

行政院國家科學委員會專題研究計畫 成果報告

以 Python 整合有限元素商業軟體於重複性分析計算架構之
開發：應用於結構可靠度分析

計畫類別：個別型計畫

計畫編號：NSC92-2211-E-002-075-

執行期間：92年08月01日至93年07月31日

執行單位：國立臺灣大學土木工程學系暨研究所

計畫主持人：呂良正

共同主持人：陳俊杉

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 93 年 11 月 1 日

行政院國家科學委員會補助專題研究計畫 成果報告
 期中進度報告

以 Python 整合有限元素商業軟體
於重複性分析計算架構之開發：
應用於結構可靠度

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 92-2211-E-002-075

執行期間： 92 年 8 月 1 日至 93 年 7 月 31 日

計畫主持人：呂良正

共同主持人：陳俊杉

計畫參與人員：林士傑

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：國立台灣大學土木工程學系

中 華 民 國 93 年 10 月 29 日

摘要

本研究利用先進的軟體設計技術：OOP、OOAD 和 UML，提出一重複性分析計算架構，結合有限元素商業軟體 ABAQUS 和可靠度分析程式從事可靠度分析。各個分析軟體間的整合，本研究採用直譯式程式語言 Python，利用其整合程式的快速、程式碼的簡潔以及物件執行期可變之彈性，實作此軟體架構。並採用一以 XML 作為介面之通用性文字檔解析器，用以解析 ABAQUS 之輸出輸入檔。本研究並以此軟體架構，探討與分析抗彎構架、板殼結構於工作載重和極限載重下之可靠度。

關鍵詞：軟體設計、Python、結構可靠度分析、反應面法

Abstract

The research uses the advanced software design technology: OOP, OOAD and UML to propose a computational framework for recursive analyses. The framework integrates commercial FEM software ABAQUS and reliability analysis codes to carry out structural reliability analysis by taking advantage of the great analysis capability of ABAQUS. The scripting language Python is used to implement the proposed software framework. The features of Python including fast gluing of different programs, clear syntax, and the flexibility of Python object during runtime contribute to the implementation of the framework. The research also uses a general text-file parser that provides a user interface based on XML to parse both the input and output files generated from ABAQUS. Finally, the proposed framework is used to study the reliability of framed structures and plate and shell structures under the service load and the limit load.

Keywords: Software design, Python, Structural Reliability Analysis, Response Surface Method

前言

由於電腦運算速度不斷提升，且能勝任繁複的迭代工作，所以在科學、工程的領域中，電腦程式模擬一直是驗證理論研究的最佳工具，使研究者能短時間得到用人力相當耗費時間之運算結果。因此在各個領域中，現今已有為數眾多為了解決科學計算或工程設計之商業套裝軟體。以結構可靠度為例，由於可靠度的分析問題除了可靠度計算之理論，如反應面法 (RSM)、蒙地卡羅法 (MCS) 和一階可靠度分析法 (FORM) 等之外，還需結構分析程式以提供不同力學問題之解析。有鑑於其分析之耗時與複雜，多種商業軟體，如 STRUREL、VaP、COMPASS 等因應而生。

然而，現有之可靠度分析商業軟體之結構分析程式，多數僅針對某些物理或力學問題而由軟體設計公司自行研發，其能解決問題之廣度與深度不若一般泛用型結構分析或有限元素分析之商業軟體，如 ABAQUS、SAP、ETABS 等。當使用者遇到那些昂貴的商業軟體無法支援之問題時，往往必須投入大量之時間自行開發。

因此，本研究使用 Python 程式語言搭起可靠度分析程式和多種泛用型有限元素分析程式之運算架構；並運用目前在資料處理中發展迅速之 XML、SQL 等語言，歸納整理出不同階層之軟體程式介面；搭配設計模式、架構模式、UML 之軟體分析與設計概念，希冀帶給使用者和研究者一個易使用、擴充性高、功能完整之軟體架構。

研究目的

觀察結構可靠度有關文獻，可發現大部分處理的問題是構架結構(framed structures)(桁架、梁、柱、剛架等)而且幾乎都是線彈性的，此原因應為一般研究可靠度的人對於分析門檻較高之板殼結構或非線性(幾何/材料)並不熟悉，就算是有些熟悉，要寫一個此類分析程式所花費的時間也相當長(可能是可靠度分析程式開發時間的百倍)。而這類較複雜的分析問題，利用目前有限元素商業軟體是很容易解決的，但是在做結構可靠度分析時，如何利用這些有限元素商業軟體卻碰到瓶頸，因為可靠度分析幾乎都需要將可靠度分析的程式與結構分析程式整合於同一程式，因此在沒有商業軟體的程式碼(source codes)情況下，幾乎是無法進行的。

一般結構可靠度的極限狀態函數(limit state function)常是隱式(implicit)的，雖然可以利用蒙地卡羅法(Monte Carlo Simulation, MCS)經由大量抽樣求得失效機率，但如果每次分析所費的時間較長(如三維或非線性問題)，此法並不可行。本計畫以反應面法(Response Surface Method, RSM)先求取近似之顯式(explicit)極限狀態函數(經由少數抽樣數目之特殊抽樣方法)，然後以一階可靠度法 FORM (First Order Reliability Method) 法迭代求得可靠度指標(Nowak and Collins 2000)。反應面法應用於可靠度分析可使真實之隱式結構反應函式以顯式之多項式模擬，但仍需要計算結構反應值(Faravelli 1989, Kim and Na 1997)。文獻上較複雜的結構之可靠度分析其結構分析程式皆由作者自行開發(Liu and Moses 1994, Sweden Technology Centre 2001)，實因使用泛用型結構分析商業軟體，將牽涉可靠度程式與結構分析程式間之溝通問題。

本研究之目的即在嘗試運用新的軟體技術，建立彈性的軟體架構，並進一步解決此程式間複雜的溝通問題，所開發的架構將有開發時間短及應用問題廣等優點。本研究擬採用的軟體示意架構基本上是以物件導向直譯式程式語言 Python 為中心的軟體架構，串連可靠度分析的方法與泛用型有限元素分析商業軟體 ABAQUS。

