3

COMPUTATIONAL INDUSTRIAL MECHANICS PROGRAMME

Sophisticated mathematical modelling aided by powerful computing and visualization has the potential to provide the cutting-edge to industry; generation of cost-effective solutions, process optimization and product design are some of the areas where modelling and simulation can play critical to enabling role. The C-MMACS Computational Industrial Mechanics Programme (CIMP) seeks to develop and apply tools of mathematical modelling and computer simulation in diverse areas of engineering.

Inside

3.1 The effect of Inertia on the Dynamics and Rheology of a Periodically Forced Spherical Particle in a Quiescent Fluid

The problem of the dynamics and rheology of a dilute suspension of periodically forced particles in a quiescent fluid has been studied by the group. This problem has fundamental implications in areas such as the statistical mechanics of driven systems with fading memory and may also have implications for the development of vibration controllers and active sensors and actuators. The solution of this system yields phase plots of particle position and velocity. In addition time series for the position and velocity of the particles are also obtained. The software for the solution was tested by showing a reflection of the phase plot on changing the initial direction of the driving force as shown in Figure 3.1. The software was also tested by comparing the results with a perturbation solution and a good comparison with the numerical solution was obtained as shown in Figure 3.3. Analysis of these time series with the TISEAN software showed a negative Kolmogorov - Sinai entropy and hence absence of chaos in the parametric regime considered. Certain features in the phase plots reflected the presence of nonlinearity in the equations governing the problem. To eliminate possible reasons for the existence of these features (kinks), minor modifications in the program were made such as increasing the resolution of the calculations. This resulted in smooth phase plots with sharp edges; 'kinks', only at, ReF = 0.01. This parameter is a measure of the amplitude of the driving force. It was observed that the kinks occurred at around the zero velocity and the minimum and maximum displacements. The magnitude of the kinks increased with Reynolds numbers Re. This suggested a possible physical reason,

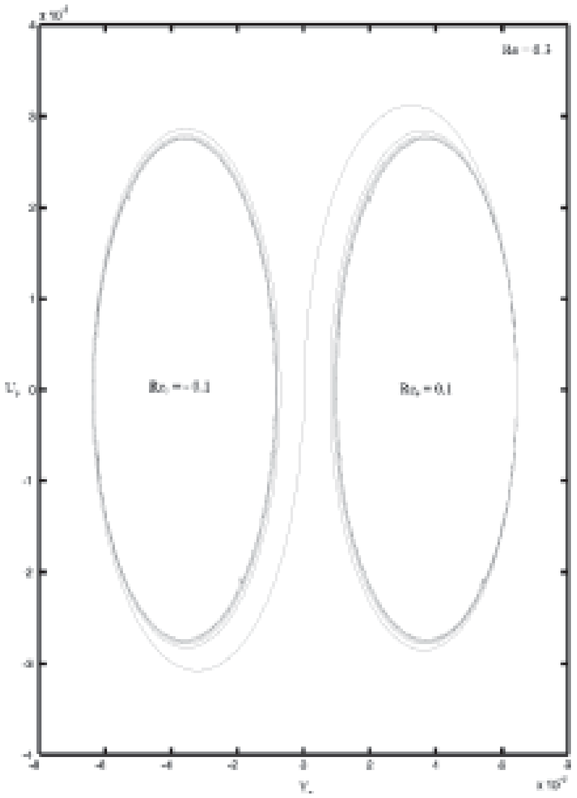


Figure 3.1 The phase plot at Re=0.3 and ReF = 0.1; showing the reflection property, i.e., if we reverse the direction of ReF, the plot also reverses as shown above.

