prevent overlapping of inclusions as a consequence of their shifting, it is recommended to distribute inclusions in individual tiles so that they are far enough from each other to allow for their safe shifting. This safety distance can be roughly estimated as (`MULTIPLE_EPS` + 1)×2×`epsilon`. Alternatively, the `epsilon` to be passed to wang2d4t3d can be estimated as the smallest distance between inclusions distributed in individual tiles divided by 2×(`MULTIPLE_EPS`+1). Note that the shifting can be suppressed by using negative value of the command line parameter `subdivisions`. In such a case, whenever a proximity criterion is not satisfied for given `epsilon`, an error message is issued and the program is prematurely terminated.

5 Compilation

The source code is written in plain C without dependence on any non-standard libraries. For compilation on Linux/Unix platforms, change to the directory containing wang2d4t3d.c and use (typically) command

```
gcc -O2 -o wang2d4t3d wang2d4t3d.c -lm
```

Should you prefer other compiler, replace `gcc` by the name of your preferred compiler and follow its syntax for proper specification of command line options. For compilation on Windows/Mac

[1]The distance of an inclusion from a tile side is considered as the smaller from distances between the tile side and line parallel to the side and touching the inclusion on either or other side.

platforms, create a project for console application in you favourite development environment or use you favourite command line compiler for Windows/Mac.

Note that all other source files are included in wang2d4t3d.c and should be therefore kept in a directory searched for header files (ideally in the same directory as wang2d4t3d.c). They should not be compiled separately.

6 Compilation Directives

There are several compilation directives influencing the behaviour and performance of wang2d4t3d. While some of the directives can be customized by basic users, the other should be modified by advanced user only or should not be modified at all. The default values of the directives can be obtained by executing `wang2d4t3d -h`. The directives are placed at the top of the source code (wang2d4t3d.c) and should not be therefore used as command line options of the compiler.

`MAX_RADIUS_RATE` – defines the minimum ratio between the minimum dimension of the tile reduced by `epsilon` and the size of the major half-axis. The value `MAX_RADIUS_RATE` may not be smaller than 2. Recommended value is 2.5.

`MAX_TILES` – controls the maximum number of tiles that can be processed by wang2d4t3d. It is expected that `MAX_TILES` is an integer power (at least the first power) of 10, in which case the maximum number of tiles is equal to `MAX_TILES - 1`. The larger value of `MAX_TILES`, the smaller number of inclusions intersecting individual sides of a tile are allowed. The most common value for `MAX_TILES` is 100.

`DEF_REL_EPSILON` – is the default relative tolerance. It should not exceed the maximal relative tolerance `MAX_REL_EPSILON`.

`MAX_REL_EPSILON` – is the maximal relative tolerance. It should be much smaller than 1.

`ARBITRARY_TILE_ID` – allows to number tiles arbitrarily by non-negative integer identification numbers.

`SEPARATE_PER_REP` – enforces specification of periodic repetitions of inclusions separately for each direction of global coordinate system.

`DETOUCH` – forces inclusion closer to the tile side than `epsilon` to be shifted so that its distance from the tile side is exactly `epsilon` and simultaneously the shift is minimized. If the inclusion is touching the tile side, then it is shifted inside the tile.

`OVERDETOUCH` – forces inclusion intersecting the tile side and being closer to that side than `epsilon` to be shifted inside tile so that its distance from the tile side is exactly `epsilon`. Note that directive `OVERDETOUCH` can be defined only if directive `DETOUCH` is defined.

`ALLOW_OVERLAP` – allows inclusions (including all their periodic repetitions) in the reference tile to freely overlap.

`CHECK_OVERLAP` – forces a check that inclusions (including all their periodic repetitions) in reference tile, if directive `ALLOW_OVERLAP` is not defined, do not overlap and that inclusions (including applied periodic repetitions) used in individual tiles do not overlap. The overlap is checked with respect to `epsilon`.

`ALLOW_OUTSIDE` – allows to accept also reference inclusions with center outside the reference tile but not by more than `epsilon`.

