DIVISION OF COMPUTER AIDED DESIGN
FACULTY OF CIVIL ENGINEERING AND ENVIRONMENTAL ENGINEERING
POZNAŃ UNIVERSITY OF TECHNOLOGY

MASTER’S THESIS, PART I

DESCRIPTION OF DEVELOPED ALGORITHMS AND PROGRAM

Adrian Kowalczewski

Thesis promoter:
Ph. D. Krzysztof Szajek

“ ⎔ The raw maths or patterns that make everything.”

Dziękuję i dedykuję Mamie, Tacie, Marcie I Panu Krzysztofowi Szajkowi.

Contents

1. The opening

1.1. Introduction

Connection is a necessary element of every structure, every type of construction requires designing connections – furthermore – very often they are crucial parts of construction. Even when every element of construction is designed for 10% of its capacity, when connections of this element are designed in the wrong way, the whole structure will most likely collapse, as the strength of the structure will be determined by the strength of connections.

Aim of the work is making computer program – in a form of add-on to program capable of calculating statics in 2D constructions – which will calculate the capacity of timber connections with metal fasteners - such as nails or bolts – with respect to European norm, EN 1995.

In a matter of designing connections, European norm leaves relatively large freedom of connection modelling for designer. Connection should satisfy spacing requirements (to prevent brittle failure) and connectors shear capacity should be greater than forces acting on them. Other properties of connection – such as - be way of connecting elements – directly, or using indirect solutions such as metal plates – are left for designer.

Interest in both civil engineering and programming, will to improve own skills, and need for computer programs calculating engineering tasks were leading factors for choice of presented thesis topic.

1.2. Aim and scope of work

Aim of the work is making computer application named "Timber connection", used for designing rectangular cross-section timber elements connections, with metal fasteners, in any two dimensional state of stress. Scope of topic include:

- Designing algorithms based on EN 1995-1-1, allowing design of timber connections.
- Implementation of algorithm in environment of Soldis Projektant (www.soldis.com.pl)
- Making this coverage, describing work of program with its manual
- Calculating project of timber hall, involving use of developed program

2. Symbols used

Latin upper case letters

$F_{ax,Ed}$ Design axial force on fastener

$F_{ax,Rd}$ Design value of axial withdrawal capacity of the fastener

$F_{ax,Rk}$ Characteristic axial withdrawal capacity of the fastener

F_d Design force

$F_{M,Ed}$ Design force from a design moment

$F_{v,0,Rk}$ Characteristic load-carrying capacity of a connector along the grain

$F_{v,Ed}$ Design shear force per shear plane of fastener

$F_{v,Rd}$ Design load-carrying capacity per shear plane per fastener

$F_{v,Rk}$ Characteristic load-carrying capacity per shear plane per fastener

M_d Design moment

$M_{y,Rk}$ Characteristic yield moment of fastener

$F_{90,d}$ Design splitting capacity

$F_{90,k}$ Characteristic splitting capacity

$F_{ax,d}$ Design load-carrying capacity of an axially loaded connection

$F_{ax,k}$ Characteristic load-carrying capacity of an axially loaded connection

Latin lower case letters

a_1 Spacing, parallel to grain, of fasteners within one row

a_2 Spacing, perpendicular to grain, between rows of fasteners

$a_{3,t}$ Distance between fastener and loaded end

$a_{3,c}$ Distance between fastener and unloaded end

$a_{4,t}$ Distance between fastener and loaded edge

$a_{4,c}$ Distance between fastener and unloaded edge

$a_{1,lim}$ Required minimal spacing, parallel to grain, of fasteners within one row

$a_{2,lim}$ Required minimal spacing, perpendicular to grain, between rows of fasteners

$a_{3,t,lim}$ Required minimal distance between fastener and loaded end

$a_{3,c,lim}$ Required minimal distance between fastener and unloaded end

$a_{4,t,lim}$ Required minimal distance between fastener and loaded edge

$a_{4,c,lim}$ Required minimal distance between fastener and unloaded edge

b Width of element

d Diameter of connector

d_h Head diameter of connector

$f_{h,l,k}$ Characteristic embedment strength of timber member *i*

$f_{ax,k}$ Characteristic pointside withdrawal strength

$f_{h,k}$ Characteristic embedment strength

$f_{head,k}$ Characteristic pull through parameter for nails

$f_{t,0,k}$ Characteristic tensile strength along the grain

$f_{u,k}$ Characteristic tensile strength

h Height of element

k_{mod} Modification factor for duration of load and moisture content

n_{ef} Effective number of fasteners

t Thickness

t_{pen} Penetration depth

Greek lower case letters

α Angle between a force and the direction of grain

α_{el} Angle between elements

γ_M Partial factor for material properties

3. Timber connections

3.1. Introduction

All connection methods in Eurocode 1995 are realized with metal type fasteners, which are hammered/driven/inserted perpendicularly to the elements, inside their overlap area. Depending on angle between the elements, this area will be rectangle or rhomboid. There are many types of dowel-type fasteners described in norm, those include: nails, square nails, bolts, screws, dowels, round staples and square staples. Each of them have different properties and formulas used for calculating coefficients, but after calculating coefficients one final algorithm applies for all connectors, thus enabling relatively easy comparison between their capacity. All rules for dowel-type fasteners connections described in Eurocode concern connections in compression or tension, with possible shear. Rules for connections subjected to bending moment were derived in [2].

3.2. Calculations according to EN 1995

Calculation of connection capacity comes down to checking two conditions:

1. Comparison of effective design load-carrying capacity of all rows of fasteners to axial force in connection.

$$\frac{F_{Ed} = N_{Ed}}{F_{v,ef,Rd}} \leq 1$$

2. Comparison of design splitting capacity to design shear force

$$F_{v,Ed} \leq F_{90,Rd}$$

While splitting check comes down to calculating $F_{90,Rd}$ from formula, calculation of effective design load-carrying capacity of all connectors, $F_{v,ef,Rd}$ involves many steps and formulas.

3.3. Bending and shearing of connections

Calculation of connections subjected to bending moment share some steps and idea of calculation bended connections in steel structures (EN 1993-1-8). Maximal force acting on the most fatigued connector is calculated and compared to connector capacity.

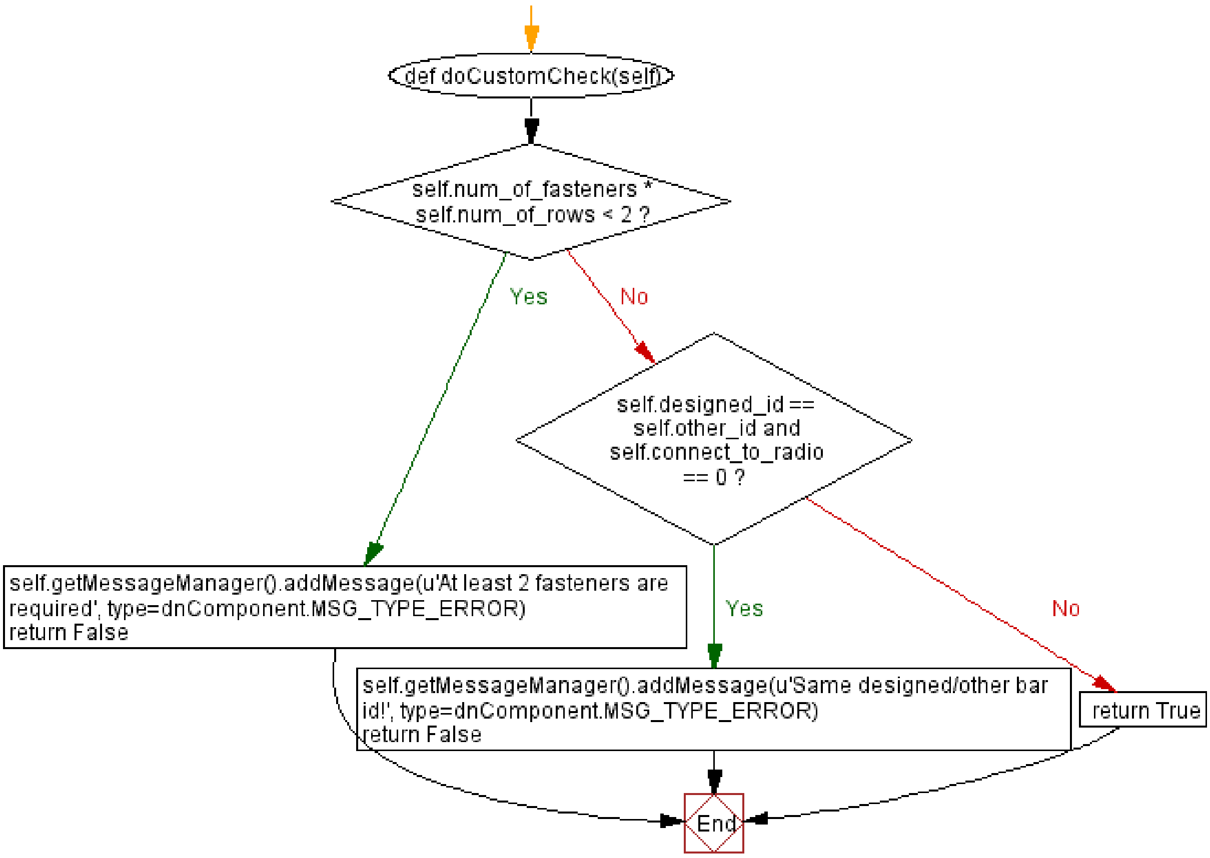
$$F_d \leq F_{v,Rk}$$

Single connector load-carrying capacity is calculated using formulas described in EN 1995, while formulas for calculating F_d are described in [2]. Greater spacing requirements are also introduced.

4. Calculation algorithm, developed solutions

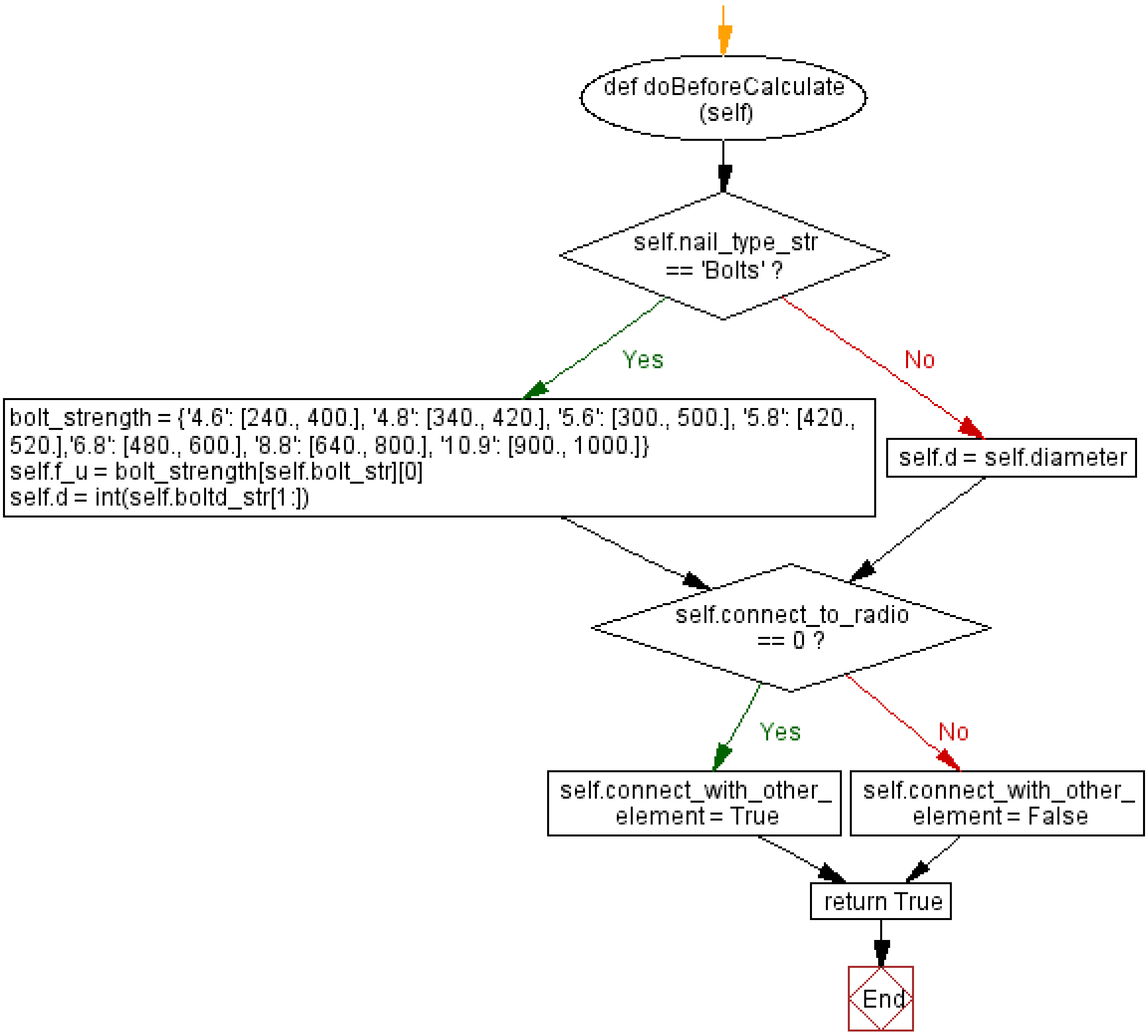
4.1. Data preparation, necessary checks

Before proceeding to calculations, program performs series of essential checks, both in domain of Soldis Projektant environment and in domain of Norm.



Algorithm 1

Above algorithm defined in doCustomCheck function, executed before main algorithm, checks if connection consists of at least 2 fasteners (Norm requirement) and if selected bars for connection design aren't the same bar. If those checks are false, program raises corresponding error.



Algorithm 2

Another algorithm executed before main algorithm prepares some data for further use in component class.

If user selected Bolts as a connector type, program should read bolt diameter and class from combo box, which replaces regular control for diameter and connector length, when bolt type connector is picked.

Another check is whether user picked to connect bar to the other bar or to calculate indirect connection, involving use of steel plates. Depending on check result, extensively used global variable connect_with_other_element will be set True or False.

4.2. Main algorithm

4.2.1. Bar angles and pairing

Algorithm 3 is first part of main algorithm. It gets object list and performs certain tasks related to determining proper bar angles and bar pairs. Steps performed by algorithm: 1. as bar angle is different for its beginning and end (figure below), dictionary consisting of proper angles (as every bar would begin in considered node) is made.

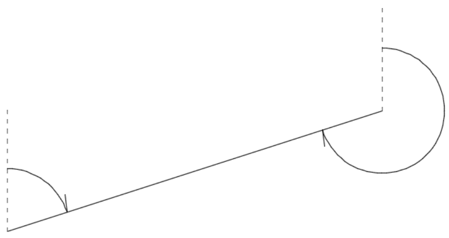


Figure 1

2. Determining bar pairs - if true angle between elements is 0 or $180^0 \pm 2^0$, bars are considered to be paired, what means they belong to one element. 3. Determining unpaired bars - done by excluding paired bars from all bars list. 4. Checking, if there's any bar belonging to more than one pair, what would mean wrongly drawn geometry (Bars invading each other). 5. Finding user selected bars on all bars list, assigning them to ele_0 (Designed bar) and ele_1 (Bar bearing the connection)

4.2.2. Elements widths and shear planes

Algorithm 4 performs tasks related to determining elements thickness and shear planes. It consists of 2 routes, one for connection with other element, second for indirect connection with steel plate. Steps performed by algorithm:

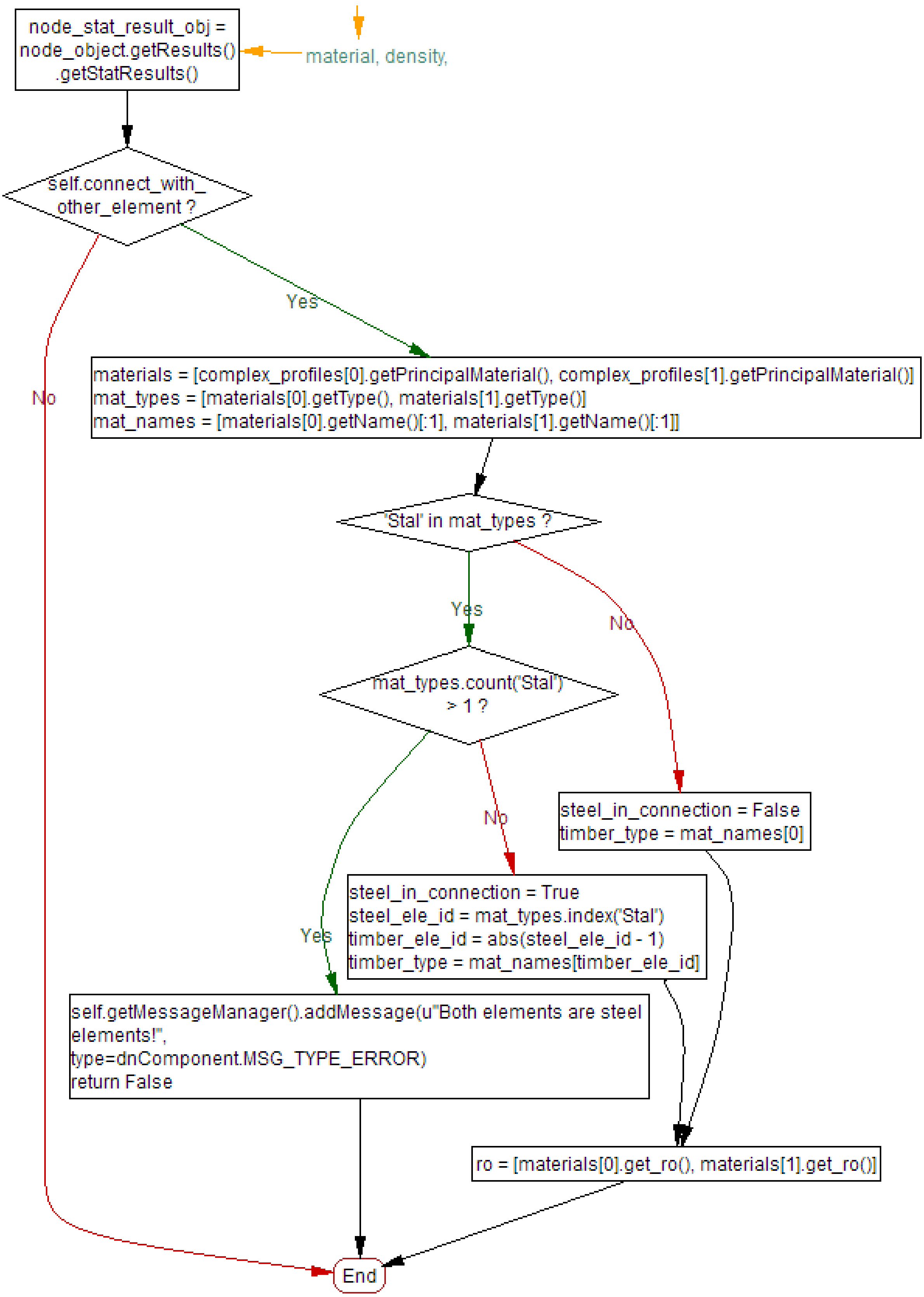
Route 1, connection with other element. 1. Assume simplest connection i.e. single shear connection. 2. If ele_0 or ele_1 profile count is greater than 1, assume that it is connection with two shear planes. Save which profile is two profiled, for further use in algorithm. 3. If both ele_0 and ele_1 consist of two profiles, return error, as it is invalid connection. 4. If profiles of two profiles element are different – return error, as such connection is inconsistent with norm. 5. If there is one shear plane, save profiles widths as t1 and t2. If there are two shear planes, save two profiles element branch width as t1 and one profile element width as t2.

Route 2, connection with steel plate. 1. If designed element is one profile element, assume plates are joined outside. If designed element is two profile element, assume one plate inside and calculate its width basing on space between element branches. 2. If user selected to design plate inside pocket drilled in one profile element (also called "concealed plate"), assume that designed element is now two profile element with steel plate inside. 3. save two profile branch width as t1, profile inside width as t2.

Last part of algorithm checks, if the space between elements in two shear plane connection is in range from 0 to 10 mm, and if any profile have more than 2 branches.

