Object-Oriented Probabilistic Finite Element Analysis and its Applications

Jin-Oh Jeong, Chan Cook Park and Uk Youn Cho

Institute for Advanced Engineering, Korea

ABSTRACT

The object-oriented approach enables software to be more extensible, reusable and easily readable compared with the procedure-based approach. The library named POOFEL (Probabilistic Object-Oriented Finite Element class Library) is presented by using Rumbaugh's OMT methodology such as object modeling and data flow diagram. When dealing with the structural reliability with respect to random variables, the probabilistic finite element analysis has shown to be an effective tool. In this paper, a perturbation-based probabilistic finite element formulation for plane elasticity is presented and implemented in the object-oriented framework. The results of J-integral by using the developed POOFEL were presented and compared with the results by using ABAQUS as tests. The numerical results of the mean and variance values of beam deflection are also presented.

1. INTRODUCTION

Most of the FEM packages are designed in the procedure-oriented paradigm and are written in the procedure-oriented languages such as FORTRAN or C. According as the size of program is growing with the complexity of engineering software, researchers or developers become to spend more time and efforts on the management or maintenance or extension of their programs. The extensibility of the application code can be more critical as the involved system is more complex, i.e., and the reusability of a part of the existing code is sometimes emphasized to reduce the development cost. Of course, the modularization of the program can be utilized to solve the extensibility. However, as far as the procedural approach is used for the large application, a great advance in extensibility may be too difficult to be achieved.

Several researchers have proposed to design the finite element analysis, matrix solver and preprocessor such as a mesh generator. For example, typical finite element analysis by S.P.Scholz[1], transient non-linear coupling problems by Rihaczek and Kroplin[2], matrix class library in C++ by G.W.Zeglinski et al.[3] have been presented. Besides, other matrix solvers such as IML++, MV++, SparseLib++ are developed by utilizing object-oriented concepts and language. Several object-oriented finite element packages such as Diffpack and FE++ were developed.

Mechanical failure of some safety-related components in the nuclear power industry is mainly due to the crack growth, which can cause catastrophic failures or slow failures like
fatigue. The probabilistic analysis of structural reliability has been used to evaluate the performance of structural components under severe environments. It has been known that there are a number of sources of uncertainties in structural design and analysis such as loading conditions, material properties, component geometry and boundary conditions. Probabilistic numerical methods in mechanics have become popular to assess those uncertainties. For example, Monte Carlo Simulation (MCS) involves direct simulation of behavior concerned with random behavior. However, the multivariate distribution functions must be known and the computing time may be impractical. Indirect approaches may be probabilistic finite element method (PFEM), second-moment analysis[4] and fast numerical integration[5] for solving problems involving random behavior. Most of PFEM applications are based on perturbation theory because it is relatively simple and tractable. Several researchers have presented to solve linear or nonlinear coupled problems in the various fields such as geotechnical engineering[6], random stiffness in simple beam[7] and structural dynamics[8]. However, due to the intrinsic complexity of probabilistic analysis, there exist very few presentations dealing with probabilistic fracture mechanics in object-oriented framework.

In this paper, a prototype of object-oriented perturbation-based probabilistic finite element class libraries (POOFEL) is proposed and implemented to evaluate the probabilistic J-integral by using object modeling technology (OMT) proposed by Rumbough[9]. In addition, OOFEL is tested by evaluating the fracture toughness in the Single-Edge Notched Tension (SENT) specimen. The numerical results are presented and compared by using ABAQUS.

2. DESIGNING OF OBJECT-ORIENTED SOFTWARE

The object-oriented technology is used to facilitate the development of software components by using the object concept. The object-oriented programming (OOP) is natural to the development of software because the reasoning in object-oriented way is closer to the real world than the procedural way.

Basic characteristics of OOP are data abstraction and encapsulation. Objects are assembled with related data and specific functions together have the boundary from outside and have the specific interface with other objects. Encapsulation is an extension of the information hiding of the structural programming. The detailed methods are effectively implemented internally behind the objects and share the necessary information with other objects. The data within the objects can be used only through the method. This methodology protects the important information of the object and has the following essential advantages. The object cannot change the variables of the other objects. Otherwise, the objects may change the original properties. The objects protect themselves from outside action and have their own methods to use their variables.

The simplification in the operation of the objects may be achieved by hiding their variables. One object only needs to know how to exchange the information with others and how to keep the information is hiding internally within the objects. Because encapsulation not only simplifies the relation between the objects but also minimizes the interaction with other objects, reusability of objects can be made easily possible. The new data structure called 'Class' is used for the reusability of objects. Class is defined as the general type of object and it has the member variables, member functions and the simplified interface with outside. Thus, it contributes to the simplification of the programming, maintenance and extensibility of the application code.

Hierarchical structure between classes is implemented through the inheritance. The
member variables and member functions in the derived classes are inherited from the base classes. Thus, it contributes to the simplification of the programming, maintenance and extensibility of the application code. The hierarchical structure can be obtained by using the inheritance of the 'IS-A' relationship between the classes.