文獻探討

放眼文獻上較有名的可靠度分析商業軟體（Sweden Technology Centre 2001，Schneider 1997），如表一中之 VaP，其並非使用反應面法作為可靠度分析方法，因此極限狀態方程式必須由使用者輸入，以致此軟體使用性有前述之限制。再如 STRUREL，其亦只提供特定某些極限狀態方程供使用者選擇，且其系統可靠度是建立在桿件可靠度上，有串聯和並聯之分類。

表一 可靠度分析常用之商業軟體

軟體名稱	擁有/開發者	可靠度分析方法	主要功能與特點
VaP	Institute of Structural Engineering, Swiss Federal Institute of Technology, Zurich / Schneider et al	蒙地卡羅法，一階可靠度法，數值積分法	視窗化之使用介面（MS-Windows only）
STRUREL	Munich University	蒙地卡羅法，一階可靠度法（FORM），二階可靠度法（SORM）	依功能不同包含了五部分軟體： STATREL：實驗資料統計分析，將資料點符合到某一種統計分佈上。 也可做時間序列之統計分析 COMREL：個別桿件之時變或非時變之可靠度分析 SYSREL：系統（桿件之串、並聯）可靠度分析 NASCOM：有限元素分析程式 NASREL：NASCOM 和 COMREL 之連結程式
CalRel	UCB (USA)	與 STRUREL 同	包含了多種機率密度函數分佈，亦包含了敏感度分析。極限狀態方程可由使用者自訂副程式中輸入
NESSUS	Southwest Research Intitute (USA)	與 STRUREL 同	本為美國太空總署（NASA）所使用，可與 ABAQUS、Patran 做連結，著重於太空中構件之潛變分析
COMPASS	Martec Ltd. (Canada)	與 STRUREL 同	提供一可選擇之破壞模式庫。極限狀態方程可由使用者自訂副程式輸入
PROBAN	DNV (Norway)	與 STRUREL 同	使用自行研發之商用結構分析軟體 SESAM
COSSAN	University Innsbruck (Austria)	重點抽樣之蒙地卡羅法	包含系統可靠度分析
ISPUD	University Innsbruck (Austria)	重點抽樣之蒙地卡羅法	針對特定可靠度指標作結構設計
proSINTAP	DNV (SAQ) / SINTAP Consortium (Sweden/EU)	蒙地卡羅法及一階可靠度法	針對裂縫之可靠度分析
UMFRAP	UMIST (UK)	一階可靠度法	可提供設計安全係數之探討
STAR 6	British Energy (UK)	與 STRUREL 同	提供破壞與倒塌機率之分析

綜合以上所述，可知雖然文獻上有一些商業化之可靠度分析軟體，但除了價格普遍偏高

(動輒數千、數萬美元)之外,由於其結構分析軟體大都為自行研發,能解決之問題的廣度便受到限制,能解決特定種類的問題而無法廣泛地探討各種領域所遭遇到不同的問題。為解決以上的盲點,讓研究者能不必或大大減少開發結構分析程式與可靠度分析程式之連結的時間,且讓可靠度分析能更廣泛地應用在各個研究領域中;本研究擬以在可靠度分析領域中被廣泛使用之反應面法配合一階可靠度方法作為可靠度分析方法,結合泛用型商業軟體 ABAQUS,以跨平台之直譯式語言 Python 作為各應用程式間連接之橋樑,透過前述多種 Python 所支援之標準語言與模組,並輔以 wxPython 做視窗化之使用者介面,發展出一套完整且自動化之可靠度分析軟體,提供研究者方便使用並期能短時間擴充、開發符合其需求之程式;更希冀此程式軟體架構能為可靠度的應用開啟嶄新的一扇門。

研究方法

為了達到以上之目的,本研究運用現今軟體設計的觀念與技術:物件導向程式語言、模式語言(UML)和設計模式(Design Patterns)。大多數的程式設計思維分為循序式程式設計,物件導向程式設計。兩者最大的不同在於前者著重於演算法,先針對問題流程中需要之功能,依序建構;後者以物件的角度,由問題之大處慢慢抽絲剝繭成各個物件往小處思考,整個大問題拆解成一個一個互相關連的物件後,在實作各個物件內部的細節。

由於本研究著眼於軟體整合,探討的是許許多多不同軟體間的黏結、溝通的介面處理問題,而非龐大複雜的數值計算程式。因此,選擇物件導向程式設計將是合宜的選擇。物件導向程式設計有三個最重要的特徵:

(1)封裝(encapsulation):

物件的封裝即是將物件內部資訊隱藏起來(information hiding),使用者僅能透過類別設計的介面來操作物件,使用者不能也不需要知道內部的實作細節。這對於減少程式碼間的耦合能有很大幫助。

(2)繼承(inheritance):

繼承最主要能讓一個類別與另一個類別建立父與子之階層關係。繼承類別者稱子類別,被繼承者稱為父類別。子類別可以使用父類別的屬性(attribute)與方法(method),並可以增加屬性與方法及修改。

(3)多型(polymorphism):

簡單來說,類別透過繼承能使用父類別的方法,但也能 override 其方法;許多中間的程式碼不需要更動,只要在客戶端餵給子類別生成的物件,中間的程式碼用到同名之方法時,會因多型之特性而呼叫到子類別之同名方法。

設計模式(design pattern)是許多程式設計師之物件導向程式設計經驗的累積;此觀念最早是由建築師 Christopher Alexander 所提出:「每一則模式都在描述某種一再出現的問題,並描述解決方案的核心,讓你能依此變化出各種招式,解決上萬個類似的問題。」(Gamma et al. 1995),後來此觀念才延伸到程式設計領域。在程式設計中,設計模式是由彼此有關連性之類別或介面所組成,而互相之間的調整能夠套用在程式設計上進而解決問題,增加程式碼之可攜性與再利用性。所以,一般來說,設計模式的組成,包含了(1)模式名字,通常是程式設計領域內慣用之簡短詞彙;(2)模式應用時機,提供可解決問題之情境;(3)解決方案之組成,既然模式是由類別組成,特定的模式必然有其特定的架構;(4)模式之實行效果。透過注入設計模式觀念,希冀本研究之軟體架構能更有彈性。