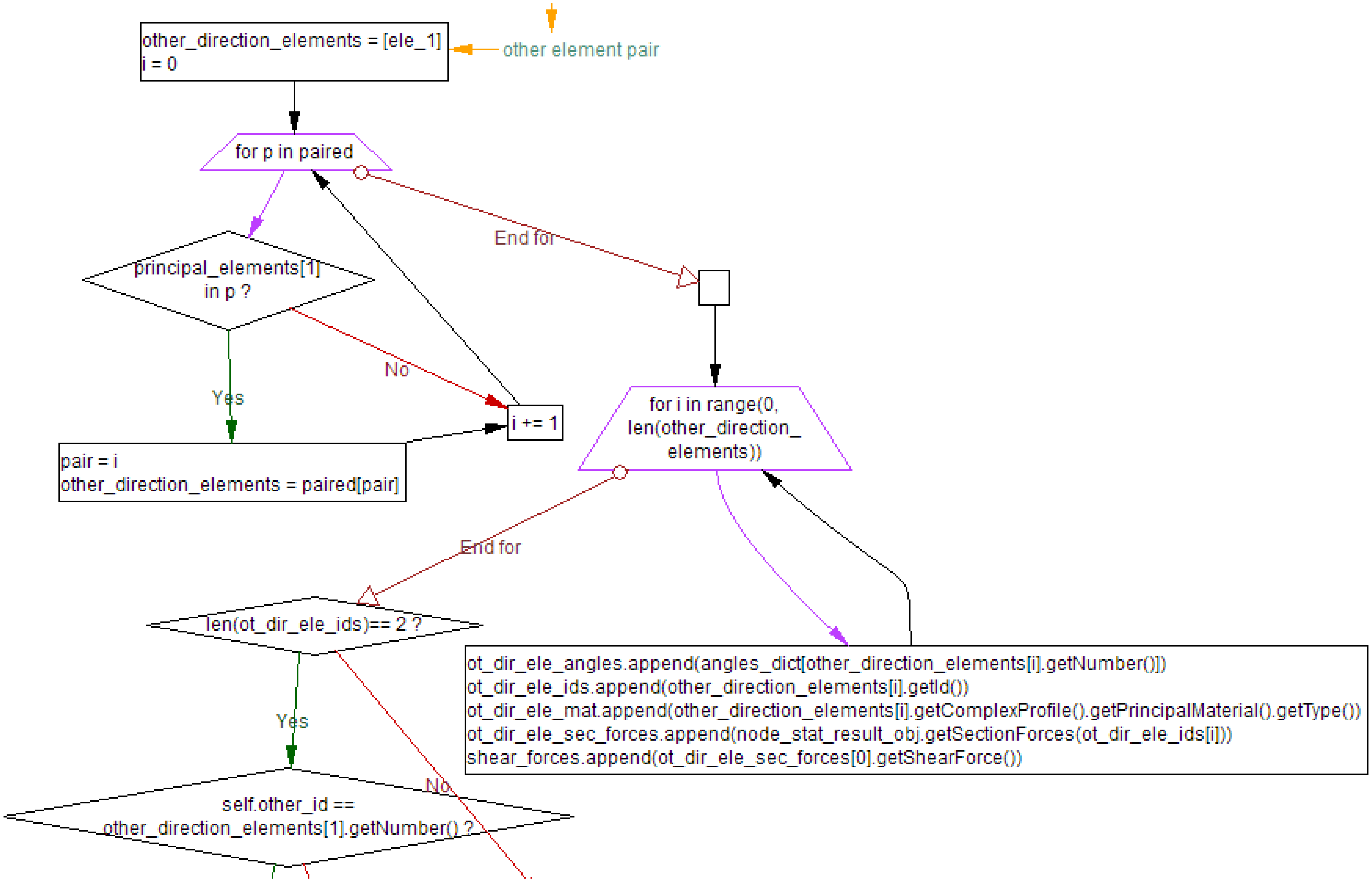
4.2.3. Connection with other bar data preparation

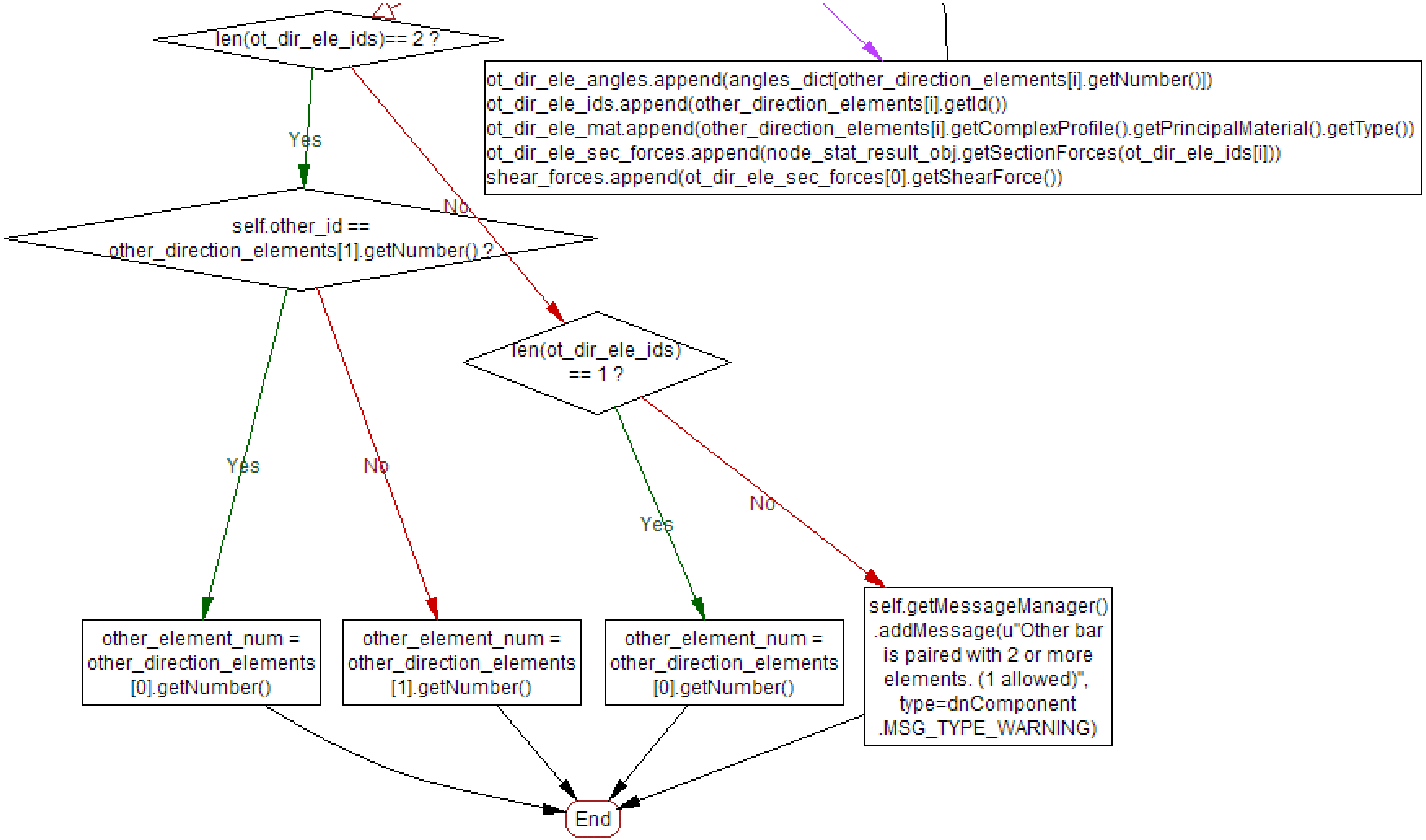
For clarity of the work this section is divided into 3 parts which are one algorithm.



Algorithm 3

Algorithm 5 gets material object from connection elements, and sets variables for further use in main algorithm. If both materials are steel, error is returned.

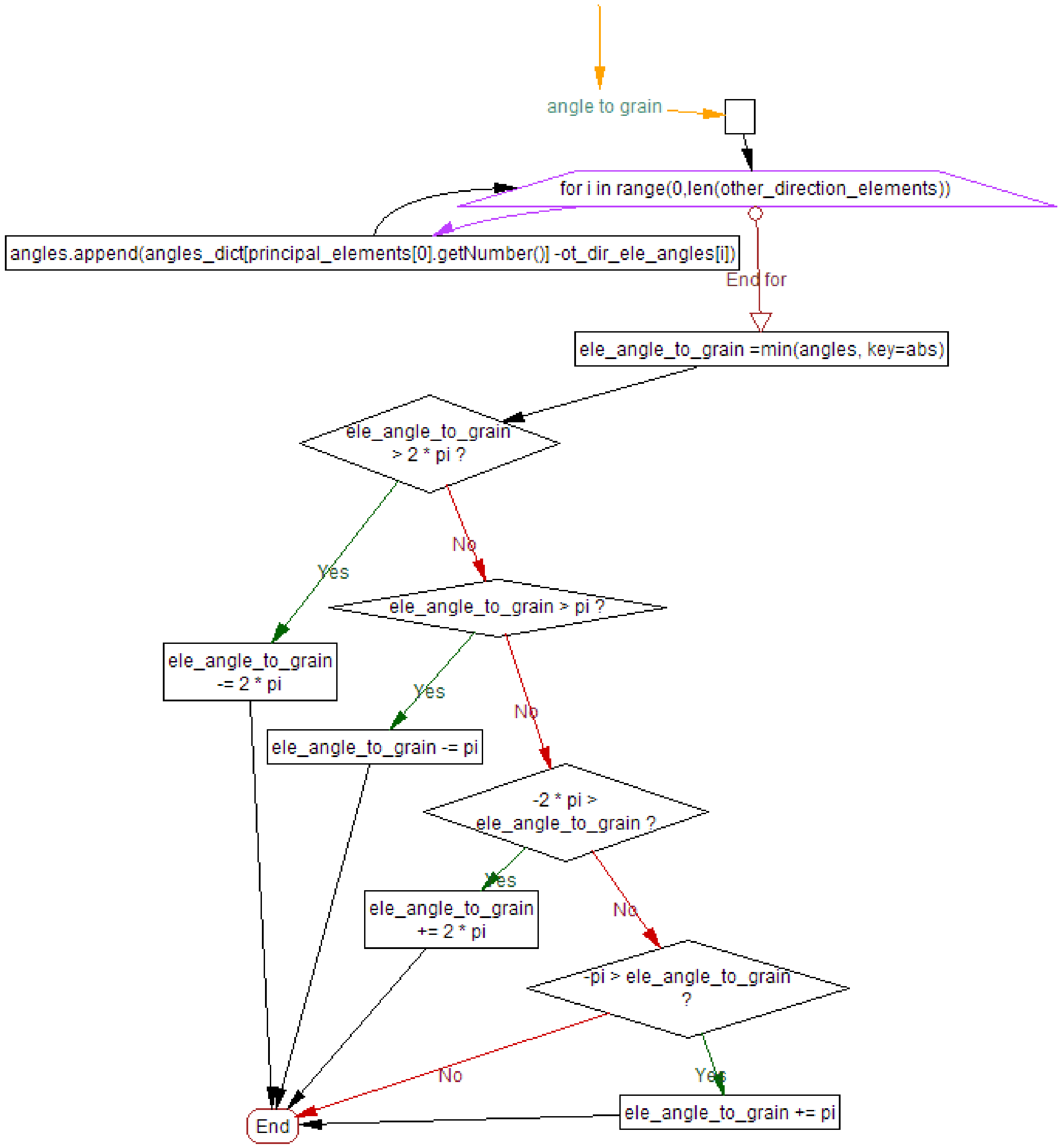

other_direction_elements = [ele_1]
i = 0
other element pair
for p in paired
End for
principal_elements[1]
in p ?
No
Yes
i += 1
pair = i
other_direction_elements = paired[pair]
for i in range(0,
len(other_direction_
elements))
End for
len(ot_dir_ele_ids)== 2 ?
ot_dir_ele_angles.append(angles_dict[other_direction_elements[i].getNumber()])
ot_dir_ele_ids.append(other_direction_elements[i].getId())
ot_dir_ele_mat.append(other_direction_elements[i].getComplexProfile().getPrincipalMaterial().getType())
ot_dir_ele_sec_forces.append(node_stat_result_obj.getSectionForces(ot_dir_ele_ids[i]))
shear_forces.append(ot_dir_ele_sec_forces[0].getShearForce())
Yes
No
self.other_id ==
other_direction_elements[1].getNumber() ?



len(ot_dir_ele_ids)== 2 ?
ot_dir_ele_angles.append(angles_dict[other_direction_elements[i].getNumber()])
ot_dir_ele_ids.append(other_direction_elements[i].getId())
ot_dir_ele_mat.append(other_direction_elements[i].getComplexProfile().getPrincipalMaterial().getType())
ot_dir_ele_sec_forces.append(node_stat_result_obj.getSectionForces(ot_dir_ele_ids[i]))
shear_forces.append(ot_dir_ele_sec_forces[0].getShearForce())
Yes
No
self.other_id ==
other_direction_elements[1].getNumber() ?
len(ot_dir_ele_ids)
== 1 ?
Yes
No
Yes
No
other_element_num =
other_direction_elements
[0].getNumber()
other_element_num =
other_direction_elements
[1].getNumber()
other_element_num =
other_direction_elements
[0].getNumber()
self.getMessageManager()
.addMessage(u"Other bar
is paired with 2 or more
elements. (1 allowed)",
type=dnComponent
.MSG_TYPE_WARNING)
End

Algorithm 4

Algorithm 6 determines whether bar bearing the connection ("other bar") has pair (by iterating paired list, made in algorithm 3 and searching other bar object in each returned pair tuple), and if it is true, it gets shear forces on both selected bearing element and its pair. Shear forces on those elements are used for checking splitting in connection in algorithm 25.



Algorithm 5

Algorithm 7 calculates other element grain direction angle to designed element grain direction. If other element has pair, lesser angle will be picked. Angle to grain direction is always less than 180^0 (π). Idea of angle to grain direction presented on figure below.

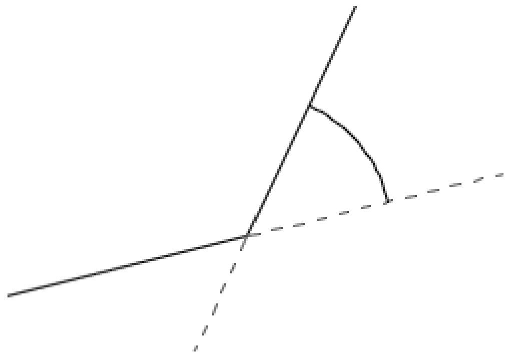
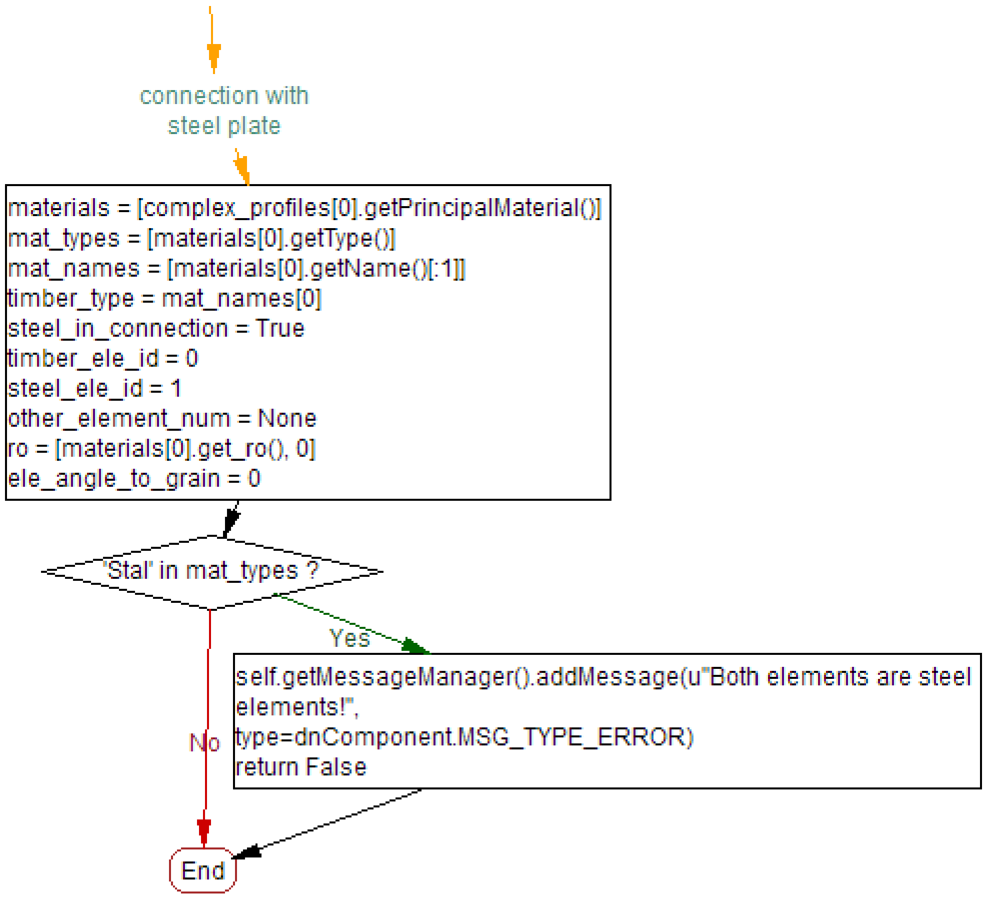


Figure 2

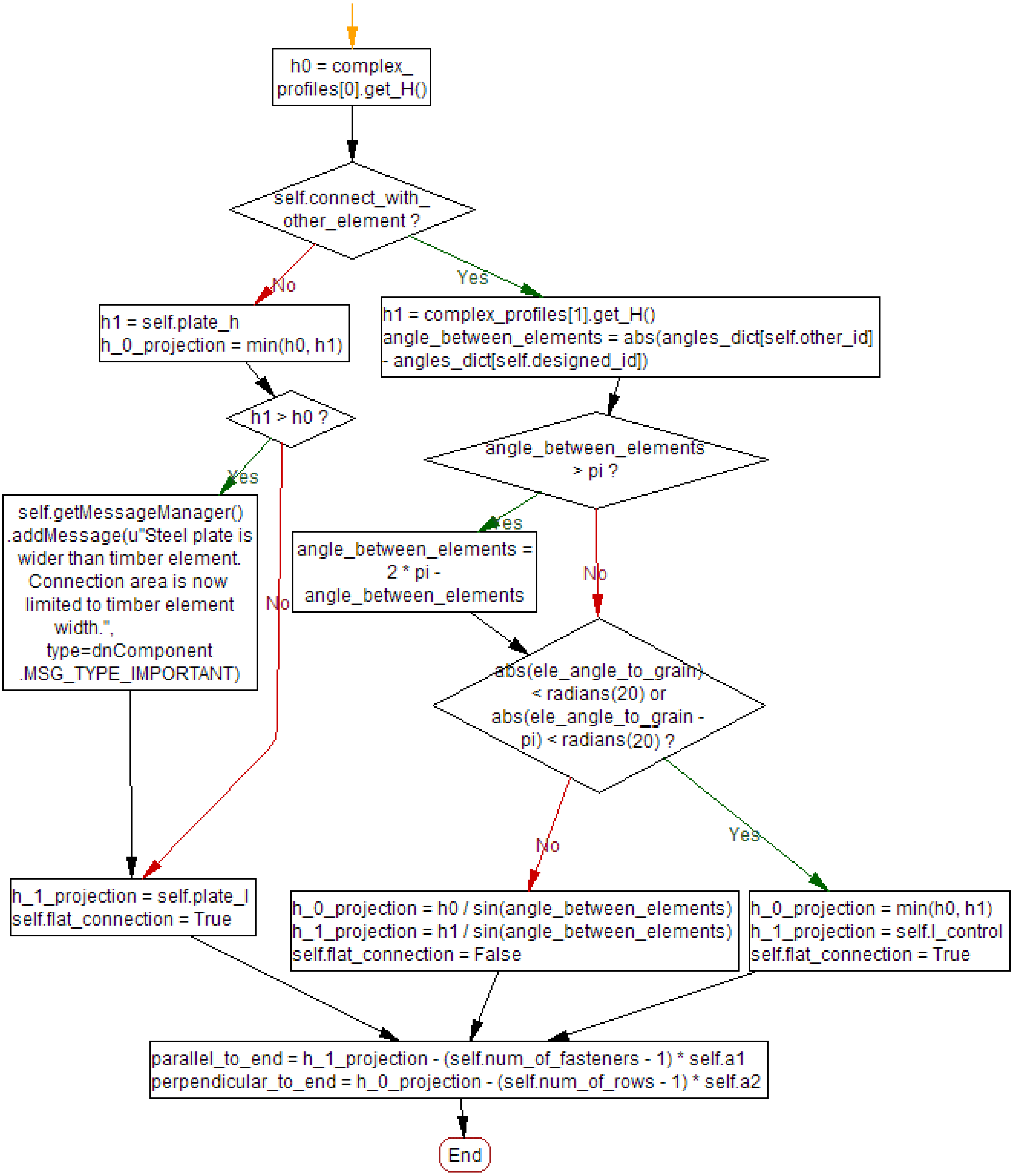
4.2.4. Connection with steel plate data preparation



Algorithm 6

Algorithm 8 prepares data in the same manner as Algorithms 5, 6, 7. Most variables are equal 0 or are independent, as designed element is always made of timber and angle between plate and designed element is 0.

4.2.5. Calculation of area of connection



Algorithm 7

Algorithm 9 calculates area of connection. If connection with other element is selected, area will be calculated by determining elements overlapping area (presented on figure below)

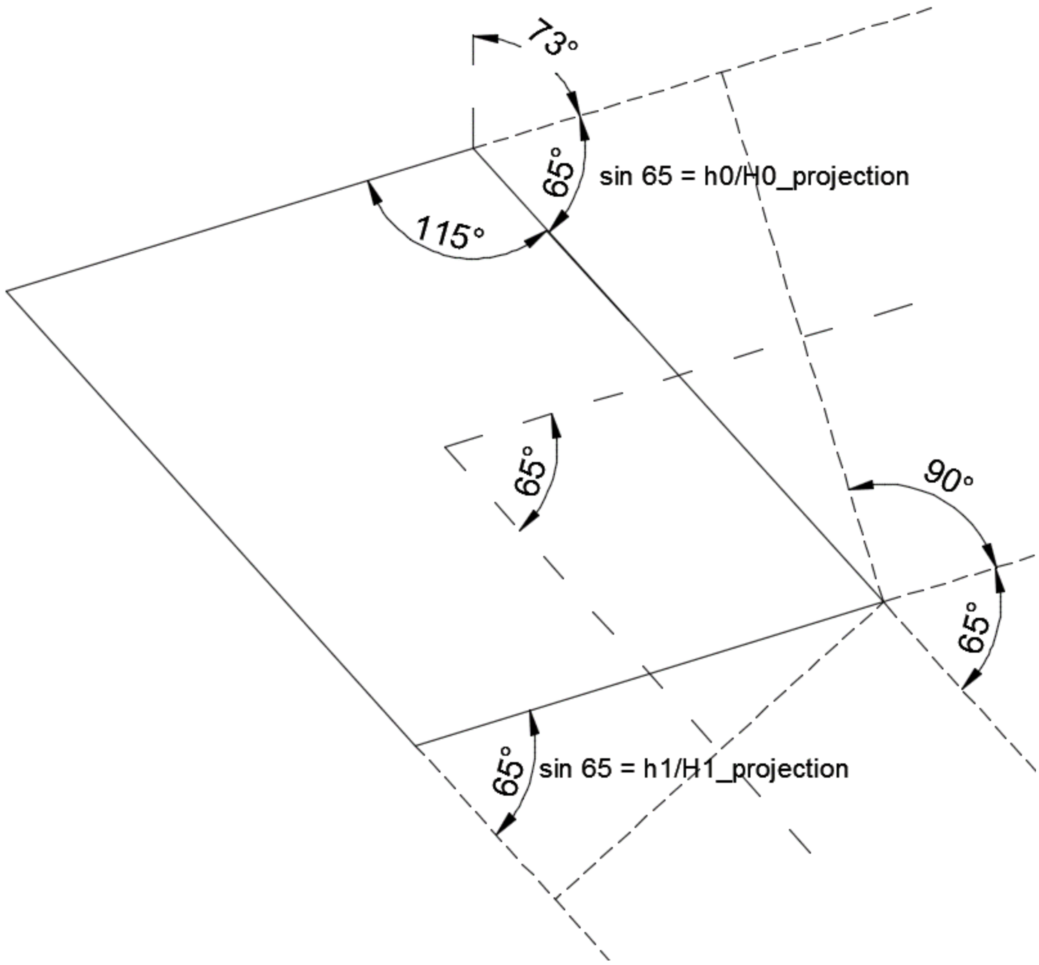


Figure 3

If connection is flat, connection area will equal A = min (h1, h2) ∗ user defined connection overlapping length (∗ plate length for connection with steel plate).

