A NONLINEAR FINITE ELEMENT TOOLBOX FOR STRUCTURAL CONTROL

Matthew W. Roberts[1], Luciana R. Barroso[2], and Loren D. Lutes[3]

ABSTRACT

The unique informational requirements of structural control often preclude the use of proprietary software. The nonlinear finite element toolbox (NLFET) is a code specifically developed for use in nonlinear structural control research. This paper provides an overview of NLFET including the data structures and algorithms used.

NLFET is implemented in MATLAB in order to make use of the powerful control and nonlinear routines and toolboxes available. The data is stored in MATLAB structures for maximum flexibility and to improve the readability of the code. Element types are defined using an object oriented design so that new elements (both linear and nonlinear) can be added easily and without necessitating changes in the core analysis code.

A small numerical example, including sample code, is presented to better illustrate the use of NLFET.

NLFET is released under the GNU general public license (GPL). The GPL fosters collaboration among software developers by requiring that the source code be available to all. It is hoped that this license arrangement will allow NLFET to become a robust, efficient, and powerful analysis tool for structural control.

Keywords: structural analysis, control, dynamics, MATLAB

INTRODUCTION

The three great virtues of a programmer are *laziness*, *impatience*, and *hubris* (Wall et al. 1996). Judging by the number of structural control papers asserting new or significant discoveries, it appears that the last virtue is well developed in the structural control community. However, much of the simulation in structural control has been done by writing custom software codes, indicating that the first two virtues are not developed fully among structural control programmers. Often, the unique informational

[1]Graduate Research Assistant, Department of Civil Engineering, Texas A&M University, 3136 TAMU, College Station, TX 77843-3136. E-mail: matt@openfem.com

[2]Assistant Professor Department of Civil Engineering, Texas A&M University, 3136 TAMU, College Station, TX 77843-3136. E-mail: lbarroso@civil.tamu.edu

[3]Professor Emeritus and Adjunct Professor, Department of Civil Engineering, Texas A&M University, 3136 TAMU, College Station, TX 77843-3136. E-mail: L-Lutes@tamu.edu

requirements of structural control preclude the use of proprietary structural analysis or finite element codes (Balmès 1999). However, much could be done to save time and reuse software for numeric simulations in structural control. In addition, the recent overview of structural control by Housner et al. (1997) found that "devices and algorithms for . . . control of *non-linear* systems" (emphasis in original) was a high priority of research investigation. The Nonlinear Finite Element Toolbox (NLFET) seeks to fill these needs by providing a nonlinear finite element code which uses an object-oriented approach, provides software that can be used at many levels, and licenses the software under the GNU General Public License (FSF 2002) so that the source code is freely available and redistributable.

OBJECT-ORIENTED DESIGN

Element definition in NLFET is object-oriented. That is, the code for each element provides the element stiffness matrix and all other information needed by the analysis engine. Thus, adding elements to NLFET is accomplished by simply writing a new subroutine which conforms to the application programming interface (API). By separating the element subroutines from the analysis code it is easy to use different analysis engines for specific problems, and changes to the analysis engine are not required when adding new elements.

The MATLAB scripting language is not specifically designed for object-oriented programming. However, the MATLAB language allows quick development and is particularly adapted to numerical computation. There are also many powerful toolboxes available for control [e.g., Control System Toolbox (1998), μ-Analysis and Synthesis Toolbox (2001)] and nonlinear analysis (Shampine and Reichelt 1997).

MODULAR ARCHITECTURE

NLFET can be used at many levels. At the most basic level, NLFET can be used to do low-level functions like assembly of the global stiffness matrix or static condensation. The assembled and/or condensed matrix can then be used with other solvers or with custom analysis codes.

NLFET can also be used for linear and nonlinear dynamic analysis. Currently, linear analysis can be accomplished using the state space routines provided with the Control System Toolbox (e.g. `lsim`, `step`). The MATLAB ode suite of solvers (`ode45`, `ode15s`, etc.) can be used for nonlinear problems.

Post processing is also available with NLFET. The graphical user interface (GUI) tools can also be used with other analysis software if the output conforms to the openly-documented formats of NLFET or by writing a data translation program. Figure 1 shows the mode shape display for a finite element model using the GUI capabilities of NLFET.

SOURCE CODE AVAILABLE

Perhaps the greatest advantage of NLFET is that the source code is freely available for modification and redistribution. By releasing the source code, the users of NLFET can help in the development and correction of programming errors. As noted by Raymond (2000), "Given enough eyeballs, all bugs are shallow." Open source projects,

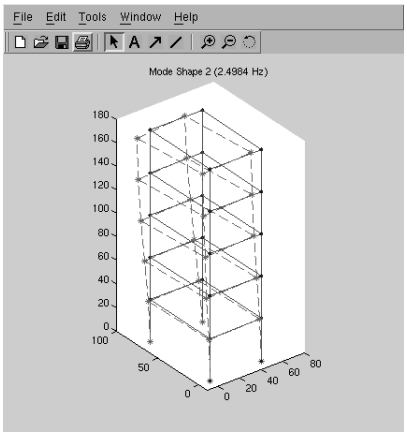

FIG. 1. Mode shape display in NLFET.

like the Linux Kernel and Apache Web Server, are renowned for the stability and robustness of the software and the speed with which coding errors are corrected.

