
etacad

Release 0.0.13

Kevin Axel Tagliaferri

Jun 12, 2025

CONTENTS:

Etacad is a module aimed at simplifying and/or automating the drawing of structural element plans based on the ezdxs library. Etacad is not a structural calculation module; it only performs drawing functions and material computation.

GETTING STARTED

1.1 Installation

If you have pip:

```
pip install etacad
```

1.2 Requirements

- ezdxf

1.3 Links

Visit the repository and tutorials on YT.

- [GitHub](#)
- [Pypi](#)
- [Video Tutorials](#)

For the following examples, we will use variable 'doc' as an instance of `ezdxf.new()`.

```
import ezdxf
```

2.1 Creating a new document

Pass the setup parameter as `True` to use the dimensions and text styles from the `ezdxf` module. By default, `etacad` uses them.

```
doc = ezdxf.new("R2010", setup=True)
```

2.2 Creating a Beam instance

Create a simple concrete beam with *Beam* class.

```
# Importing Beam class.
from etacad import Beam

# Creating a basic Beam instance.
beam = Beam(width=.2,
            height=.35,
            length=4,
            x=1,
            y=2,
            as_sup={.008: 2},
            as_inf={.012: 2, .008: 1},
            stirrups_db=.008,
            stirrups_sep=.15,
            cover=.025)
```

The *x* and *y* parameters correspond to the default point where the drawings will be plotted unless specified when executing the drawing methods.

2.2.1 Beam bars

You can add more steel bars by specifying their diameter as the key and the quantity as the dictionary values. These will be symmetrically arranged along the width or effective height of the section. You can add bars vertically using the parameters *as_right* and *as_left*. The cover is considered at the bar's centerline.

```
beam = Beam(width=.2,
            height=.35,
            length=4,
            as_sup={.01: 2, .008: 1},
            as_right={.008: 2},
            as_inf={.012: 2, .008: 1},
            as_left={.008: 2},
            stirrups_db=.008,
            stirrups_sep=.15,
            cover=.025) # The cover is considered at the bar's centerline.
```

You can also add an anchorage length for the bars using *anchor_sup*, *anchor_right*, *anchor_inf*, and *anchor_left*, passing a single value for all the bars on the side or a list specifying a value for each bar.

2.2.2 Beam stirrups

The stirrups are handled differently than the bars. You can specify a single diameter and spacing (as in the previous example), which will distribute the stirrups along the beam, or you can provide a list of diameters, requiring additional parameters such as the X coordinate where each stirrup begins and the reinforcement length. Additionally, you can provide a list of the anchorage lengths for each stirrup or a float for all stirrups.

```
beam = Beam(width=.2,
            height=.35,
            length=4,
            as_sup={.01: 2, .008: 1},
            as_inf={.012: 2, .008: 1},
            stirrups_db=[.008, .01, .008],
            stirrups_sep=[.15, .1, .15],
            stirrups_length=[.6, 2, .6],
            stirrups_anchor=.125,
            stirrups_x=[.2, 1, 3.2],
            cover=.025)
```

2.2.3 Beam columns intersections

Intersections of the columns can be taken into account; we need to introduce three additional parameters: *columns*, *columns_pos*, and *columns_symbol*, the latter being optional.

```
beam = Beam(width=.2,
            height=.35,
            length=4,
            as_sup={.008: 2},
            as_inf={.012: 2, .008: 1},
            stirrups_db=.008,
            stirrups_sep=.15,
            columns=[(.2, .35), (.25, .35)],
            columns_pos=[0, 3.65],
            columns_symbol=["C1", "C2"],
            cover=.025)
```

2.2.4 Longitudinal drawing of the beam

The Beam class offers methods for drawing longitudinal and cross sections, which are customizable through the method parameters and the *settings* parameter. Each method returns a dictionary that may or may not contain other nested dictionaries (depending on the method and class). The keys that do not have dictionaries contain a list of DXF entities.

```
beam.draw_longitudinal(document=doc,  
                        x=2,  
                        y=1,  
                        unifilar_bars=False)
```

See more about the parameters of the *Beam.draw_longitudinal()* function for further details.

2.2.5 Transversal drawing of the beam

```
beam.draw_transverse(document=doc,  
                     x=2,  
                     y=1,  
                     unifilar=False)
```

See more about the parameters of the *Beam.draw_transverse()* function for further details.

2.2.6 Longitudinal rebar detailing drawing

```
beam.draw_longitudinal_rebar_detailing(document=doc,  
                                       x=2,  
                                       y=1,  
                                       unifilar=True)
```

See more about the parameters of the *Beam.draw_longitudinal_rebar_detailing()* function for further details.

2.2.7 Transversal rebar detailing drawing

```
beam.draw_transverse_rebar_detailing(document=doc,  
                                     x=2,  
                                     y=1,  
                                     unifilar=True)
```

See more about the parameters of the *Beam.draw_transverse_rebar_detailing()* function for further details.

2.2.8 Table rebar detailing

```
beam.draw_table_rebar_detailing(document=doc,  
                               x=2,  
                               y=1)
```

See more about the parameters of the *Beam.draw_table_rebar_detailing()* function for further details.

2.3 Creating a Column instance

Create a simple concrete column with *Column* class.

```
# Importing Column class.
from etacad import Column

# Creating a basic Column instance.
column = Column(width=.25,
                depth=.25,
                height=6,
                x=2,
                y=2,
                as_sup={.012: 3},
                as_right={.012: 1},
                as_inf={.012: 2},
                as_left={.012: 1},
                stirrups_db=.008,
                stirrups_sep=.15,
                cover=.025)
```

The *x* and *y* parameters correspond to the default point where the drawings will be plotted unless specified when executing the drawing methods.