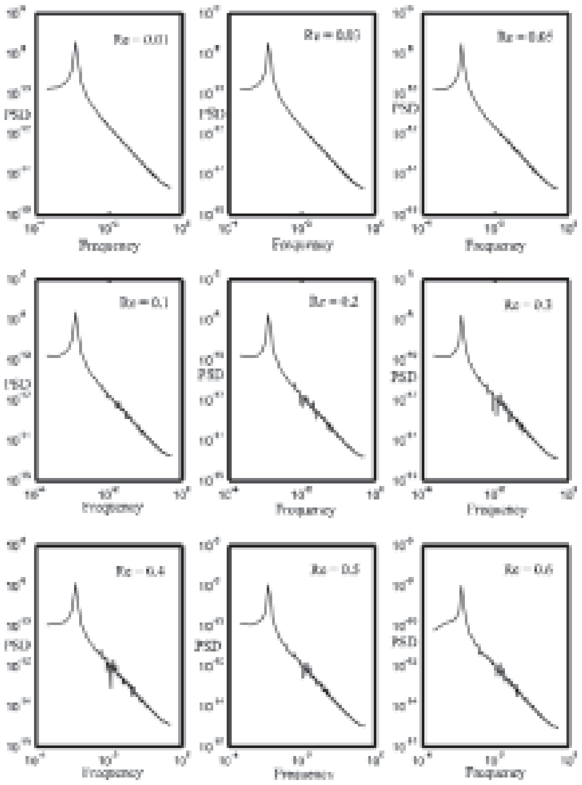


Figure 3.2 Power spectrum of the time series at ReF = 0.01, showing the occurrence of new frequencies in this region.

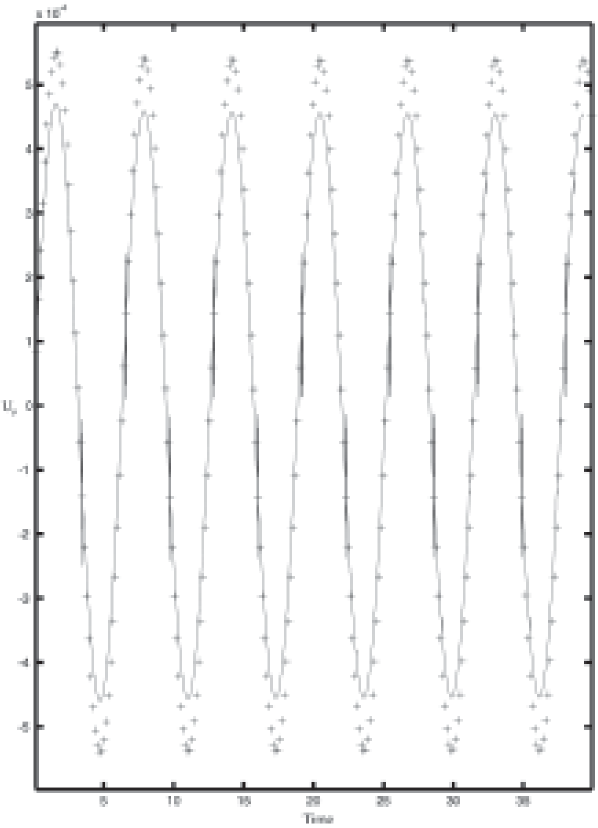


Figure 3.3 Comparison between the numerical solution (-) and the perturbation solution (+) of the particle velocity time series.

as ReF was very low and hence the effect of the nonlinear term was significant. The phase plots were computed and it was observed that at ReF = 0.01, new frequencies occurred in the power spectrum in addition to the fundamental frequency as shown in Figure 3.2. This was a clear effect of nonlinearity and an indication of the existence of a bifurcation at this point. In all other cases, it was observed that increasing the nonlinearity just increased the span of the phase plots. A computation of the various experimentally computable results such as velocity amplitude, mean displacement and amplitude ratios with the Stokes' flow case etc, and their dependence with Reynolds numbers Re and the amplitude of the periodic force ReF. was performed. The expression for a rheological parameter namely the mean normal stress difference was derived and its behavior with Re, ReF and the volume fractions was computed.