NUMERICAL EXAMPLE

Overview

In order to demonstrate the use of NLFET, a simple numerical example is presented. Data pertaining to the structure is stored in a MATLAB structure variable named `data` in this particular case. The example structure, a single-bay frame with associated mass, is shown in Figure 2. The node numbers of the finite element model are circled and the element numbers are within squares. The columns (elements 1 and 2) are assumed to remain elastic so they are modeled using Bernoulli beam elements. Element 3 uses a nonlinear beam element which can undergo bilinear, hysteretic yielding at the ends, as idealized by the plastic hinges near nodes 2 and 3. Element 4 is a viscous damper providing passive control of the frame. Application of the boundary conditions reduces the problem to three degrees of freedom: the displacement of the mass in the x-direction, X, and the rotation of nodes 2 and 3, θ_2 and θ_3, respectively. A sinusoidal excitation, F, is applied to the mass in the x-direction. The frequency of the force is chosen to be close to resonance so as to build up response and induce yielding.

Active and semiactive control are possible with NLFET (Roberts 2002), but are not included here for simplicity.

Defining Material and Section Properties

NLFET separates the definition of material and section properties for maximum flexibility and coding efficiency. Material definitions are stored as an array of structures. The structural members in the example (elements 1, 2, and 3) are made of steel material which is added to the data structure with the following code (all units are inches and pounds)

```
% Steel 36 ksi
data.materials(1).E     = 29000000;
```

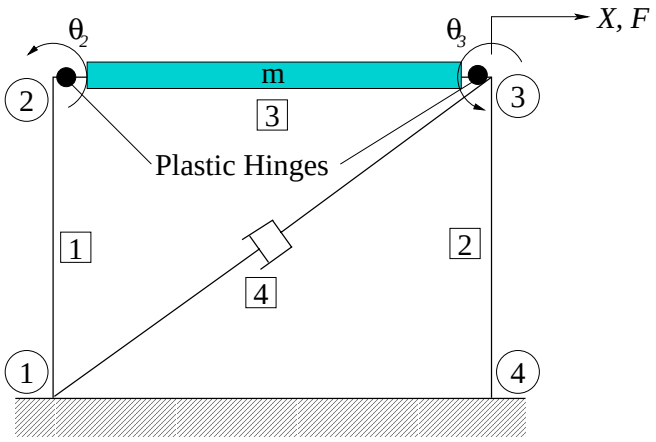

FIG. 2. Simple numerical example frame for dynamic analysis.

```
data.materials(1).Fy    = 36000;
data.materials(1).name  = '36ksi steel';
data.materials(1).alpha = 0;
```

Here, `E` is Young's modulus, `Fy` is the yield stress, `name` is the name of the material, and `alpha` is the coefficient of the post-yielding incremental stiffness of the material (setting it to zero here indicates the material is perfectly plastic). The damper element material requires only the damping coefficient, `c`, in the material definition:

```
data.materials(2).c       = 478.9;
```

The value for `c` used here is chosen so that the effective damping of the structure increases to 5% ($\xi = 0.05$) from an uncontrolled value of 2%.

The columns use a W14x109 section which is defined in the data structure with the following code

```
% W14x109 (columns)
data.sections(1).A=32.0;
data.sections(1).I=1240;
data.sections(1).name = 'W14x109';
```

where `A` is the cross-sectional area and `I` is the moment of inertia. The beam uses a W24x68 section:

```
% W24x068 (beam)
data.sections(2).A=20.1;
data.sections(2).I=1830;
data.sections(2).Z=177.0;
data.sections(2).name = 'W24x68';
```

Note that because the beam can yield, a plastic modulus (`Z`) must be given.

Element Definition

With the materials and sections defined, the definition of the elements is done with the following code

```
data.elements(1).material = 1;
data.elements(1).section  = 1;
data.elements(1).nodes    = [ 1 2];
data.elements(1).type     = 'beam2d';

data.elements(2) = data.elements(1);
data.elements(2).nodes    = [ 4 3];

data.elements(3).material = 1;
data.elements(3).section  = 2;
data.elements(3).nodes    = [ 2 3];
data.elements(3).type     = 'nlbeam2d';

data.elements(4).material = 2;
data.elements(4).nodes    = [ 1 3];
data.elements(4).type     = 'vdamper2d';
```

The `material` and `section` are the indexes into the materials and sections array, respectively. The `nodes` field is a connectivity vector for the element, and `type` indicates the element type (a corresponding `.m` file must be defined: `beam2d.m`, `nlbeam2d.m`, etc.)

