

NLPAK

The Nonlinear SSI Package

How to use NLPAK with simple example for Piled raft

The NLPK is an add-in tool to the PLPAK that allows analyze piled rafts on nonlinear piles.

Nonlinear settlement of pile consists of two main parts. Settlement due to interaction effects and other due to pile load test nonlinear relation between load and settlement. Settlement due to interaction effects is a linear settlement. Settlement due to nonlinear pile load test is a nonlinear settlement. Hence, nonlinear analysis could be performed on nonlinear settlement only. NLPK allows user to perform nonlinear analysis with or without separation of those two types of settlements. NLPK provides two approaches of analysis:

1- Full working load analysis

In this approach, nonlinear settlement is calculated at a certain load level.

2- Incremental failure analysis

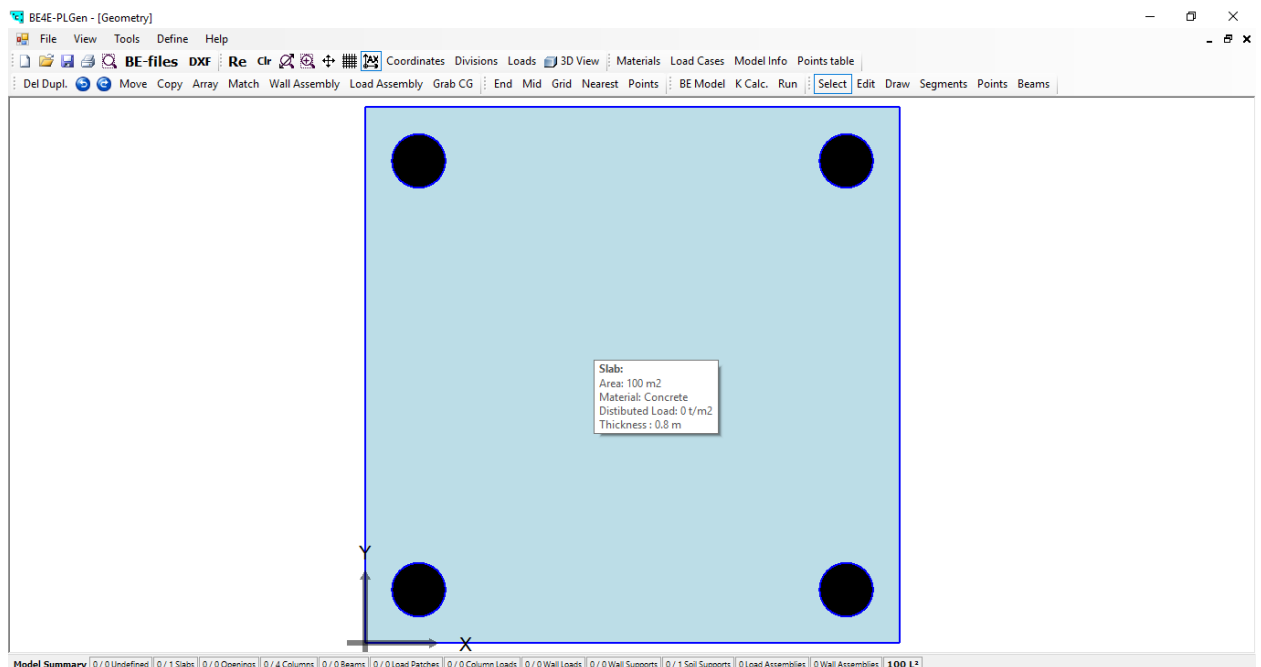
In this approach, nonlinear analysis is performed incrementally, and each pile reach its maximum capacity is removed. In addition, soil could be removed after reach its bearing capacity. Load settlement response of the raft is plotted simultaneously through the analysis.

For more clarification, The NLPK is described using simple example, solved two times using the two analysis approaches. For input file structure, see appendix 1. It has to be noted that this input file could be exported automatically using Pile Editor tool in P-PPAK tool.