2.3.1 Column bars

It works the same as the *Beam bars* section. The consideration here is that symmetrical columns are generally used, and all the bar parameters (*as_sup*, *as_right*, *as_inf*, *as_left*) should be utilized. It must be noted that the bars at the edges apply to two faces of the section, as considered in this example.

2.3.2 Column stirrups

It works the same as the *Beam stirrups* section.

2.3.3 Column beams intersections

Intersections of the beams can be taken into account; we need to introduce three additional parameters: *beams*, *beams_pos*, and *beams_symbol*, the latter being optional.

```
column = Column(width=.25,
                depth=.25,
                height=6,
                x=2,
                y=2,
                as_sup={.012: 3},
                as_right={.012: 1},
                as_inf={.012: 2},
                as_left={.012: 1},
                stirrups_db=.008,
                stirrups_sep=.15,
                beams=[(.25, .25), (.25, .25)],
                beams_pos=[0, 5.75],
                beams_symbol=["B1", "B2"],
                cover=.025)
```

2.3.4 Longitudinal drawing of the column

The Column class offers methods for drawing longitudinal and cross sections, which are customizable through the method parameters and the *settings* parameter. Each method returns a dictionary that may or may not contain other nested dictionaries (depending on the method and class). The keys that do not have dictionaries contain a list of DXF entities.

```
column.draw_longitudinal(document=doc,
                        x=3,
                        y=1,
                        unifilar_bars=False)
```

See more about the parameters of the *Column.draw_longitudinal()* function for further details.

2.3.5 Transversal drawing of the column

```
column.draw_transverse(document=doc,
                      x=5,
                      y=1,
                      unifilar=False)
```

See more about the parameters of the *Column.draw_transverse()* function for further details.

2.3.6 Longitudinal rebar detailing drawing

```
column.draw_longitudinal_rebar_detailing(document=doc,
                                         x=2,
                                         y=1,
                                         unifilar=True)
```

See more about the parameters of the *Column.draw_longitudinal_rebar_detailing()* function for further details.

2.3.7 Transversal rebar detailing drawing

```
column.draw_transverse_rebar_detailing(document=doc,
                                       x=6,
                                       y=1,
                                       unifilar=True)
```

See more about the parameters of the *Column.draw_transverse_rebar_detailing()* function for further details.

2.3.8 Table rebar detailing

```
column.draw_table_rebar_detailing(document=doc,
                                  x=2,
                                  y=1)
```

See more about the parameters of the *Column.draw_table_rebar_detailing()* function for further details.

API REFERENCES

API References.

```
class etacad.bar.Bar(reinforcement_length: float, diameter: float, x: float = 0, y: float = 0, direction:
    Direction = Direction.HORIZONTAL, orientation: Orientation =
    Orientation.BOTTOM, transverse_center: tuple = None, left_anchor: float = 0,
    right_anchor: float = 0, mandrel_radius: float = 0, bend_longitud: float = 0,
    bend_angle: float = 0, bend_height: float = 0, element_type: ElementTypes =
    ElementTypes.BAR, denomination: str = None, position: str = None, quantity: float =
    None)
```

Longitudinal bar element, computes geometrics and physics props and manages dxf drawing methods.

Parameters

- **reinforcement_length** (*float*) – Length of the stirrup reinforcement.
- **diameter** (*float*) – Diameter of stirrup bar.
- **x** (*float*) – X coordinate of the bottom left corner Stirrup bounding box point.
- **y** (*float*) – Y coordinate of the bottom left corner Stirrup bounding box point.
- **left_anchor** (*float*) – Left anchor length of stirrup.
- **right_anchor** (*float*) – Right anchor length of stirrup.
- **mandrel_radius** (*float*) – Mandrel radius of bar.
- **direction** (*Direction*) – Direction of the stirrup (Horizontal or Vertical).
- **orientation** (*Orientation*) – Orientation of the stirrup (top, right, down, left).
- **bend_longitud** – Bending longitud of the bar at center of bar.
- **bend_angle** (*float*) – Bending declination angle.
- **bend_height** (*float*) – Bending height of the bar at center of bar.
- **transverse_center** (*tuple*) – Transverse center of the drawing of cross-section.

Type

float

Variables

- **reinforcement_length** – Total length of the reinforcement bar.
- **diameter** – Diameter of the bar.
- **radius** – Radius of the bar (half of the diameter).
- **length** – Overall length of the bar, including bends and anchors.
- **x** – X coordinate of the bottom left corner of the bounding box.
- **y** – Y coordinate of the bottom left corner of the bounding box.
- **direction** – Direction of the bar (horizontal or vertical).

- **orientation** – Positioning of the bar (top, right, down, left).
- **transverse_center** – Coordinates of the transverse center of the cross-section.
- **left_anchor** – Length of the left anchor.
- **right_anchor** – Length of the right anchor.
- **mandrel_radius** – Radius of the mandrel used for bending.
- **mandrel_radius_ext** – External radius of the mandrel plus the bar's diameter.
- **bend_longitud** – Length of the bar bend at its center.
- **bend_angle** – Angle of the bar bend.
- **bend_height** – Height of the bar at the bend.
- **bending_projection** – Projection length caused by the bar's bend.
- **box_width** – Width of the bounding box containing the bar.
- **box_height** – Height of the bounding box containing the bar.
- **weight** – Weight of the bar, calculated using a steel density of 7850 kg/m³.
- **element_type** – Type of the element (BAR).
- **denomination** – Optional denomination for the bar element.

data() → dict

Collects and returns the essential attributes of the bar element in a dictionary format.

Returns

Dictionary containing key attributes of the bar such as denomination, length, diameter and weight.

Return type

dict

draw_longitudinal(*document: Drawing, x: float = None, y: float = None, unifilar: bool = False, dimensions: bool = True, denomination: bool = True, settings: dict = {'text_denomination_distance': 0.1, 'text_denomination_height': 0.05, 'text_dim_distance_horizontal': 0.05, 'text_dim_distance_vertical': 0.05, 'text_dim_height': 0.05}*) → dict

Draws the longitudinal view of the bar in a DXF document.

