13th World Conference on Earthquake Engineering
Vancouver, B.C., Canada
August 1-6, 2004
Paper No. 78

RECENT DEVELOPMENTS IN SNAPS3D – A FINITE ELEMENT ANALYSIS PROGRAM FOR 3-DIMENSIONAL NONLINEAR TIME-HISTORY ANALYSIS OF PRECAST CONCRETE STRUCTURES

Serdar ASTARLIOGLU[1], Andrew SCANLON[2], Ali M. MEMARI[3]

SUMMARY

This paper describes the enhancements made to the finite element analysis program (SNAPS3D) developed to carry out materially nonlinear static or dynamic analysis of three-dimensional precast concrete building structures. The finite element library consists of three-dimensional solid elements, shell elements to model wall panels and horizontal diaphragms, frame elements to model beams and columns, and connection elements to model the interface between diaphragm and wall panels as well as the interface between beams and columns. The material library consists of linear elastic, elastoplastic, uniaxial concrete, and shear-slip material models.

INTRODUCTION

The approach taken by most building codes towards the design of precast concrete structures in seismic zones has been to emulate the design principles used in monolithic structures as a result of uncertainties concerning the performance of precast concrete structures during earthquakes. When dry connections are used, the behavior of a precast concrete structure under seismic load can differ from a similar reinforced concrete structure to a great extend, since highly localized nonlinear behavior in the form of rocking and sliding is observed at the connection regions.

While a two-dimensional approach with the assumption of rigid floor diaphragms may be sufficient for simple or coupled shear wall layouts, in case of flexible floor diaphragms and unsymmetrical lateral load resisting system arrangements, this approach may not provide sufficient accuracy. Because of this, previous analytical research into the behavior of precast structures has been limited to two-dimensional cases, where the wall systems were isolated from the rest of the structure with the assumption of rigid floor diaphragms (Kianoush [1,2], Caccese [3], and Pekau [4]).

SNAPS3D has been developed to perform nonlinear dynamic analysis of three-dimensional precast concrete building structures and to determine the effect of diaphragm flexibility, rocking and sliding mechanisms on the overall response of the building. This paper summarizes the changes and enhancements made to SNAPS3D.

[1] S. Astarlioglu, The Pennsylvania State University, USA, serdarA@engr.psu.edu
[2] A. Scanlon, The Pennsylvania State University, USA, axs21@psu.edu
[3] A. M. Memari, The Pennsylvania State University, USA, amm@psu.edu

DESCRIPTION OF SNAPS3D

SNAPS3D was initially developed (Astarlioglu [5]) to model the behavior of large panel precast structures. The finite element library consisted of three-dimensional solid elements to model wall panels and horizontal diaphragm components, beam elements to model regular beams and coupling beams, spring elements to model the behavior of joints between panels and diaphragms, and truss elements to model vertical post-tensioning. While solid, beam, and truss elements exhibited only linear-elastic behavior, spring elements had nonlinear material behavior to model potential crushing of concrete, sliding of panels at horizontal and vertical joints, and joint openings.

The source code for SNAPS3D was written in FORTRAN 90 using a procedural programming approach where different tasks were performed by specialized subroutines. The program consisted of the following subroutines: input, loading, incremental loading, stiffness matrix assembly, solution of equations, residual force calculations, convergence check, and output results, in addition to modules for storage of global arrays such as nodal coordinates, element connectivity, material properties, and boundary conditions. Figure 1 shows the initial program layout (Owen [6]).

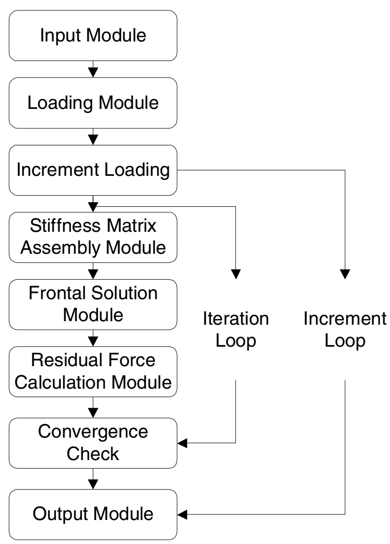

Figure 1. Flow Chart for SNAPS3D

The new version of SNAPS3D, which will be described in the following sections, is radically different from the initial version summarized above in two aspects: Element / material library and programming approach.