使用設計模式概念當然不能缺少共通的語言 UML。因為物件導向之設計概念為分析、設計、實作三部分反覆再修改之反覆性設計(iterative design)。因此,設計圖表、文字文件必須要保存完整且有一致性,才不至於使程式設計師之間的溝通不良,造成誤解;程式也不易傳承與維護,UML 便提供了解決方案(Albir 1999)。

UML 不只是一種用來描繪圖形的表示法，他是一種語言，但並非程式語言，而是模式語言。UML 除了富有語意之文件格式外，亦提供了多種圖形 (diagram)，以圖像敘述個物件的狀態、互相間的關連性等，供工程師能很快明瞭架構和其運作。其中包含了 (1) 類別圖 (class diagram)，可表示類別之屬性、類別方法等和不同類別間之關係 (如繼承等)；(2) 物件圖 (object diagram)，用來表示物件之實例變數，顯示靜態物件之快照 (snapshot)；(3) 互動圖 (interaction diagram)，提供某一使用例 (use case) 時間序上之交互關係、變數傳遞流程等。透過分析與設計，詳實記錄成 UML 文件後再來實作，然後再進入反覆修改，方能有一個好的軟體架構誕生 (Gamma et al.1995; Larman 2001; Mark 1996)。

其次，為了整合不同之商業軟體，本研究利用直譯式程式語言 Python 的 JAVA 版本：Jython (Pedroni 2002)，來作為整合軟體架構之核心語言。Python 是由 Guido van Rossum 於 1991 年所發表，除了具備前述本研究所需條件外，其餘如：(1) 跨平台，(2) 程式碼易讀，(3) 完整物件導向技術，(4) 高度模組化之結構，(5) 類別庫支援多，包含數值運算、圖像處理等皆是其優點。且由於其核心程式碼是公開的，全世界不斷有人致力於使其功能更加完備，所以不斷更新的函式庫及類別庫使其更有效率。

而 Jython 則是將 Python 直譯器以 JAVA 程式語言來實作。因此，程式設計師可以在 Jython script 中直接使用 JAVA 的內建的類別、自訂類別，以及 AWT 和 SWING 等視窗繪圖函式庫。以往為人所質疑之 JVM 和直譯式程式語言之速度，也隨著 JVM 效率日益改善，相對的 Jython 的執行速度也大幅改善。由於本研究擬使用之文字解析器是用 JAVA 寫成，再者考慮往後程式使用者介面或整合環境設計之必要，因此本研究選擇 Jython 作為軟體整合之核心語言。

本研究可靠度分析程式為自行撰寫之 FORTRAN 程式，雖有程式碼，但為了模擬使用商業軟體的情形，亦只使用編譯過之執行檔 (.exe)，將程式當作黑盒子來呼叫。

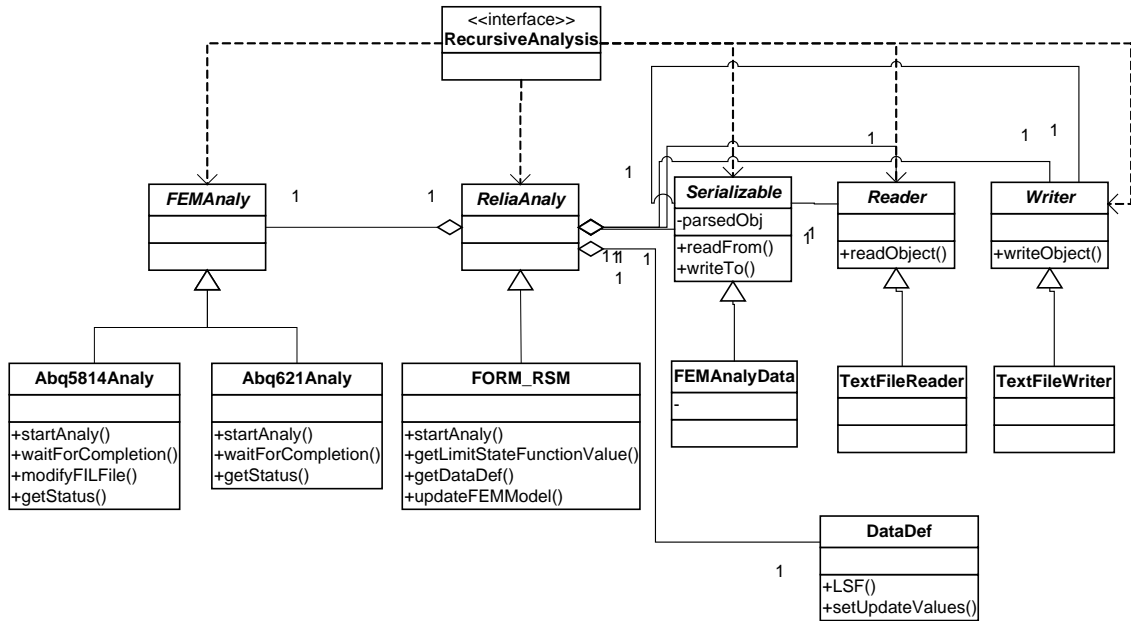
由於可靠度分析過程中須呼叫有限元素商業軟體，為了讓可靠度程式使用方便，本研究以分段式之程式撰寫邏輯將可靠度分析程式拆成兩段：一段是在有限元素商業軟體分析前，一段是在有限元素商業軟體分析後，已求得極限狀態方程之值；其判別則以一寫在文字檔 "index.txt" 裡之數字指標來做判別。

本研究之可靠度分析程式使用反應面法 (RSM) 搭配一階可靠度分析方法之一階二次矩法 (FOSM)。反應面法之樣本點選取提供兩種選擇：軸向法與二因子法。

結果與討論

本研究提出之重複性分析計算架構如圖一所示；圖一為 UML 裡之設計類別圖 (Design Class Diagram)。此架構主要分為三大模組：FEMAnaly、ReliaAnaly 和 Seriliable 模組。ReliaAnaly、FEMAnaly 兩者皆為抽象類別 (Abstract Class)，其實作類別各自發展，繼承該抽象類別，兩者間無須知曉對方之實作內容，僅以抽象類別當介面來溝通，此為 Bridge Pattern。另外 Seriliable 利用了 Serializer 設計模式。