Parameters

- **document** (*Drawing*) – The DXF document to draw on.
- **x** (*float, optional*) – X coordinate for the drawing, defaults to self.x.
- **y** (*float, optional*) – Y coordinate for the drawing, defaults to self.y.
- **unifilar** (*bool, optional*) – Whether to draw a unifilar representation (simplified view), defaults to False.
- **dimensions** (*bool, optional*) – Whether to include dimensions in the drawing, defaults to True.
- **denomination** (*bool, optional*) – Whether to include the denomination label, defaults to True.
- **settings** (*dict, optional*) – Dictionary of settings for dimensioning. Defaults to *BAR_SET_LONG*.

Returns

Dict of drawing entities for the longitudinal view.

Return type

dict

draw_transverse(*document: Drawing*, *x: float = None*, *y: float = None*, *settings: dict = {'text_denomination_distance': 0.1, 'text_denomination_height': 0.05, 'text_dim_distance_horizontal': 0.05, 'text_dim_distance_vertical': 0.05, 'text_dim_height': 0.05}*) → dict

Draws the transverse section of the bar in a DXF document.

Parameters

- **document** (*Drawing*) – The DXF document to draw on.
- **x** (*float, optional*) – X coordinate for the drawing, defaults to self.x.
- **y** (*float, optional*) – Y coordinate for the drawing, defaults to self.y.
- **settings** (*dict, optional*) – Dictionary of settings for dimensioning. Defaults to *BAR_SET_TRANSVERSE*.

Returns

Dict of drawing entities for the transverse section.

Return type

dict

extract_data(*labels: list[str] = None*)

Extracts specific data attributes based on the provided list of labels. If no labels are provided, it defaults to extracting “denomination”, “length”, “diameter”, and “weight”.

Parameters

labels (*list[str], optional*) – List of attribute names to extract. Defaults to common attributes if not provided.

Returns

A list of values corresponding to the requested labels.

Return type

list

class etacad.stirrup.**Stirrup**(*width: float, height: float, diameter: float, reinforcement_length: float, spacing: float, x: float = 0, y: float = 0, mandrel_radius_top: float = 0, mandrel_radius_bottom: float = 0, anchor: float = 0, direction: Direction = Direction.HORIZONTAL, orientation: Orientation = Orientation.BOTTOM, element_type: ElementTypes = ElementTypes.STIRRUP, denomination: str = None, position: str = None*)

Stirrup element, computes geometrics and physics props and manages dxf drawing methods.

Parameters

- **width** (*float*) – External width of stirrup.
- **height** (*float*) – External height of stirrup.
- **diameter** (*float*) – Diameter of stirrup bar.
- **reinforcement_length** (*float*) – Length of the stirrup reinforcement.
- **spacing** (*float*) – Stirrup spacing.
- **x** (*float*) – X coordinate of the bottom left corner Stirrup bounding box point.
- **y** (*float*) – Y coordinate of the bottom left corner Stirrup bounding box point.
- **mandrel_radius_top** (*float*) – Mandrel radius of stirrup bar at the top.
- **mandrel_radius_bottom** (*float*) – Mandrel radius of stirrup bar at the bottom.
- **anchor** (*float*) – Anchor length of stirrup.
- **direction** (*str*) – Direction of the stirrup (Horizontal or Vertical).

- **orientation** (*str*) – Orientation of the stirrup (top, right, down, left).

Variables

- **reinforcement_length** – Length of the stirrup reinforcement.
- **spacing** – Stirrup spacing.
- **mandrel_radius_top** – Mandrel radius of stirrup bar at the top.
- **mandrel_radius_ext_top** – Mandrel radius plus diameter of stirrup bar at the top.
- **mandrel_radius_bottom** – Mandrel radius of stirrup bar at the bottom.
- **mandrel_radius_ext_bottom** – Mandrel radius plus diameter of stirrup bar at the bottom.
- **anchor** – Anchor length of stirrup.
- **quantity** – Number of stirrups that forms de reinforcement.
- **diameter** – Diameter of stirrup bar.
- **direction** – Direction of the stirrup (Horizontal or Vertical).
- **orientation** – Orientation of the Stirrup (top, right, down, left).
- **length** – Length of the bar to fork the stirrup.
- **weight** – Weight of the stirrup, considering 7850 kg / m3.
- **box_width** – Width of the box that contains the stirrup.
- **box_height** – Height of the box that contains the stirrup.

data()

Collects and returns the essential attributes of the stirrup element in a dictionary format.

Returns

Dictionary containing key attributes of the bar such as denomination, length, diameter, weight and

quantity. :rtype: dict

draw_longitudinal (*document: Drawing, x: float = None, y: float = None, unifilar=True*) → dict

Draw the longitudinal reinforcement of the stirrup in the dxf file.

Parameters

- **document** (*Drawing*) – Document in which it will be drawn.
- **x** (*float*) – X coordinate of the bottom corner of the drawing.
- **y** (*float*) – Y coordinate of the bottom corner of the drawing.
- **unifilar** (*bool*) – Single-line drawing.

Returns

None.

Return type

None

draw_transverse (*document: Drawing, x: float = None, y: float = None, unifilar: bool = False, dimensions: bool = False, settings: dict = {'dim_style': 'EZ_M_10_H25_CM', 'text_dim_distance_anchor': 0.1, 'text_dim_distance_horizontal': 0.05, 'text_dim_distance_vertical': 0.1, 'text_distance_length_count': 0.1, 'text_length_count_height': 0.05}*) → dict

Draw the cross-section of the stirrup in the dxf file.