Element Library

Solid Element

In the original program, 8-node solid elements shown in Figure 2 were utilized to model the wall panels and floor diaphragms. Unfortunately, this approach requires through the thickness meshing of these structural members, resulting in high number of elements. For this reason, solid elements were replaced with continuum based shell elements as the primary finite element for modeling these members. However, solid elements are still available in SNAPS3D for reasons, which will be discussed in programming approach section.

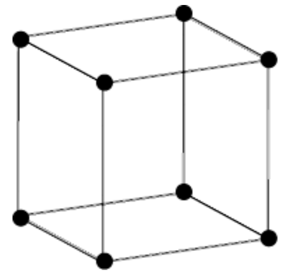

Figure 2. 8-node solid element.

Shell Element
The shell elements implemented in SNAPS3D are based on continuum-based shell approach pioneered by Ahmad [7] for analysis of curved shells. Since these wall panels and floor diaphragms are flat (not curved) structural members, the formulation of the element was modified to incorporate drilling degrees of freedom (Cook [8]). The resulting element is a sub-parametric finite element in which the geometry interpolation is based on the shape functions of an 8-node solid element, while the displacement interpolation is based on a 12-node solid element (with additional nodes at centers of side faces). The element, shown in Figure 3, has eight master nodes that have three translational degrees of freedom (dof) and four slave (internal) nodes that have three translational and three rotational dof.

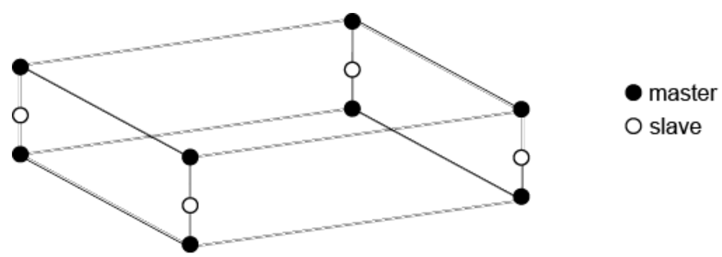

Figure 3. 8-node shell element with 4 slave nodes.

Frame Element
Degenerated continuum based approach used in development of shell element has also been used in the development of the frame element, shown in Figure 4. Further details on this type of frame element are available in Belytscko [9]. In this case, 4 master nodes at each end of the frame element are condensed into a single slave node with three translational and three rotational dof. This element can be used to model rectangular beams and columns.

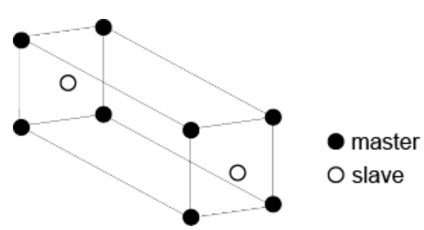

Figure 4. 8-node frame element with 2 slave nodes.

Connection Element
The connection element, shown in Figure 5, is very similar to the frame element in most aspects. The major difference is that the connection element does not use gauss integration points for stiffness matrix integration. Instead, the integration points have to be provided by the user. This element can also be used to model beams and columns that do not have a rectangular cross section such as I and L beams.

Alternatively, using the nonlinear material models that are described in the next section, it is possible to use this element to model column to beam and wall panel to horizontal floor connection interfaces. In this mode, different materials can be assigned to each integration point. Unlike the spring element, which requires a load-deflection (lumped) relationship to define nonlinear behavior, the connection element requires stress-strain (distributed) relationship.

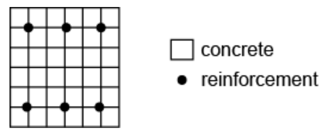

Figure 5. Cross-section of connection element.

The elements described above allow the modeling of not only be used to model connections in precast panel buildings, which was the original objective of SNAPS3D, but precast frames with PRESSS type connections (Priestley [10]) as well. Figure 6 shows a typical finite element mesh at a platform type wall panel / floor diaphragm connection. Figure 7 shows a typical finite element mesh at a frame connection.