Next step of algorithm is calculation of distances to the end / edge of connection area. Graphical presentation on figure below.

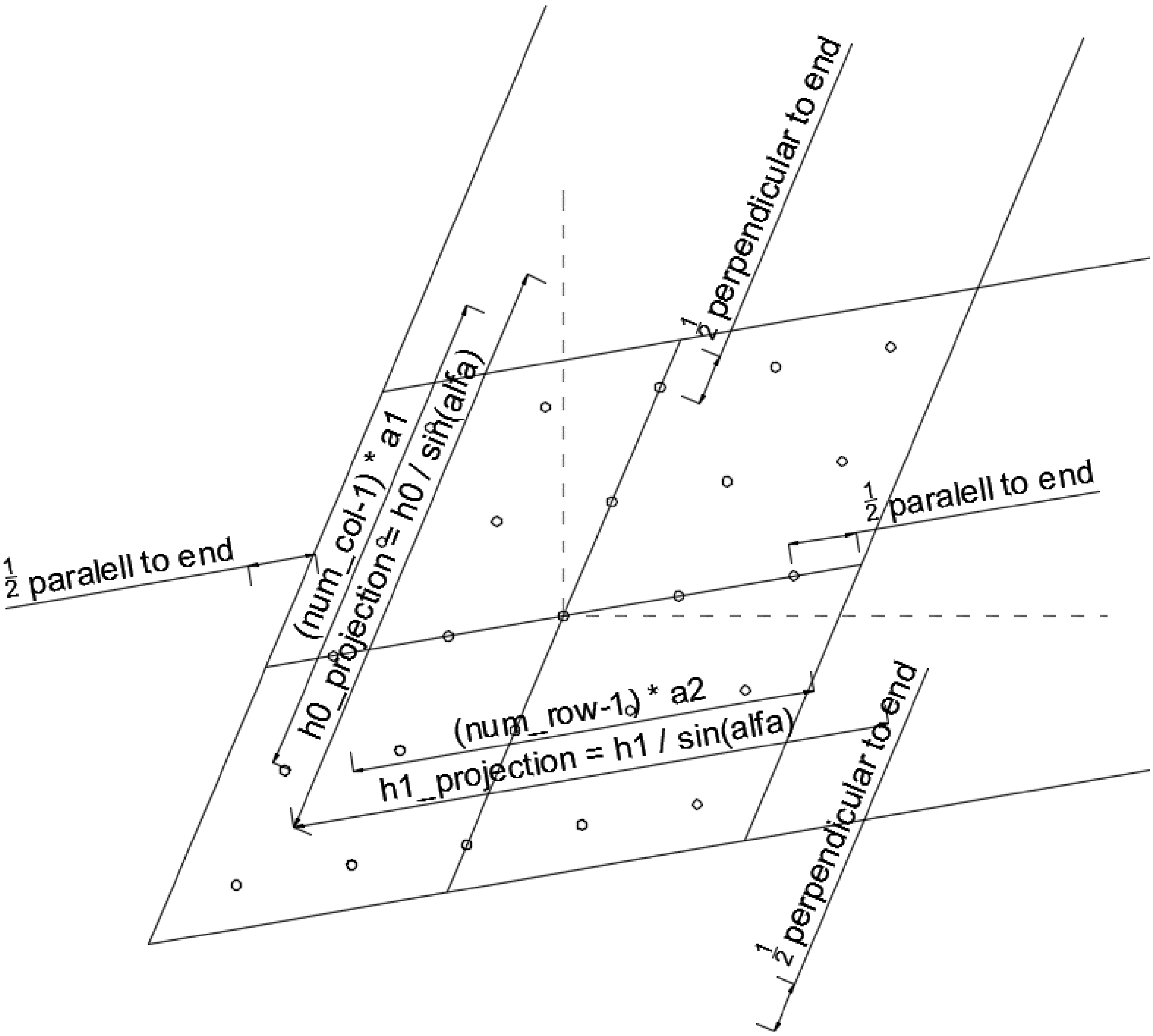
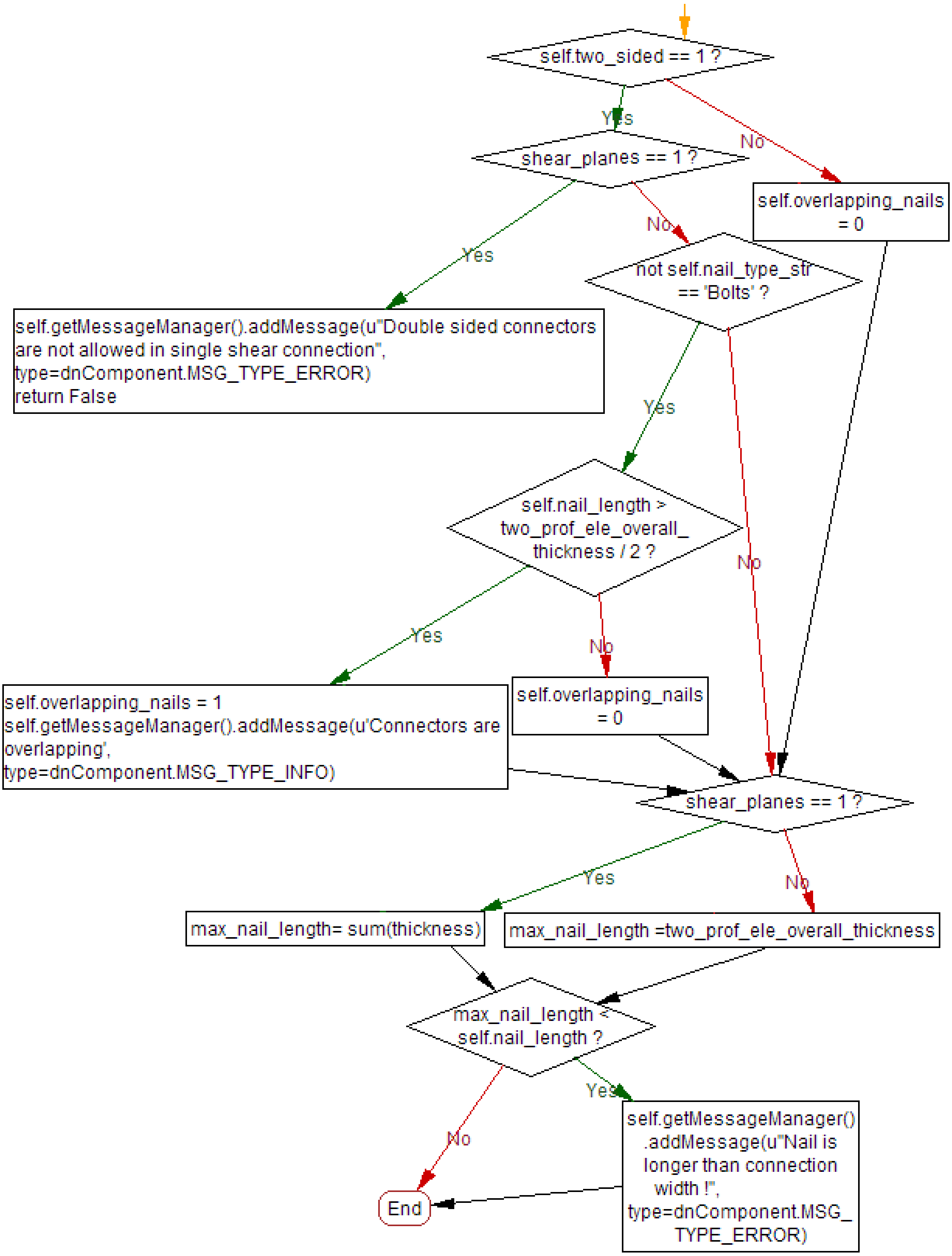


Figure 4

4.2.6. Connector length & overlapping check



Algorithm 8

Algorithm 10 determines, whether connectors are overlapping (possible with checking double sided connectors when there are two shear planes) and if connector length does not exceed element overall thickness.

4.2.7. Calculation of coefficients for Nails & Screws ($d \leq 6\ mm$)

For clarity of the work and easier description this section is divided into 5 parts which are one algorithm. Other connectors algorithms are presented in one continuous algorithm.

Algorithm 11 (below) calculates t_1 and t_2 for connection. According to norm:

- t_1 is:

  - The nail headside material thickness where the connection is in single shear.

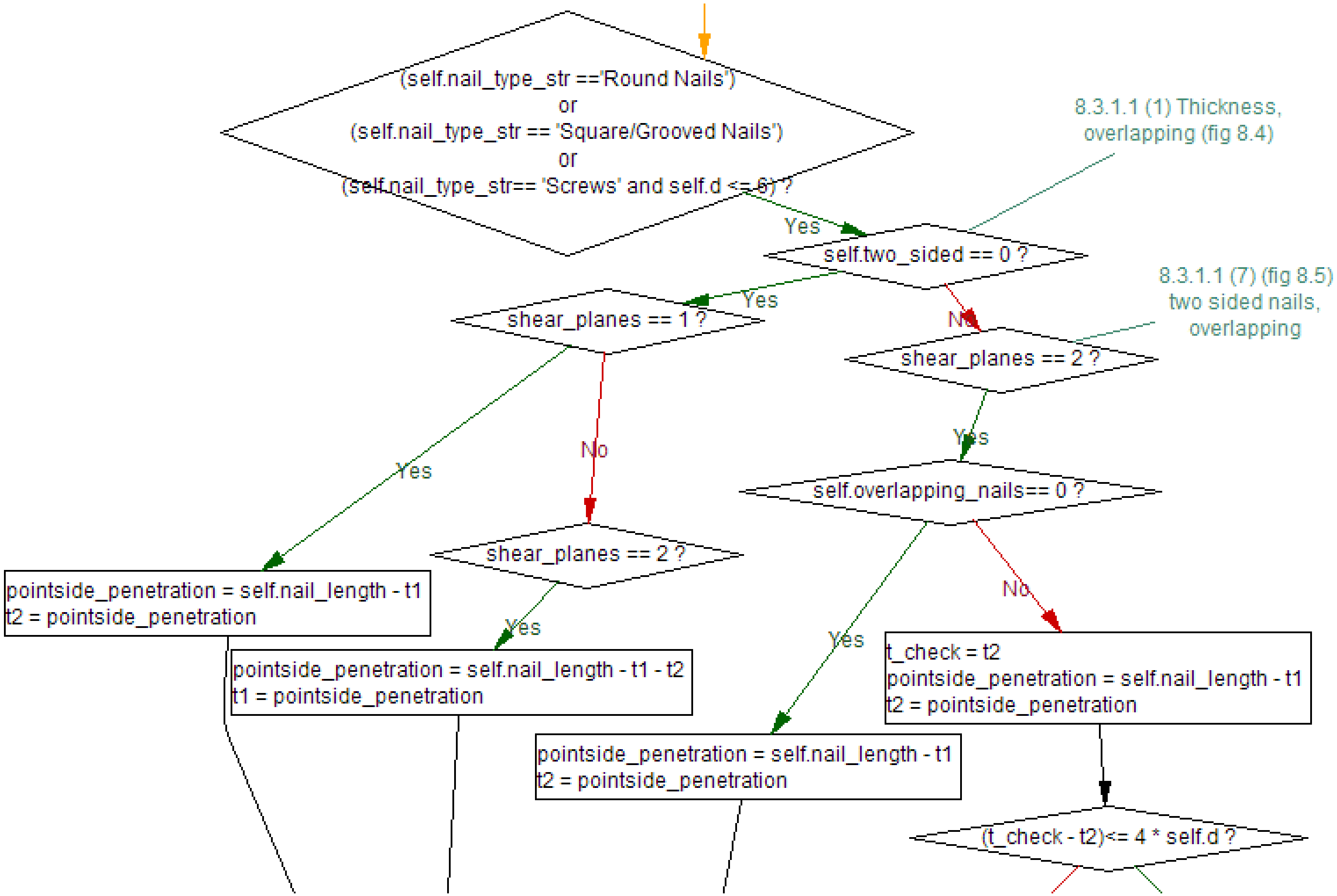
  - The minimum of the nail headside material thickness and the nail pointside penetration in a double shear connection.
- t_2 is:

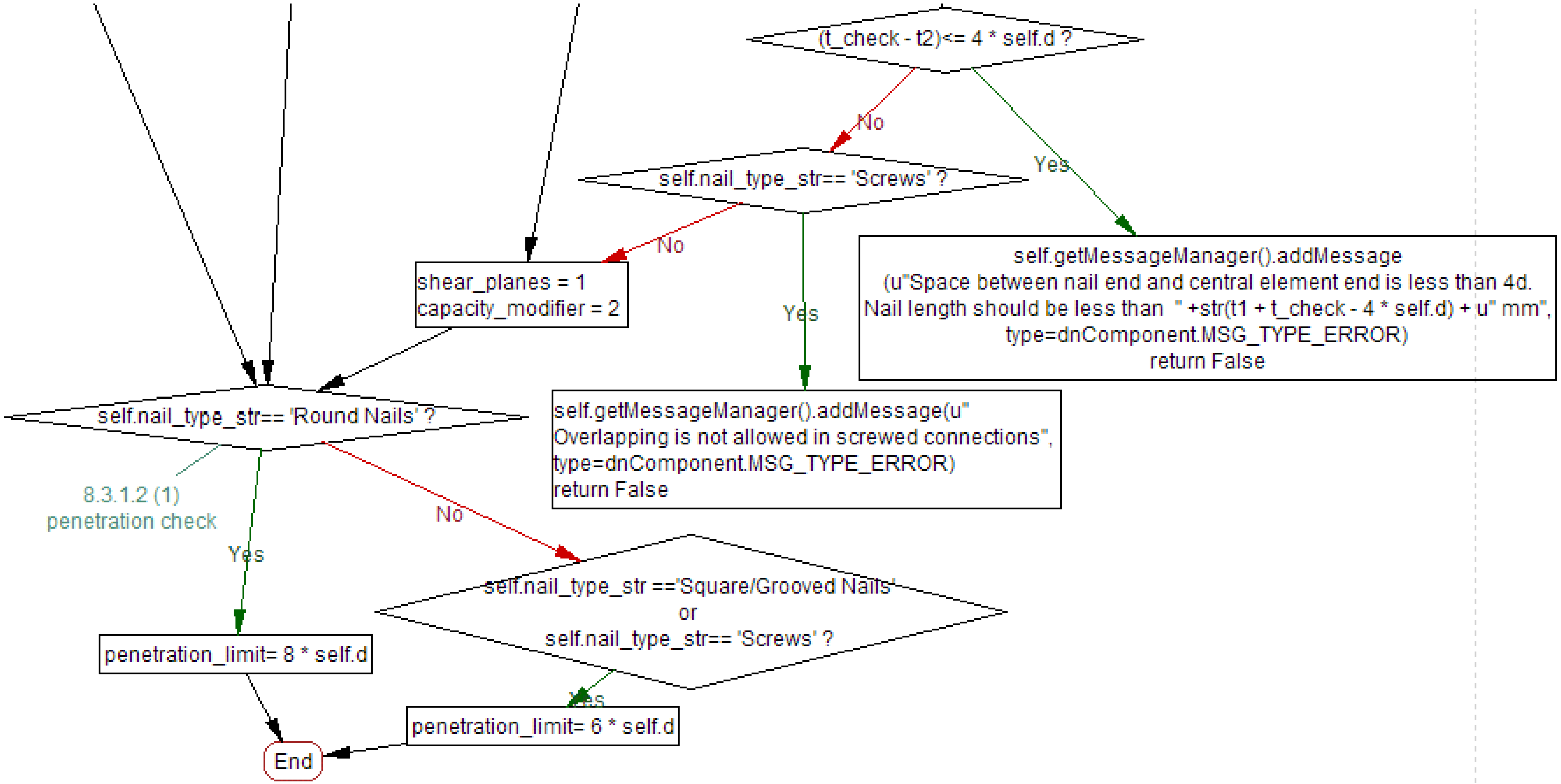
  - The nail pointside penetration where the connection is in single shear.

  - The central member thickness for a connection in double shear.

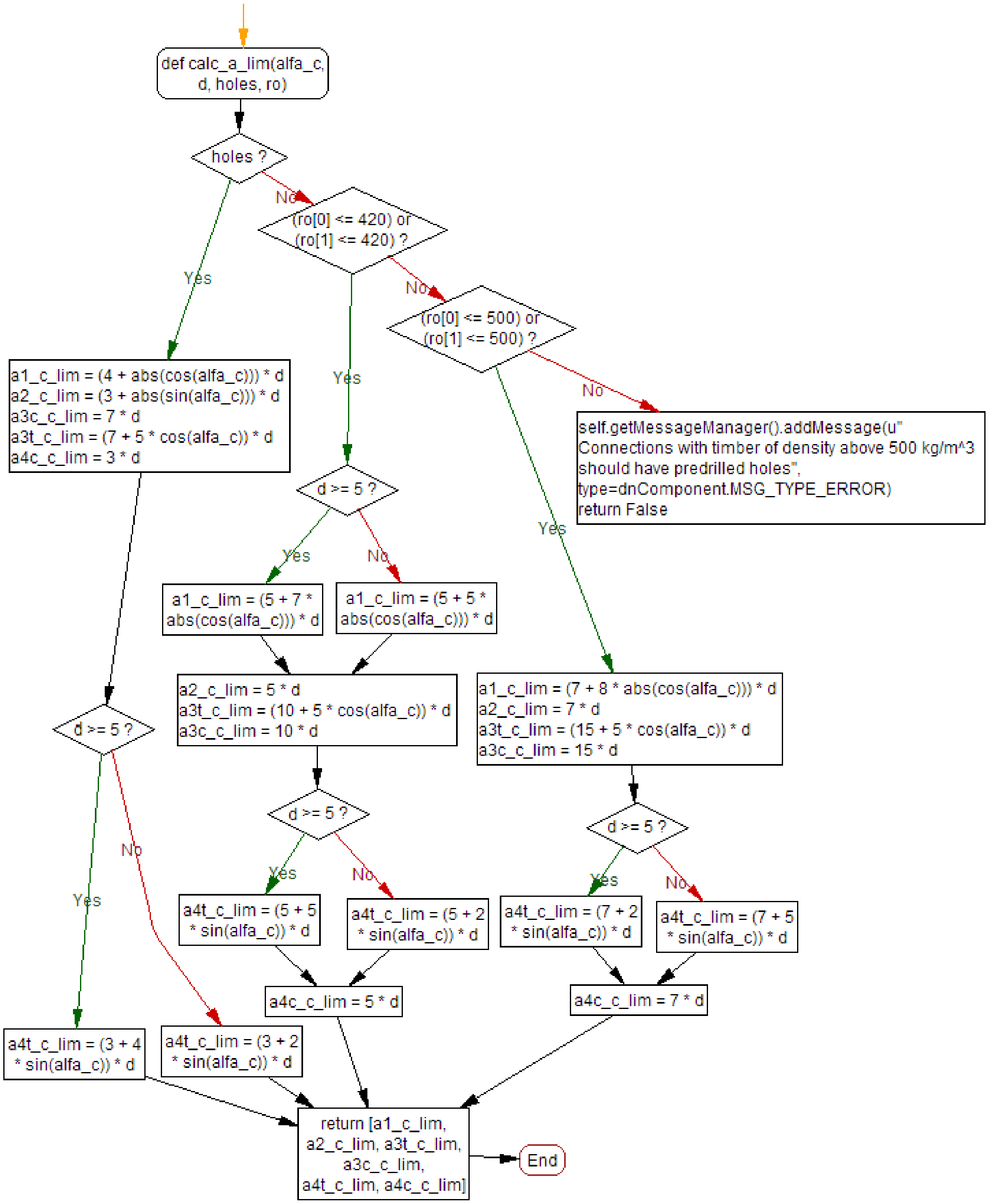
Depending on previous calculations, proper path will be chosen in order to determine t_1, t_2 and connector penetration length. If double sided connectors were chosen, algorithm will perform necessary checks and afterwards will treat the connection as two single shear connections (final capacity will be multiplied by capacity_modifier, set to 2 by this operation.).