RecursiveAnalysis 為一個接收參數、生成所有物件的控制者介面，而 ReliaAnaly 則接收其所有生成之物件：有限元素分析模組 FEMAnaly、有限元素分析資料模組 FEMAnalyData 等等來統籌操作，形成 "has a" 之 composition 關係。



圖一 可靠度分析軟體架構類別圖

而 ReliaAnaly 類別之方法，除了 startAnaly 基本啟動可靠度分析之外，尚須 getLimitStateFunctionValue 針對每組設計變數擷取相對應之極限狀態方程值；getDataDef 擷取更新有限元素模型資料。這兩個方法所需之資訊則利用 DataDef 類別裡，LSF：定義和計算極限狀態方程值以及 setUpdateValues：定義設計變數與非設計變數之隨機變數之間的關係。UpdateFEMModel 方法則是做更新有限元素商業軟體之模型之用。

而類別屬性方面，由於現今設計之架構 ReliaAnaly 需統籌整個可靠度分析，所以 FEMObj、Serializable、Reader、Writer 等皆為此類別之 attribute。Beta 和 Pf 則儲存可靠度分析結果之可靠度指標和破壞機率。而 TOTAL_FEMANALY_CALL 是用來計算計算呼叫有限元素商業軟體次數之 counter。

有限元素分析模組 (FEMAnaly) 則僅著眼於有限元素商業軟體的驅動和分析；並且記錄分析狀態的資訊。各個有限元素商業軟體驅動方式不同，透過繼承 FEMAnaly，使用者可 override startAnaly 方法；而圖一中 waitForCompletion 方法是為了確保有限元素商業軟體完全分析結束，可靠度分析程式才會繼續進行後續的計算；這是因為雖然程式的運作理論上是照順序來，應不會有重疊的情形發生，但是有時候有限元素模型非常龐大、例如進行動力或非線性分析時，雖然有限元素分析已完畢，但輸出結果會多花一些時間，而可靠度分析又需要從其輸出結果擷取資訊，有時便會因為這段時間差而出現錯誤，導致分析程式中斷。因此，我們增設此一方法讓使用者針對各個不同有限元素商業軟體的特性來實作。

本研究就以上所提出之重複性分析軟體架構，連結有限元素商業軟體 ABAQUS。ABAQUS 為學界、業界相當知名之有限元素商業軟體，由 HIBBIT, KARLSSON & SORENSEN, INC.公司 (HKS) 所出品。其元素、材料和分析類型相當豐富。從一般熟悉的線性結構分析、挫屈分析到幾何、材料非線性分析、動力分析、裂縫破壞分析等各種力學問題皆有提供解決方案。此外，ABAQUS 可以根據使用者之需求，自訂 user subroutine 或參數。由於 ABAQUS 所用之理論與程式品管相當嚴謹，故成為業界、研究單位最常用與信賴之有限元素商業軟體之一。

啟動有限元素商業軟體方法迥異；然而本研究所用之有限元素商業軟體 ABAQUS 提供從命令列執行之方式。我們便把這啟動方式定義在 startAnaly 方法裡。直接在 Jython 程式碼裡鍵入：

```
os.system(ABAQUS_install_path+"abq641 job="+FileName)
```

ABAQUS_install_path 為 ABAQUS 在電腦上之安裝路徑，FileName 為欲執行之輸入檔名稱。Jython 裡”os.system”方法會在命令列執行括弧內的字串，如此能重複啟動 ABAQUS 分析。

由於幾乎所有的有限元素商業軟體其分析結果、模型資訊等資料，皆提供文字檔的形式供使用者讀取，因此，為了能通用地解析各種類型之文字檔，本研究利用直譯式程式語言 Jython 能直接匯入 Java 類別物件的優點，整合莊勝欽（2003）所開發之通用型文字檔解析器。此解析器透過 XML 語言作為使用者介面，每個欲解析之文字檔需搭配撰寫一描述檔，此描述檔由使用者自訂欲解析資料之宣告、解析方式。

此文字解析器並無設計「寫」的功能，但作可靠度分析時，每次迭代更新之隨機變數的值，都要用來更新有限元素模型。理論上，我們可照原本之架構，開發一套”write”之更新指令，提供使用者在描述檔裡使用。但此作法相當耗費時間。事實上，解析文字檔的動作和更新文字檔的動作大同小異：從檔案開頭開始巡禮，到目標位置，一個做 read 動作，一個則做 write 動作。因此，若將更新文字檔的動作簡化來看，只有兩個要素：欲更新之值，和欲更新值在文字檔中之位置。欲更新之值，由可靠度分析程式算得；而欲更新值在文字檔中之位置，當通用性文字解析器在解析文字檔時，已走過一遍；只要在解析文字檔時，一併記錄欲更新之位置，即可達到更新文字輸入檔之目的。

為了以上之需求，本研究在該通用型文字檔解析器中之 Parsing Action 裡增加一個指令：AssignFilePosition。並在 XML 描述檔裡追加 Assign 這個子元素。其撰寫格式如下：

```
<Assign Target="aPositionName" Method="AssignFileStreamPosition">
```

當解析至欲更新之位置時，便可以此指令將此檔案位置記錄至”aPositionName”。由於檔案位置是由 byte 數計算，為整數，因此，使用者必須事前宣告一個變數名稱為 aPositionName”之 Int 來儲存檔案位置。圖二展示 Assign 元素之用法。

本研究利用直譯式程式語言 Jython 之特性，在執行期動態組合物件；此一以 Jython 物件溝通資料訊息之介面使可靠度分析中極限狀態方程式之組成更具彈性及通用性。其將解析之結果儲存在一樹狀結構內，但由於其資料單元（Variable、Table、Array 和 DataObject）皆為物件，取得其儲存之資料皆須透過類別方法，所以資料指涉之方式相當冗長與麻煩，為了使使用者能方便使用解析有限元素分析輸出檔之結果，本研究針對該通用性文字解析器，寫一個轉換程式，將其轉成 Jython 物件，以期方便使用。