T R Ramamohan, I S Shivakumara and K Madhukar

3.2 Object-Oriented Design of FEM code "FACS"

Recently there has been growing interest in applying object-oriented approaches to large-scale programs with a view to treating various complexities within these. Some investigations have also been reported in the area of the Finite Element Method (FEM). It has been demonstrated that object-oriented programming can provide strong support to desirable features of FEM systems such as reusability, extensibility, easy maintenance, etc.

The object-oriented approach attempts to manage the complexity inherent in real-world problems by abstracting out knowledge and encapsulating it within objects. Objects are self-contained entities (physical or conceptual) composed of both functions and data. That is, an object retains certain information and knows how to perform certain operations. Objects which share the same behavior are said to belong to the same class. The various features of this approach consist of a class mechanism with inheritance, virtual function call mechanism and polymorphism. Another form of commonality can be expressed through templates. A class template is a special type of class definition that allows the programmer to generate an entire family of related classes, each of which is suited for working with a specific kind of data. The above is a very brief introduction to the concept of object-oriented programming. A detailed account of object-oriented programming can be found in many computer journals, language user guides and other literatures.

In the present investigation, object-oriented techniques have been applied in the development of a FEM software ``FACS'' for analyzing engineering problems. Although ``FACS'' was initially the acronym of Finite element Analysis of Composite Structures and was developed for finite element analysis of laminated composite structures, it presently consists of about 212 thousand lines of code and can solve a considerable range of general kinds of structure analysis, heat transfer and metal working problems and has led to several research publications in these fields. C++ is used in the development of the program which has several features to support object-oriented programming and can provide high computing efficiency because of its compatibility with C. The code, developed under both Windows and Linux environments, can be downloaded from the direct link (www.facssoft.com) for education purpose by the student community.

A brief discussion of the present object-oriented design is as follows. The present framework consists of several basic classes such as *ElemType, Material, Node, Element,* etc. which are traditional classes used for the representation of finite elements. Several specific classes are derived from these abstract classes. For example, a class *Elem3DStBrick8Lay* defining three dimensional eight noded structural layered brick element is virtually inherited from multiple base classes each having own set of base classes, as depicted in Figure 3.4. In most of the earlier investigations, these primitive objects directly interact with the problem domain. However, it can be revealed from the real world concept that in the FEM domain, some super objects can be identified which are either aggregates of the same objects or a superset of different objects. Therefore, we create an interface between the primitive objects and the problem domain by defining classes such as *ElemTypeGroup, MaterialGroup, NodeGroup, ElementGroup*, etc., which deal with groups of similar objects. For example, *ElementGroup* class deals with the lists of elements and performs several tasks including assembly of element stiffness matrices and load vectors, and solution of the system equations. This class is derived from a *LinkedList* class template and so inherits all its operations for the proper management of the list.

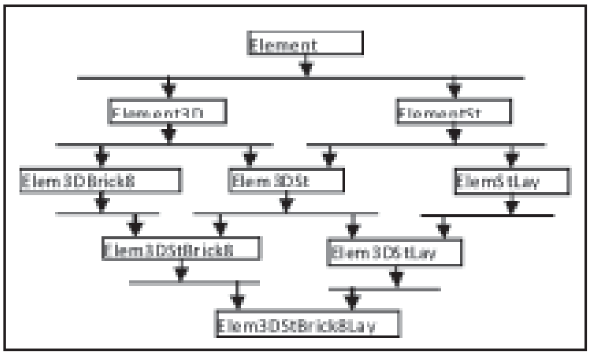


Figure 3.4 An example of inheritance of class Elem3DStBrick8Lay.