Parameters

- **document** (*Drawing*) – Document in which it will be drawn.
- **x** (*float*) – X coordinate of the bottom corner of the drawing.
- **y** (*float*) – Y coordinate of the bottom corner of the drawing.
- **unifilar** (*bool*) – Single-line drawing.
- **dimensions** (*bool*) – Dimensions drawing.
- **settings** (*dict, optional*) – Dictionary of settings for dimensioning. Defaults to *STIRRUP_SET_TRANSVERSE*.

Returns

None.

Return type

None

extract_data(*labels: list[str] = None*)

Extracts specific data attributes based on the provided list of labels. If no labels are provided, it defaults to extracting “denomination”, “length”, “diameter”, “weight” and quantity.

Parameters

labels (*list[str], optional*) – List of attribute names to extract. Defaults to common attributes if not provided.

Returns

A list of values corresponding to the requested labels.

Return type

list

class etacad.concrete.**Concrete**(*vertices: list[tuple[float, float]], height: float = None, length: float = None, x: float = 0.0, y: float = 0.0, specific_weight: float = 2400, element_type: ElementTypes = ElementTypes.CONCRETE*)

A class used to represent a concrete section with various geometric and physical attributes.

Parameters

- **vertices** (*list[tuple[float, float]]*) – List of tuples representing the vertices of the concrete polygon.
- **height** (*float, optional*) – Height of the concrete section. Defaults to None.
- **length** (*float, optional*) – Length of the concrete section. Defaults to None.
- **x** (*float, optional*) – X-coordinate for the concrete section’s placement. Defaults to 0.0.
- **y** (*float, optional*) – Y-coordinate for the concrete section’s placement. Defaults to 0.0.
- **specific_weight** (*float, optional*) – Specific weight of the concrete. Defaults to *CONCRETE_WEIGHT*.
- **element_type** (*ElementTypes, optional*) – Type of the structural element (e.g., *CONCRETE*).

Variables

- **dim3D** (*float*) – The third dimension (either height or length) of the concrete section.
 - **volume** (*float*) – Volume of the concrete section.
 - **weight** (*float*) – Weight of the concrete section.
 - **box_width** (*float*) – Width of the bounding box in the longitudinal direction.
-

- **box_height** (*float*) – Height of the bounding box in the longitudinal direction.
- **box_width_transverse** (*float*) – Width of the bounding box in the transverse direction.
- **box_height_transverse** (*float*) – Height of the bounding box in the transverse direction.
- **element_type** (*ElementTypes*, *optional*) – Type of the structural element (e.g., CONCRETE).

Raises

TypeError – If neither height nor length is provided.

3.1 Methods

draw_longitudinal(*document*, *x=None*, *y=None*, *dimensions=True*, *dimensions_inner=True*, *settings=CONCRETE_SET_LONG*)

Draws the concrete section in the longitudinal direction with optional dimensioning.

draw_transverse(*document*, *x=None*, *y=None*, *dimensions=True*, *dimensions_boxing=True*, *dimensions_inner=True*,

settings=CONCRETE_SET_TRANSVERSE)

Draws the concrete section in the transverse direction with optional dimensioning.

polygon

Property method to return a Polygon object created from the vertices of the concrete section.

draw_longitudinal(*document: Drawing*, *x: float = None*, *y: float = None*, *dimensions: bool = True*, *dimensions_inner: bool = True*, *settings: dict = {'dim_style_boxing': 'EZ_M_10_H25_CM', 'dim_style_inner': 'EZ_M_10_H25_CM', 'text_dim_distance_horizontal': 0.25, 'text_dim_distance_vertical': 0.25}*) → dict

Draws the concrete section in the longitudinal direction with optional dimensioning.

Parameters

- **document** (*Drawing*) – The drawing document to which the concrete section will be added.
- **x** (*float*, *optional*) – X-coordinate for the placement of the concrete section. Defaults to None, in which case *self.x* is used.
- **y** (*float*, *optional*) – Y-coordinate for the placement of the concrete section. Defaults to None, in which case *self.y* is used.
- **dimensions** (*bool*, *optional*) – Flag to indicate whether to draw dimensions. Defaults to True.
- **dimensions_inner** (*bool*, *optional*) – Flag to indicate whether to draw inner dimensions. Defaults to True.
- **settings** (*dict*, *optional*) – Dictionary of settings for dimensioning. Defaults to *CONCRETE_SET_LONG*.

Returns

A dictionary with keys “concrete_elements”, “dimensions”, and “all_elements”, each containing the

corresponding drawing elements. :rtype: dict

draw_transverse(*document*: *Drawing*, *x*: *float* = *None*, *y*: *float* = *None*, *dimensions*: *bool* = *True*, *dimensions_boxing*: *bool* = *True*, *dimensions_inner*: *bool* = *False*, *settings*: *dict* = {'dim_style_boxing': 'EZ_M_10_H25_CM', 'dim_style_inner': 'EZ_M_10_H25_CM', 'text_dim_distance_horizontal': 0.25, 'text_dim_distance_vertical': 0.25, 'text_dim_inner_perpendicular_distance': 0.05}) → *dict*

Draws the concrete section in the transverse direction with optional dimensioning.

Parameters

- **document** (*Drawing*) – The drawing document to which the concrete section will be added.
- **x** (*float*, *optional*) – X-coordinate for the placement of the concrete section. Defaults to *None*, in which case *self.x* is used.
- **y** (*float*, *optional*) – Y-coordinate for the placement of the concrete section. Defaults to *None*, in which case *self.y* is used.
- **dimensions** (*bool*, *optional*) – Flag to indicate whether to draw dimensions. Defaults to *True*.
- **dimensions_boxing** (*bool*, *optional*) – Flag to indicate whether to draw boxing dimensions. Defaults to *True*.
- **dimensions_inner** (*bool*, *optional*) – Flag to indicate whether to draw inner dimensions. Defaults to *False*.
- **settings** (*dict*, *optional*) – Dictionary of settings for dimensioning. Defaults to *CONCRETE_SET_TRANSVERSE*.