以圖三為例，我們所希望之指涉資料方式為 parameter.Area 或 CLoad.Load 等方式來取的資料。這樣的指涉方式在程式裡可看成有一個物件叫做 parameter，其中有屬性為 Area，也是一個儲存資料的物件，其中又有屬性；這樣一來，當我們使用此物件要得到 Area 的值時，其指涉方式便與前例我們希望的相同。

因此，轉換的步驟便是一層一層走過通用性文字解析器所解析之樹狀結構之節點，一層層增加 Jython 物件。而下一層之節點則為這一層 Jython 物件之屬性，屬性名稱則為該節點名稱。這樣的作法，所用程式語言需能動態增加物件之屬性，也就是在執行期組合物件來使用。由於每個分析問題欲解析之資料皆不相同，其解析結果之樹狀結構層樹、節點數也都不同；此轉換程式必定需靠執行期視該解析結果，動態組合物件。Scripting 程式語言強大的功能便在此發揮。

一般編譯式程式語言之類別屬性或變數在編譯期便確定，不可能在執行期出現原本不知道名稱之變數或屬性。但 scripting 程式語言可以輕鬆做到。Jython 將類別所有屬性名稱以字串儲存在一類別內建串列裡：__dict__。因此，當我們欲增加物件之屬性時，只需將該屬性名稱擴增至__dict__裡即可。

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited with XMLSPY v5 U (http://www.xmlspy.com) by William (Perso
- <TextFileDescription>
  <DataTemplate />
  - <DataDeclaration>
    <Datum Type="String" Name="Title" />
    - <Datum Type="Table" Name="Nodes" Count="4">
      - <Fields>
        <Field Type="Int" Name="NodeNum" />
        <Field Type="Double" Name="X" />
        <Field Type="Int" Name="X_Pos" />
        <Field Type="Double" Name="Y" />
        <Field Type="Int" Name="Y_Pos" />
      </Fields>
      - <UserDefinedParsing>
        - <Loop Count="4" Index="i">
          <Read Target="i.NodeNum" Method="ReadString" />
          <Pass Method="PassAfterChar" Parameter="," />
          <Read Target="i.X" Method="ReadString" />
          <Assign Target="i.X_Pos" Method="AssignFileStreamPosition" />
          <Pass Method="PassAfterChar" Parameter="," />
          <Assign Target="i.Y_Pos" Method="AssignFileStreamPosition" />
          <Read Target="i.Y" Method="ReadString" />
        </Loop>
      </UserDefinedParsing>
    </Datum>
  </DataDeclaration>
+ <ParsingStream>
</TextFileDescription>

```

圖二 Assign 元素使用範例

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited with XMLSPY v5 U (http://www.xmlspy.com) by Wil
- <TextFileDescription>
  - <DataDeclaration>
    - <Datum Type="Table" Name="parameter" Count="2">
      - <Fields>
        <Field Type="Double" Name="Area" />
        <Field Type="Int" Name="Area_FileStreamPosition" />
        <Field Type="Double" Name="Ix" />
        <Field Type="Int" Name="Ix_FileStreamPosition" />
        <Field Type="Double" Name="Iy" />
        <Field Type="Int" Name="Iy_FileStreamPosition" />
      </Fields>
      + <UserDefinedParsing>
    </Datum>
    - <Datum Type="Table" Name="CLoad" Count="1">
      - <Fields>
        <Field Type="Double" Name="Load" />
        <Field Type="Int" Name="Load_FileStreamPosition" />
      </Fields>
      + <UserDefinedParsing>
    </Datum>
  </DataDeclaration>
- <ParsingStream>
  <Parse For="parameter" />
  <Parse For="CLoad" />
</ParsingStream>
</TextFileDescription>

```

圖三 XML 描述檔範例

以此轉換程式為例，我們先定義一個空類別 PyObj，沒有任何屬性或方法。若我們產生一個 PyObj 的物件 s，以圖三這個例子來看，我們需先增加 parameter 和 CLoad 這兩個 Table，因此可以如下面方式來實作：

```
s.__dict__.append("parameter")
s.__dict__.append("CLoad")
```

依此類推，當我們要指涉第一個斷面積時，便可使用 s.parameter.Area[0]的方式，相當簡潔，且由於皆用使用者在撰寫描述檔時所自訂之名稱，對於使用者使用上相當自然與方便，不會因為程式寫固定一些儲存資料之陣列符號或名稱，而讓使用者常常在使用上要思索某個 array index 是代表哪一個物理問題裡的數值。注意，由於 parameter 是 Table，如前面所提到，table 是由多個 field 之 array 所組成，因此我們轉換時，用 Jython 裡之內建型別"list"來儲存，所以需用中括號（"[]"）和數字引數來指涉資料。

有限元素模型更新之部分，筆者利用解析有限元素輸入檔時一併紀錄之資料位置，搭配從 DataDef 模組裡所定義之欲更新數值，來做有限元素輸入檔更新動作。從 DataDef 模組我們設計回傳兩個串列（list）給主程式，再將這兩個串列指派給 Writer 使用。由於通用性文字解析器在記錄資料位置時，是以 byte 數來記錄；因此，筆者為了一致和正確性，以 binary 模式開啟檔案串流，依照每個欲更新數值和其所對應之位置，加以寫入檔案中。

有了 Jython 物件能使用，如何實作 DataDef 模組讓使用者能對於解析資料作一些簡單的數值運算（加、減、乘、除、冪方、加總等）、邏輯運算等。原本之構想為設計一個圖形使用者介面（GUI），提供一些數學運算和邏輯運算功能讓使用者組合。這樣一個需求下之環境，正與 scripting 的環境類似；再加上 Jython 能動態納入 script 模組至主程式使用；因此，本研究便以 Jython script 做為定義資料之介面。