Last step of algorithm 11 is assigning penetration limit for chosen connector type.


(self.nail_type_str =='Round Nails')
or
(self.nail_type_str == 'Square/Grooved Nails')
or
(self.nail_type_str== 'Screws' and self.d <= 6) ?
8.3.1.1 (1) Thickness, overlapping (fig 8.4)
Yes
self.two_sided == 0 ?
Yes
No
shear_planes == 1 ?
8.3.1.1 (7) (fig 8.5) two sided nails, overlapping
shear_planes == 2 ?
Yes
No
Yes
self.overlapping_nails== 0 ?
shear_planes == 2 ?
pointside_penetration = self.nail_length - t1
t2 = pointside_penetration
Yes
No
Yes
pointside_penetration = self.nail_length - t1 - t2
t1 = pointside_penetration
t_check = t2
pointside_penetration = self.nail_length - t1
t2 = pointside_penetration
pointside_penetration = self.nail_length - t1
t2 = pointside_penetration
(t_check - t2)<= 4 * self.d ?

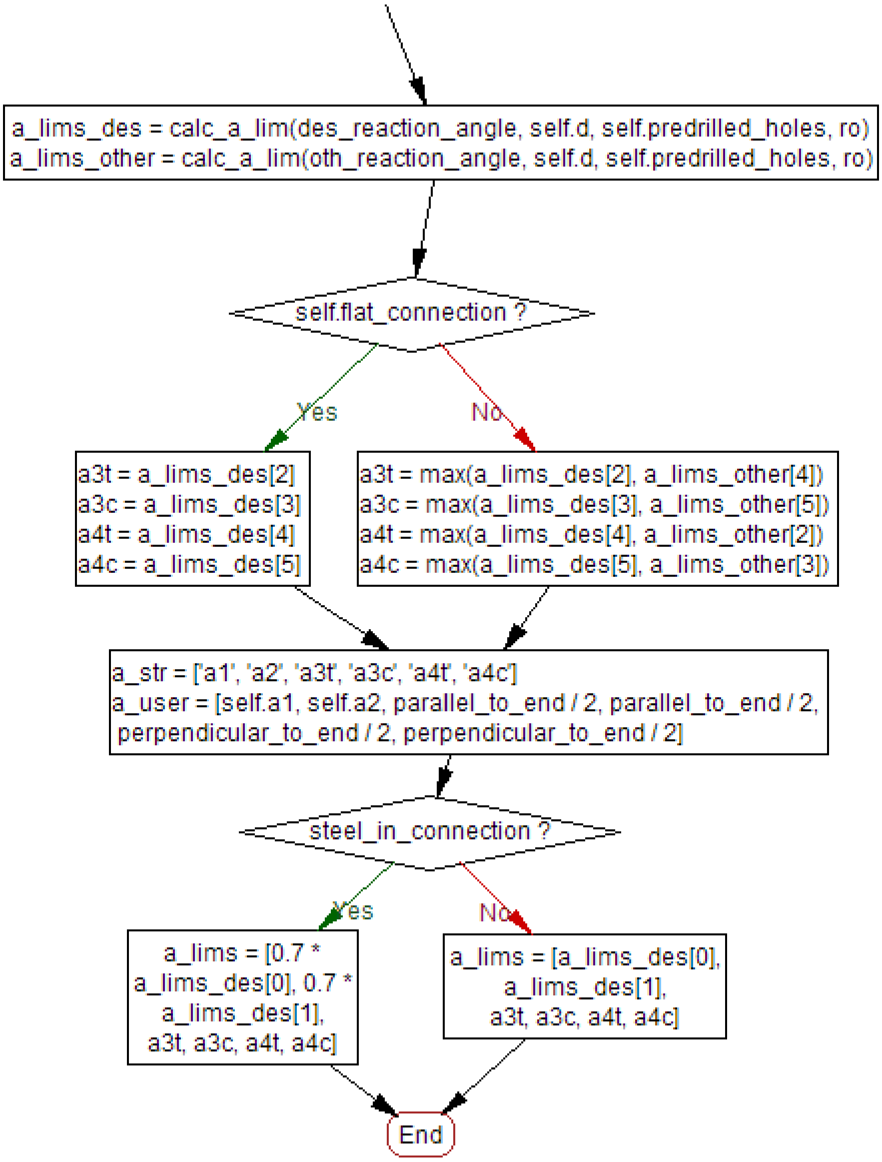

(t_check - t2)<= 4 * self.d ?
No
Yes
self.nail_type_str== 'Screws' ?
No
Yes
shear_planes = 1
capacity_modifier = 2
self.getMessageManager().addMessage
(u"Space between nail end and central element end is less than 4d.
Nail length should be less than " +str(t1 + t_check - 4 * self.d) + u" mm",
type=dnComponent.MSG_TYPE_ERROR)
return False
self.getMessageManager().addMessage(u"
Overlapping is not allowed in screwed connections",
type=dnComponent.MSG_TYPE_ERROR)
return False
self.nail_type_str== 'Round Nails' ?
8.3.1.2 (1)
penetration check
Yes
No
self.nail_type_str =='Square/Grooved Nails'
or
self.nail_type_str== 'Screws' ?
penetration_limit= 8 * self.d
penetration_limit= 6 * self.d
End

Algorithm 9



Algorithm 10

Algorithm 13 describes function calc_a_lim, calculating connection connectors spacing limits. Algorithm directly corresponds to table 8.2 in norm.

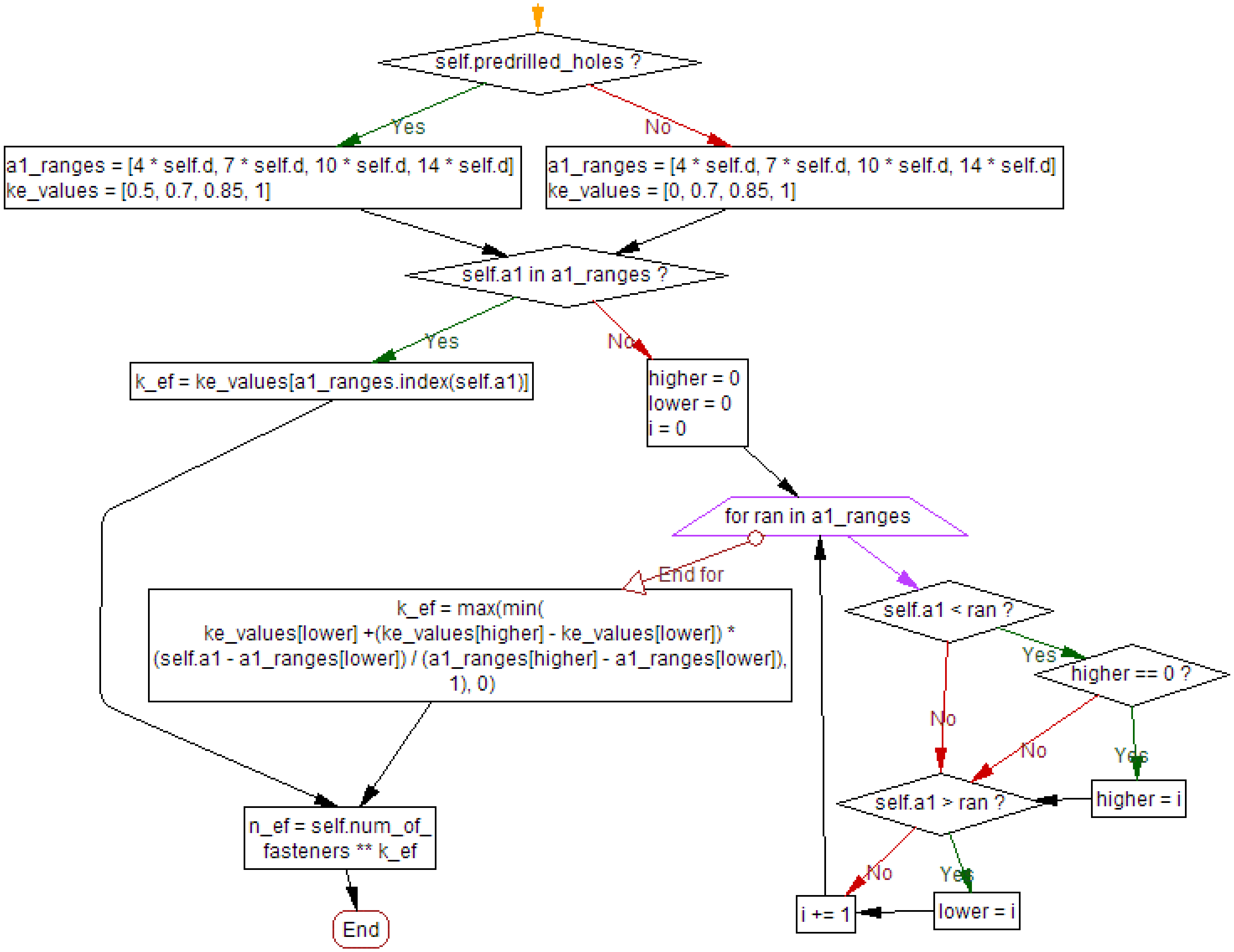


Algorithm 11

Algorithm 13 uses previously defined function calc_a_lim to calculate spacing limits for both designed and other element. If connection is flat, only designed element values are taken, as all limits correspond to other connection limits. For non-flat connection it should be taken into consideration that end of designed element is edge of other element and vice versa. Thus, it is needed to pick maximal value between designed element spacing limit and corresponding other element spacing limit. Relations between spacing limits ("o" states for other element value):

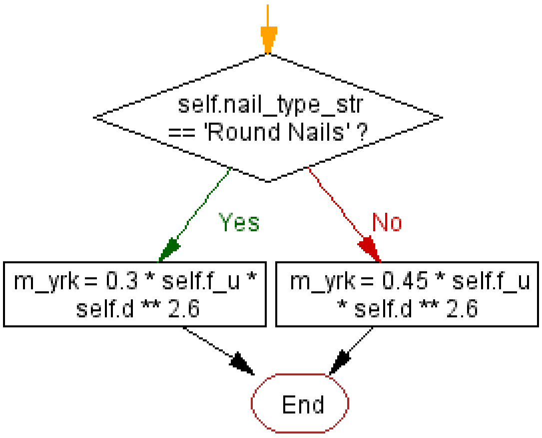
$$a_{3,t} \sim a_{o,4,t}\ , \qquad a_{3,c} \sim a_{o,4,c}\ , \qquad a_{4,c} \sim a_{o,3,c}\ , \qquad a_{4,c} \sim a_{o,3,c}$$

Further steps of algorithm include making list of string representation of each spacing value and writing down user-defined spacings for comparison further in code. If there is steel element in connection a_1 and a_2 requirement is lowered to 70%.



Algorithm 12

Algorithm 14 task is calculating k_{ef} coefficient, crucial for $n_{ef} = n^{k_{ef}}$ value, which directly corresponds to connection capacity. Algorithm iterates through a1_ranges values, comparing them to user-defined a_1 value. Every time when a_1 is bigger than one from iteration, algorithm will write down current iterator position as "lower" value. After encountering a1_range value which will be greater than a_1, algorithm will write down this value, but only once. After such preparation it is possible to linearly interpolate between k_{ef} values for respective a_1, as lower and higher value (x_0, x_1) and corresponding k_{ef} (y_0, y_1) are known and can be used in interpolation formula $y = y_0 + (y_1 - y_0) \cdot \frac{x - x_0}{x_1 - x_0}$. After calculating and limiting k_{ef} to 1, n_{ef} is calculated.



Algorithm 13

Last step (Algorithm 15) of preparing data for nail / screw connection calculation is simple calculation of characteristic yield moment of a fastener.

4.2.1. Calculation of coefficients for Staples

Staples algorithm (Algorithm 16) bases on algorithm for nails, albeit there are few differences: 1. as staples are treated as two nails connected with the crown, minimum spacings and end distances requirements are larger. 2. There is also Theta factor (angle between staple crown direction and grain direction), corresponding with $a_{1,lim}$ value and overall connection capacity. 3. Formula for $m_{y,rk}$ is different, and it does not depend on f_u, however, it is required for f_u to be greater than 800 N/mm^2. 4. Doubled staple capacity is represented by multiplying capacity_modifier variable by 2 (or 1.4, if theta < 30^0)

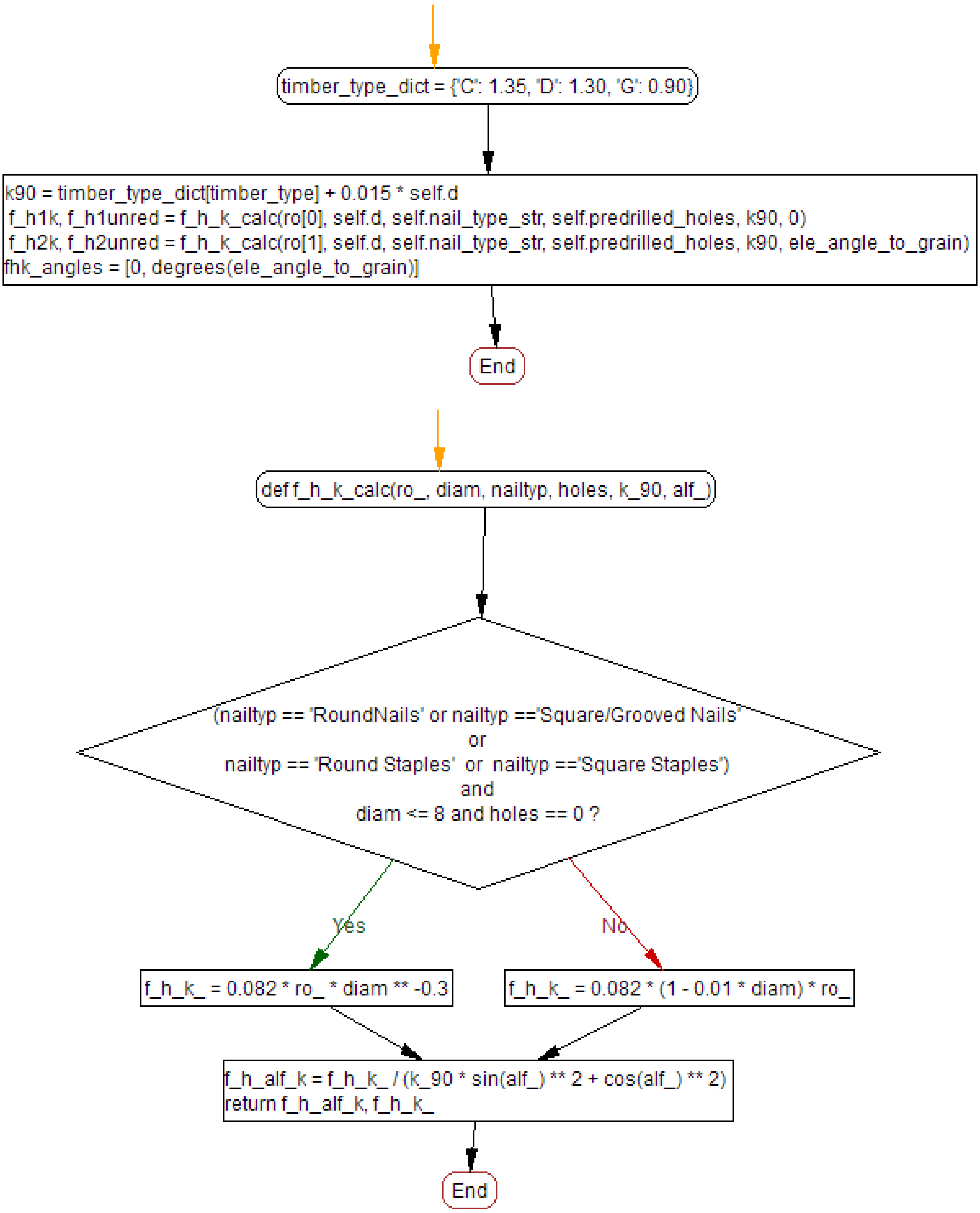
4.2.2. Calculation of coefficients for Bolts and Screws ($d > 6\ mm$)

Bolts and screws (d > 6 mm) algorithm (Algorithm 17) is based on algorithm for nails, albeit there are few differences: 1. Norm require bolts to be in 6 < d < 30 mm range. 2. For bolts, whole path responsible for calculating t_1, t_2 and penetration length is omitted, as, by definition and logic, bolts are always needed to be as long as the connection. For screws, path similar to one in Algorithm 11 is used. 2. Minimum spacings and end distances requirements are different. 3. Formula for n_{ef} is different, and it does not depend on k_{ef} .

4.2.3. Calculation of coefficients for Dowels

Dowels algorithm (Algorithm 18) is based on algorithm for bolts, the differences are: 1. t_1 and t_2 calculations, path from screws algorithm is applied. 2. Minimum spacing and end distances requirements.

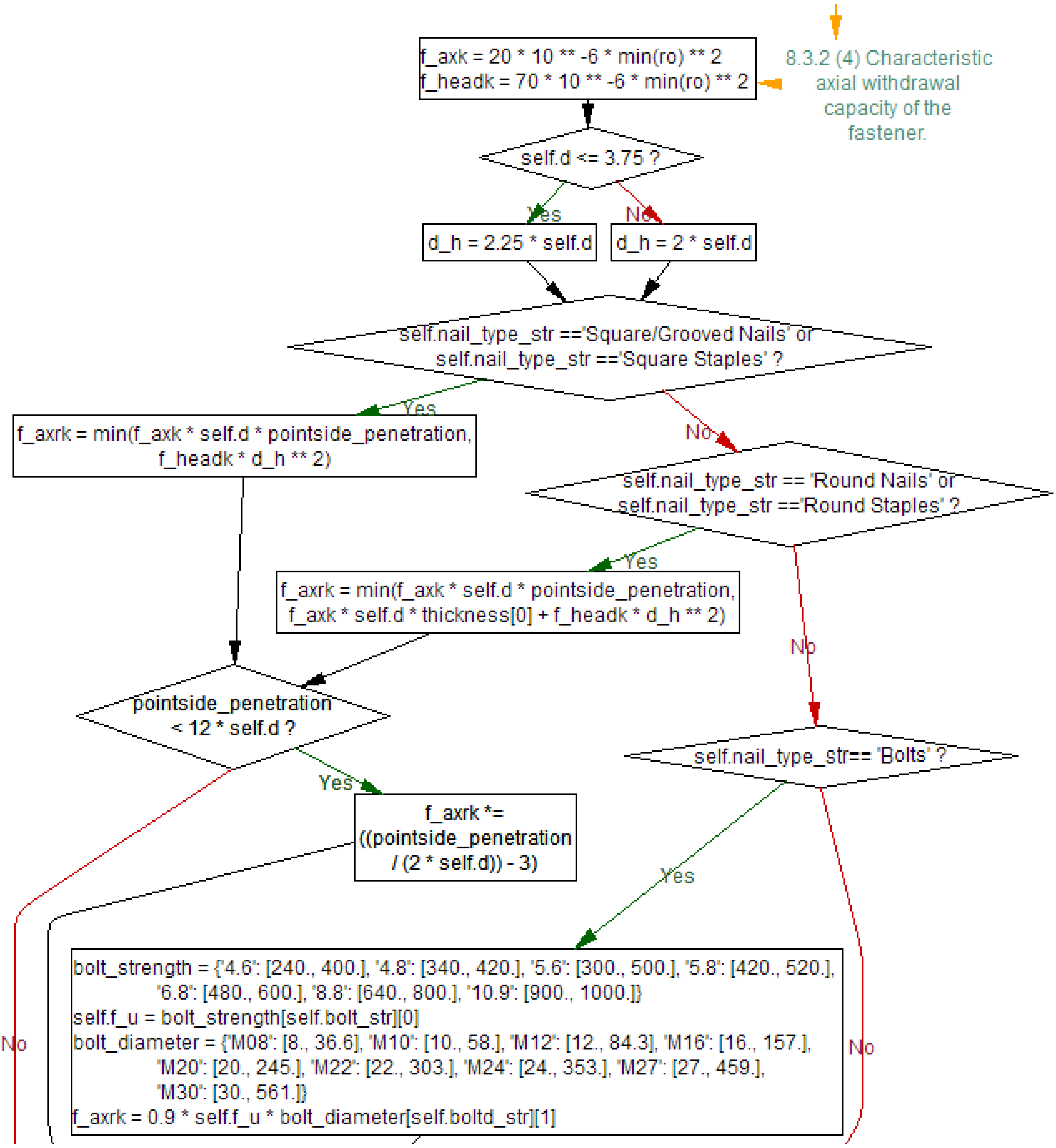
4.2.4. Calculation of embedment strength

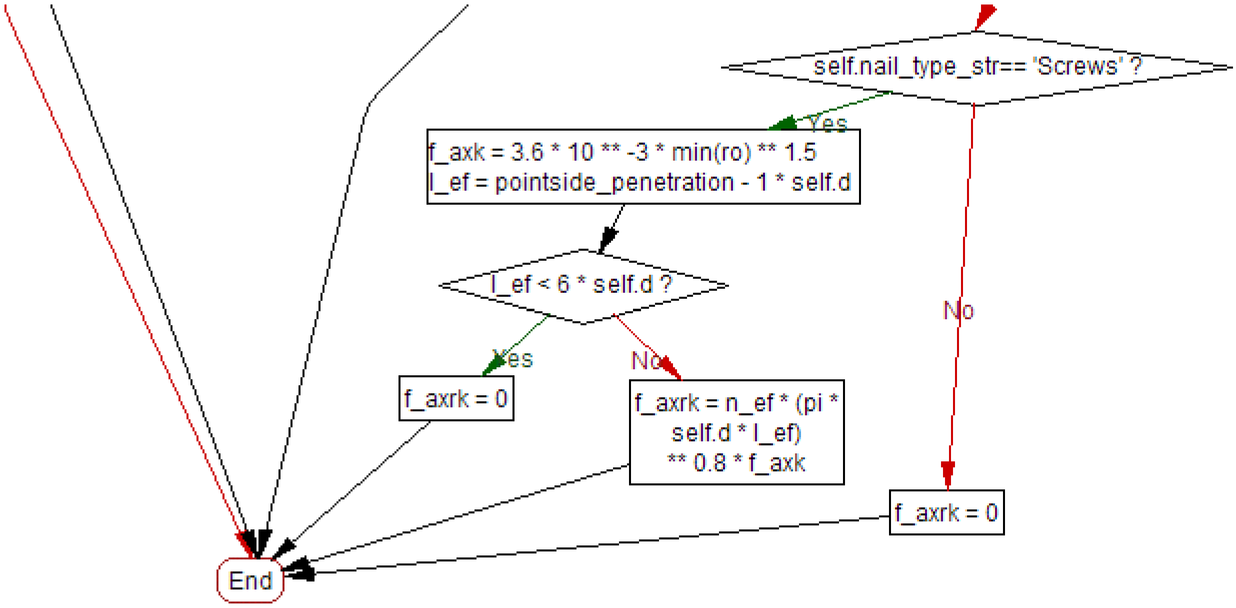


Algorithm 14

Algorithm 19 is responsible for calculating chosen connector type embedment strength ($f_{h,k}$ & $f_{h,\alpha,k}$). Function f_h_k_calc calculates connector embedment strength both when the load is parallel to grain and when it is rotated by an angle. First part of embedment factor k_{90} is determined by using dictionary containing first letter of material name and respective value for this material (C for softwoods, D for hardwoods, G for GL (i.e. LVL)) and getting dictionary value for designed elements material. For designed element angle of load is 0^0 (compression/tension), for other element it is an angle between elements grains.