Returns

A dictionary with keys “concrete_elements” and “all_elements”, each containing the corresponding drawing elements.

Return type

dict

property polygon

Property to get a Polygon object created from the vertices of the concrete section.

Returns

A Polygon object representing the concrete section.

Return type

Polygon

```
class etacad.beam.Beam(width, height, length, x: float = 0, y: float = 0, direction: Direction =
    Direction.HORIZONTAL, orientation: Orientation = Orientation.RIGHT,
    number_init: int = None, as_sup: dict = None, anchor_sup: float | list = 0,
    as_right: dict = None, anchor_right: float | list = 0, as_inf: dict = None,
    anchor_inf: float | list = 0, as_left: dict = None, anchor_left: float | list = 0,
    cover=0.025, concrete_specific_weight=2400, stirrups_db: float | list = None,
    stirrups_sep: float | list = None, stirrups_length: list = None, stirrups_anchor:
    float | list = None, stirrups_x: list = None, columns: list[list[float, float]] = None,
    columns_pos: list[float] = None, columns_symbol: list[str] = None,
    nomenclature: str = '#', denomination: str = None, element_type: ElementTypes =
    ElementTypes.BEAM)
```

Beam element, computes geometrics and physics props and manages dxf drawing methods (longitudinal, transversal, reinforcement detailing, etc.)

Parameters

- **width** (*float*) – Width of the beam.

- **height** (*float*) – Height of the beam.
- **x** (*float*) – X-coordinate of the bottom left corner of the beam concrete shape.
- **y** (*float*) – Y-coordinate of the bottom left corner of the beam concrete shape.
- **direction** (*str*) – Direction of the beam (HORIZONTAL or VERTICAL).
- **orientation** (*str*) – Orientation of the beam (TOP, RIGHT, DOWN, LEFT).
- **as_sup** (*dict*) – Dictionary containing reinforcement information for the top side.
- **anchor_sup** (*float | list*) – Anchor length for the top reinforcement or list of lengths.
- **as_right** (*dict*) – Dictionary containing reinforcement information for the right side.
- **anchor_right** (*float | list*) – Anchor length for the right reinforcement or list of lengths.
- **as_inf** (*dict*) – Dictionary containing reinforcement information for the bottom side.
- **anchor_inf** (*float | list*) – Anchor length for the bottom reinforcement or list of lengths.
- **as_left** (*dict*) – Dictionary containing reinforcement information for the left side.
- **anchor_left** (*float | list*) – Anchor length for the left reinforcement or list of lengths.
- **cover** (*float*) – Concrete cover for the reinforcement.
- **concrete_specific_weight** – Specific weight of the concrete. Defaults to CONCRETE_WEIGHT.
- **stirrups_db** (*float | list*) – Diameter of stirrups or list of diameters.
- **stirrups_sep** (*float | list*) – Separation between stirrups or list of separations.
- **stirrups_length** (*list*) – List of lengths for stirrups.
- **stirrups_anchor** (*float | list*) – Anchor length of stirrups or list of anchor lengths.
- **stirrups_x** (*list*) – List of x-coordinates for stirrups.
- **columns** (*list[list[float, float]]*) – List of lists containing column width and height.
- **columns_pos** (*list[float]*) – List of positions (X-coordinate of start) for columns.
- **columns_symbol** (*list[str]*) – List of symbols representing columns.
- **nomenclature** (*str*) – Nomenclature prefix used for labeling elements.
- **number_init** (*int*) – Initial number for labeling elements.
- **element_type** (*ElementTypes*) – Type of the structural element (e.g., BEAM).

Variables

- **width** – Width of the beam.
- **height** – Height of the beam.
- **length** – Length of the beam.
- **x** – X-coordinate of the bottom left corner of the beam concrete shape.
- **y** – Y-coordinate of the bottom left corner of the beam concrete shape.
- **direction** – Direction of the element (HORIZONTAL or VERTICAL).
- **orientation** – Orientation of the element (e.g., TOP, RIGHT, DOWN, LEFT).

- **as_sup** – Dictionary containing reinforcement information for the top side.
- **max_db_sup** – Maximum diameter of the top reinforcement bars.
- **anchor_sup** – Anchor length for the top reinforcement or list of lengths.
- **number_init_sup** – Initial numbering for the top reinforcement bars.
- **as_right** – Dictionary containing reinforcement information for the right side.
- **max_db_right** – Maximum diameter of the right reinforcement bars.
- **anchor_right** – Anchor length for the right reinforcement or list of lengths.
- **number_init_right** – Initial numbering for the right reinforcement bars.
- **as_inf** – Dictionary containing reinforcement information for the bottom side.
- **max_db_inf** – Maximum diameter of the bottom reinforcement bars.
- **anchor_inf** – Anchor length for the bottom reinforcement or list of lengths.
- **number_init_inf** – Initial numbering for the bottom reinforcement bars.
- **as_left** – Dictionary containing reinforcement information for the left side.
- **max_db_left** – Maximum diameter of the left reinforcement bars.
- **anchor_left** – Anchor length for the left reinforcement or list of lengths.
- **number_init_left** – Initial numbering for the left reinforcement bars.
- **cover** – Concrete cover for the reinforcement.
- **concrete_specific_weight** – Specific weight of the concrete.
- **concrete** – Concrete instance class.
- **stirrups_db** – Diameter of stirrups or list of diameters.
- **stirrups_sep** – Separation between stirrups or list of separations.
- **stirrups_length** – List of lengths for stirrups.
- **stirrups_anchor** – Anchor length of stirrups or list of anchor lengths.
- **stirrups_x** – List of x-coordinates for stirrups.
- **columns** – List of lists containing column width and height.
- **columns_pos** – List of positions (X-coordinate of start) for columns.
- **columns_symbol** – List of symbols representing columns.
- **bars_as_sup** – List of top reinforcement bars as entities.
- **bars_as_right** – List of right reinforcement bars as entities.
- **bars_as_inf** – List of bottom reinforcement bars as entities.
- **bars_as_left** – List of left reinforcement bars as entities.
- **all_bars** – List of all reinforcement bars as entities.
- **stirrups** – List of stirrups as entities.
- **all_elements** – List of all elements/entities in the structure.
- **nomenclature** – Nomenclature prefix used for labeling elements.
- **number_init** – Initial number for labeling elements.
- **element_type** – Type of the structural element (e.g., BEAM).