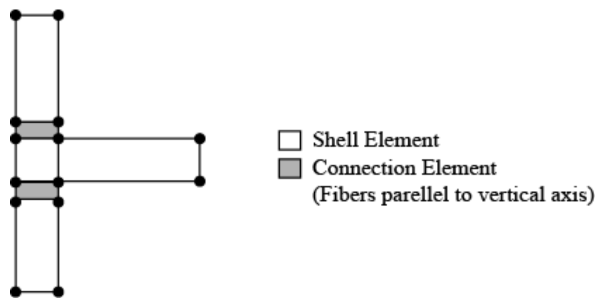

Figure 6. FE mesh for a wall to slab connection.

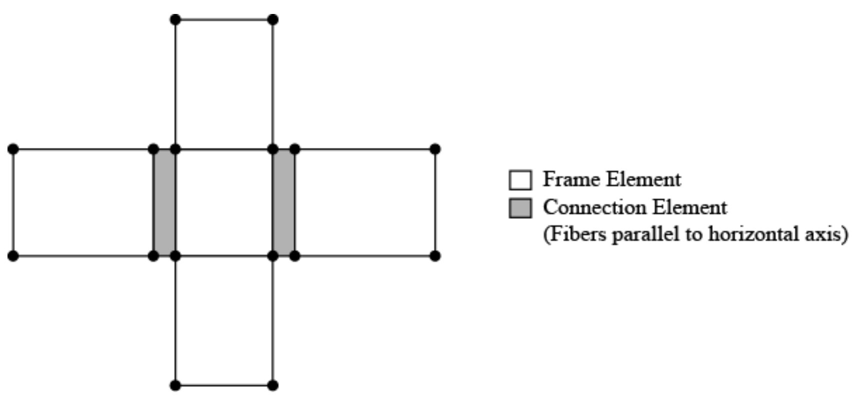

Figure 7. FE mesh for a column-beam connection.

Material Library

Aside from the linear elastic material model, SNAPS3D has uniaxial concrete (Figure 8), steel (Figure 9), and shear-slip (Figure 10) material models to simulate the nonlinear behavior of connection interfaces under seismic load. Of these models, concrete and steel models can be assigned to different integration points on the same element to represent the axial stress-strain relationship, while shear-slip model has to be assigned to all the integration points.

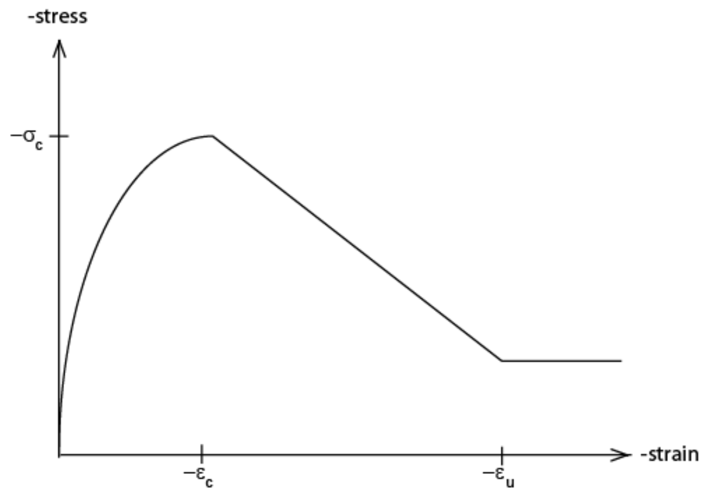

Figure 8. Uniaxial concrete stress-strain curve.

The nonlinear part of the concrete stress-strain relationship in Figure 8 is in the following form (Darwin [11]):

$$\sigma_i = \frac{E_0 \varepsilon_i}{1 + \left(\frac{E_0}{E_s} - 2\right)\left(\frac{\varepsilon_i}{\varepsilon_c}\right) + \left(\frac{\varepsilon_i}{\varepsilon_c}\right)^2} \tag{3}$$

Where,

ε_i	:	Uniaxial strain in the principal stress direction
σ_c, ε_c	:	Maximum compressive stress and corresponding strain
E_0	:	Initial tangent modulus
E_s	:	Secant modulus

The tensile strength of concrete is neglected, and the unloading/reloading is assumed to occur parallel to initial tangent modulus.

