

NLDYNA-Nonlinear Dynamic Analysis.

Introduction

The study of structural systems considering inelastic response was, until very recently, only possible through the use of commercial packages with subsequent limitations as a research tool. For instance, commercial packages are very rigid since not all of them allow the addition of independent elements and/or material models as often required in research activities. In top of that restriction the used solvers in commercial codes are of the black-box type making its extension to general tasks an impossible goal. With the recent development of high level languages (like Python) it is now possible to develop very efficient in-house implementations. This project describes a general in-house finite element assembler and solver aimed at studying the non-linear response of dynamic systems. The code is intended to be used in the testing of material models and/or complex kinematic formulations commonly required in research activities. The code has the following advantages:

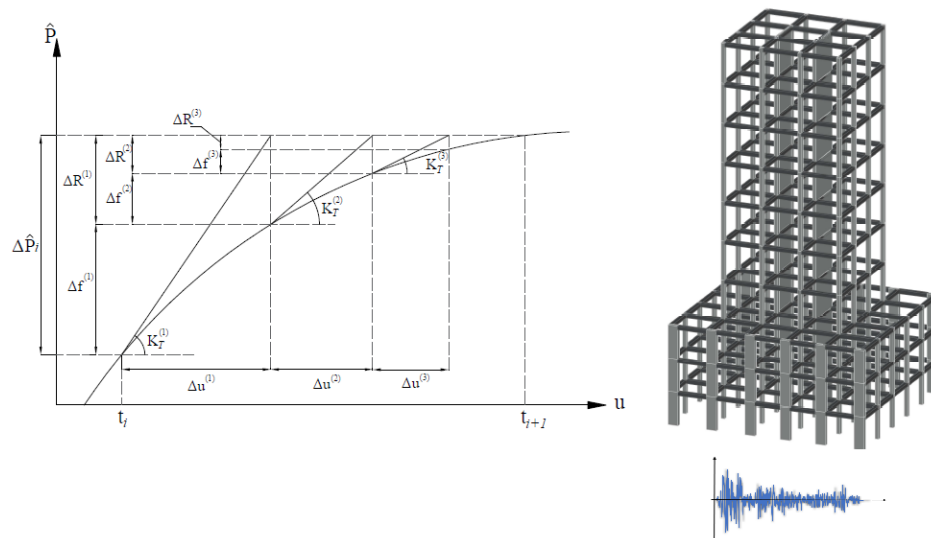
- **It is multiphysics oriented:** The code is just a general dynamic Newton-Raphson solver where the physical context is provided by the user in terms of material and/or element models.
- **The implementation has been fully parametrized:** It does not have an implicit space dimensionality and problems with an arbitrary number of degrees of freedom per node can be solved.
- **Python based user elements and material models:** The implementation of user elements and user constitutive models is highly simplified in comparisson with commercial codes as it is connducted in a high level language like Python.
- **Easily coupled with independent scripts:** Since the code is fully open and written in a modular structure it can be coupled with external independent scripts required in specific analysis and design problems.

Nonlinear dynamic analysis of generalized finite element problems

NLDYNA is a generalized finite element program for the solution of time-dependent non-linear problems. The code is able to handle static and dynamic analysis problems assumed of hyperbolic nature. It is generalized as it can solve user defined problems in different physical contexts through the implementation of user elements and user constitutive responses.

In **NLDYNA** a dynamic problem is splitted into several time increments and each increment is solved by a Newton-Raphson algorithm. Time stepping is conducted by an implicit Wilson θ -method. The solution of linear static problems takes place in a single increment and a single iteration.

A model is defined in **NLDYNA** through 7 easy to write input data files containing basic problem parameters: nodal point data, element data, loads data and general constraints assignments data. The model can use elements available in the code's library, specific user defined elements or a combination of both. Similarly, a model can also use **NLDYNA's** available elements in combination with user defined material models.



The following sections describe aspects related to the implementation and general use:

- **General problem formulation:** Time stepping [scheme \(02_Formulation.ipynb\)](#) and overall program algorithms.
- **Nonlinear Dynamic Analysis program:** General [description \(03_NLDYNA.ipynb\)](#) of input parameters, output requests and general concepts.
- **User elements subroutines:** Example of implementation of a [UEL \(04_UEL_subroutine.ipynb\)](#) subroutine.
- **User material subroutines:** Example of implementation of a [UMAT \(05_UMAT_subroutine.ipynb\)](#) subroutine.

The following sections describe simple to follow applied examples:

- **Example 01:** Linear [static \(06_Example01.ipynb\)](#) analysis of a 2D frame.
- **Example 02:** Linear [dynamic \(07_Example02.ipynb\)](#) analysis of a linear 2D frame.
- **Example 03:** Linear [dynamic \(08_Example03.ipynb\)](#) analysis of a linear 3D frame.
- **Example 04:** [Nonlinear static \(09_Example04.ipynb\)](#) analysis of a simple 1D mass-spring system.
- **Example 05:** [Nonlinear dynamic \(10_Example05.ipynb\)](#) analysis of a 2D frame.
- **Example 06:** Design of laterally loaded [pile \(11_Example06.ipynb\)](#).