draw_longitudinal(*document: Drawing*, *x: float = None*, *y: float = None*, *concrete_shape: bool = True*, *bars: bool = True*, *columns: bool = True*, *columns_axes: bool = True*, *stirrups: bool = True*, *middle_ave: bool = True*, *middle_ave_symbol: str = 'A'*, *dim: bool = True*, *dim_style: str = 'EZ_M_25_H25_CM'*, *unifilar_bars: bool = False*, *unifilar_stirrups: bool = True*, *settings: dict = {'concrete_settings': {'dim_style_boxing': 'EZ_M_25_H25_CM', 'dim_style_inner': 'EZ_M_10_H25_CM', 'text_dim_distance_horizontal': 0.6, 'text_dim_distance_vertical': 0.1}, 'text_dim_height': 0.05, 'text_dim_horizontal_distance': 0.05}) → dict*

Draws the longitudinal section of the beam.

Parameters

- **document** (*Drawing*) – The DXF document where the beam will be drawn.
- **x** (*float, optional*) – X-coordinate of the left bottom corner of beam concrete shape.
- **y** (*float, optional*) – Y-coordinate of the left bottom corner of beam concrete shape.
- **concrete_shape** (*bool*) – If True, the concrete shape of the beam is drawn.
- **bars** (*bool*) – If True, the reinforcement bars are drawn.
- **columns** (*bool*) – If True, the columns are drawn.
- **columns_axes** (*bool*) – If True, the axes of the columns are drawn.
- **stirrups** (*bool*) – If True, the stirrups are drawn.
- **middle_ave** (*bool*) – If True, the middle axis of the beam is drawn.
- **middle_ave_symbol** (*str*) – Symbol to represent the middle axis.
- **dim** (*bool*) – If True, dimensions are drawn.
- **dim_style** (*str*) – The dimension style to be used.
- **unifilar_bars** (*bool*) – If True, the reinforcement bars are drawn as unifilar.
- **unifilar_stirrups** (*bool*) – If True, the stirrups are drawn as unifilar.
- **settings** (*dict*) – Dictionary of drawing settings. Default is *BEAT_SET_LONG*.

Returns

A list of graphical entities representing the longitudinal section of the beam.

Return type

list

draw_longitudinal_rebar_detailing(*document: Drawing*, *x: float = None*, *y: float = None*, *unifilar: bool = True*, *columns_axes: bool = True*, *settings: dict = {'text_height': 0.05}) → dict*

Draws the longitudinal rebar detailing for the beam.

Parameters

- **document** (*Drawing*) – The DXF document where the detailing will be drawn.
- **x** (*float, optional*) – X-coordinate of the starting point for drawing.
- **y** (*float, optional*) – Y-coordinate of the starting point for drawing.
- **unifilar** (*bool*) – If True, the rebar is drawn as unifilar.
- **columns_axes** (*bool*) – If True, the axes of the columns are drawn.
- **settings** (*dict*) – Dict with beam longitudinal rebar drawing settings.

Returns

A list of graphical entities representing the longitudinal rebar detailing.

Return type

list

draw_transverse(*document: Drawing, x: float = None, y: float = None, x_section: float = None, unifilar: bool = False, dimensions: bool = True, settings: dict = {'concrete_settings': {'dim_style_boxing': 'EZ_M_10_H25_CM', 'dim_style_inner': 'EZ_M_10_H25_CM', 'text_dim_distance_horizontal': 0.1, 'text_dim_distance_vertical': 0.1, 'text_dim_inner_perpendicular_distance': 0.05}, 'text_dim_distance': 0.05, 'text_dim_height': 0.05}*) → dict

Draws the transverse section of the beam at a given x-section.

Parameters

- **document** (*Drawing*) – The DXF document where the beam will be drawn.
- **x** (*float*) – X-coordinate of the starting point for drawing.
- **y** (*float*) – Y-coordinate of the starting point for drawing.
- **x_section** (*float, optional*) – The x-coordinate of the section to be drawn.
- **unifilar** (*bool*) – If True, the bars are drawn as unifilar.
- **dimensions** (*bool*) – If True, dimensions are drawn.
- **settings** (*dict*) – Dict with beam transverse drawing settings.

Returns

A dict of graphical entities representing the transverse section of the beam.

Return type

dict

draw_transverse_rebar_detailing(*document: Drawing, x: float = None, y: float = None, x_section: float = None, unifilar: bool = False, dimensions: bool = True*) → dict

Draws the transverse rebar detailing for the beam at a given x-section.

Parameters

- **document** (*Drawing*) – The DXF document where the detailing will be drawn.
- **x** (*float, optional*) – X-coordinate of the starting point for drawing.
- **y** (*float, optional*) – Y-coordinate of the starting point for drawing.
- **x_section** (*float, optional*) – The x-coordinate of the section to be drawn.
- **unifilar** (*bool*) – If True, the rebar is drawn as unifilar.
- **dimensions** (*bool*) – If True, dimensions are drawn.

Returns

A dict of graphical entities representing the transverse rebar detailing.