Figure 9 shows the elastic-perfectly plastic stress strain relationship assumed for steel. The compression and tension branches of the stress-strain relationship are assumed to be identical, unloading/reloading is assumed to take place parallel to the initial modulus, and strain hardening is not considered.

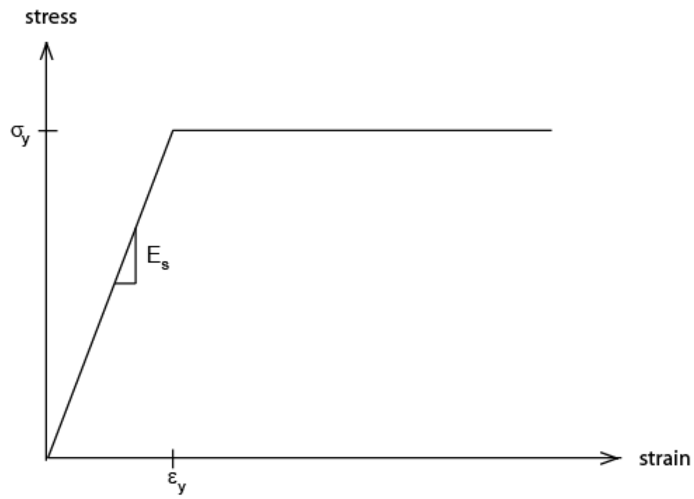

Figure 9. Steel stress-strain model.

Figure 10 shows the shear-slip relationship used in this study. This relationship is only activated when the integration point under consideration is under compressive stresses (σ). If the normal stresses become zero (gap opens), then the shear resistance is lost, till the gap closes again. The parameters defining the relationship are:

G_d	:	Tangent shear modulus of the connection material
G_s	:	Tangent shear modulus of the slip surface
μ	:	Coefficient of friction
σ	:	Compressive stress acting on the point

The unloading/reloading is assumed to take place parallel to the initial modulus.

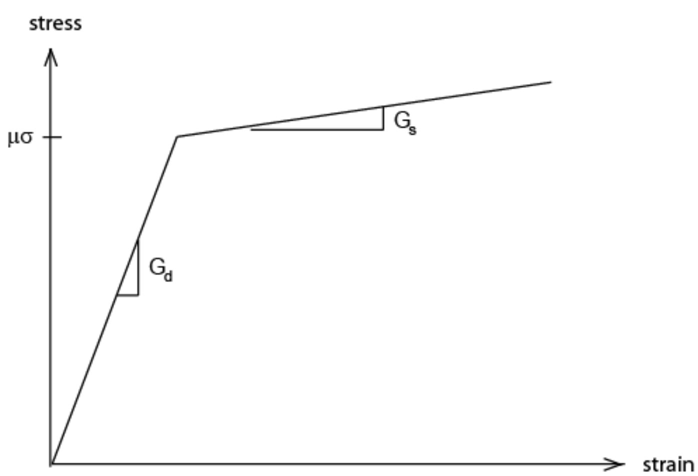

Figure 10. Shear-slip relationship

Programming Approach

The source code for SNAPS3D has been completely rewritten in C#, an object-oriented programming language developed by Microsoft [12]. In this approach, the program is organized into objects that store both methods (equivalent to FORTRAN subroutine/function) and data, and a group of objects that have common behaviors, methods, and operators are called classes. This results in a source code, which is easier to maintain and modify (by adding new elements and/or materials) as reported by Forte [13]. SNAPS3D contains classes for nodes, elements, materials, solvers, windows forms (for input and output), and matrices. The concepts of inheritance and polymorphism are also used throughout the program. Figure 11 shows the implementation of inheritance for derivation of element classes.

Each element contains properties that define its nodes, integration points, and material(s), as well as methods to calculate the shape functions, strain displacement matrix, constitutive matrix, stiffness matrix, mass matrix, and internal force vector. The *element* base class provides abstract information on these properties and methods. Each subclass derived from *element* base class must implement its own version of any of the methods mentioned above. The *solid* element and *truss* element classes, which directly inherit from *element* class, have to implement their own versions of the methods for evaluating the shape functions and strain-displacement relationships, etc.