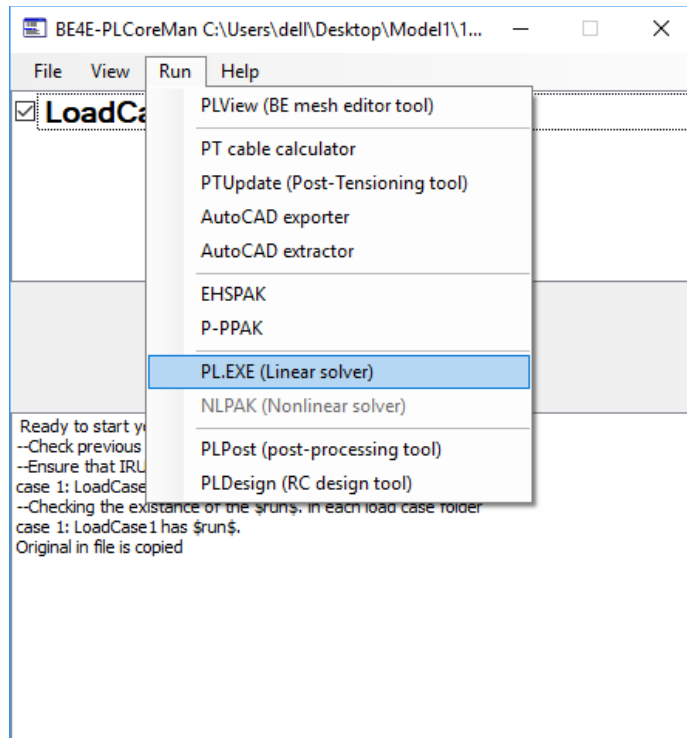
Problem 1:

This problem is 10×10 m piled raft with thickness 0.8 m supported on four piles each 0.5 m radius subjected to its own weight. The ultimate pile load is assumed to be 100 ton.

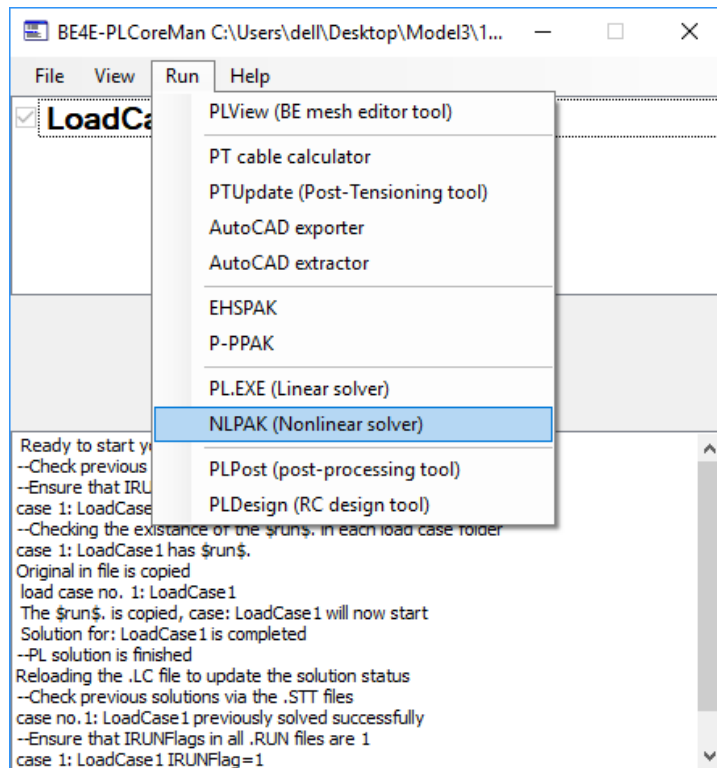
1- Generate Gen model.