`CHECK_COMPATIBILITY` – enforces verification of compatibility of tile sides with the same nonzero code. The compatibility is checked against the side (reference side) on which the code appeared in the geometry file for the first time. It is strongly recommended to use directive REDUCED_ESTIMATE if directive CHECK_COMPATIBILITY is defined to aleviate memory demands for the compatibility check.

REPORT_MISSING_REDUNDANT – reports the list of missing and redundant inclusions (with respect to the reference side) for each matching tile side violating the compatibility check. Note that directive REPORT_MISSING_REDUNDANT can be defined only if directive CHECK_COMPATIBILITY is defined. Note that the information about missing and/or redundant inclusions may be contradictive. This happens for the inclusions containing or passing through a tile corner. To indicate such a situation, question mark is added to the report.

CHECK_ISOLATED – checks for setups resulting in isolated nodes at tile corners in the generated finite element mesh. Generally these nodes must be present in the mesh to allow for tile mesh assembly after which these nodes usually become not isolated. In some cases, however, these nodes may remain isolated even in the assembled mesh and should be removed by postprocessing. If such setup is identified, a warning is issued.

USE_TILE – forces that each t3d model entity (except virtual ones) refers to a particular tile. This directive must be defined.

USE_SIDE – forces explicit side specification for t3d patches on tile sides. This directive may not be defined.

USE_CODE – forces explicit code specification for t3d tiles. This directive may not be defined.

USE_SIMPLE – forces simple mirroring between sides of tiles with the same code in order to override discrepancies due to round-off error or numerical instabilities. This directive may not be defined.

MULTIPLE_EPS – defines the multiple of the epsilon up to which the shift of the inclusion in each particular direction is performed when resolving intersection of the inclusion with tile sides and corners. The value MULTIPLE_EPS may not be smaller than 1. Recommended value is 2. Note that value less than 2 may prevent overdetouching (see directive OVERDETOUCH).

SETUP – controls the way how the individual tiles are arranged in space in order to not interfere with each other when being discretized in T3d. The possible values are GRID resulting in distribution of tiles in a regular 2D grid (if there are not enough tiles to fill the grid fully, some of its entries will remain empty) or DIAGONAL, in which case the tiles are distributed along diagonal of a square. Obviously, the setting of SETUP is irrelevant if only one tile is processed. Note that no matter whether GRID or DIAG is used, the resulting meshes of individual tiles are placed in the position of the reference tile. Keep in mind however, that the resulting meshes are generally different depending on the setting of SETUP. Note also that the GRID (DIAGONAL) setup results in the smallest (largest) problem dependent tolerance.

REDUCED_ESTIMATE – enables better estimate of maximal number of inclusions in a tile and maximal number of inclusions intersecting or touching tile sides by accounting for overlapping of inclusions and overlapping of intersections of inclusions with tile sides in a global sense. It is strongly recommended to use this directive to aleviate memory demands for huge problems with overlaping (reference) inclusions. The performance of the directive REDUCED_ESTIMATE is controlled by directives INTERVALS_2D, INTERVALS_1D, INTERVAL_RATE_2D, INTERVAL_RATE_1D, SERIES_2D and SERIES_1D - see below.

INTERVALS_2D – defines in how many intervals to devide the area of the tile. Minimal values is 10.

INTERVALS_2D – defines in how many intervals to devide the length of individual sides of the tile. Minimal values is 10.

INTERVALS_RATE_2D – defines the ratio between the size of the last (largest) and the first (smallest) 2D interval. The minimal values is 1.

INTERVALS_RATE_1D – defines the ratio between the size of the last (largest) and the first (smallest) 1D interval. The minimal values is 1.

SERIES_2D – defines which series is used for 2D intervals. The possible values are CONSTANT_SERIES, ARITHMETIC_SERIES and GEOMETRIC_SERIES, standing for intervals of equal size, intervals of size forming arithmetic series and intervals of size forming geometric series, respectively. Note that constant series can be prescribed also as arithmetic or geometric series with the ratio of the size of the last (largest) and the first (smallest) interval set to 1.

