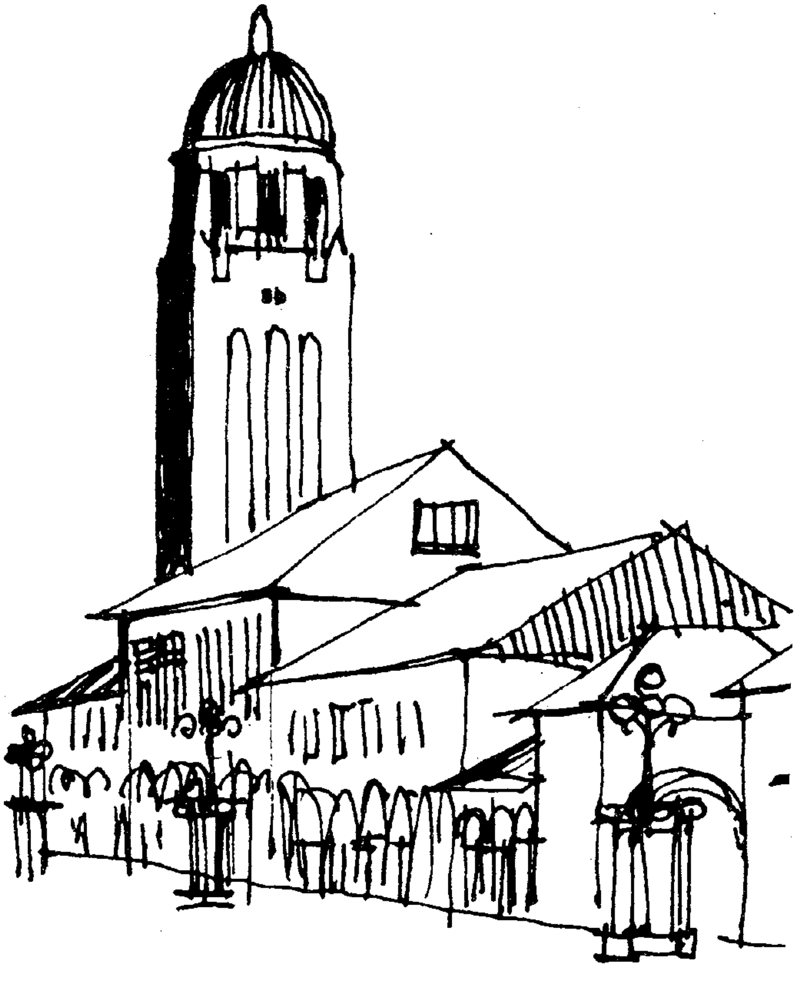
The John A. Blume Earthquake Engineering Center

Department of Civil Engineering
Stanford University

SOLUTION TECHNIQUES FOR LINEAR AND NONLINEAR DYNAMICS OF STRUCTURES MODELED BY FINITE ELEMENTS



by
Hojjat Adeli

Report No. 23
June 1976

50272-101

REPORT DOCUMENTATION PAGE

1. REPORT NO.
NSF/RA-761737

2.

3. Recipient's Accession No.
PB298265

4. Title and Subtitle
Solution Techniques for Linear and Nonlinear Dynamics of Structures Modeled by Finite Elements

5. Report Date
June 1976

6.

7. Author(s)
H. Adeli

8. Performing Organization Rept. No.
23

9. Performing Organization Name and Address
Stanford University
The John A. Blume Earthquake Engineering Center
Department of Civil Engineering
Stanford, California 94305

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.
(C)
(G)

12. Sponsoring Organization Name and Address
Engineering and Applied Science (EAS)
National Science Foundation
1800 G Street, N.W.
Washington, D.C. 20550

13. Type of Report & Period Covered
Technical

14.