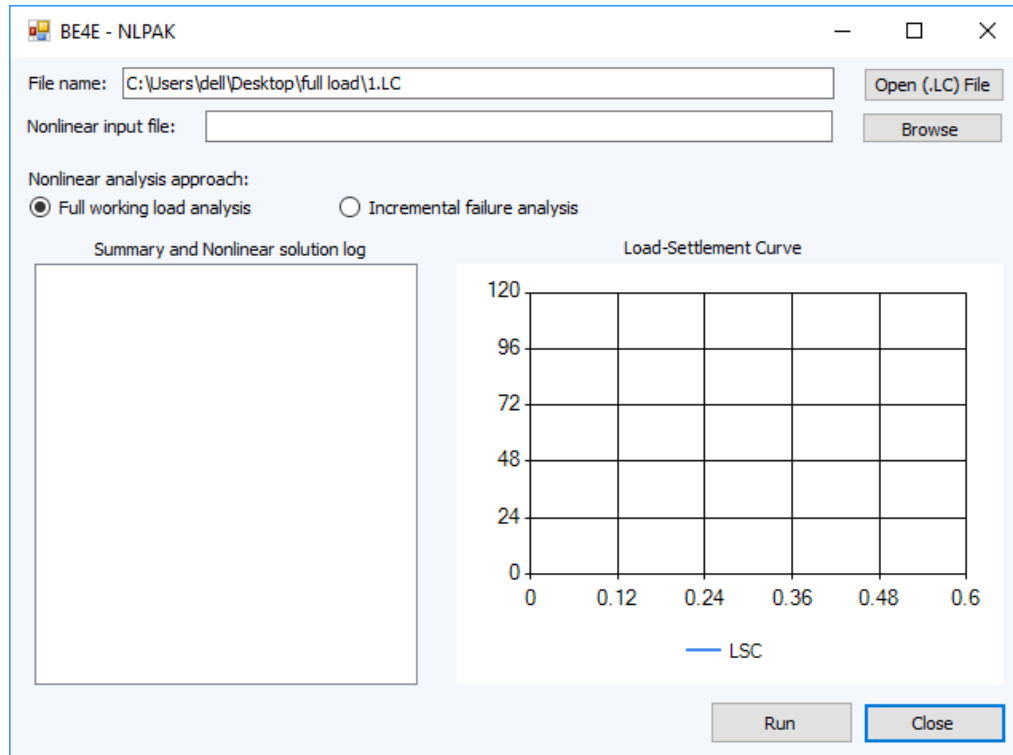


- 2- Run the problem from PLGen or load it from PLCoreMan.
- 3- From PLCoreMan run EHSPAK and P-PPAK to extract SSI stiffness in (PL\$MATK\$.-4).
- 4- Run PL.exe (Linear solver).