而本研究軟體架構（圖一）內 DataDef 模組之 setUpdateValues 方法接收兩個參數，除了解析資料轉換成之 Jython 物件 s 之外，還有更新過的設計變數 x（x 為串列，名稱亦可自訂）。同樣由這些物件需組合成欲更新之隨機變數值（本例之 new 串列，名稱使用者可自訂）和相對應更新位置之串列（pos 串列，名稱亦可自訂），再回傳給主程式。接著便以數個例題介紹本研究開發之程式介面的使用。

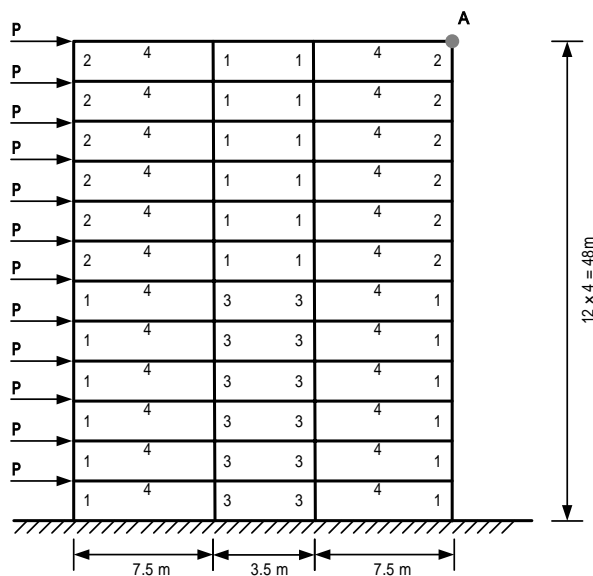
本研究應用此軟體架構探討傳統抗彎構架結構於工作載重下之可靠度分析的問題。工作性要求主要建立在正常使用下，結構物功能發揮狀況與使用者使用舒適度。探討結構於工作載重下之可靠度分析問題，最常見定義極限狀態函式為容許位移（ δ_a ）與結構在工作載重下靜力分析位移（ δ ）之差：

$$G = \delta_a - \delta$$

如圖四為十二層三跨構架（趙國藩 1996）：材料彈性模數為 $E = 2.0 \times 10^7$ kN/m²。桿件斷面慣性矩與桿件斷面積之關係為： $I_i = \alpha_i A_i^2$ （ $i=1, 2, \dots, 5$ ）。隨機變量為桿件斷面積 A_1 、 A_2 以及側向集中力 P；其隨機性質如表二所示。極限狀態方程為：

$$G = 0.096 - u_A$$

其中 0.096 為最大容許水平位移， u_A 為節點 A 之水平位移。由表三之分析結果比較可發現本研究之程式與文獻分析結果相符。



圖四 十二層三跨構架

表二 十二層三跨構架隨機變量性質

Random Variable	Mean	COV	Distribution Type	α_i
$A_1(m^2)$	0.25	10 %	Lognormal	0.08333
$A_2(m^2)$	0.16	10 %	Lognormal	0.08333
$A_3(m^2)$	0.36	10 %	Lognormal	0.08333
$A_4(m^2)$	0.20	10 %	Lognormal	0.26670
$A_5(m^2)$	0.15	10 %	Lognormal	0.20000
$P(kN)$	30.0	25 %	Ext. Type I	

表三 十二層三跨構架之可靠度分析結果

文獻及使用方法	Reliability Index
趙國藩 (1996)	1.4531
本研究 (純二次次反應面搭配 FORM)	1.4545

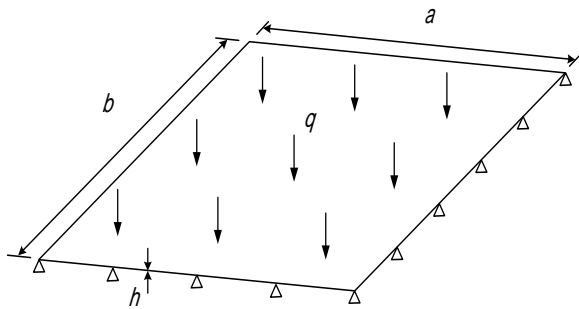
本研究亦應用此軟體架構探討文獻上甚少著墨之版殼結構可靠度分析的問題。首先探討板殼結構於工作載重下之可靠度分析問題：如圖五為一四邊簡支承版結構受均佈載重 q ， $a=6m$ ， $b/a=1.5$ ，板厚 $h=0.1m$ 。極限狀態方程之形式與上例相同：

$$G = 0.02 - u_{center}$$

其中 0.02 (m) 為最大容許水平位移， u_{center} 為板結構中點之垂直變位。

表四 四邊簡支承板結構之隨機變量性質

Random Variable	Mean	COV	Distribution Type
$h(m)$	0.1	10 %	Normal
$E(GPa)$	20.0	10 %	Normal
$q(kN/m^2)$	1.0	20 %	Normal



圖五 四邊簡支承之板結構

本研究以 ABAQUS 之 Shell Element 來模擬此板結構；所用之元素代號為 S8R，如圖六所示，其為二次、六個自由度（三個位移、三個轉角）reduced integration（四個積分點）之有限元素。以 ABAQUS CAE 建結構模型（如圖七）。