4.2.5. Withdrawal capacity of connectors





Algorithm 15

Algorithm 20 calculates withdrawal capacity ($F_{ax,Rk}$) for chosen connector type. Algorithm consists of four paths (Round nails/staples, Square nails/staples, Bolts, Screws (Dowels are assumed to be unable to take tensile loading)). Each path follows steps described in Norm with additional improvements described in [2]. Bolts capacity is calculated with respect to EN 1993.

4.2.6. Preparation for bending calculation

Before connection bending can be calculated, it is necessary to "solve" all connection connectors (i.e. calculate their coordinates and radiuses list). This task is solved using developed function calc_connectors_with_displacement, which is one of the most complex functions developed in program. Graphical explanation of function steps is presented and explained below.

- Explanation for connection with elements at an angle to each other

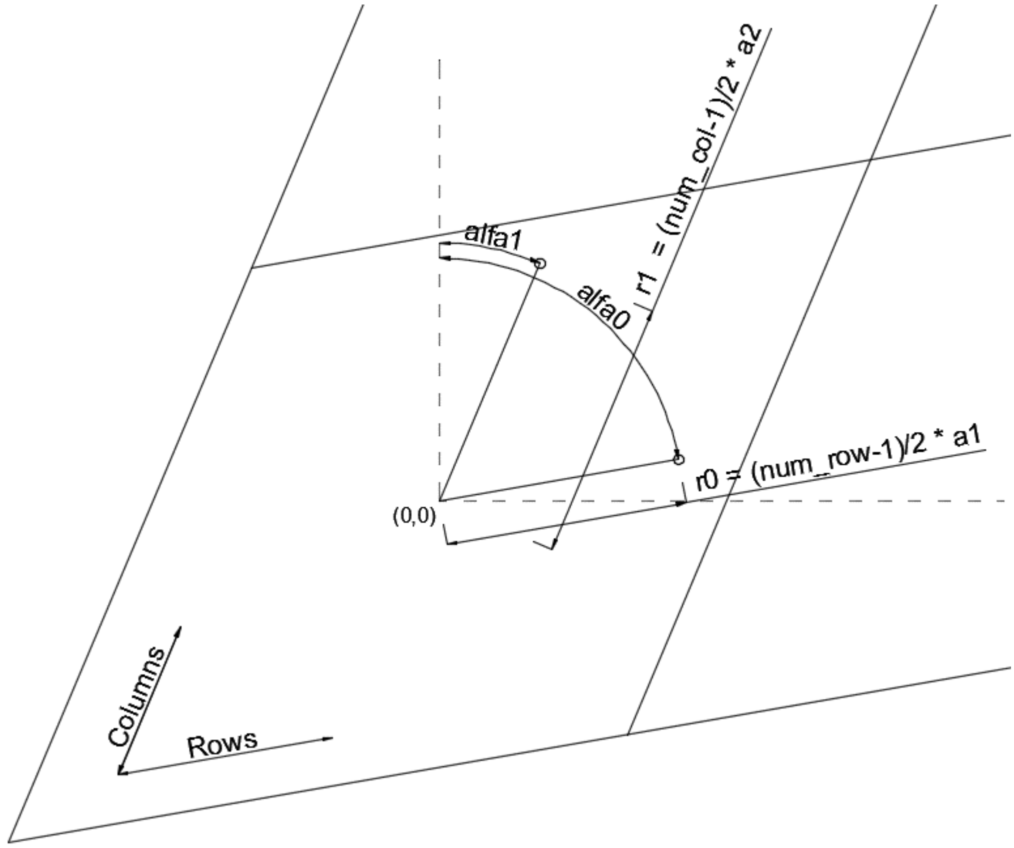


Figure 5

First step of solving connectors is calculation of boundary connectors coordinates (Figure 5). It is carried out by calculating connector radius $r_i = \frac{n-1}{2} * a_i$ where n is amount of fasteners in direction (columns or rows) and a_i is fasteners spacing in respective direction.

After calculating connectors radiuses, their Cartesian system coordinates are calculated from trigonometrical relations. (Formulas and idea presented on Figure 6)

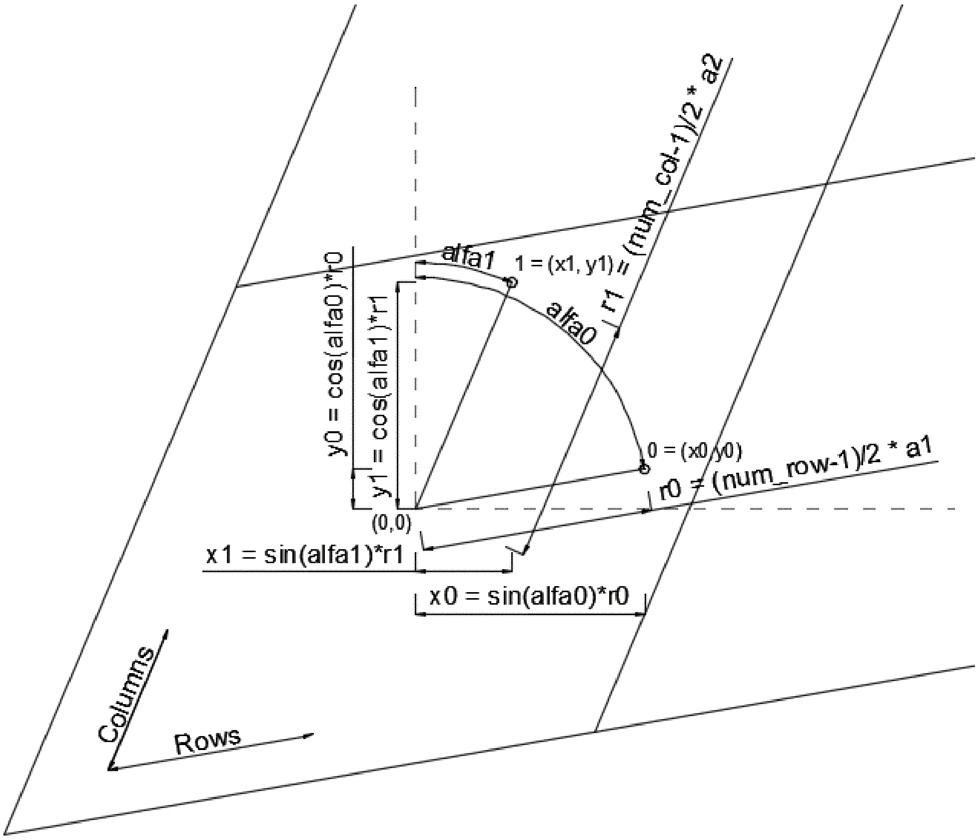


Figure 6

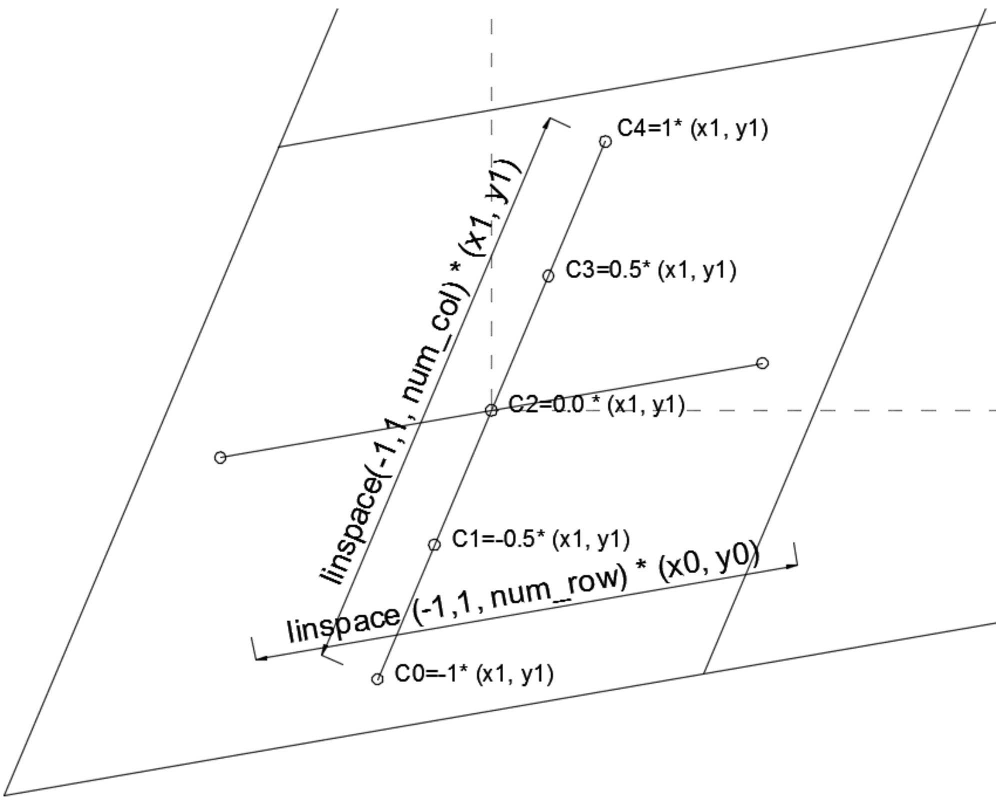


Figure 7

Next step (Figure 7) involves creating linearly spaced vector for connectors directions. Idea is to obtain linear relation between connectors, by dividing 1D vector of coordinates (-1, 1) by connectors amount. Result of this operation for exemplary 5 connectors will be list (-1, 0.5, 0, 0.5, 1). Having such list, it is possible to calculate coordinates of all connectors in direction, by simply multiplying list by positive boundary coordinate.

$$Coords = linspace(-1,1,n) * (x_i, y_i)$$

$$for\ n = 4, \qquad (x_1, y_1) = (9,3),$$
$$\rightarrow Coords = linspace(-1,1,4) * (9,3) = (-1,-0.33,0.33,1) * (9,3)$$
$$= ((-9,-3),(-3,-1),(3,1),(9,3))$$

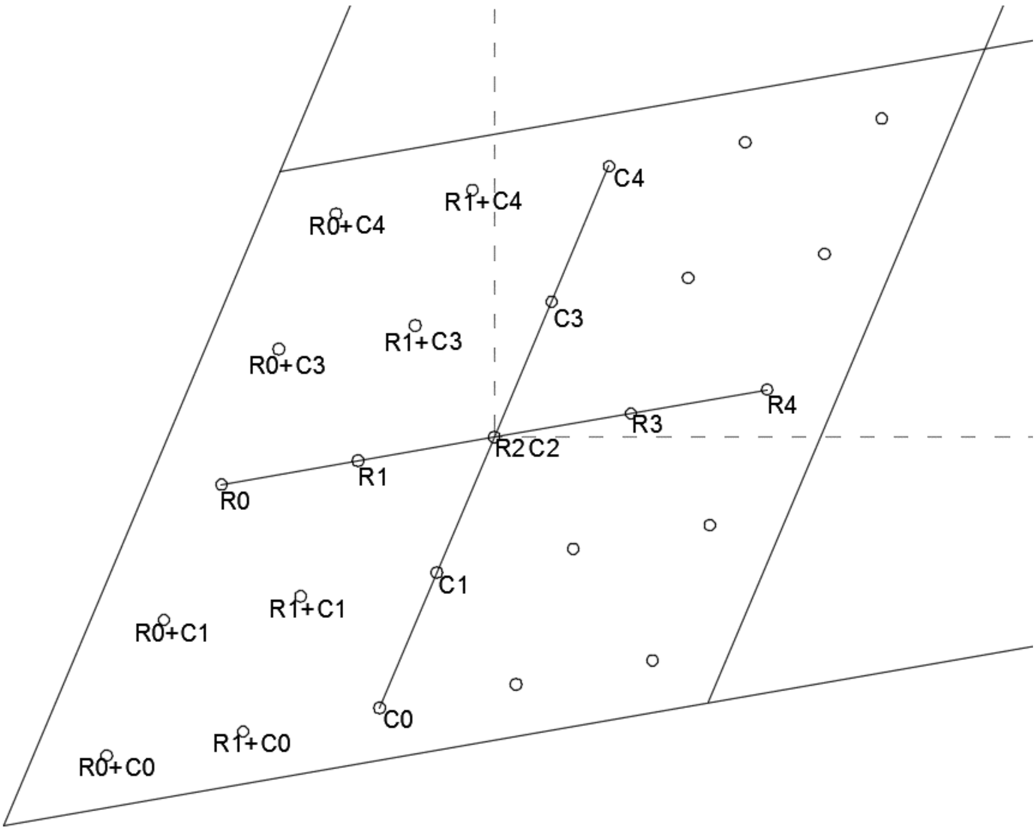


Figure 8

After obtaining coordinates of connectors in one row and column rest of the connectors can be obtained by iterating through one direction connectors coordinates and adding to them other direction coordinates list (Figure 8). Exemplary one iteration for row:

- Iterator value, $R1 = (-1,-1)$

- Column coordinates, $C = (C0, C1, \dots, C4) = ((-2,-4),\ (-1,-2),\ (0,0),\ (1,2),\ (2,4))$

$R1_{col} = R1 + C = (-1,-1) + \big((-2,-4),(-1,-2),(0,0),(1,2),(2,4)\big) =$
$((-3,-5),(-2,-3),(-1,-1),(0,1),(1,3))$

After iterating through all connectors, whole matrix of connectors in connection is obtained.

Last step is calculation of connectors radiuses and their square sum, basing on generated coordinates.

- Explanation for flat connection (flat angle connection)

Flat connection procedure is identical as for non-flat connection, with exception for column to vertical angle, which is set in a way that rows are perpendicular to columns (Figure 10)

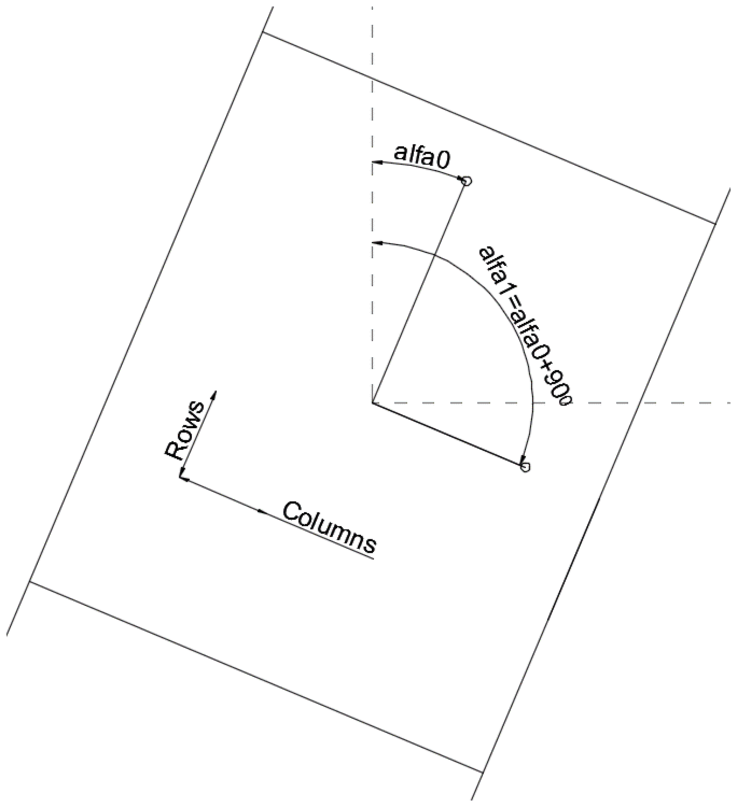


Figure 9

- Explanation for connection with steel plate

Similar to flat connections, columns angle is fixed to perpendicular to rows, displacement of plate is also included after generating connectors matrix. (Figure 11).