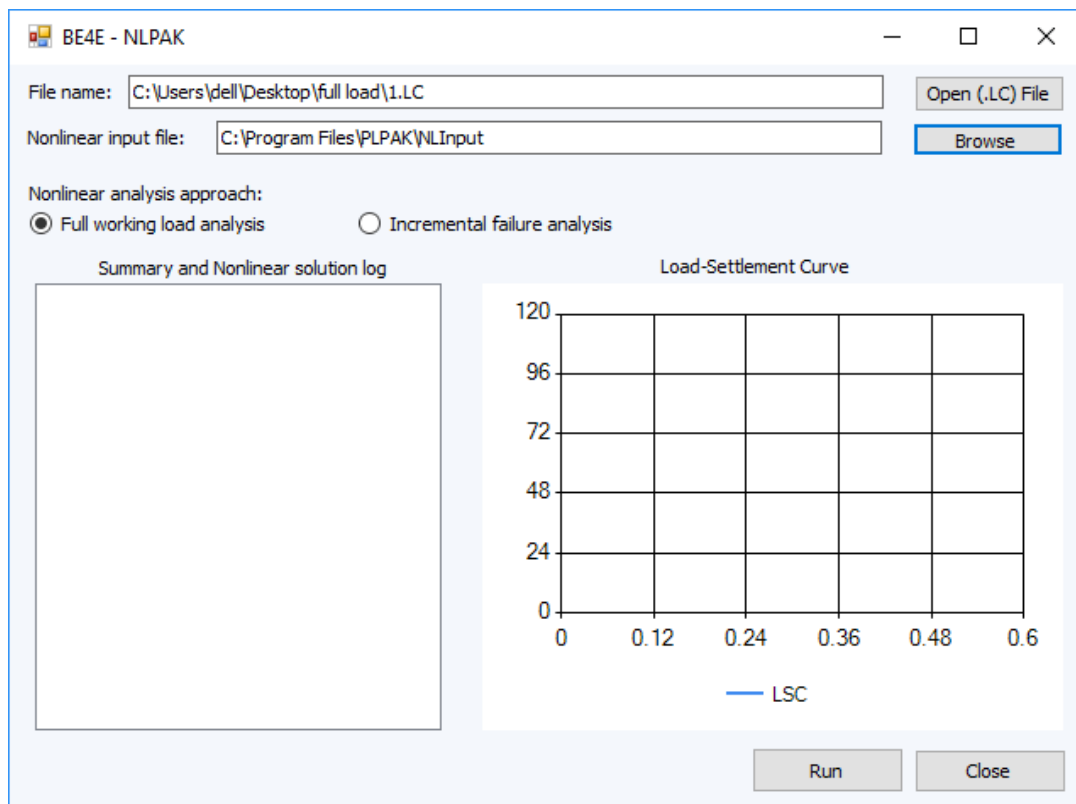


- 5- After run PL.exe (Linear solver) NLPAK button will be activated. Run NLPAK (Nonlinear solver).