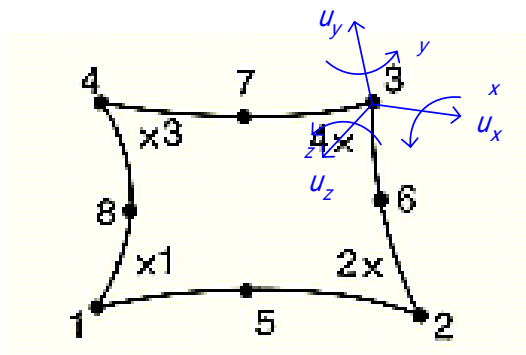
由於此例題之中點垂直變位有近似解（Timoshenko 1959）：

$$w_{\max} = \alpha \frac{qa^4}{D}$$

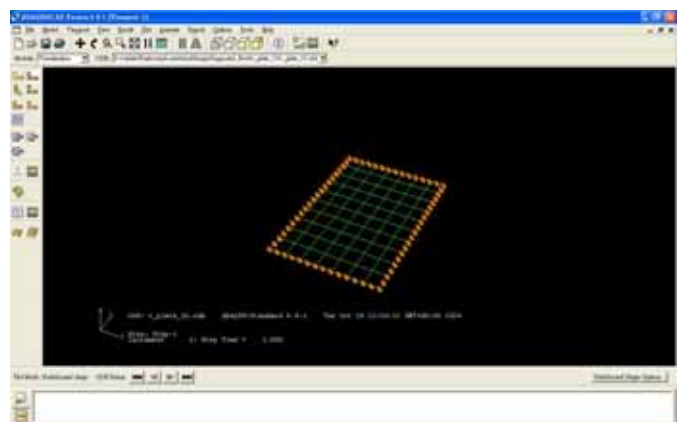
where $\alpha = 0.0072$ for $b/a=1.5$, $\nu = 0.3$

$$D = \frac{Eh^3}{12(1-\nu^2)}$$

因此極限狀態方程可表示成顯式的方程式，即可以一階二次矩法（FOSM）來求得可靠度指標。圖八為此題 ABAQUS 分析結果輸出檔之 XML 描述檔。透過通用性文字解析器和本研究之包裝而形成 Jython 物件，便可由使用者使用來定義極限狀態方程（如圖九）；使用者直接以描述檔內宣告資料之名稱（如本題之 U）來組合極限狀態方程即可。表五即為以顯式極限狀態方程和 FOSM 相對於 ABAQUS 和 FOSM 和 RSM 所做之可靠度分析結果比較。



圖六 ABAQUS shell element S8R



圖七 四邊簡支承板結構之 ABAQUS 模型

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited with XMLSPY v5 U (http://t
- <TextFileDescription>
- <DataDeclaration>
  <Datum Type="Double" Name="U" />
  </DataDeclaration>
+ <ParsingStream>
</TextFileDescription>

```