Shell and *frame* element classes inherit from *solid*, and add new properties as internal (slave) nodes, which do not exist in the *solid* element class. These subclasses overwrite some of the methods, such as the method for calculating constitutive matrix, from the *solid* element class with their own versions. The method for calculating the stiffness matrix on the other hand is not overwritten and both *shell* and *truss* element classes use the solid version of the method for stiffness matrix calculation.

The other important benefit of object-oriented approach is polymorphism. For example, in the case of static analysis, the class for assembling the global stiffness matrix, all the elements are accessed by their base class name (*element*). As a result, addition of a new element class does not require the assembly class to be changed.

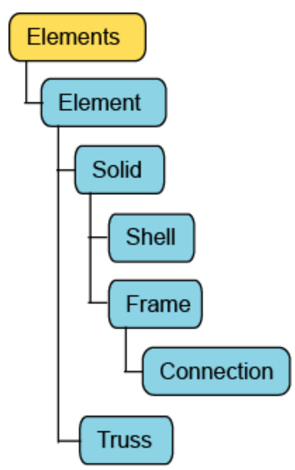

Figure 11. Elements namespace and element class hierarchy.

The way the matrices are handled is also quite different from the original version of the program. In the object-oriented version, there are two matrix classes: *submatrix* and *matrix*. *Submatrix* is identical to a FORTRAN style matrix and is filled with real numbers and when an instance of this class is created, enough memory to hold all the elements (in this case real numbers) of the matrix has to be allocated. *Matrix* class on the other hand, is a matrix of *submatrix* objects. When an instance of *matrix* class is created, memory to hold all the elements (in this case *submatrix* objects) is not automatically allocated. All stiffness matrices in SNAPS3D (element or global) are of *matrix* type.

RESULTS

SNAPS3D is still in development/verification phase. Results will be made available when program testing and verification is completed.

CONCLUDING REMARKS

Previous analytical studies on behavior of precast structures have been limited to two-dimensional models. The computer program being developed in this study promises to provide useful insights into the three-dimensional behavior of precast concrete large panel structures and frames under seismic load.

REFERENCES

1. Kianoush MR, Scanlon A. “Inelastic Seismic Response of Precast Concrete Large Panel Coupled Shear Wall System.” University of Alberta, 1986.
2. Kianoush MR, Elmorsi R, Scanlon A. “Response of Large Panel Precast Wall Systems: Analysis and Design.” PCI Journal 1996: 90-108.
3. Caccese V, Harris HG. “Seismic Resistance of Precast Concrete Shear Walls.” Earthquake Engineering and Structural Dynamics 1987; 15: 661-677.
4. Pekau OA, Hum D. “Seismic Response of Friction Jointed Precast Panel Shear Walls.” PCI Journal 1991; 36(2): 56-71.
5. Astarlioglu S, Memari AM, Scanlon A. “Modeling Strategies for Three Dimensional Analysis of Precast Panel Buildings Under Seismic Load.” Proceedings of the 12th World Conference on Earthquake Engineering, Auckland, New Zealand, Paper No. 1356, 2000.

6 Owen DRJ, Hinton E. “Finite Elements in Plasticity: Theory and Practice.” Swansea: Pineridge Press, 1980.

7. Ahmad S, Irons BB, Zienkiewicz OC. “Analysis of Thick and Thin Structures by Curved Finite Elements.” International Journal for Numerical Methods in Engineering 1970; 2: 419-451.
8. Cook RD. “Four-Node Flat Shell Element: Drilling Degrees of Freedom, Membrane-Bending Coupling, Warped Geometry, and Behavior.” Computers and Structures 1994; 50(4): 549-555.
9. Belytscko T, Liu WK, Moran B. “Nonlinear Finite Elements for Continua and Structures.” West Sussex: John Wiley and Sons Ltd, 2000.
10. Priestley MJN, Sritharan S, Conley JR, Pampanin S. “Preliminary Results and Conclusions from the PRESSS Five-Story Precast Concrete Test Building.” PCI Journal 1999: 42-67.
11. Darwin D, Pecknold D. “Inelastic Model for Cyclic Biaxial Loading of Reinforced Concrete.” University of Illinois at Urbana-Champaign, 1974.
12. Microsoft Corp. http://www.microsoft.com.
13. Forte BWR, Foschi RO, Stiemer SF. “Object-Oriented Finite Element Analysis.” Computers and Structures 1990; 34(3): 355-374.