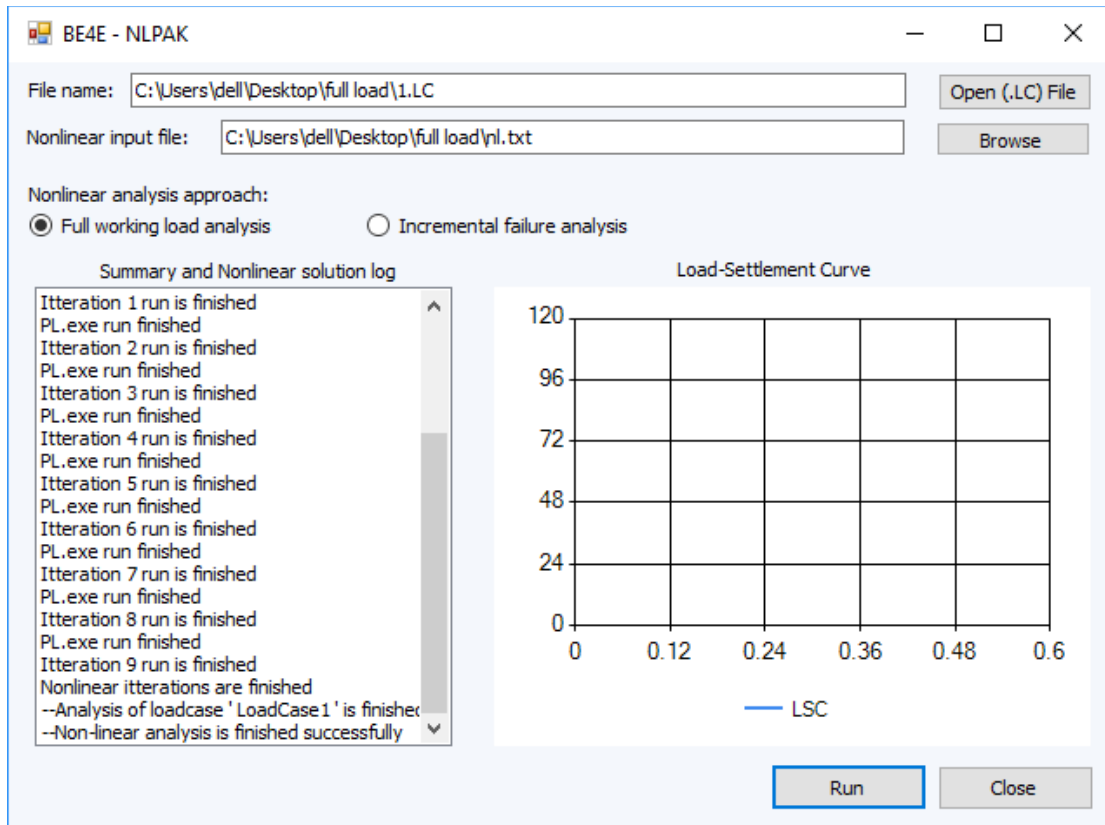




a. Load nonlinear input file of the problem (File structure see appendix 1).

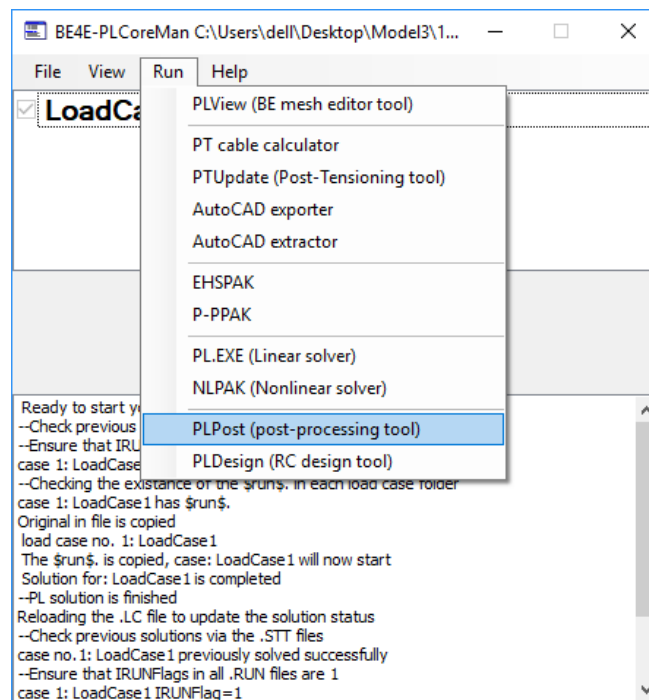


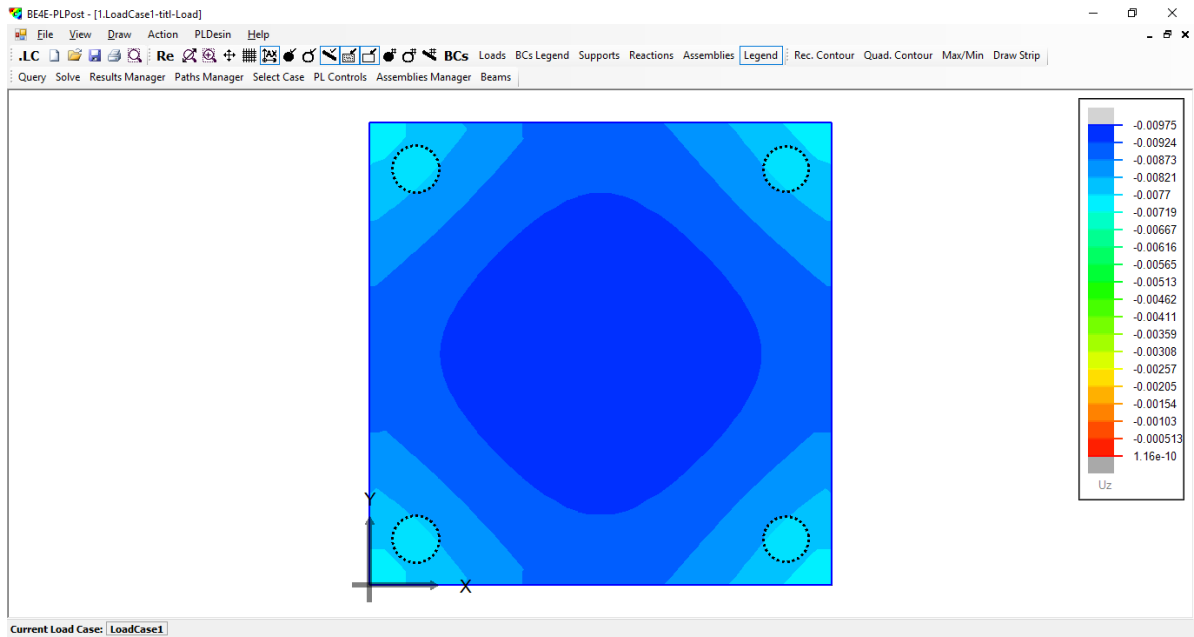
b. Press button Run. Iterations will start until reach the final solution.