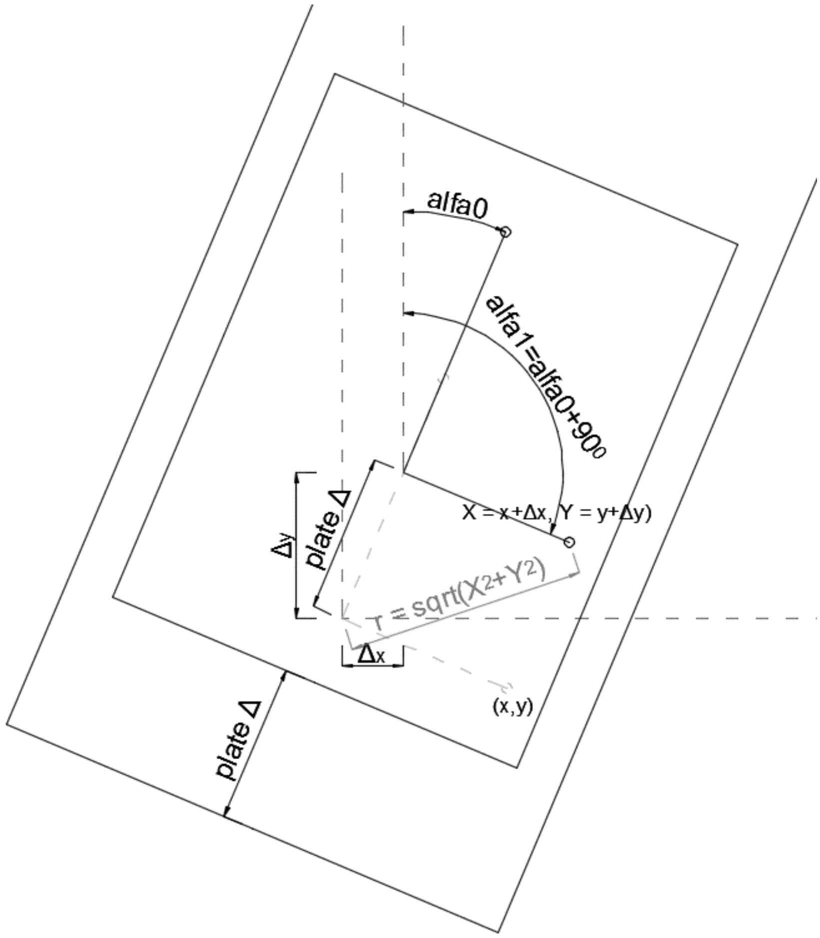
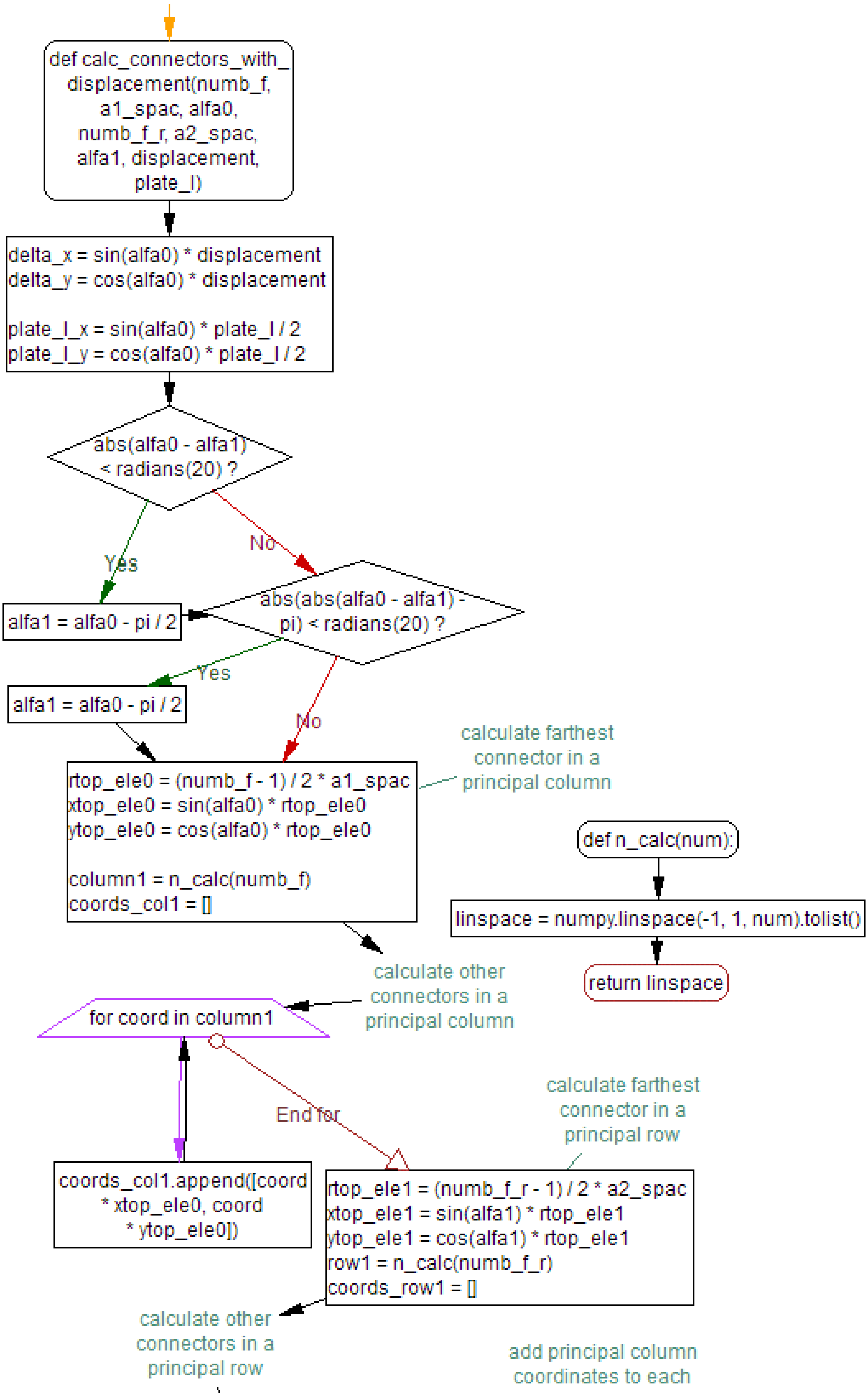
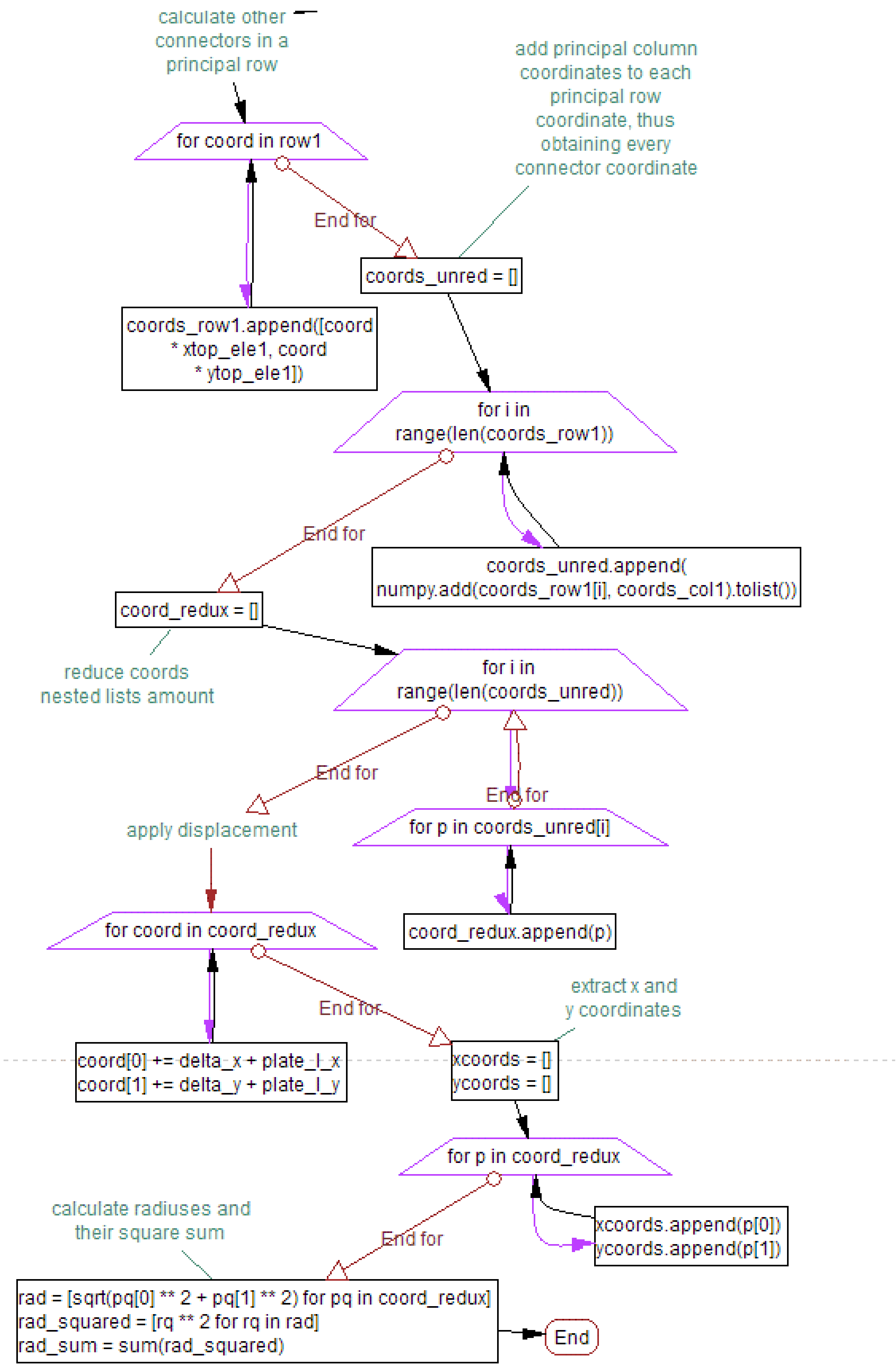


Figure 10

- Function algorithm





Algorithm 16

4.2.7. Calculation of bending

Calculations of bending are carried out according to rules introduced in [2]. There are no rules for bended dowel-type connections in Eurocode 1995. Initial step for calculation of bending is solving connectors, carried out by calling out previously described calc_connectors function. After obtaining connectors matrix, boundary connector is found, by finding connector with largest radius. (Algorithm 22). Algorithm 24 calculates spacings and end distances limits for bending in order to prevent brittle failure, as described in [2]. Algorithm 23 task is to calculate spacings and end distances limits and to calculate resultant force F_d acting on most fatigued connector (boundary connector). Formulas for resultant force [2]:

$$F_{m,d,max} = M_d \frac{r_{max}}{\sum_{i=1}^{n} r_i^2}$$

$$F_{h,d} = \frac{H_d}{n}, \qquad F_{v,d} = \frac{V_d}{n}$$

$$F_d = \sqrt{\left(F_{v,d} + F_{m,d,max} \cdot \frac{x}{r_{max}}\right)^2 + \left(F_{h,d} + F_{m,d,max} \cdot \frac{y}{r_{max}}\right)^2}$$

$$\alpha = \arccos\left(\frac{F_{h,d} + F_{m,d,max} \cdot \frac{y}{r_{max}}}{F_d}\right)$$

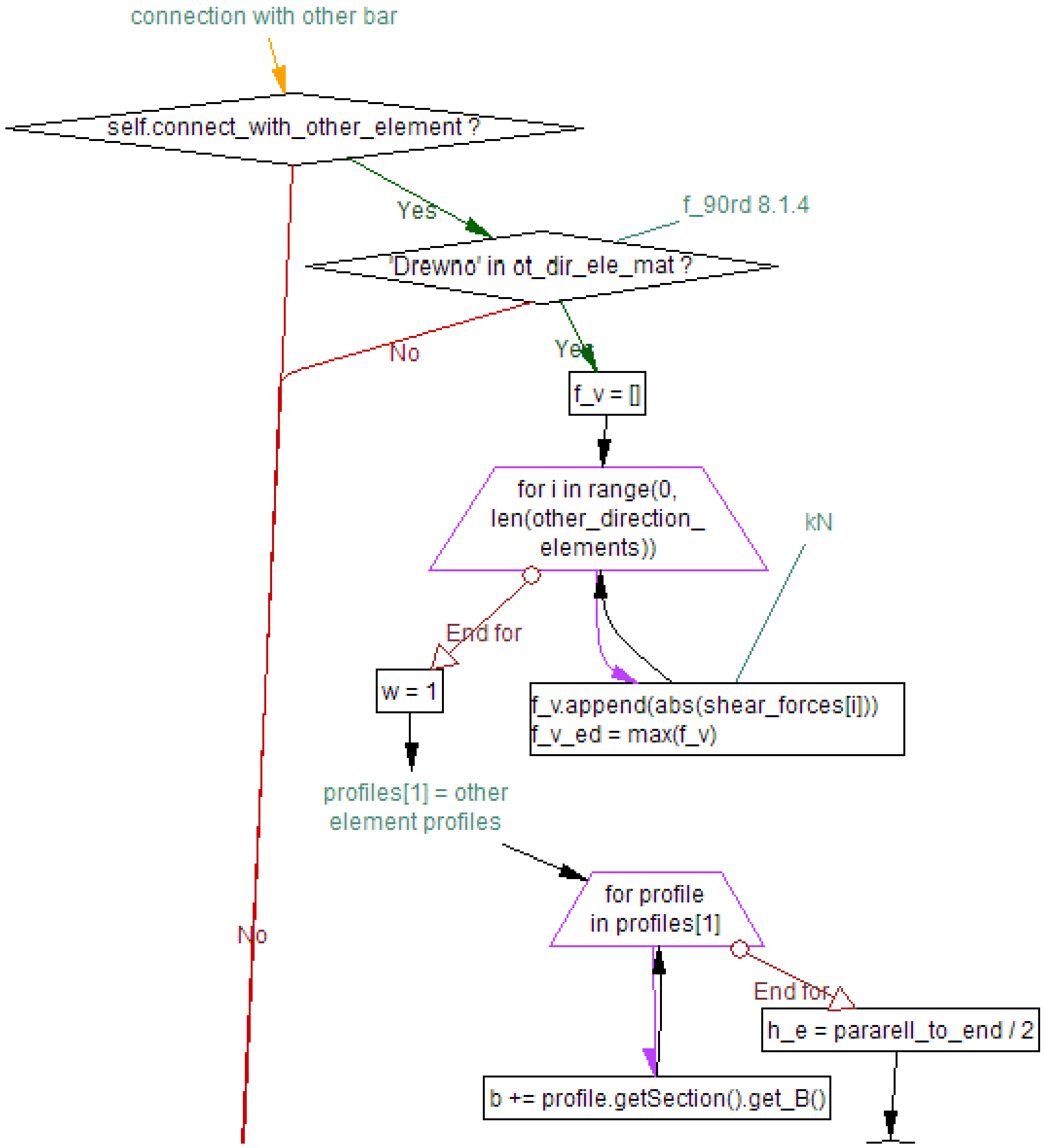
Where α is an angle of inclination of force to the horizontal.

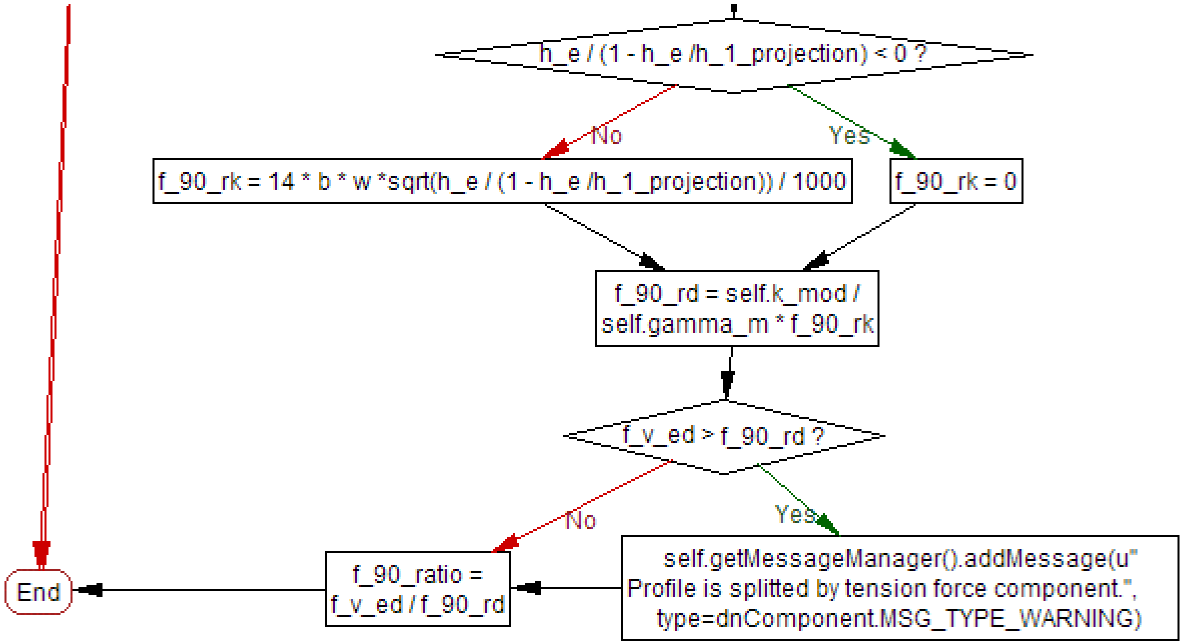
After calculating resultant force, algorithm will update embedment strength values with respect to newly calculated angle of resultant force, α.

4.2.8. Calculation of splitting capacity

If designed element is connected to other timber element Algorithm 24 (below) will calculate connection splitting capacity with respect to Norm.

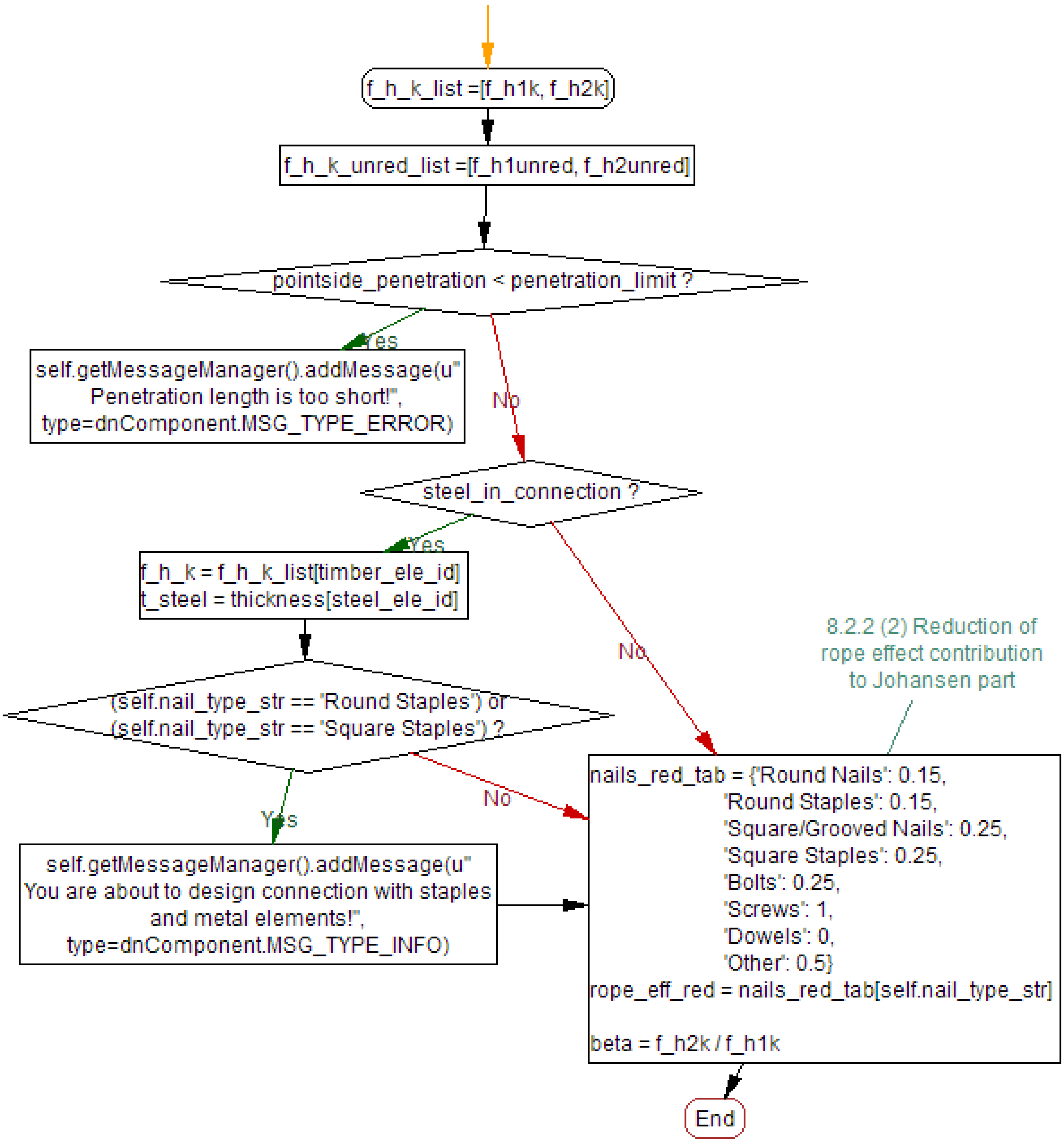
If final capacity, $F_{90,Rd}$ will be lesser than other elements shear forces, $F_{v,Ed}$ (calculated in Algorithm 6) error will be risen.


connection with other bar
self.connect_with_other_element ?
Yes
f_90rd 8.1.4
'Drewno' in ot_dir_ele_mat ?
No
Yes
f_v = []
for i in range(0, len(other_direction_ elements))
kN
End for
w = 1
f_v.append(abs(shear_forces[i]))
f_v_ed = max(f_v)
profiles[1] = other element profiles
for profile in profiles[1]
No
End for
h_e = pararell_to_end / 2
b += profile.getSection().get_B()



Algorithm 17

4.2.9. Pointside penetration check & assigning rope effect contribution reduction

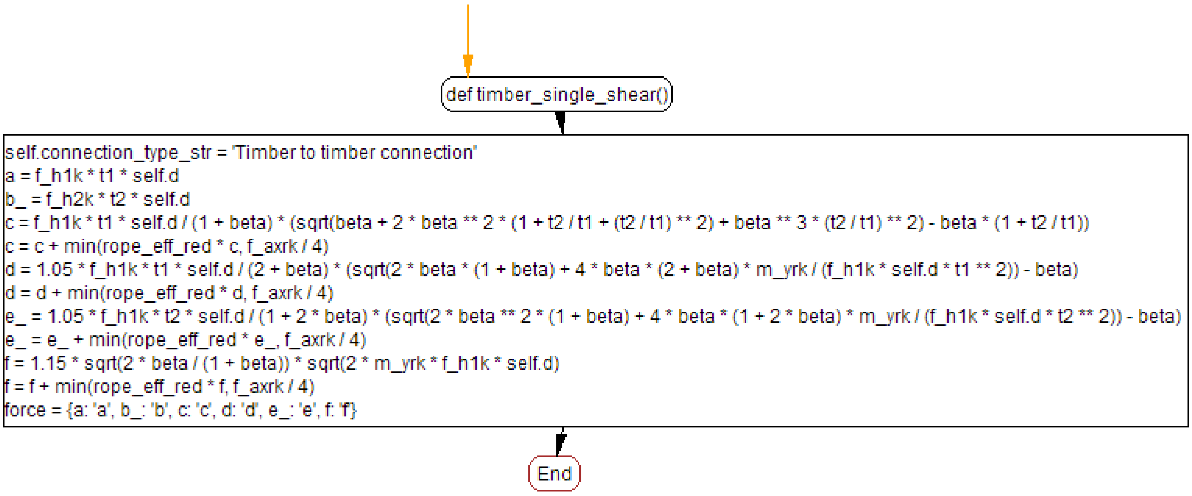


Algorithm 2618

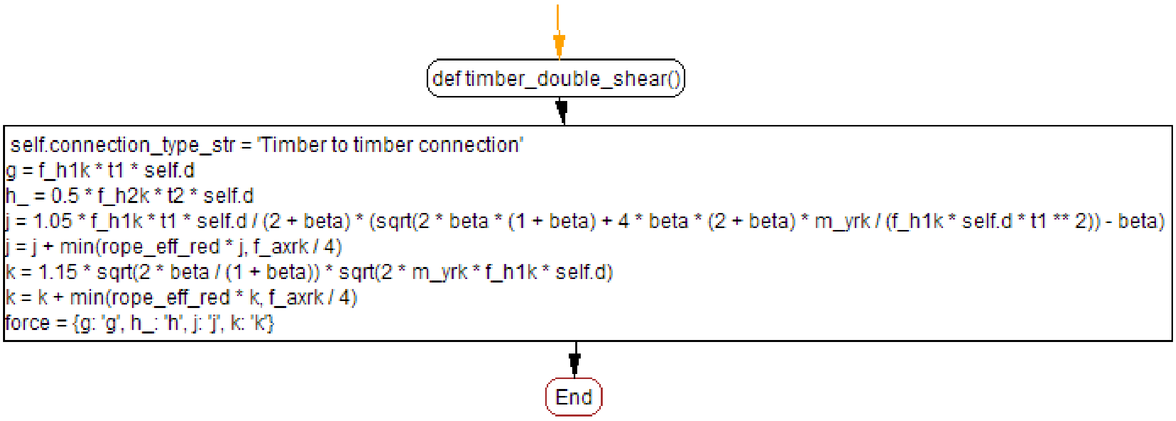
Algorithm 26 checks, if previously calculated (sections 4.2.7 to 4.2.10) pointside penetration length is greater than its limit. If there is steel in connection, algorithm will ignore its contribution to embedment strength by using $f_{h,k}$ calculated for timber element only. Steel element thickness is also saved for further use in functions for calculating load-carrying capacity of connectors in steel-to-timber connections. Second part of the algorithm sets rope effect reduction factor for further use in connector load-carrying capacity calculateion.