15. Supplementary Notes

16. Abstract (Limit: 200 words)

Competitive solution techniques for linear and nonlinear dynamic analysis of structures by the finite element method are described. The accuracy, stability, and efficiency of the solution procedures are examined by comparing the results from a plane stress sample problem. An efficient operational procedure is developed for the isoparametric quadrilateral, the type of element used, in order to avoid matrix multiplications. The use of a lumped mass approach, resulting in a diagonal mass matrix, is shown to be more efficient than the non-diagonal mass formulation because the equations of motion are uncoupled in the acceleration terms. A comparison of four solution techniques indicates that direct linear extrapolation with trapezoidal rule is the best technique for linear dynamic analysis. For nonlinear analysis, both material and geometric nonlinearities are included in the finite element formation. An investigation involving three implicit and three explicit methods is evaluated and recommendations are made. Algorithms for these solution techniques are developed and are implemented in three computer programs. The appendices contain an operational procedure for calculating the stiffnesses and equivalent nodal loads and a lumped mass matrix for an isoparametric quadrilateral.

17. Document Analysis a. Descriptors

Dynamic structural analysis
Stiffness methods
Matrix methods

Computer programming
Linear systems
Nonlinear systems

Algorithms
Mathematical models
Finite element analysis
Analysis (mathematics)

b. Identifiers/Open-Ended Terms

NODIMP
NODEXP
RESPPSQ4

Isoparametric quadrilateral elements
Nodal loads

c. COSATI Field/Group

18. Availability Statement
NTIS

19. Security Class (This Report)

20. Security Class (This Page)

21. No. of Pages

22. Price
A10-A01

(See ANSI-Z39.18)

See Instructions on Reverse

SOLUTION TECHNIQUES FOR LINEAR AND NONLINEAR DYNAMICS OF STRUCTURES MODELED BY FINITE ELEMENTS

By

Hojjat Adeli

Technical Report No. 23

The John A. Blume Earthquake Engineering Center

Department of Civil Engineering

Stanford University

June 1976

Any opinions, findings, conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Intentionally Blank

PREFACE

Except for minor modifications and the omission of Appendices C, D, and E (which contain listings of computer programs), this report is the dissertation for my Ph.D. degree in Civil Engineering at Stanford University.

I would like to express my sincere appreciation to my adviser, Professor William Weaver, Jr., for his valuable guidance, encouragement, and support during the course of this research. I also wish to thank Professors James M. Gere and Haresh C. Shah for reviewing the dissertation.

Financial support for this project was provided by a scholarship from Tehran University and a grant from Lockheed Missiles and Space Company. Funds to use the computer were provided by the School of Engineering at Stanford.

Finally, my thanks to Ms. Suzanne Bennett and Ms. Iona Williams for typing the dissertation.

H. Adeli

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Page

CHAPTER 1

INTRODUCTION

1.1 Introduction

Dynamic analysis is rapidly becoming a common consideration in the design of structures, especially in determining response to earthquake ground motions. Methods for linear dynamic analysis of complex structures (where the material is assumed to be linearly elastic and displacements are small) were developed during the last two decades and are now well known. However, in many cases inelastic behavior of structures must be taken into account in order to obtain an economic and safe design. Nonlinear dynamic analysis of structures is a rather new field, and many researchers are actively investigating different aspects of the subject. Some of the important applications of nonlinear analysis are found in the design of missiles, aircraft, nuclear reactors, transportation vehicles, multi-story buildings located in seismic regions, etc.

Dynamic analysis of complex structures by the finite element method is performed in two major steps. The first step is to develop a finite number of equations of motion, and the second step is to solve these equations for the response at the nodes and the stresses within the elements. For large problems with several hundreds (or thousands) of degrees of freedom, the selection of an efficient algorithm for solving the equations of motion becomes a very important factor. This is especially true for nonlinear analysis, for which the cost of computations is an order of magnitude higher than that for linear analysis.

1.2 Objectives

The objectives of this project are to determine the most efficient techniques for linear and nonlinear dynamic analysis of structures modeled by finite elements. The selection of solution techniques to be compared is based on the information available in the literature (see Chapter 2). The accuracy, stability, and efficiency of the solution techniques are examined by comparing the results from a plane stress sample problem.

For linear analysis of the sample problem, results for the following solution techniques are compared: (1) direct linear extrapolation with the trapezoidal rule, (2) central difference predictor, (3) two-cycle iteration with the trapezoidal rule, and (4) normal mode method.