The objects may have polymorphic characteristics which is represented by dynamic binding implemented through the virtual function in C++. Certain methods (member function in C++) can be interpreted as other functions by using the polymorphism. Virtual functions defined in base classes or derived classes are redefined in the lower class. Because virtual functions are executed by the trigger of a run-time object in tree of the hierarchical structure, the core part of the application does not need any modification even if any newly derived class is added.

With the suitable combination of encapsulation, inheritance and polymorphism, reusability, readability and simplification of coding can be achieved. Thus, researchers can concentrate on the physical modeling rather than on implementation of the application.

Even though there exist many object-oriented methodologies by various authors such as Yourdon, Booch and Rumbaugh, OMT by Rumbaugh is applied to the design of probabilistic finite element method in this paper.

3. PROBABILISTIC FINITE ELEMENT ENGINE BY OMT

3.1 Object modeling

A short statement of general finite element analysis is needed to choose the candidate classes for which are derived from the list of important nouns summarized the problem. The possible basic classes are extracted from the candidate classes to obtain the engine for the general-purpose finite element analysis within the framework of object-oriented programming. These basic classes are the basis on the object modeling. The extracted basic classes are used to indicate the relationship between the objects. We remove the several classes from the candidate classes such as unimportant classes, ambiguous classes, attribute or implementation.

Fig. 1 Object modeling diagram for OOFEM engine by using OMT methodology

The association between the classes needs to be defined. The attribute of classes can be extracted from the description of the classes. Fig. 1 shows the object modeling diagram based on the OMT terminology
3.2 Functional modeling

Relative priority between the dynamic and functional modeling is given according to the considered system. If the reactions to the user, such as input from the keyboard, are a minor consideration to the overall system analysis, in other words, in case of procedure-oriented analysis, the dynamic analysis is seldom used in the object-oriented modeling. The functional modeling is only considered in this study.

The processes between the objects should be defined to construct the functional modeling. Because they are related with the operation in the object modeling, the processes for the FE analysis need to define the operation of the classes. After the data flow is defined for each class, the beginning and ending points for each process are clarified. They may be another process, an actor or data storage. A data flow diagram is shown in Fig. 2 after the above analysis is performed on each operation of the object modeling. Fig. 3 shows data processes of probabilistic finite element analysis and consequential class definition is shown in Fig. 4.
4. FORMULATION FOR PFEM

4.1 Probabilistic finite element formulation for plane elasticity problem

In case of plane elasticity problem, material properties like Young's modulus and external load can be considered as random variables. Besides, dimensions of substrates and geometry can be also regarded as random variables. More complicatedly, the spatial randomness of those variables can be considered. In this paper, only Young's modulus and surface traction are considered as random variables. Applying the above conditions to the equations from stochastic potential energy minimization principles[11], we can get total six equations which consist of one from zeroth order equality, Eq. (1), three from first order, Eq. (2) and another one from second order equality, Eq. (3) which is transformed into Eq. (4) if random variables are independent among each other.

\[ \frac{\partial d}{\partial \sigma} = \frac{\partial G}{\partial \sigma} = 0 \quad \frac{\partial d}{\partial \sigma} = \frac{\partial G}{\partial \sigma} = 0 \quad \frac{\partial d}{\partial \sigma} = \frac{\partial G}{\partial \sigma} = 0 \quad \frac{\partial d}{\partial \sigma} = \frac{\partial G}{\partial \sigma} = 0 \quad \frac{\partial d}{\partial \sigma} = \frac{\partial G}{\partial \sigma} = 0 \quad \frac{\partial d}{\partial \sigma} = \frac{\partial G}{\partial \sigma} = 0 \]

\[ \left[ \frac{\partial d}{\partial \sigma}, \frac{\partial d}{\partial \sigma}, \frac{\partial d}{\partial \sigma}, \frac{\partial d}{\partial \sigma}, \frac{\partial d}{\partial \sigma}, \frac{\partial d}{\partial \sigma} \right] = \left[ \frac{\partial G}{\partial \sigma}, \frac{\partial G}{\partial \sigma}, \frac{\partial G}{\partial \sigma}, \frac{\partial G}{\partial \sigma}, \frac{\partial G}{\partial \sigma}, \frac{\partial G}{\partial \sigma} \right] \]

\[ K \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 d}{\partial b_i \partial b_j} \text{cov}(b_i, b_j) = -2 \left( \frac{\partial^2 K}{\partial E \partial E} \text{cov}(E, E) + \frac{\partial^2 K}{\partial E \partial P_1} \text{cov}(E, P_1) + \frac{\partial^2 K}{\partial E \partial P_2} \text{cov}(E, P_2) \right) \]

\[ K \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 d}{\partial b_i \partial b_j} \text{cov}(b_i, b_j) = -2 \frac{\partial^2 K}{\partial E \partial E} \text{cov}(E, E) \]