Return type

dict

```
class etacad.column.Column(width, depth, height, diameter: float = None, column_type: int =
    ColumnTypes.RECTANGULAR, x=0, y=0, direction: Direction =
    Direction.VERTICAL, orientation: Orientation = Orientation.RIGHT,
    as_sup: dict = None, as_right: dict = None, as_inf: dict = None, as_left: dict
    = None, anchor_sup: list = 0, anchor_right: list = 0, anchor_inf: list = 0,
    anchor_left: list = 0, cover=0, concrete_specific_weight: float = 2400,
    stirrups_db: list = None, stirrups_anchor: list = None, stirrups_sep: list =
    None, stirrups_length: list = None, stirrups_x: list = None, beams: list =
    None, beams_pos: list = None, beam_symbol: list = None, nomenclature: str
    = '#', denomination: str = None, number_init: int = None, element_type:
    ElementTypes = ElementTypes.COLUMN)
```

Column element, computes geometrics and physics props and manages dxf drawing methods (longitudinal, transversal, reinforcement detailing, etc.)

Parameters

- **width** (*float*) – Width of the column (rectangular section) in units of length.
- **depth** (*float*) – Depth of the column (rectangular section) in units of length.
- **height** (*float*) – Height of the column in units of length.
- **diameter** (*float*, *optional*) – Diameter of the column if circular (optional), defaults to None.
- **column_type** (*int*) – Type of column section (rectangular or circular), defaults to RECTANGULAR.
- **x** (*float*) – X-coordinate of the column's position, defaults to 0.
- **y** (*float*) – Y-coordinate of the column's position, defaults to 0.
- **direction** (*Direction*) – Orientation of the column (VERTICAL or HORIZONTAL), defaults to VERTICAL.
- **orientation** (*Orientation*) – Column's orientation (RIGHT, LEFT, etc.), defaults to RIGHT.
- **as_sup** (*dict*, *optional*) – Dictionary representing the longitudinal steel in the upper part of the column.
- **as_right** (*dict*, *optional*) – Dictionary representing the longitudinal steel in the right side of the column.
- **as_inf** (*dict*, *optional*) – Dictionary representing the longitudinal steel in the lower part of the column.
- **as_left** (*dict*, *optional*) – Dictionary representing the longitudinal steel in the left side of the column.
- **anchor_sup** (*list*, *optional*) – List of anchorage values for the upper part of the column.
- **anchor_right** (*list*, *optional*) – List of anchorage values for the right side of the column.
- **anchor_inf** (*list*, *optional*) – List of anchorage values for the lower part of the column.
- **anchor_left** (*list*, *optional*) – List of anchorage values for the left side of the column.
- **cover** (*float*, *optional*) – Concrete cover of the column in units of length.
- **concrete_specific_weight** (*float*) – Specific weight of the concrete used in the column.
- **stirrups_db** (*list*, *optional*) – List of diameters of the stirrups used in the column.

- **stirrups_anchor** (*list, optional*) – List of anchorage values for the stirrups.
- **stirrups_sep** (*list, optional*) – List of spacing values between stirrups.
- **stirrups_length** (*list, optional*) – List of lengths of stirrups.
- **stirrups_x** (*list, optional*) – List of x-coordinates for stirrups placement.
- **beams** (*list, optional*) – List of crossing beams interacting with the column.
- **beams_pos** (*list, optional*) – List of positions for the crossing beams.
- **beam_symbol** (*list, optional*) – List of symbols used to represent crossing beams.
- **nomenclature** (*str, optional*) – Column identification string or nomenclature.

Variables

- **as_sup** (*dict*) – Dictionary containing the top longitudinal reinforcement details.
- **as_right** (*dict*) – Dictionary containing the right longitudinal reinforcement details.
- **as_inf** (*dict*) – Dictionary containing the bottom longitudinal reinforcement details.
- **as_left** (*dict*) – Dictionary containing the left longitudinal reinforcement details.
- **max_db_sup** (*float*) – Maximum diameter of the top longitudinal bars.
- **max_db_right** (*float*) – Maximum diameter of the right longitudinal bars.
- **max_db_inf** (*float*) – Maximum diameter of the bottom longitudinal bars.
- **max_db_left** (*float*) – Maximum diameter of the left longitudinal bars.
- **max_db_hz** (*float*) – Maximum horizontal diameter among the longitudinal bars.
- **max_db_vt** (*float*) – Maximum vertical diameter among the longitudinal bars.
- **anchor_sup** (*list*) – List of top bar anchorage lengths.
- **anchor_right** (*list*) – List of right bar anchorage lengths.
- **anchor_inf** (*list*) – List of bottom bar anchorage lengths.
- **anchor_left** (*list*) – List of left bar anchorage lengths.
- **bars_as_sup** (*list*) – List of bars in the top longitudinal reinforcement.
- **bars_as_right** (*list*) – List of bars in the right longitudinal reinforcement.
- **bars_as_inf** (*list*) – List of bars in the bottom longitudinal reinforcement.
- **bars_as_left** (*list*) – List of bars in the left longitudinal reinforcement.
- **number_init_sup** (*int*) – Initial number for top longitudinal bars.
- **number_init_right** (*int*) – Initial number for right longitudinal bars.
- **number_init_inf** (*int*) – Initial number for bottom longitudinal bars.
- **number_init_left** (*int*) – Initial number for left longitudinal bars.
- **cover** (*float*) – Concrete cover thickness.
- **stirrups_db** (*list*) – List of stirrup diameters.
- **stirrups_anchor** (*list*) – List of stirrup anchorage lengths.
- **stirrups_sep** (*list*) – List of stirrup separations.
- **stirrups_length** (*list*) – List of stirrup lengths.
- **stirrups_x** (*list*) – List of stirrup X-coordinates.
- **stirrups** (*list*) – List of stirrup elements.
- **beams** (*list*) – List of crossing beams.