Several other classes, in addition to those discussed above, need to be defined in a complete finite element library. For example, solid model classes such as *Keypoint and KeypointGroup, Line and LineGroup, Area and AreaGroup and Volume and VolumeGroup,* are defined that perform modal generation and meshing. Classes *Load* and *Constraint* are also defined dealing with loading and constraints as applied to solid model objects and/or nodes and elements. As has become common practice now, some mathematical variables required in FEM domain such as vector, matrix, etc. have been represented in template form so that they can take variable type (integer, float, double, etc.) as an argument. Engineering variables such as strains and stresses have also been abstracted out. Several utility classes are also defined for the purpose of processing of finite element results and also for managing the finite element objects. The object oriented approach has facilitated the natural extension of the code in implementation of several other features. For example, the code consists of some classes which perform interpretation of FEM commands written in C.

The code also includes several inherent classes for text editing facility which can be used for preparing command file inside the program and also for preparing documentation related to the software.

The general conclusion is that use of object-oriented programming with C++ is attractive for the development of the present finite element application program.

Surendra Kumar

3.2 Mesh Distortion Sensitivity of Hybrid Stress Finite Elements

Finite element results are severely affected by the quality of elements. Finite element meshes are systematically updated in applications like shape optimization, crash analysis, metal forming, fluid flow analysis and large displacements. Remeshing becomes expensive and it can lead to inaccurate solutions that are projected onto a new mesh. So the accuracy of the element and the efficiency of the computations should not be compromised in case of severe distortion. This necessitates the development of elements which are insensitive to mesh distortion. The negative Jacobian is one of main causes of inaccurate elements and the classical finite elements are not capable of handling this situation.

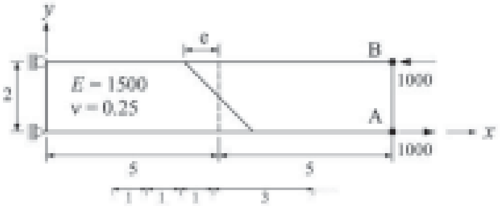
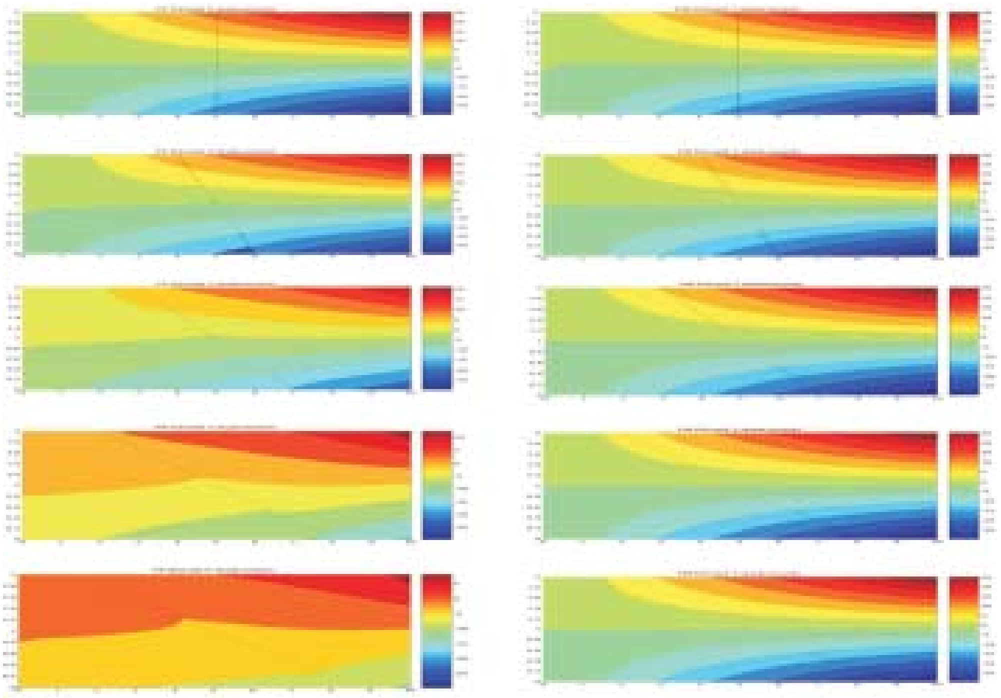