SIMULATION RESULTS

The simulation is 15 seconds long and includes a linear analysis for comparison with the results of the nonlinear analysis. The MATLAB `lsim` command is used for the linear analysis, and the ODE suite of nonlinear, differential equation solvers (Shampine and Reichelt 1997) is used for the nonlinear analysis. Both analysis algorithms require a state space formulation so that second-order differential equations of motion must be converted to first-order differential equations. This results in a state-space system with six states, $X, \theta_2, \theta_3, \dot{X}, \dot{\theta}_2, \dot{\theta}_3$. In addition, the nonlinear analysis requires two additional states, ϕ_2 and ϕ_3 which represent the angle of rotation "inside" the plastic hinges at nodes 2 and 3, respectively.

Figure 3 shows the rotation at node 2 (θ_2) for the linear and nonlinear analysis. The rotation on the inside of the plastic hinge (ϕ_2) is also plotted in the figure. The "plateaus" in the ϕ_2 state indicates that the beam is yielding at the plastic hinges. The total run time for the nonlinear case was over 11 minutes compared to 0.23 seconds for the linear analysis. Thus the nonlinear analysis is over 2800 times slower, indicating that a different solution algorithm (i.e. Newmark-Beta) may be more efficient.

CONCLUSION

The nonlinear finite element toolbox has been developed to ease numerical simulation in structural control. The main advantages of this software are the object-oriented

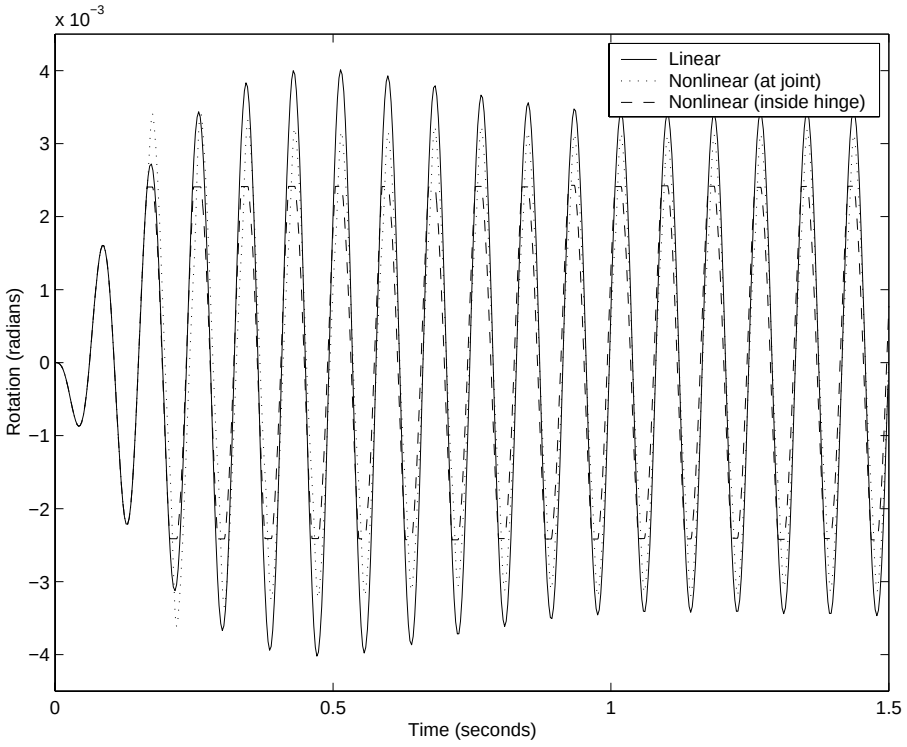

FIG. 3. Rotation at node 2 from a linear and nonlinear analysis.

design, overall modular architecture, and the availability of the source code, released under the GNU General Public License. A simple dynamic analysis example has been presented to illustrate the use of NLFET. Further information, including the full code for the example shown here, can be found by visiting `http://www.nlfet.org/`.

REFERENCES

Balmès, E. (1999). "Sensors, degrees of freedom, and generalized modeshape expansion methods." *International Modal Analysis Conference (IMAC)*, Orlando. 628–634.

Control System Toolbox (1998). The MathWorks, Inc., Natick, Massachusetts.

Free Software Foundation (FSF) (2002). *GNU General Public Licence (GPL)*. World Wide Web, `http://www.gnu.org/licenses/gpl.html`.

Housner, G. W., Bergman, L. A., Caughey, T. K., Chassiakos, A. G., Claus, R. O., Masri, S. F., Skelton, R. E., Soong, T. T., Spencer, Jr., B. F., and Yao, J. T. P. (1997). "Structural control: Past, present and future." *Journal of Engineering Mechanics*, 123(9), 897–971.

μ-Analysis and Synthesis Toolbox (2001). The MathWorks, Inc., Natick, Massachusetts.

Raymond, E. S. (2000). *The Cathedral and the Bazaar*. `http://www.tuxedo.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/`.

Roberts, M. W. (2002). “Control of a three-dimensional nonlinear building with magnetorheological dampers,” PhD thesis, Texas A&M University. In preparation.

Shampine, L. F. and Reichelt, M. W. (1997). “The MATLAB ode suite.” *SIAM Journal on Scientific Computing*, 18(1), 1–22.

Wall, L., Christiansen, T., and Schwartz, R. L. (1996). *Programming Perl*. O’Reilly & Associates, Inc., Sebastopol, CA, 2nd edition.