SERIES_1D – defines which series is used for 1D intervals. The possible values are CONSTANT_SERIES, ARITHMETIC_SERIES and GEOMETRIC_SERIES, standing for intervals of equal size, intervals of size forming arithmetic series and intervals of size forming geometric series, respectively. Note that constant series can be prescribed also as arithmetic or geometric series with the ratio of the size of the last (largest) and the first (smallest) interval set to 1.

MIN_MAX_HALVING – allows to split inclusion also with respect to the major and minor axes. The feature is currently not fully implemented and should not be used.

EXCESS_SIZE – defines in which way the size by which inclusions may exceed the tile is considered. The possible values are MAX_ALLOWED_SIZE, assuming maximal possible inclusion size (see directive MAX_RADIUS_RATE), and MAX_PRESENT_SIZE, assuming the largest (in terms of the size of the major axis) reference inclusion no matter whether the inclusion is used in individual tiles or whether it is intersecting the tile side. Note that MAX_PRESENT_SIZE setting results in smaller problem dependent tolerance.

OFFSET – defines the relative distance between tiles (with respect to the minimal tile dimension) to ensure that the individual tiles arranged in the space do not interfere with each other when being discretized in T3d. The value of OFFSET should be in range $\langle 0.5;1.0 \rangle$. Recommended value is 0.5. Note that the distance between tiles is further enlarged to account for inclusions exceeding the tiles. Although the part of the inclusion outside the tile is obviously not subjected to the discretization, it may influence the mesh density control space. The real distance between the tiles may be further increased due to arrangement of tiles in space which must comply with certain criteria to minimize the dispersion of specified mesh density along the corresponding pairs of matching sides.

BUFFER_SIZE – defines the size of the input line buffer. Recommended value is 1024.

7 Example of Geometry File

```
Arbitrary comment

# Overall numbers
2 18 4

# Reference tile
0 0  1 1  1  0.5 0.2

# Reference inclusions
 1  0.320 -0.475  1.257 -0.124 0.084 0.073 0.2 0.1  0 0
 2  0.283  0.471 -0.045  1.415 0.149 0.105 0.2 0.1  0 0
 3 -0.443  0.054  0.471  0.460 0.119 0.111 0.2 0.1  0 0
 4 -0.146 -0.273  0.790 -0.621 0.106 0.081 0.2 0.1  0 0
 5 -0.303  0.451  0.686  0.593 0.144 0.102 0.2 0.1  0 0
```

```
 6 -0.143  0.162  0.930  0.390 0.081 0.061 0.2 0.1 0 0
 7  0.101 -0.252 -0.496  0.548 0.084 0.067 0.2 0.1 0 0
 8  0.255 -0.032  0.804  0.803 0.083 0.074 0.2 0.1 0 0
 9 -0.138  0.058  0.861 -0.099 0.099 0.070 0.2 0.1 0 0
10  0.218  0.254  1.217  0.102 0.115 0.062 0.2 0.1 0 0
11 -0.458  0.041  0.535  0.152 0.083 0.076 0.2 0.1 0 0
12 -0.365 -0.470  0.591 -0.178 0.093 0.078 0.2 0.1 0 0
13  0.211 -0.110  1.209 -0.228 0.077 0.055 0.2 0.1 0 0
14  0.469 -0.230  0.657  0.752 0.095 0.074 0.2 0.1 0 0
15 -0.020  0.246  0.953 -0.080 0.089 0.070 0.2 0.1 0 0
16 -0.113 -0.185  0.828  0.298 0.076 0.068 0.2 0.1 0 0
17 -0.013  0.499 -0.600  1.309 0.087 0.062 0.2 0.1 0 0
18 -0.479 -0.252 -0.697  0.723 0.085 0.074 0.2 0.1 0 0

# Tile 1
 1  1 1 3 4  13 0
 1  2
 2  2
 3  0
 3  1
 5  2
 6  0
10  0
12  2
13  0
16  0
17  0
18  0
18  1

# Tile 2
 2  2 2 4 3  14 0
 1  0
 2  0
 4  0
 5  0
 7  0
 8  0
 9  0
11  0
11  1
12  0
14  0
14  1
15  0
17  2
```

For the above example to work, it is expected that directive SEPARATE_PER_REP is not defined and MAX_RADIUS_RATE is set to 2.5.