Figure 3.5 Two Element Distortion Test under Endmoment



(a) (b)

Figure 3.6 Horizontal Displacement contours for element distortion test (a) Horizontal displacement Contour of Classical FE for various *e* values (b) Horizontal displacement Contour of Hybrid Stress Element for various *e* values

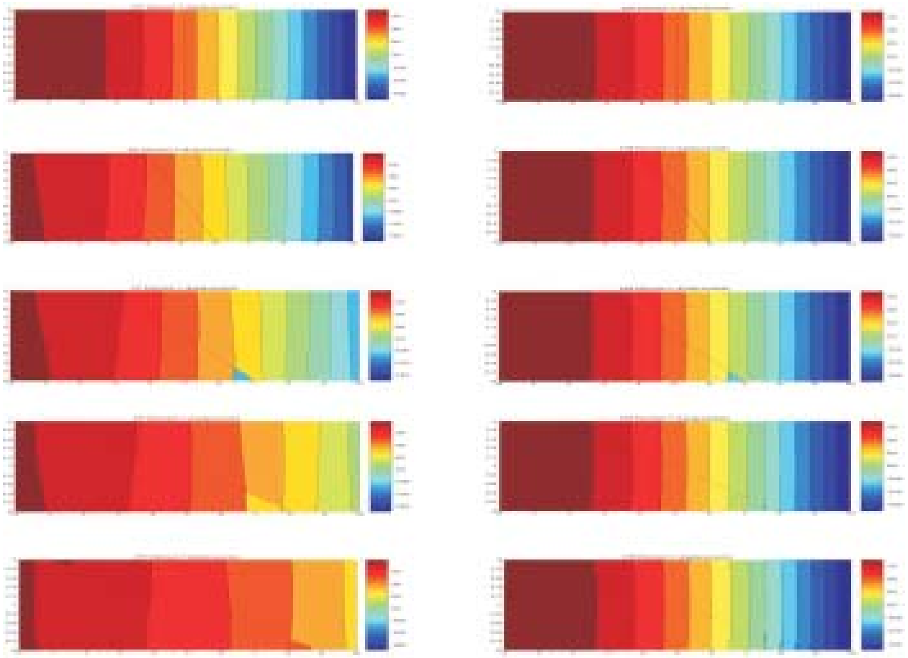


Figure 3.7 Vertical Displacement contours for element distortion test (a) Vertical displacement Contour of Classical FE for various *e* values (b) Vertical displacement Contour of Hybrid Stress Element for various *e* values.

Development of accurate plane elements under loads and geometries leading to mesh distortion is still not complete among researchers. Among the techniques proposed for improving the quadrilateral elements, the use of non-conforming displacement is the most successful in pure bending problems. To overcome mesh distortion problems, a quadratic element model is developed with non-conforming displacement modes and modified shape functions. This present refined 8-noded hybrid stress plane element consists of two additional non-conforming modes with classical membrane degrees of freedom to improve its behavior. Further the shape functions are modified with quadratic polynomials to account for distortion.

The performance of the present element is evaluated for various benchmark problems like eigen value test, patch test and element distortion sensitivity test. The two element cantilever beam problem is a well known example for testing the distortion sensitivity of the element. A distortion parameter *e* is introduced (Figure 3.5) for the mesh distortion effect. The value of *e* varies from 0,1,2,3 and 4. When the distortion parameter is zero, the classical finite element results and present hybrid stress elements produce same displacement values. When the distortion parameter is non-zero for mesh distortion, the present hybrid stress element produces excellent results compared to the classical finite elements. With the increase of the mesh distortion parameter value, the classical finite element solution becomes severly affected (Figure 3.6 and Figure 3.7). However the present element results are not affected by distortion. The reason for the hybrid stress elements success over the classical finite element (FE) is that the hybrid stress element tries to satisty the continuity and completeness (complete quadratic displacement function) conditions and it was derrived using assumed stress function and quadratic completeness.

V Senthilkumar and G Prathap