

Program Description:

Enoch is an object-oriented framework written in C++. It provides an application program interface (API) that can be used within other programs to represent and analyze finite element models. There is no user interface built into Enoch—it is up to the application programmer to gather data from users and pass it on to the Enoch code.

A structural model consists of an array of Node objects and an array of Element objects. The Node objects store position, displacement, velocity, acceleration and mass values for an arbitrary number of degrees of freedom. The Element class is associated with a set of Node classes and defines an interface for getting an element's stiffness matrix.

A structural model is passed to an Enoch object that performs a Newmark-Beta Time History Analysis. Loads are specified for each degree of freedom at each time step, allowing arbitrary, complex load conditions. The damping matrix is generated using Rayleigh's model of damping.

If all the elements are linear, the stiffness and damping matrices are generated at the beginning of the analysis and remain constant throughout. Structural models incorporating non-linear elements are included in the analysis by updating the global stiffness matrix at each time step. The linear elements are assembled into a global linear stiffness matrix, which remains constant throughout the time-history analysis. At each time step, the tangent stiffness matrices of each nonlinear element are acquired through the interface defined in the Element class.

Non-linear elements are added by creating a new class derived from the Element class. When an Enoch object requests the element's stiffness matrix, any computation to generate the tangent stiffness matrix may be performed.

The abstraction of elements allows many different elements, both linear and non-linear, to be included in the same analysis. Since the time-history analysis interacts with each element through the same interface, new elements can be developed without changing the analysis code. Using the same interface, it is also possible to develop a new analysis module that uses the existing elements.

Modeling:

To investigate the behaviour of multi-degree of freedom structure, a shear wall was modeled. The wall was idealized as three beams stacked vertically, as shown in Figure 1. The vertical and horizontal displacements of the base of the wall were fixed, which the rotational degree of freedom was connected to a fixed node through a plastic hinge.

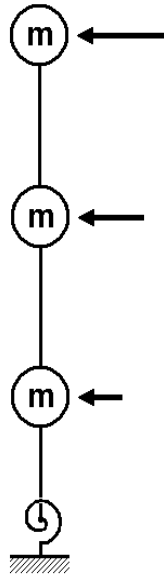


Figure 1 - Shear Wall Model

The wall was a reinforced concrete shear wall. To determine section properties, a representative wall was modeled in Response 2000. The wall was 300mm thick, and 6000mm wide, with longitudinal reinforcement all along the wall but concentrated at either end. Stirrups enclosed the longitudinal reinforcement in six equally sized hoops. Properties determined by response 2000 were as follows:

A [mm ²]	1800000
I [10 ⁹ mm ⁴]	1.67
E [MPa]	30 000

Furthermore, the story height was set at 5m and the time step used was 0.003s.

The wall was modeled with beam elements that only account from bending deformation of the wall. Though this is not the best way to model a shear wall, is adequate to demonstrate the difference between the plastic response of the hinge and the elastic response of the wall.**2.**

The plastic hinge was used to only account for the plastic deformations, and not linear ones. As such it had a high initial stiffness of 5000000 kN*m/rad to ensure that the elastic deformations were primarily those of the beams. A yield moment of 25000 kN*m and a post-yield stiffness of 30000 kN*m/rad were also used.

The model was subjected to a push-pull analysis. Load was applied to each node in an inverted triangular distribution, with the loads on the first and second story 1/3 and 2/3 of the third story load respectively. The load at the top node of the wall rose from 0 to 200 kN, decreased to -200 kN and rose again to zero.

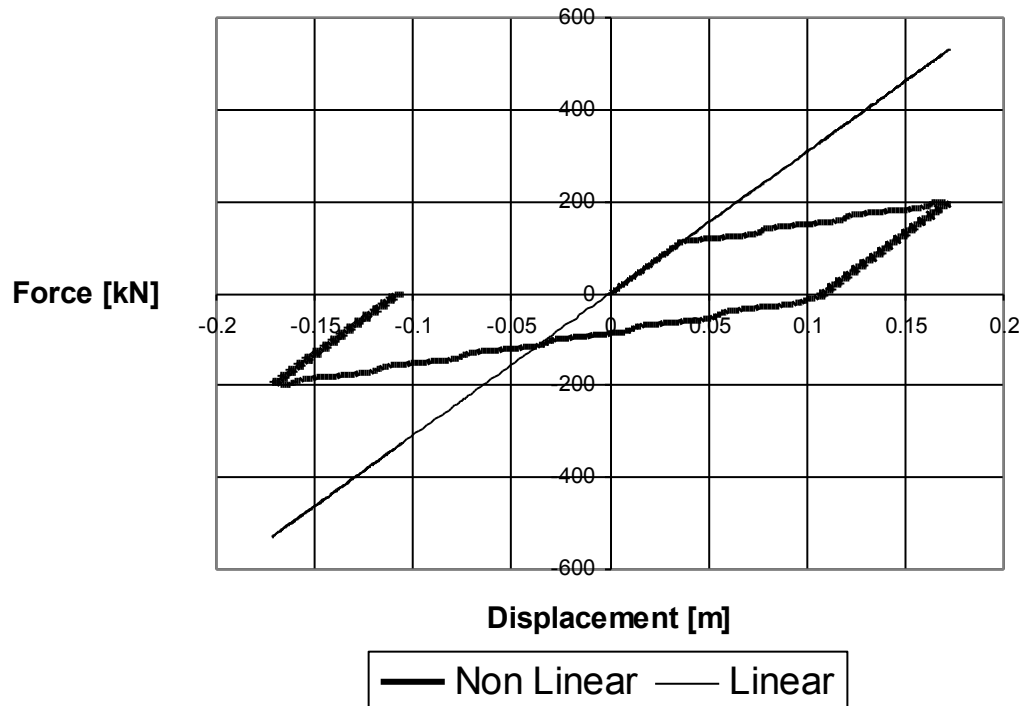


Figure 2 - Force Deflection Response of Top of Wall

Figure 2 shows the force-displacement response of at the top of the wall. Though the beam element connected to the top of the wall responds linearly throughout the loading, the effect of the hinge's plastic response can be seen clearly. This shows that isolating non-linear behaviour to one part of a structure reduces the load imposed on the rest of the structure.

Limitations

The Enoch framework is far from complete. Currently, though damping must be specified in terms of Rayleigh's alpha and beta coefficients, the solutions diverge unless beta equals zero. Only lumped mass is considered, making Enoch unsuitable for modeling some torsional problems.

In general, writing custom finite element analysis code is time-consuming. The program must be both written and validated. The payoff from a flexible, extensible modeling tool is huge and a decision must be made about whether the payoff is worth the trouble. A

standardized framework reduces the overhead cost of writing custom code, while maintaining the flexibility of an open model.

Future Work

There still remains much work to be done on this program. The data structure used to store the loading information needs to be changed. Currently the load is stored as matrix with (number of degrees of freedom) x (number of time steps) elements. This is very inefficient, especially for the case of a constant load. A better solution would be to develop a Load class and associate an instance of that class with each degree of freedom. The Load class would be inherited for different types of loads, for instance ConstantLoad, EarthquakeLoad, SinusoidalLoad, etc... This would simplify the task of using many different loads.

The data output behaviour currently leaves much to be desired. The ultimate solution would be for Enoch to make the data available to another class which would deal with the task of logging the data.

Finally, more validation is needed. Once the above changes have been made, many structures should be tested in many load cases to ensure that the program gives accurate results and to gain an understanding of how small the time step needs to be to guarantee convergence.