For nonlinear analysis, both material and geometric nonlinearities are included in the finite element formulation. Three implicit solution techniques are investigated in this work. They are (1) Newmark-Beta method, (2) Houbolt's method, and (3) Park's stiffly-stable method. In addition, the following explicit methods are compared: (1) central difference predictor, (2) two-cycle iteration with the trapezoidal rule, and (3) fourth-order Runge-Kutta method.

Algorithms for these solution techniques are developed and are implemented in three computer programs. The first program is for linear analysis, the second is for nonlinear analysis by implicit methods, and the third is for nonlinear analysis by explicit methods.

CHAPTER 2

SOLUTION TECHNIQUES FOR DYNAMIC ANALYSIS OF STRUCTURES

2.1 Introduction

For the solution of linear equations of motion, one can employ either the normal-mode method of dynamic analysis or some step-by-step numerical integration procedure. However, for nonlinear equations of motion, the use of numerical integration procedures appears to be mandatory. The normal-mode method is well known in the literature and is described briefly in Section 4.1.5. In this chapter, step-by-step integration procedures applied to dynamic analysis of structures are described. Information available in the literature is summarized, and this research consists of investigating and comparing the more promising methods.

In numerical integration procedures, time derivatives are usually approximated as difference equations involving one or more increments of time. If a given formula expresses the response in terms of previously-determined values of displacement, velocity, or acceleration, it is called a predictor (or an explicit formula); otherwise, it is called a corrector (or an implicit formula).

An estimate of the error (and its order) in the use of a difference formula can be obtained by comparing the formula with the expansion of Taylor's series [47,55]. For example, if the difference formula matches up to and including the $(\Delta t)^4$ term in the Taylor's series, it is called a fourth-order method; and its truncation error is the next term in the Taylor's series. Although this local truncation error gives an estimate of the accuracy of the method, the propagated error associated with numerical round-off has a random character and may be much more important. The repetitive use of an approximation formula causes error accumulation that may artifically magnify or attenuate the response. Also, when the order of the difference equation is higher than that of the differential equation, extraneous terms may be introduced into the approximate solution [8]. These spurious solutions, accompanied by round-off error over a sequence of time steps, may dominate the response and give unstable results. While the truncation error can always be estimated or bounded, numerical instability is significantly more difficult to analyze. Almost all of the existing techniques for examining the stability of numerical integration methods apply only to linear systems [2,40,48,55]. Moreover, stability in linear systems does not insure stability in nonlinear systems.

For linear systems, a numerical integration procedure is called unconditionally stable if the solution is bounded regardless of the size of the time step. It is called conditionally stable if the solution is bounded for any time step smaller than a so-called critical time step. Finally, if the solution approaches zero as the number of time steps approaches infinity, the numerical integration procedure is said to be asymptotically stable (or A-stable).

2.2 Central Difference Predictor

The second-order central difference formula [48] is as follows:

$$\ddot{d}_{i-1} = \frac{1}{(\Delta t)^2}\left(d_{i-2} - 2d_{i-1} + d_i\right) \qquad (i > 1) \qquad (2.1)$$

where d_i is the displacement at the end of the ith uniform time step of size Δt . This formula yields the predictor

$$d_i = -d_{i-2} + 2d_{i-1} + (\Delta t)^2 \ddot{d}_{i-1} \qquad (2.2)$$

Dahlquist [7] showed that no explicit multistep integration procedure (including the central difference predictor) can be asymptotically stable. Kreig [27] found that among the second-order predictors, the central difference formula has the largest stable time step of all, for which

$$(\Delta t)_{cr} = 2/\omega_{max} \qquad (2.3)$$

where ω_{max} is the highest angular frequency of the analytical model.

Key and Beisinger [26] used this method for the linear analysis of thin shells. In addition, Krieg and Key [28] showed that using a diagonal mass matrix improves the accuracy and efficiency of the central difference procedure. Successful use of this method in nonlinear analysis has also been reported by Key [25]. He used artificial viscosity to control the instability of the formula.