6- Close NLPK and go back to PLCoreMan.

7- Show results on PLPost.



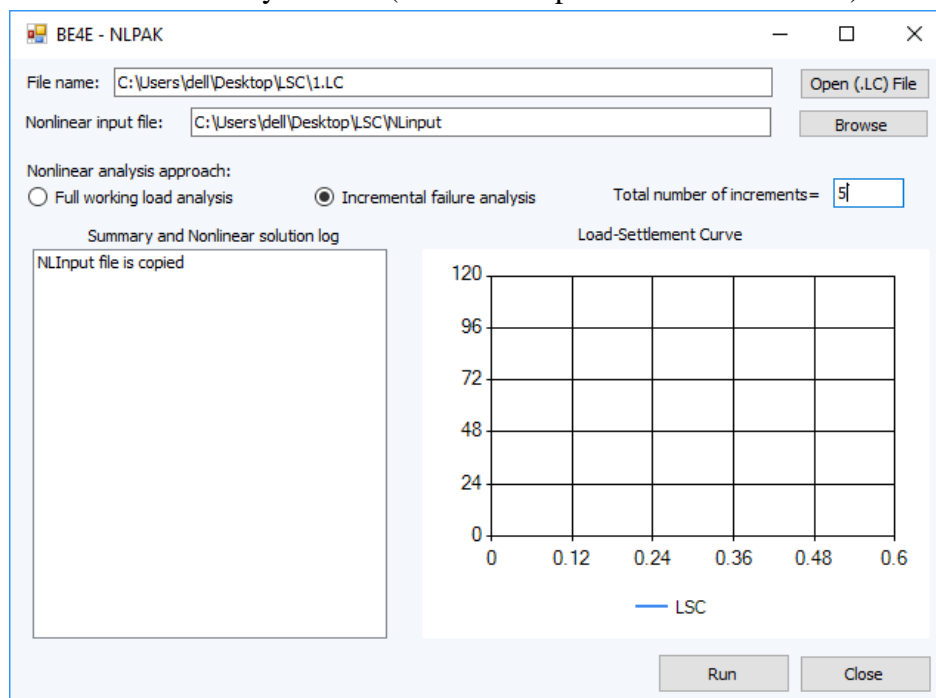


Problem 2:

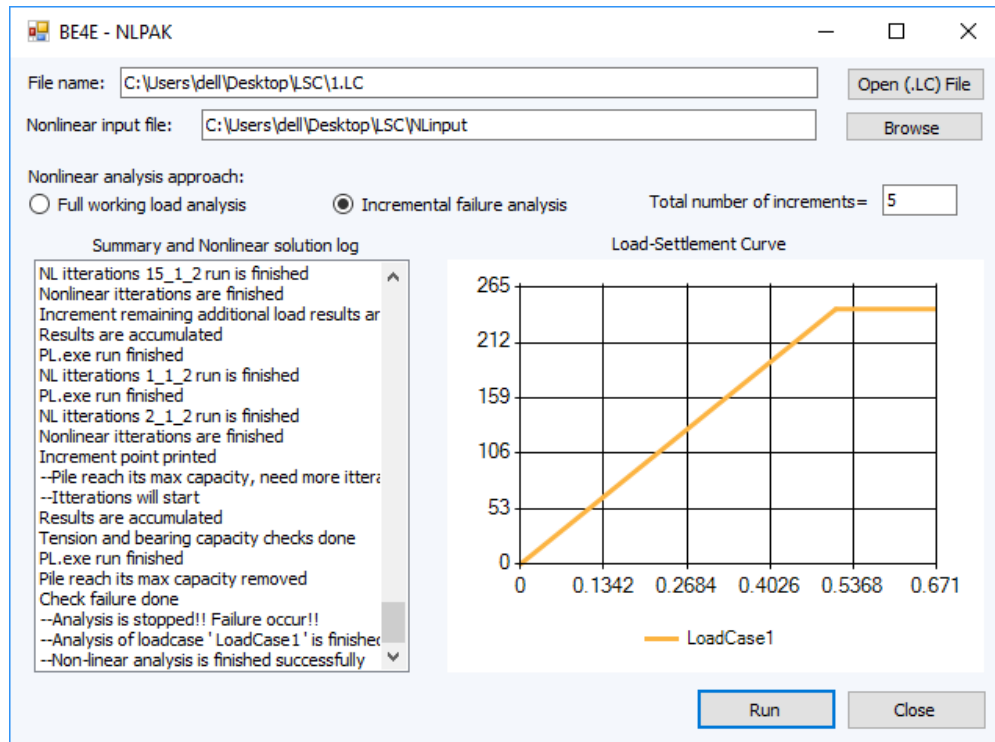
Problem 1 is solved again using the second approach (incremental failure analysis). The ultimate pile load is assumed ($Q_{ult}=100$ ton) and the failure working load is assumed ($Q_{all}= 45$ ton). The same steps of problem 1 is followed till step 5.a:

5- Run NLPK (Nonlinear solver) and choose **Incremental failure analysis**,

- Load nonlinear input file of the problem (File structure see appendix 1).
- Choose number of increments you want (in this example it was 5 increments).



c. Press button Run. Iterations will start until reach the final solution. Load-settlement curve will be plotted automatically during the analysis.



d. All the remaining steps are the same as example 1.

Appendix 1 Input file Structure

1- Nonlinear input file (Full working load approach):

<ul style="list-style-type: none"> - 0/1 0: without separation, 1: with separation. - 0/1 0: without including soil nonlinearity, 1: with including soil nonlinearity. - Qult, Kinitial Qult, Kinitial. Qult, Kinitial 	<div style="border: 1px solid black; padding: 10px; width: fit-content; margin: auto;"> <p>Ultimate load capacity and initial stiffness (obtained from load-settlement curve) for each pile/soil cell.</p> </div>
---	--

Input file of problem 1 as example:

```
0
0
100 1000
100 1000
100 1000
```

2- Nonlinear input file (Incremental failure approach):

<ul style="list-style-type: none">- 0/1 0: without including soil nonlinearity, 1: with including soil nonlinearity.- Soil bearing capacity- Q_{all}, Q_{ult}Q_{all}, Q_{ult}......Q_{all}, Q_{ult}	<ul style="list-style-type: none">- Allowable working load & ultimate load (obtained from load-settlement curve) for each pile cell.- Bearing capacity & ultimate load (obtained from load-settlement curve) for each soil cell.
--	---

Input file of problem 2 as example:

```
0
0
45 100
45 100
45 100
45 100
```