```

def LSF(s):
    G = 0.02-abs(s[1].U)
    return G

```

圖八 四邊簡支承板結構 ABAQUS 輸出檔之 XML 描述檔

圖九 定義極限狀態方程之 Jython script

表五 四邊簡支承板結構之可靠度分析結果

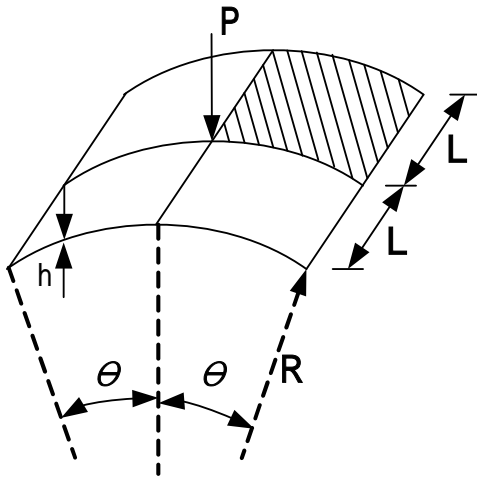
可靠度分析方法	Reliability Index
利用 Timoshenko (1959) 近似解之顯式極限狀態方程搭配 FOSM	3.0964
本文 (純二次次反應面搭配 FORM)	3.0759

本研究亦應用此軟體架構探討殼結構於考慮幾何非線性之極限載重下之可靠度分析的問題：如圖十為一 cylindrical shell，四邊簡支承， $L=254\text{mm}$ ， $R=2540\text{mm}$ ， $\theta=0.1\text{rad}$ ，彈性模數為 3.10275N/mm^2 ，浦松氏比為 0.3，殼厚 $h=6.35\text{mm}$ ，中心點受集中載重 P 。

表六為隨機變量之統計參數。我們定義此結構系統破壞之極限狀態函式：

$$\begin{aligned}
 G &= R - S \\
 &= P_u - P_w
 \end{aligned}$$

其中 P_u 為可靠度分析系統受垂直集中力之極限強度， $P_w = \frac{P_0}{S.F.}$ 可視為工作載重， P_0 設為平均值下分析所得之極限載重， $S.F.$ 為安全因子。

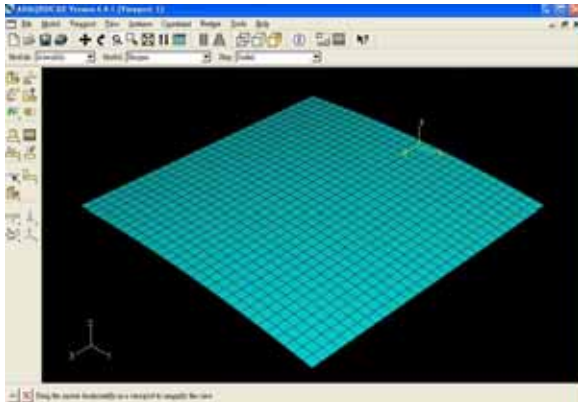


圖十 四邊筒支承之殼結構

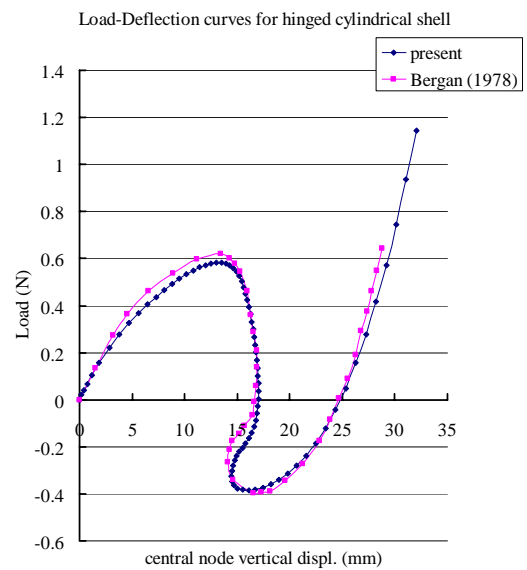
表六 四邊筒支承板結構之隨機變量性質

Random Variable	Mean	COV	Distribution Type
h(mm)	6.35	10 %	Normal
E(N/mm ²)	3.10275	15 %	Normal

使用 ABAQUS 之 shell element 編號 S8R 作有限元素模型（如圖十一）。為了計算考慮幾何非線性之極限強度，使用 ABAQUS 所提供之弧長法（RIKS Method），能夠完整記錄載重-位移具 snap-back 的行為；圖十二為 ABAQUS 分析結果與 Bergan et al. (1978) 的比較。



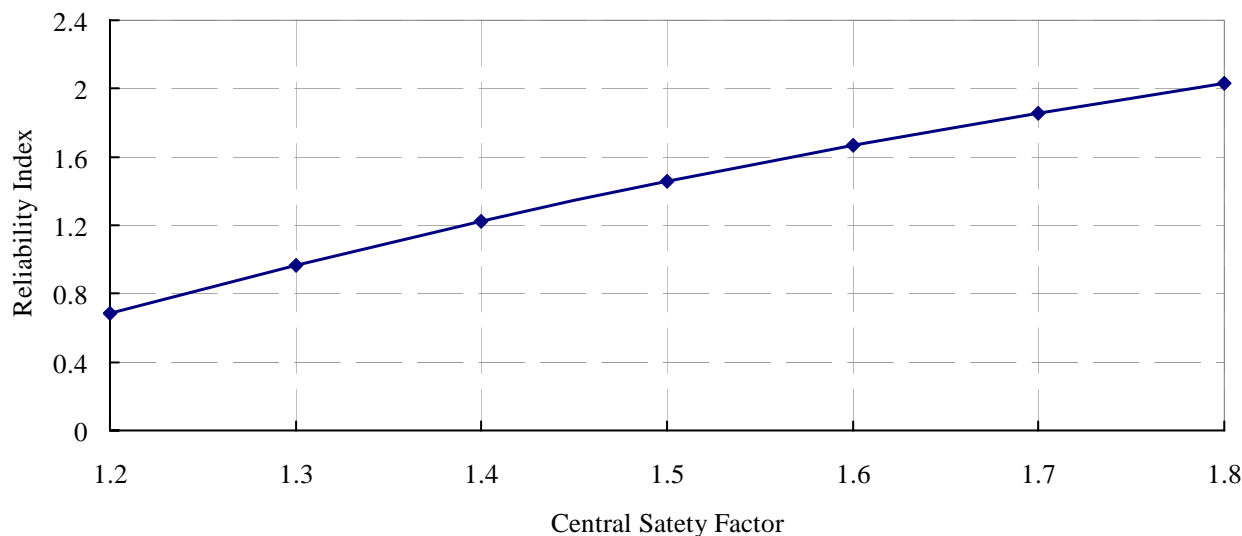
圖十一 四邊筒支承殼結構之 ABAQUS 模型



圖十二 四邊筒支承殼結構之載重-位移圖

依次改變極限狀態方程中之安全因子，可得可靠度指標與安全因子之關係（如圖十三）。

The Relationship between Safety Factor and Reliability Index



圖十三 可靠度指標與安全因子之關係

計畫成果自評

本計畫已按原計畫內容執行，所提出之重複性分析軟體架構以及建議使用之軟體技術，其整合不同商業軟體之效率與彈性已在本計畫所探討之應用問題中展現。依照本計畫之軟體架構所實作之可靠度分析程式，亦達到本計畫預計之目標，即能運用此軟體架構進行複雜結構行為（版、殼結構及非線性分析）下之可靠度分析。本研究之方法目前於文獻上並沒有，因此近期也將整理結果投稿到國際期刊。

參考文獻

- ABAQUS/CAE User's Manual, Hibbitt, Karlsson & Sorensen, Inc., Pawtucket, Version 6.4.
- ABAQUS/Standard User's Manual, Hibbitt, Karlsson & Sorensen, Inc., Pawtucket, Version 6.4.
- Albir S., *UML in a nutshell*, O'REILLY, 1999
- Bergan, P.G., Horrigmoe, G., Krakeland, B. and Soreide, T.H., "Solution techniques for non-linear finite element problems", *Int. J. Num. Meth. Engng.*, Vo. 12, pp 1677-1696, 1978.
- Faravelli, L., "Response surface approach for reliability analysis", *Journal of Engineering Mechanics*, ASCE, Vol. 115, No. 12, pp. 2763-2781, 1989.
- Gamma, E., Helm, R., Johnson, R., and Vlissides, J. M., *Design Patterns-Elements of Reusable Object-Oriented Software*, Addison Wesley Longman Inc., 1995, USA.
- Kim, S. H., and Na, S. W., "Response surface method using vector projected sampling points", *Structural Safety*, Vol. 19, No. 1, pp. 3-19, 1997.
- Larman, C., *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process*, 2nd Ed., Prentice-Hall, Inc., USA, 2001.
- Liu, Y.-W. and Moses, F., "A sequential response surface method and its application in reliability

- analysis of aircraft structural systems”, *Structural Safety*, Vol. 16, pp. 39-46, 1994.
- Mark, L. F. , *A Good Architecture for Object-Oriented Information Systems: Structures , Design, and Patterns*, OOPSLA '96-Tutorial 23, CA, USA, 1996.
- Nowak, A. S. and Collins, K. R., *Reliability of Structures*, McGraw Hill, 2000, USA.
- Pedroni, S., *Jython Essentials*, O'Reilly, USA, 2002
- Sweden Technology Centre, *Methods, Applications and Software for Structural Reliability Assessment*, Sweden Technology Centre, Corus UK Limited, UK, 2001.
- Schneider, S., *Introduction to Safety and Reliability of Structures*, IABSE, 1997, Switzerland.
- Timoshenko, S., *Theory of Plates and Shells*, McGraw-Hill, 1959, USA.
- 莊勝欽，2003，*工程分析後處理圖形程式語言之開發與應用*，國立台灣大學土木工程研究所碩士論文。
- 趙國藩，1996，*工程結構可靠度理論與應用*，大連理工大學出版社。