4.2 A formulation for probabilistic characteristics of J integral

J integral can be expressed as below Eq. (5) whereas the domain integral method[12] is applied to get numerical expression. In that case, the crack is stationary and thermal stress is ignored.

\[ J = \iint \left[ \sigma_{ij} \frac{\partial u_i}{\partial x_j} - \frac{\partial q_j}{\partial x_i} \right] dA \]

Where 2nd order perturbation is applied to get stochastic characteristics of J integral value, the response variables, \( \sigma_{ij} \), W can be expanded with random variables, b's and J can also be expanded in Eq.(6).

\[ J = J + \frac{\partial J}{\partial b_a} \Delta b_a + \frac{1}{2} \frac{\partial^2 J}{\partial b_a \partial b_b} \Delta b_a \Delta b_b \]

Replacing the response variables with the expanded terms and rearranging Eq. (5) with the same order terms, J and J derivatives with b's are found. Eq. (8),
(9) and (10) shows the equivalence of the J derivatives.

\[
J = \left[ \sigma_y \frac{\partial A_y}{\partial x} - W \frac{\partial H}{\partial x} \right] dA \quad (8)
\]

\[
\frac{\partial J}{\partial b_y} = \int \left[ \left( \frac{\partial \sigma_y}{\partial b_y} \frac{\partial A_y}{\partial x} + \sigma_y \frac{\partial}{\partial b_y} \left( \frac{\partial A_y}{\partial x} \right) \right) \frac{\partial H}{\partial x} - \frac{\partial W}{\partial b_y} \frac{\partial H}{\partial x} \right] dA \quad (9)
\]

\[
\frac{\partial^2 J}{\partial b_y \partial b_y} = \int \left[ \left( 2 \frac{\partial \sigma_y}{\partial b_y} \frac{\partial}{\partial b_y} \left( \frac{\partial A_y}{\partial x} \right) + \frac{\partial^2 \sigma_y}{\partial b_y^2} \frac{\partial A_y}{\partial x} \right) \frac{\partial H}{\partial x} + \frac{\partial^2 W}{\partial b_y \partial b_y} \frac{\partial H}{\partial x} \right] dA \quad (10)
\]

Once J and its derivatives are obtained, we can get the expectation value and variance[13] of J integral from the below Eq. (11) and Eq. (12).

\[ E[J] = \bar{J} + \frac{1}{2} \sum_{a,b} \frac{\partial J}{\partial b_a} \frac{\partial J}{\partial b_b} \left[ \text{cov}[b_a, b_b] \right] \quad (11) \]

\[ \text{Var}[J] = \sum_{a,b} \left( \frac{\partial^2 J}{\partial b_a \partial b_b} \right) \left[ \text{cov}[b_a, b_b] \right] \quad (12) \]

5. APPLICATIONS

5.1 Simple beam bending

To test the implemented probabilistic finite element code, the result of the code is compared with the result of Monte Carlo simulation in the deflection of a beam which suffers concentrated load at the end of the beam. Table 2 shows the expectation and standard deviation of the deflection.

<table>
<thead>
<tr>
<th>Table 2 Comparison of the results between PFEM and MCS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inputs</strong></td>
</tr>
<tr>
<td>Young's Modulus</td>
</tr>
<tr>
<td>Load</td>
</tr>
<tr>
<td>MCS</td>
</tr>
<tr>
<td>Young's Modulus</td>
</tr>
<tr>
<td>Sampled</td>
</tr>
<tr>
<td>Load Sampled</td>
</tr>
<tr>
<td>MCS result</td>
</tr>
<tr>
<td>PFEM result</td>
</tr>
</tbody>
</table>

5.2 J-integral

To verify the result of numerical J integral, it was compared with the result of ABAQUS in case of Single Edge Notched Tension condition. Since there exists a semi-empirical formula for fracture toughness in that case, the result was also compared with the formula[14]. Deformation of SENT conditioned specimen can be seen in Fig. 5 and Fig. 6 shows that the result of the implemented code coincides with the result of ABAQUS and semi-empirical formula.
6. CONCLUSION

An object-oriented class library is designed, implemented in C++ and applied to evaluate J-integral in SENT specimen. A probabilistic formulation for J-integral using the results of PFEM analysis is also presented but the results is not yet presented. The result of J-integral will be given through further implementation. Also, various diagrams such as object model and functional model for PFEM are presented. A deterministic evaluation of J-integral in object-oriented framework is presented and shows a good coincidence with results by using ABAQUS. A perturbation-based finite element formulation for plane elasticity is given and implemented and applied to a beam deflection problem. The result was compared with that of MCS.

ACKNOWLEDGMENT

This work has been taken in conjunction with the Development Program of Engineering Key Technology and supported by MOST and DAEWOO in Korea.

REFERENCES