YLPA manual

Vladimír Vančík
Faculty of Civil Engineering, CTU Prague, Department of Mechanics

1 Installation

The program requires Python 2.7 with the following libraries:

- Enthought Traits, Enthought TraitsUI ([1])
- Matplotlib ([2])
- Scipy, Numpy ([3])

Users with no knowledge of Python may use one of the free distributions such as Enthought Canopy ([4]) or Anaconda ([5]), where the libraries Matplotlib, Scipy and Numpy are usually preinstalled. Then you just open the command prompt and run

```
pip install traits
```

and

```
pip install traitsui
```

which should install the remaining libraries.

When the prerequisites are satisfied, unpack the directory *ylpa.zip*, open the command prompt, enter the unpacked directory and run YLPA with

```
python ylpa.py
```

2 Usage

The program YLPA (Yield-Line Plate Analysis) is designed to calculate the limit load coefficient for a mechanism with one degree of freedom. The program has a graphical user interface and is executed from the command prompt where it also writes the results. The GUI window (Fig. 1) is divided into three parts:

- The object tree structure – top left part. Each node represents one object. Right-clicking an object initiates its context menu.
- Editing window of the active object – bottom left part. If the object has some editable values, its editing interface displays here.

- The plotting window – right part. The *Replot* button plots the mechanism according to the currently active object (e.g. if the active object is a segment, it is highlighted with red color). The *Clear* button clears the previous plot.

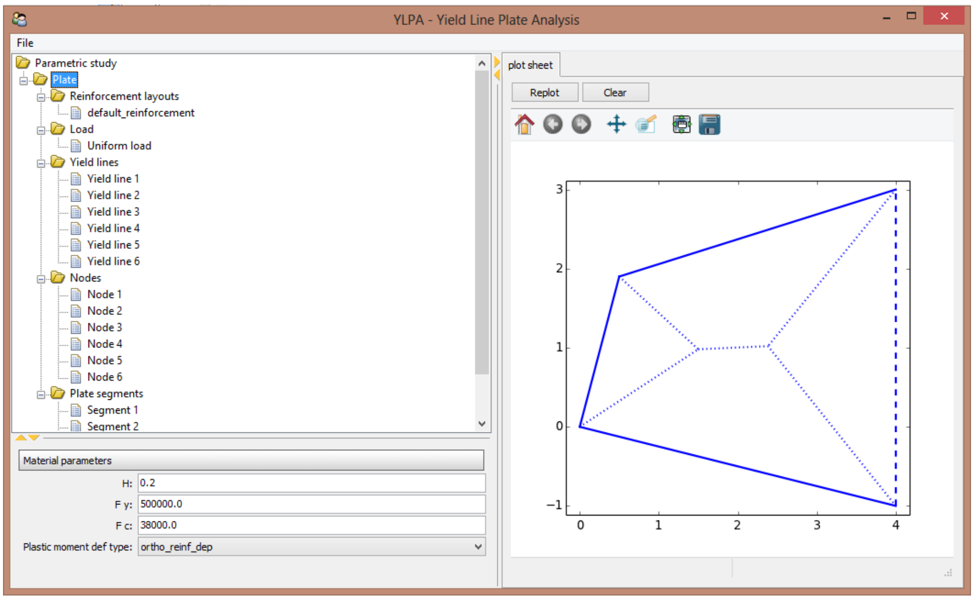

Figure 1: Graphical user interface of YLPA

2.1 Plate input

2.1.1 Geometry

A kinematic mechanism of a plate is defined by segments, nodes and yield lines.

Nodes can be added from the context menu of the *Nodes* object of the tree structure. The position of a node is set in its editing window. For a supported node, set the deflection to zero (parameter $w = 0$). The nodes are numbered in the order of their creation.

Segments can be added from the context menu of the *Plate segments* object of the tree structure. Boundary node numbers are specified in the editing window in an anticlockwise order.

Yield lines can be added from the context menu of the *Yield lines* object of the tree structure. The editable parameters of a yield line are the two end nodes and the reinforcement layout. Typically, the yield lines lie on every clamped edge and on every edge shared by two segments. A positive yield line is displayed as a dotted line and negative yield line as a dashed line.

2.1.2 Load

Load can be added from the context menu of the *Loads* object of the tree structure. Supported load types are uniform load (*PlateLoadUniform*) and force load (*PlateLoadNodalForce*). Each load object has an editable value of its multiplier (*load factor multiplier*) and the force loads have an editable position (coordinates x, y).