4.2.10. Load-carrying capacity of connectors

- Timber-to-timber connections

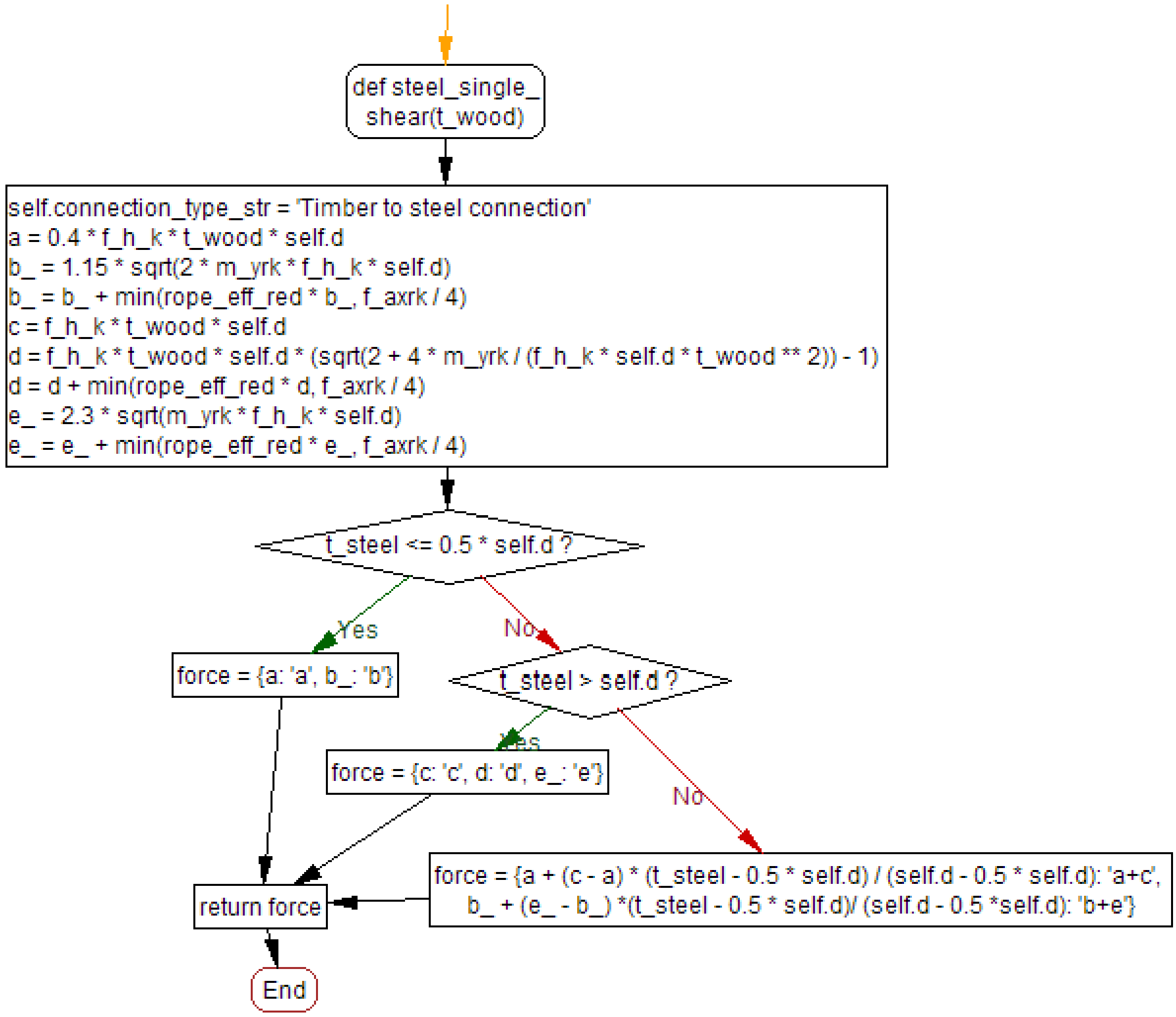


Algorithm 19

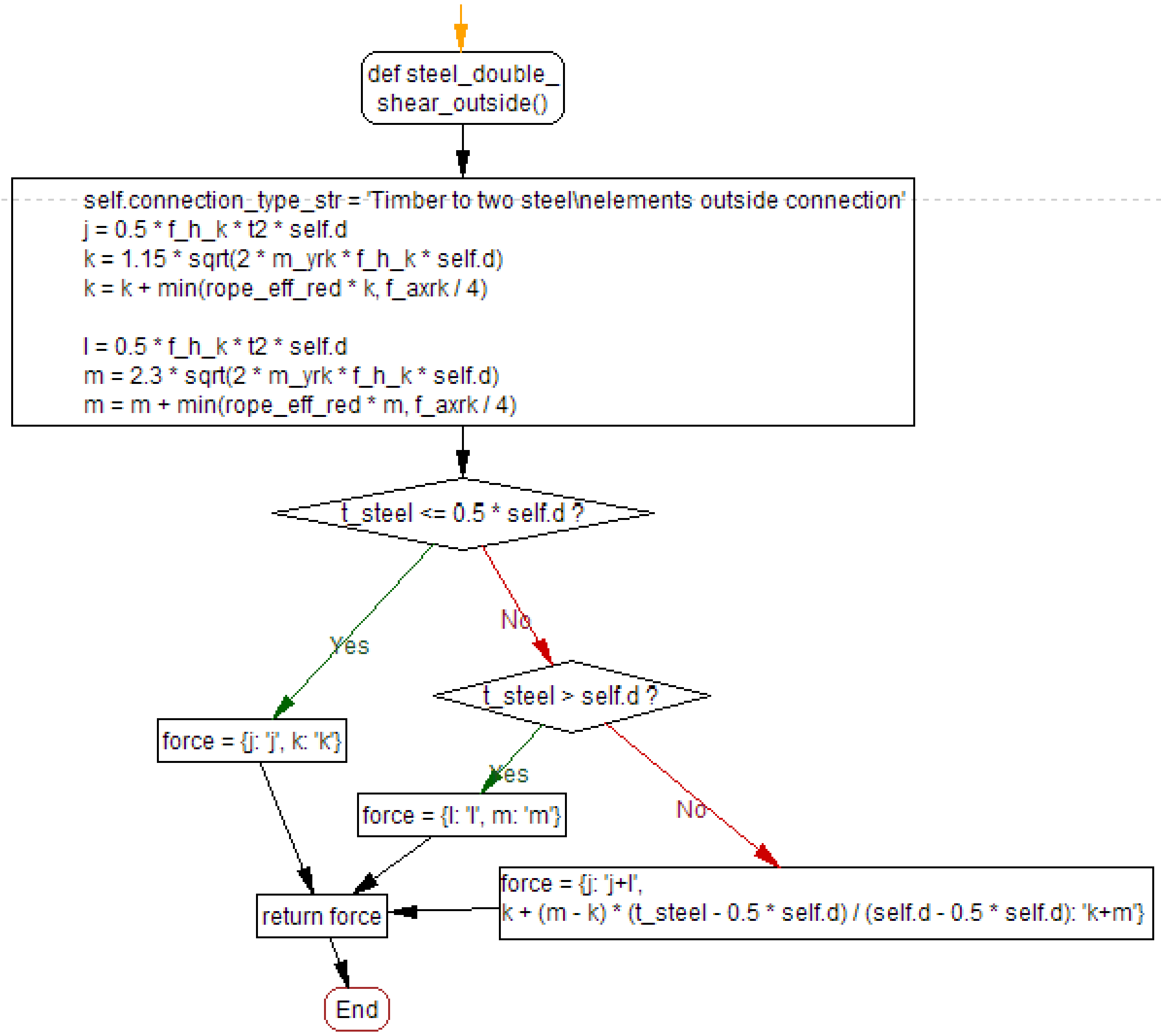


Algorithm 20

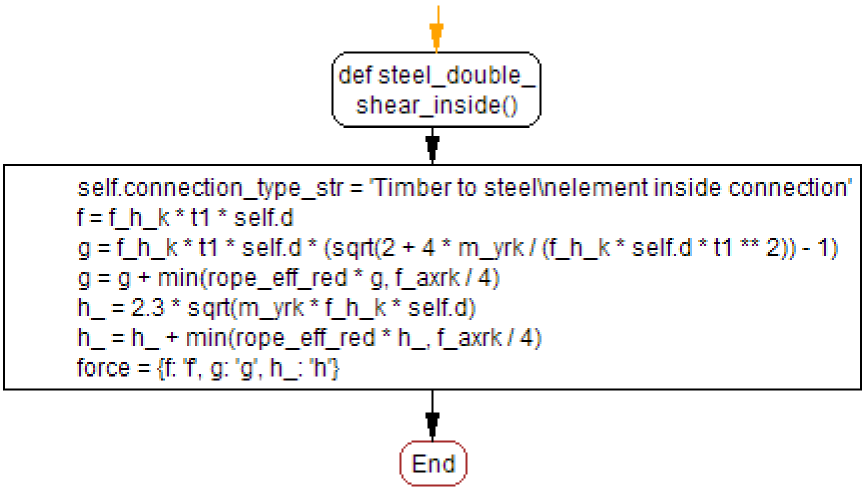
- Steel-to-timber connections



Algorithm 21



Algorithm 22



Algorithm 23

Algorithms 26 to 30 are algorithms calculating all possible destruction modes for given connection type. All equations are based on EN 1995.

After calculating Johansen yield theory part of connector (first term in equation), function will add minimum of (Johansens part · rope effect reduction) or connector withdrawal capacity ($F_{ax,Rk}$) to overall connector capacity in certain mode. It is related with norm requirement for rope effect contribution, which cannot be larger than certain % of Johansen part (limits introduced in Algorithm 25).

Single shear connection with steel plate requires thickness of timber element to be given, it is given with t_wood variable.

4.2.11. Final capacity calculation

Final capacity algorithm (Algorithm 32) calls out function corresponding to currently defined connection type.

If there are no steel elements in the connection, algorithm will call function for failure modes in timber-only connections, depending whether it is connection in single or double shear, algorithm will respectively call out Algorithm 27 or Algorithm 28.

If connection contains steel element, algorithm will check how many shear planes does the connection have, if it will be one shear plane, Algorithm 29 will be called out. If it will be two shear planes, depending if the steel element is two branch element (i.e. it is outside) Algorithm 30 will be called out, if steel element will be one branch element (i.e. it is inside) Algorithm 31 will be called out.

After obtaining values for connection failure modes ($F_{v,Rk}$), algorithm will proceed to calculation of connection axial capacity, using formula from norm with additional modifications, coming from way of program developing

$$F_{v,ef,Rk} = n_{ef} * \min(F_{v,Rk}) * n_{sp} * n_r * capacity\_modifier$$

$$F_{v,ef,Rd} = \frac{k_{mod} * F_{v,ef,Rk}}{\gamma_M}$$

$$ratio = \frac{F_{ed}}{F_{v,ef,Rd}}$$

Where n_{ef} is effective number of connectors in a row, n_{sp} is number of shear planes, n_r is amount of connector rows, and capacity_modifier is factor described in previous sections.

After calculating axial capacity, if it was chosen, algorithm will calculate bending resistance.

$$F_{v,ef,Rk,bend} = \min(F_{v,Rk}) * capacity\_modifier$$

$$F_{v,ef,Rd,bend} = \frac{k_{mod} * F_{v,ef,Rk,bend}}{\gamma_M}$$

$$ratio_{bend} = \frac{F_d}{F_{v,ef,Rd,bend}}$$

Where F_d is the design load on the connector per shear plane

After calculating bending resistance, if it was chosen, algorithm will calculate combined lateral-axial capacity.

$$F_{ax,Rd} = \frac{k_{mod} * F_{ax,Rk}}{\gamma_M}$$

$$if\ nails \rightarrow use_{axial} = \frac{F_{ax,Ed}}{F_{ax,Rd}} + \max(ratio, ratio_{bend})$$

$$else \rightarrow use_{axial} = \left(\frac{F_{ax,Ed}}{F_{ax,Rd}}\right)^2 + (max(ratio, ratio_{bend}))^2$$

Where $F_{ax,Ed}$ is overall axial load on all connectors (perpendicular to connection plane), $F_{ax,Rd}$ is design axial withdrawal capacity, calculated in Algorithm 20.

5. Developed program

5.1. Programming environment

Program was developed in Python language, which is Open Source high-level object-oriented programming language, implemented in the early 1990s. In order to cooperate with Soldis Projektant environment program was developed with respective directives, described in Soldis developer manual, obtainable from Soldis Projektant homepage (http://www.soldis.com.pl/). Program also uses widespread libraries for graphical interface programming – ttk and Tkinter.

5.2. Program capabilities

Designed program is capable of calculating timber connections with respect to EN 1995 requirements and improvements described in [2]. Possible supported connections:

- By connection material:

  Timber, Steel and timber

- By an angle between elements:

  Straight angle ("flat" connection), any other angle (connection with elements at an angle to each other)

- By method of load bearing:

  Direct, Indirect (With one branch element – using two steel plates, or using concealed steel plate. With two branch element - Using regular steel plate)

- By shear planes between elements:

  One or two shear planes

- By connector type:

  Nails, Square nails, Staples, Square staples, Bolts, Screws, Dowels

- By method of calculation:

  Lateral capacity (EN), combined lateral-axial capacity (EN), combined capacity including bending ([2])

What gives overall 504 possible connection combinations.

5.3. Program manual

5.3.1. Adding developed module to Soldis Projektant

On attached CD, module is already added and ready-to-use with portable version of Soldis Projektant.

In order to manually install module, copy self-extracting archive "TimberConnection.exe" to your Soldis Projektant main folder (by default "C:\Program Files (x86)\Soldis PROJEKTANT") and extract it with default settings. Allow files overwriting when prompted.

5.3.2. Adding module inside Soldis Projektant

First step is defining structure, in which designing the timber connection will be possible. Exemplary structure (Project calculated as part II of this thesis) is included on the CD and ready to open. For detailed rules explaining drawing and customizing structure in Soldis Projektant consult Soldis Projektant Manual.

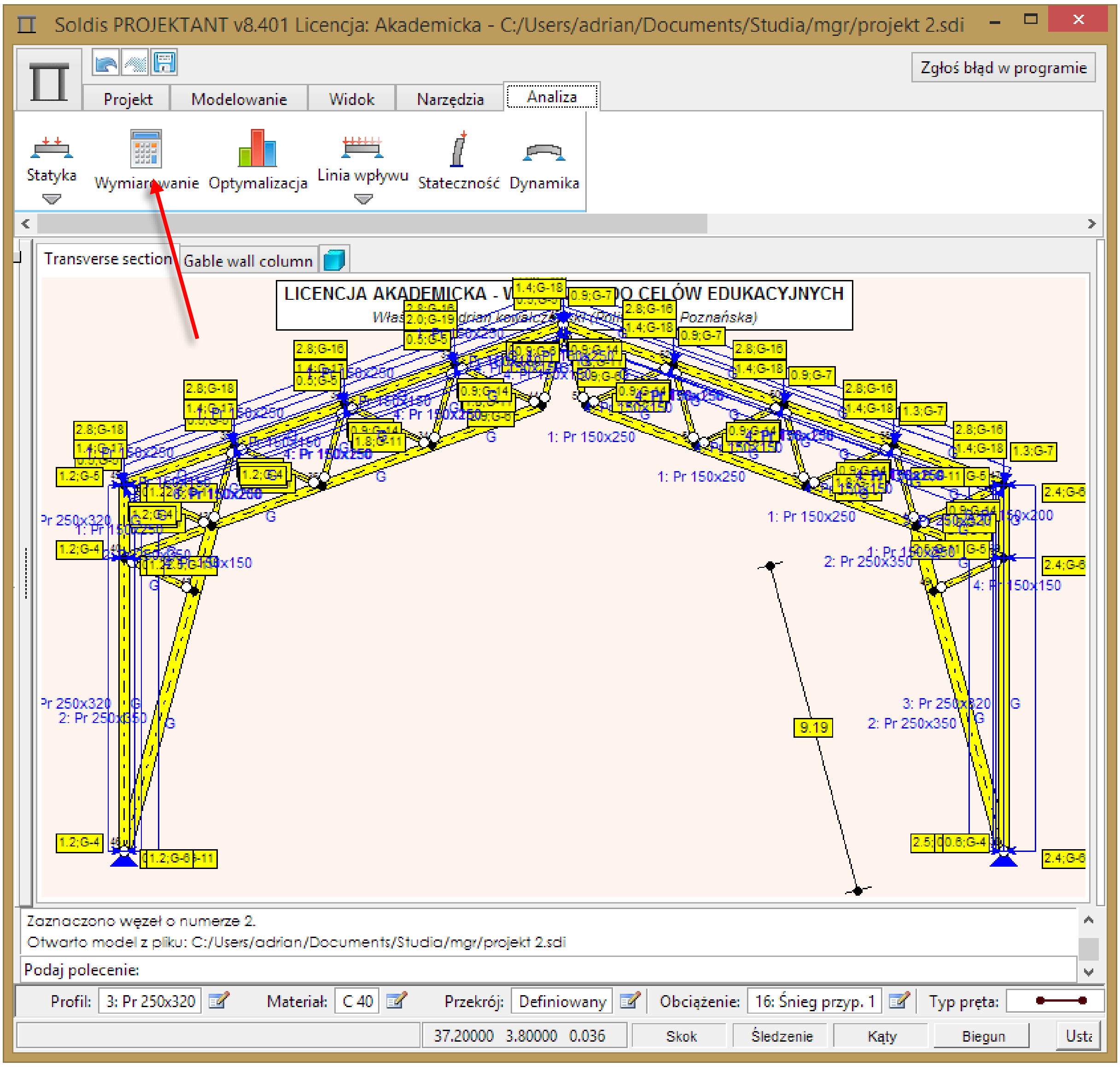


Figure 11

After defining (or opening) structure, proceed to *Analiza* module, and pick *Wymiarowanie* (Figure 12). After *Wymiarowanie* module will load, choose *Dodaj regułę* and pick Timber Connection EN 1995 (Figure 13). Pick node on which connection is about to be designed, and click *Zastosuj* (Figure 14).

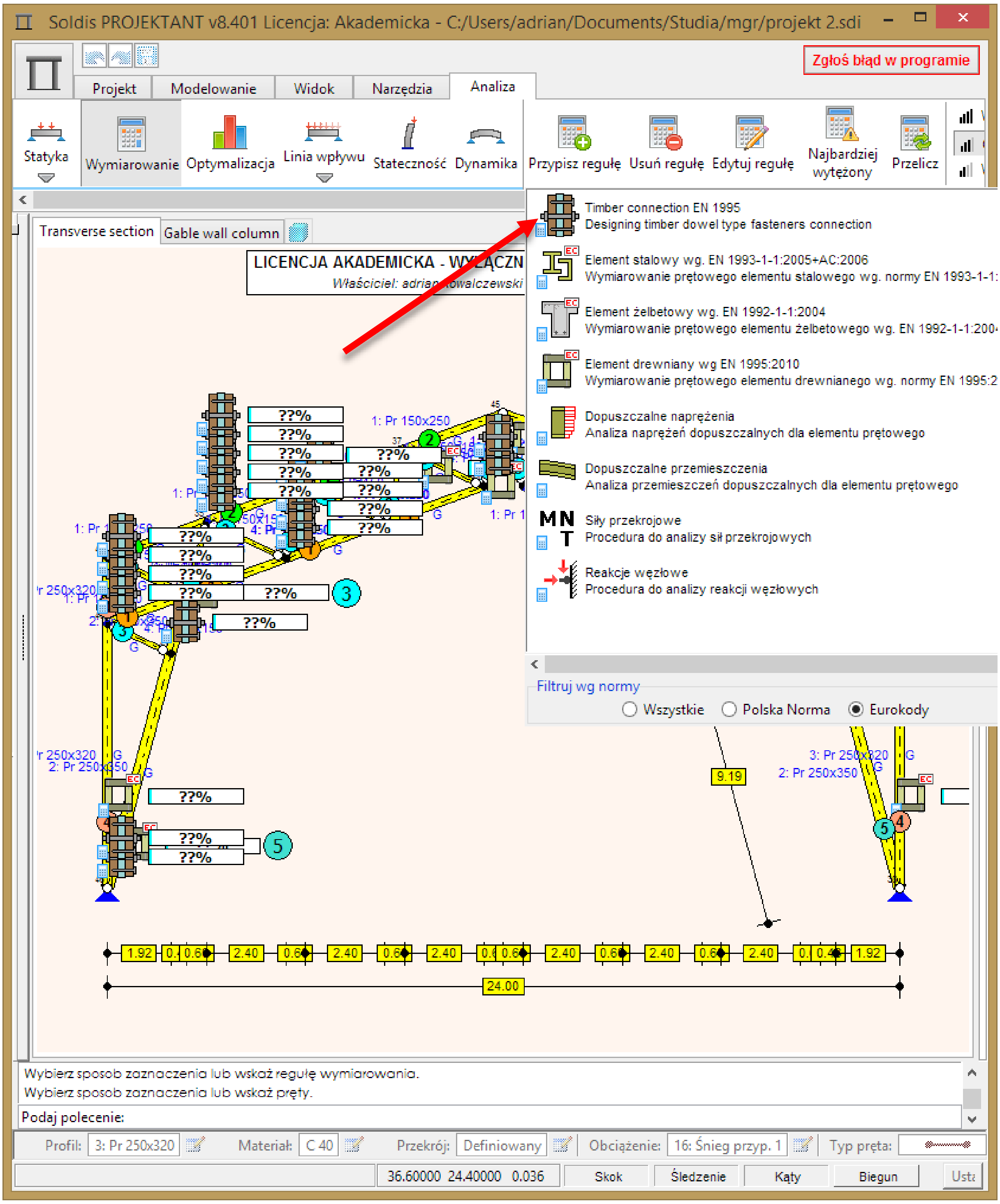

Soldis PROJEKTANT v8.401 Licencja: Akademicka - C:/Users/adrian/Documents/Studia/mgr/projekt 2.sdi
Zgłoś błąd w programie
Projekt
Modelowanie
Widok
Narzędzia
Analiza
Statyka
Wymiarowanie
Optymalizacja
Linia wpływu
Stateczność
Dynamika
Przypisz regułę
Usuń regułę
Edytuj regułę
Najbardziej wytężony
Przelicz
Transverse section
Gable wall column
Timber connection EN 1995
Designing timber dowel type fasteners connection
Element stalowy wg. EN 1993-1-1:2005+AC:2006
Element żelbetowy wg. EN 1992-1-1:2004
Element drewniany wg EN 1995:2010
Dopuszczalne naprężenia
Analiza naprężeń dopuszczalnych dla elementu prętowego
Dopuszczalne przemieszczenia
Analiza przemieszczeń dopuszczalnych dla elementu prętowego
Siły przekrojowe
Procedura do analizy sił przekrojowych
Reakcje węzłowe
Procedura do analizy reakcji węzłowych
Filtruj wg normy
Wszystkie
Polska Norma
Eurokody
Wybierz sposób zaznaczenia lub wskaż regułę wymiarowania.
Wybierz sposób zaznaczenia lub wskaż pręty.
Podaj polecenie:
Profil:
3: Pr 250x320
Materiał:
C 40
Przekrój:
Definiowany
Obciążenie:
16: Śnieg przyp. 1
Typ pręta:
36.60000 24.40000 0.036
Skok
Śledzenie
Kąty
Biegun