- **beams_pos** (*list*) – List of crossing beams' positions.
- **beam_symbol** (*list*) – List of crossing beams' symbols.
- **all_bars** (*list*) – List of all bars in the column.
- **all_elements** (*list*) – List of all elements in the column.
- **box_width** (*float*) – Width of the bounding box for the column.
- **box_height** (*float*) – Height of the bounding box for the column.
- **nomenclature** (*str*) – Nomenclature for the column reinforcement.
- **element_type** (*int*) – Type of element, set to COLUMN by default.

draw_longitudinal (*document: Drawing, x: float = None, y: float = None, concrete_shape: bool = True, bars: bool = True, beams: bool = True, beams_axes: bool = True, stirrups: bool = True, middle_ave: bool = True, middle_ave_symbol: str = 'A', dim: bool = True, dim_style: str = 'EZ_M_25_H25_CM', unifilar_bars: bool = False, unifilar_stirrups: bool = True*) → dict

Draws the longitudinal view of the column, including concrete shape, beams, stirrups, and bars. Also includes dimensioning and optional middle axes.

Parameters

- **document** (*Drawing*) – The drawing document to which the elements will be added.
- **x** (*float, optional*) – X-coordinate for the drawing position. Defaults to column's x position.
- **y** (*float, optional*) – Y-coordinate for the drawing position. Defaults to column's y position.
- **concrete_shape** (*bool*) – Whether to draw the concrete shape. Defaults to True.
- **bars** (*bool*) – Whether to draw the longitudinal bars. Defaults to True.
- **beams** (*bool*) – Whether to draw crossing beams. Defaults to True.
- **beams_axes** (*bool*) – Whether to draw the axes for beams. Defaults to True.
- **stirrups** (*bool*) – Whether to draw the stirrups. Defaults to True.
- **middle_ave** (*bool*) – Whether to draw the middle axis. Defaults to True.
- **middle_ave_symbol** (*str*) – Symbol to use for the middle axis. Defaults to "A".
- **dim** (*bool*) – Whether to add dimensions to the drawing. Defaults to True.
- **dim_style** (*str*) – Style of the dimensions. Defaults to "EZ_M_25_H25_CM".
- **unifilar_bars** (*bool*) – Whether to draw bars in unifilar view. Defaults to False.
- **unifilar_stirrups** (*bool*) – Whether to draw stirrups in unifilar view. Defaults to True.

Returns

A dict of entities drawn on the document.

Return type

dict

draw_longitudinal_rebar_detailing (*document: Drawing, x: float = None, y: float = None, unifilar: bool = True, beam_axes: bool = True, settings: dict = {'bar_settings': {'text_denomination_distance': 0.05, 'text_denomination_height': 0.05, 'text_dim_distance_horizontal': 0.05, 'text_dim_distance_vertical': 0.05, 'text_dim_height': 0.05}, 'spacing': 0.3, 'text_height': 0.05}*) → dict

Draws the longitudinal rebar detailing for the column.

Parameters

- **document** (*Drawing*) – The DXF document where the detailing will be drawn.
- **x** (*float, optional*) – X-coordinate of the starting point for drawing.
- **y** (*float, optional*) – Y-coordinate of the starting point for drawing.
- **unifilar** (*bool*) – If True, the rebar is drawn as unifilar.
- **beam_axes** (*bool*) – If True, the axes of the beam are drawn.
- **settings** (*dict*) – Dict with column longitudinal rebar drawing settings.

Returns

A dict of graphical entities representing the longitudinal rebar detailing.

Return type

dict

draw_transverse(*document: Drawing, x: float = None, y: float = None, y_section: float = None, unifilar: bool = False, dimensions: bool = True, settings: dict = {'concrete_settings': {'dim_style_boxing': 'EZ_M_10_H25_CM', 'dim_style_inner': 'EZ_M_10_H25_CM', 'text_dim_distance_horizontal': 0.05, 'text_dim_distance_vertical': 0.05}, 'text_dim_height': 0.05}*) → dict

Draws the transverse view of the column at a given y-section. generate a drawing from the transverse perspective.

Parameters

- **document** (*Drawing*) – The DXF document where the column will be drawn.
- **x** (*float*) – X-coordinate of the starting point for drawing.
- **y** (*float*) – Y-coordinate of the starting point for drawing.
- **y_section** (*float, optional*) – The y-coordinate of the section to be drawn.
- **unifilar** (*bool*) – If True, the bars are drawn as unifilar.
- **dimensions** (*bool*) – If True, dimensions are drawn.
- **settings** (*dict*) – Dict with column transverse drawing settings.

Returns

A dict of entities representing the transverse view of the column.

Return type

dict

draw_transverse_rebar_detailing(*document: Drawing, x: float = None, y: float = None, y_section: float = None, unifilar: bool = False, dimensions: bool = True, settings: dict = {'bar_settings': {'text_denomination_distance': 0.05, 'text_denomination_height': 0.05, 'text_dim_distance_horizontal': 0.05, 'text_dim_distance_vertical': 0.05, 'text_dim_height': 0.05}, 'spacing': 0.3, 'text_height': 0.05}*) → list

Draws the transverse rebar detailing for the column at a given y-section.

Parameters

- **document** (*Drawing*) – The DXF document where the detailing will be drawn.
- **x** (*float, optional*) – X-coordinate of the starting point for drawing.
- **y** (*float, optional*) – Y-coordinate of the starting point for drawing.

- **y_section** (*float*, *optional*) – The y-coordinate of the section to be drawn.
- **unifilar** (*bool*) – If True, the rebar is drawn as unifilar.
- **dimensions** (*bool*) – If True, dimensions are drawn.
- **settings** (*dict*) – Dict with column longitudinal rebar drawing settings.

Returns

A dict of graphical entities representing the transverse rebar detailing.

Return type

dict

(~) If you want to pay me for a beer, coffee, or something else: [C|_](#)

PYTHON MODULE INDEX

e

etacad.bar, ??
etacad.beam, ??
etacad.column, ??
etacad.concrete, ??
etacad.stirrup, ??