2.1.3 Material values

The plate is by default made of reinforced concrete. On the *Plate* object the strength of concrete and steel and the plate height is set. The *Reinforcement layouts* object contains user-defined reinforcement layout characterized by the values of reinforcement ratios at the top and bottom surfaces in two perpendicular directions and the orientation of the main reinforcement direction with respect to the positive axis x. A reinforcement layout may be assigned to yield lines.

2.1.4 Definition of the plastic moment

Definition of the plastic moment is set on the *Plate* object. Currently YLPA offers following options:

- ortho_reinf_indep – orthotropic reinforcement with the compressive zone independent of the yield line orientation.
- ortho_reinf_indep_uni – orthotropic reinforcement with the compressive zone independent of the yield line orientation and unified for both orthotropic directions.
- ortho_reinf_dep – orthotropic reinforcement with the compressive zone dependent on the yield line orientation.
- ortho_reinf_dep_duct – same as ortho_reinf_dep with ductility check performed after the computation.
- simple – unit moment in all directions. Useful for the calculation of isotropic plates where the resulting limit load coefficient only has to be multiplied by the value of the actual limit bending moment.

2.2 Optimization input

The optimization parameters are managed by the *Parametric study* object (its editing window is shown in Fig.. 2) and divided into node parameters (geometrical optimization, left part of the editing window) and plate parameters (reinforcement optimization, right part of the editing window). Those two groups are further split into parameters, which are the actual optimization variables and optimization data, that specify the values to be optimized and their dependency on one of the parameters.

The node parameters have editable default value and boundaries. For node optimization data we set the node number and the name of the optimized parameter (coordinate) as a string, then the master parameter number, the multiplier (designated as m) and the default value (designated as d) – in every optimization step the value specified by name and node numberis calculated as $d + mp$ with p being the current value of the master parameter.

The reinforcement optimization is constrained by the *optimization constrain value*, which must be equal to the sum of optimization parameters multiplied by their respective constrain multipliers. Those multipliers serve to take into account the cases when various reinforcement layouts would cover areas of different sizes. The optimization constrain value represents the total reinforcement volume. The reinforcement optimization data are similar to node optimization data but instead of node number we specify the name of a reinforcement layout.

2.3 Running the calculation

The calculation is ran by pressing the *Calculate* button in the plotting window or the context menu and the resulting limit load coefficient is written in the command prompt. If the definition of the limit plastic moment is ortho_reinf_dep_duct the output of the ductility check is appended to the calculation result.

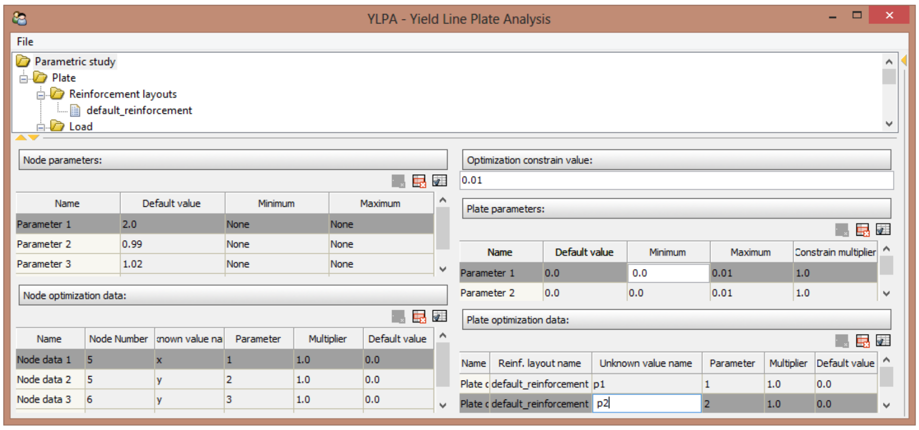

Figure 2: User interface – editing window for optimization

References

[1] Enthought Tool Suite website, http://code.enthought.com/projects/

[2] Matplotlib website, http://matplotlib.org/

[3] Numpy and Scipy documentation website, http://docs.scipy.org/doc/

[4] Enthought Canopy website, https://www.enthought.com/products/canopy/

[5] Continuum Analytics website, http://www.continuum.io/