Figure 12

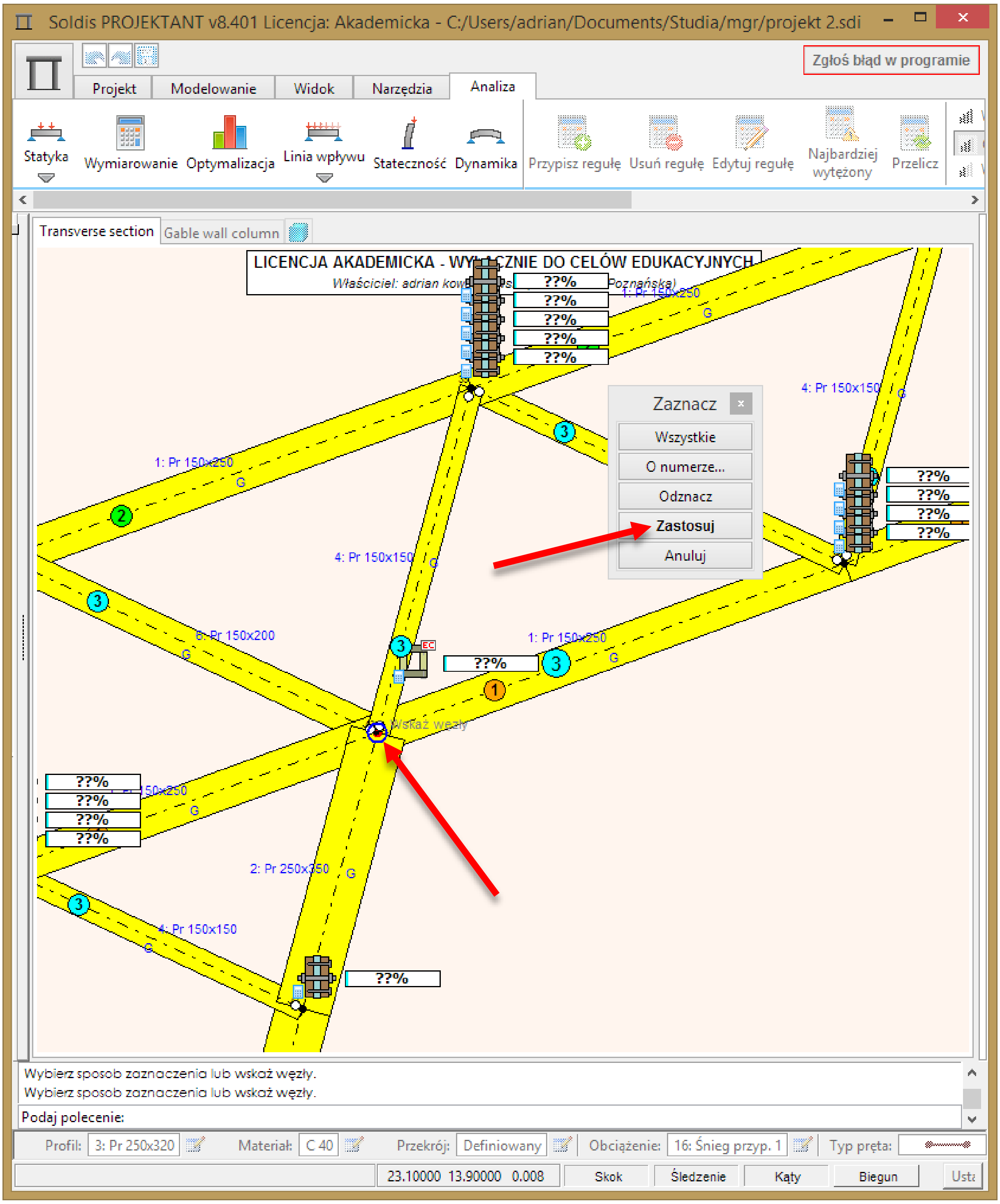


Figure 13

After following steps, module window will open (Figure 15)

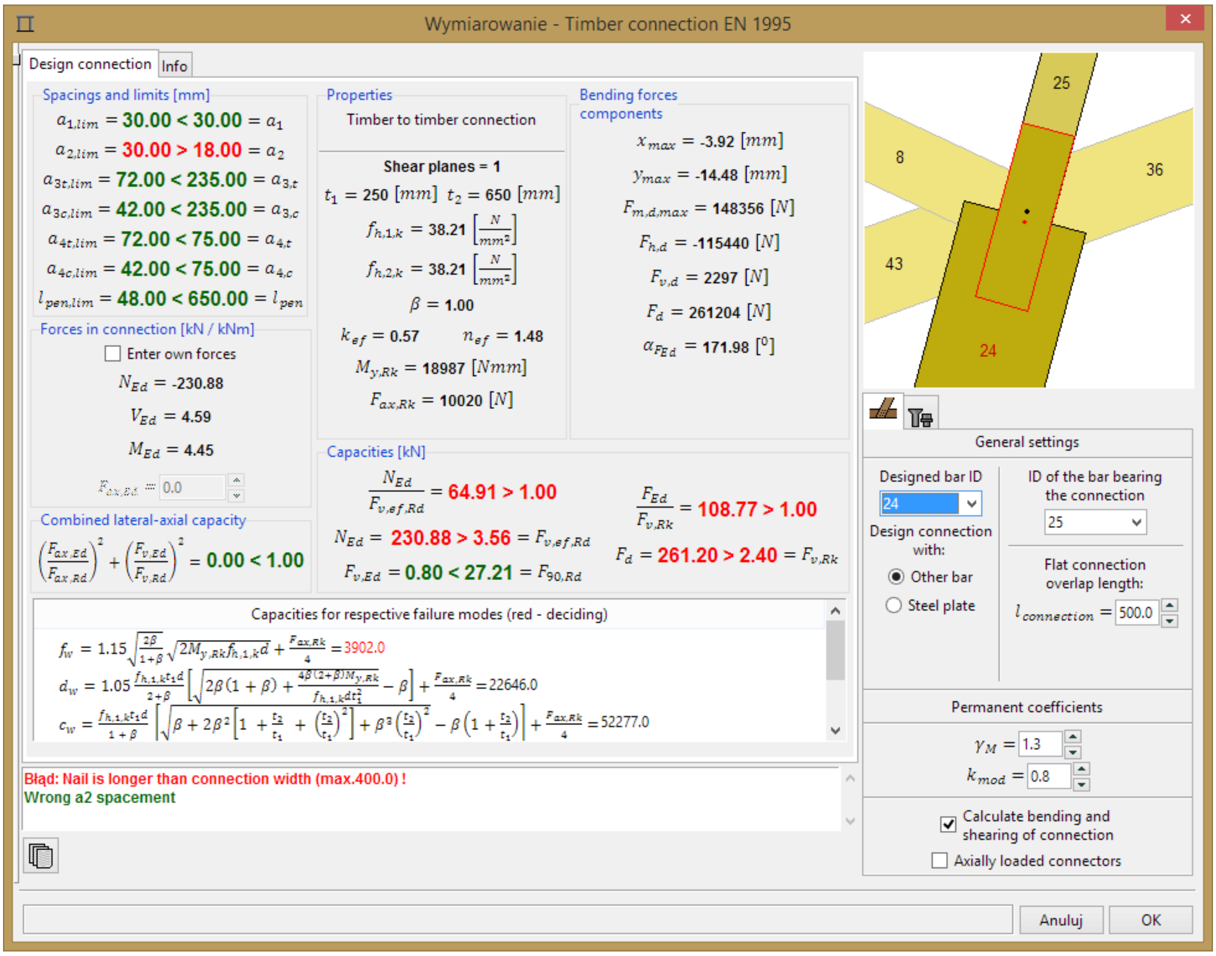


Figure 14

Each category of results is grouped inside labelled frames. Module will load with default values, which in all likelihood won't fulfill any requirements. Step-by-step procedure describing use of module will be now described.

5.3.3. Step-by-step procedure of using module

1. Begin with first tab of side bar. Tab is called "General Settings". Select bar which forces are about to be transferred with designed connection (Figure 17). List is containing all elements reaching the selected node. Use overview graph (Figure 16) to identify bar of interest.

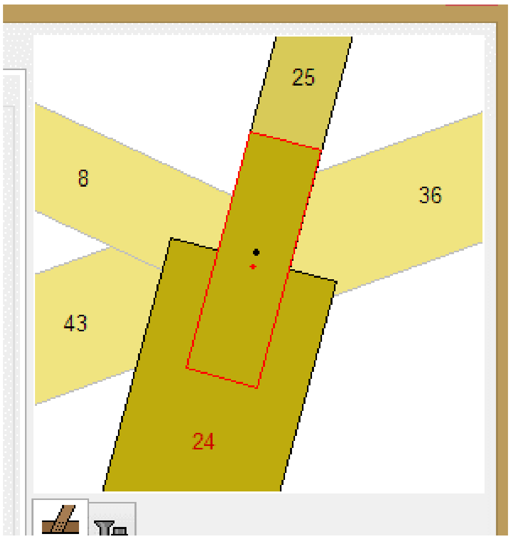


Figure 15

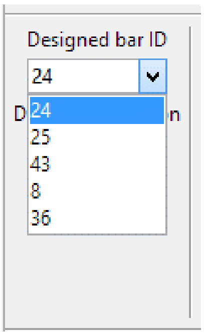


Figure 16

2. Pick type of connection of interest (either connection with Other bar or steel plate) If Other bar is selected, select the bar and if connection is flat – select its overlapping length

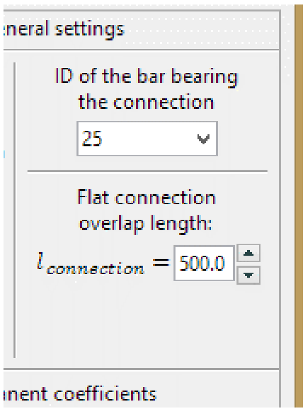


Figure 17

If steel plate is selected, determine steel plate parameters (Figure 19). Notice that overview graph will update, showing idea of shape of designed steel plate. If

Designed Element is one branch element, steel plates will show outside the element (dark plate color, straight bold line outline). If Designed Element will be two branch element, steel plate will show inside the element, what is recognizable by soft color and dashed outline. Plate thickness will also be locked, as it is needed for plate to fit inside branches space, what is calculated automatically. When steel plates are selected, they will be drawn on other elements, just to give plain idea of connection look. Option “Steel plate inside hole drilled in profile” will proceed to designing concealed plate, thus treating one profile element as two profile element with space inside equal tplate (More info in 4.2.2 section)

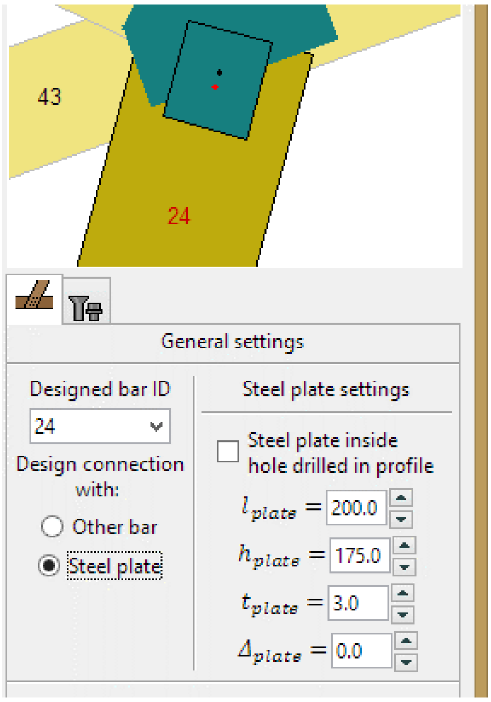


Figure 18

3. Enter desired γ_M and k_{mod} values, if other than default (Figure 20).

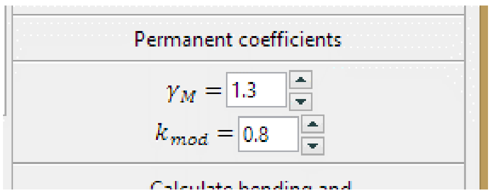


Figure 19

4. Select desired additional analysis types – Calculation of bending and shearing of connection and/or Axially loaded connectors (i.e. loaded by force perpendicular to connection plane) (Figure 21). If axially loaded analysis is chosen, enter overall axial connectors load inside $F_{ax,Ed}$ field, which will become available (Figure 22)

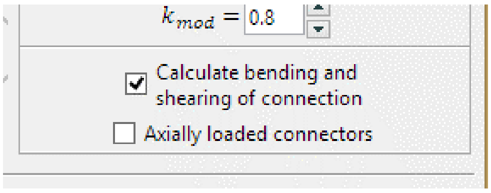


Figure 20

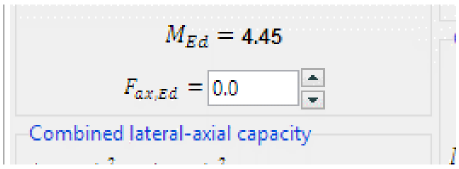


Figure 21

5. Proceed to second tab of side panel. Tab is called “Connector settings” (Figure 23).

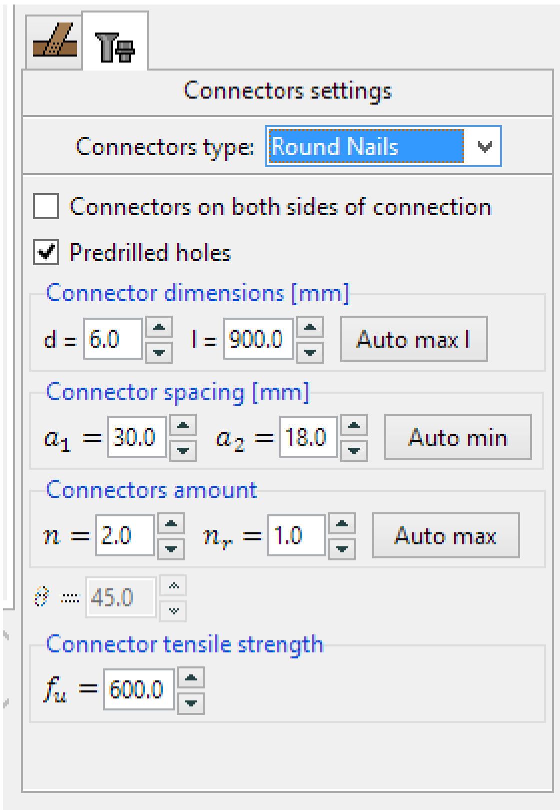


Figure 22

6. Begin with selecting desired connectors type. Notice that tab will shift depending on choice. Some options will become grayed-out with specific connector type, as they can’t be designed in such manner, or given factor is unused with given connector type.

Connectors on both sides of connection will assume that connectors amount is doubled and connectors are used on both sides of connection (front, which is visible on graphical preview and back, which is mirror image of front)

It is recommended to always use predrilled holes due to drastic a_{lim} decreasing. This example will design connection with bolts.

7. Set desired connector diameter and its length button "Auto max l" will automatically set connector length to overall connection width. Connector length should be set in a way that its penetration length will be greater than limit value.

   Notice that when bolts are selected this frame will be swapped to frame with bolt diameter and class (Figure 24). There is no need to set bolt length as it is always greater than connection length.

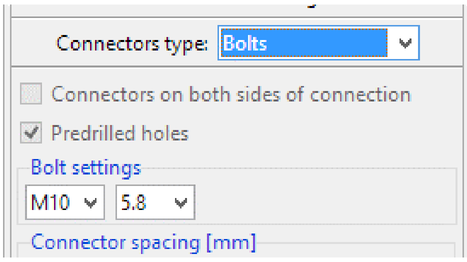


Figure 23

Notice that when staples are selected, θ control will be available (Figure 25). Set staple crown angle to grain direction with it.

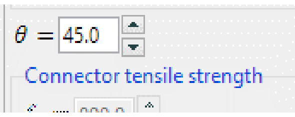


Figure 24

8. Set connectors spacing (Figure 26). They should be greater than limits displayed in "Spacings and limits [mm]" frame (Figure 27). Use "Auto min" button to automatically set $a_1=a_{1,lim}$ and $a_2=a_{2,lim}$. Notice, that overview graph will update as spacings will be changed.

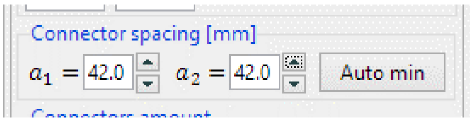


Figure 25

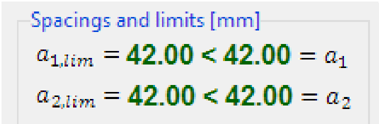


Figure 26

9. Set connectors amount in a row and amount of rows of connectors (Figure 28). Connectors must be within allowable area, determined by four a_{lim} limits (Figure 29). Button "Auto max" will fill connection area with as many connectors as possible with respect to a_{lim} requirements.

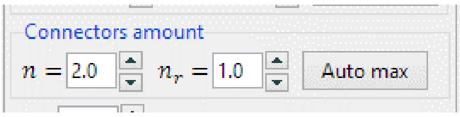


Figure 27

Figure 28

10. If it is available for selected connector type, set connector tensile strength (figure 30)

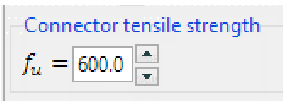


Figure 29

11. After setting all values, check if capacity requirements are fulfilled (Figure 31). If they are positive, your connection is designed properly. If not, try changing connector diameter, connectors amount, increase k_{ef} value to maximum (a_1 value). Notice which mode of failure is deciding one, and try increasing values of coefficients in this mode (Figure 31). If capacity check is still negative, change elements widths, in order to increase connection area.

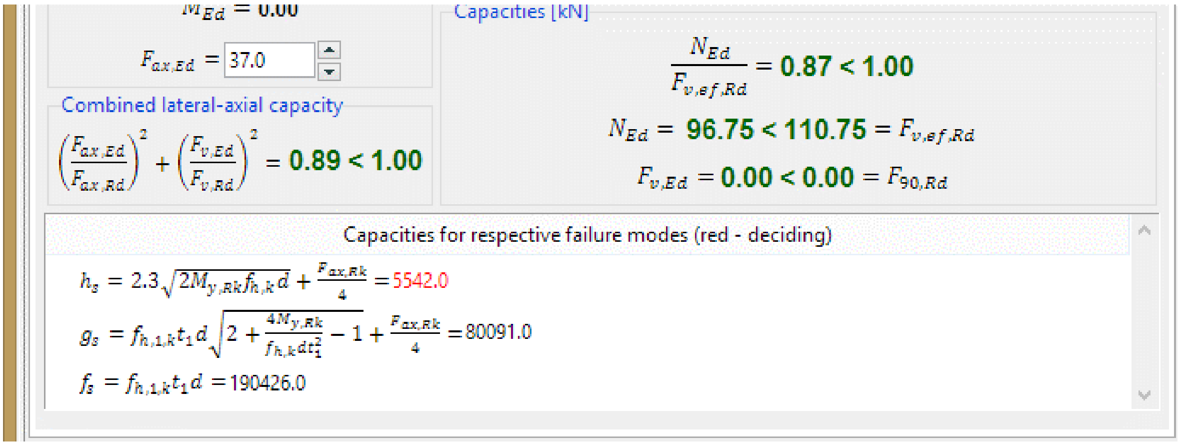


Figure 30

6. Bibliography

[1] Norm: EN 1995-1-1:2004

[2] *STRUCTURAL TIMBER DESIGN to Eurocode 5*, J. Porteous, A. Kermani , Ed. 2nd , Wiley-Blackwell, ISBN: 978-0-470-67500-7

[3] Norm: EN 1993-1-8

[4] *Manual for the design of timber building structures to Eurocode 5*, A. Page, Ed. 1st, IStructE / TRADA, ISBN 978-0-901297-440

[5] *Budownictwo ogólne, tom 4, konstrukcje budynków*, und. red. W. Buczkowski, Arkady, Ed. 1st , ISBN 978-83-213-4591-8

[6] *Dive into python*, M. Pilgrim, internet acces: http://diveintopython.org/