Weeks [56] compared the explicit central difference procedure with the implicit Houbolt and average-acceleration methods (to be discussed in the following sections) for linear as well as geometrically nonlinear problems. He found the central difference procedure generally superior. Witmer et al [58] searched for the optimal predictor-corrector method for systems of second-order differential equations. They also concluded that the central difference predictor is best. The main disadvantage of this method is that for a fine network of elements a very small time step is required to obtain stable results.

2.3 Newmark-Beta Method

Newmark [39] proposed the following implicit expressions for velocity and displacement for calculating the dynamic response of structures:

$$\dot{d}_i = \dot{d}_{i-1} + \gamma\Delta t\, \ddot{d}_i + (1-\gamma)\,\Delta t\, \ddot{d}_{i-1} \tag{2.4}$$

$$d_i = d_{i-1} + \beta(\Delta t)^2\, \ddot{d}_i + (0.5-\beta)(\Delta t)^2\, \ddot{d}_{i-1} + \Delta t\, \dot{d}_{i-1} \tag{2.5}$$

Because of its versatility, this approach is sometimes referred to as the generalized acceleration method. By setting γ equal to 0.5 and assigning different values to the parameter β, several well-known procedures can be obtained. They are the linear acceleration method (when $\beta = 1/6$), the average acceleration method or trapezoidal rule (when $\beta = 1/4$), and the constant acceleration method (when $\beta = 1/8$). The most widely used form of this method is the average acceleration method, which is unconditionally stable for linear systems [40]. Dahlquist [7] showed that the trapezoidal rule is the most accurate of the stable second-order formulas. Nickell [41] examined several step-by-step integration procedures and apparently found the trapezoidal rule to be the most attractive of all for both linear and nonlinear problems. Dunham et al [9] also reached the same conclusion.

In the Newmark-Beta method artificial damping can be provided by the δ control parameter, as recommended by Goudreau [15]. This parameter is related to γ by the equation

$$\gamma = 0.5 + \delta \qquad (\delta \geq 0) \tag{2.6}$$

It is shown in Reference 15 that for unconditional stability of the Newmark-Beta method we must choose the parameter β from the following relation:

$$\beta \geq 0.25 \; (1 + \delta)^2 \tag{2.7}$$

Nagarajan and Popov [37,38] have used this method because of the flexibility in selecting the amount of artificial damping through the use of the δ parameter.

2.4 Houbolt Method

Based on a third-order interpolation of displacements, Houbolt [19], presented the following multi-step implicit formulas for velocity and acceleration in terms of displacements:

$$\dot{d}_i = (-2d_{i-3} + 9d_{i-2} - 18\, d_{i-1} + 11d_i) \,/\, (6\Delta t) \tag{2.8}$$

$$\ddot{d}_i = (-d_{i-3} + 4d_{i-2} - 5d_{i-1} + 2d_i) \,/\, (\Delta t)^2 \tag{2.9}$$

Like the average-acceleration method, the Houbolt method is unconditionally stable for linear systems [22]. Stricklin et al [51] compared this method against the fourth-order Runge-Kutta method and the Newmark-Beta method with various values of parameters for geometrically nonlinear problems, using the pseudo-force technique. They found the Houbolt method to be the optimum one. However, as Weaver pointed out [55], these comparisons involved the solution of simultaneous equations for all methods studied; and if this time-consuming process is avoided (for explicit methods), we may reach a different conclusion. In Reference 51 it was also shown that the unconditional stability of the Houbolt procedure and the average-acceleration method does not exist in nonlinear problems. McNamara [33] has also found the Houbolt method to be the most suitable scheme for solving nonlinear problems. However, in contrast with References 33 and 51, Weeks [56] concluded that the average-acceleration procedure is more economical than the Houbolt method.

2.5 Iteration With the Trapezoidal Rule

The trapezoidal rule for displacements and velocities are

$$d_i = d_{i-1} + \Delta t(\dot{d}_{i-1} + \dot{d}_i)/2 \tag{2.10}$$

$$\dot{d}_i = \dot{d}_{i-1} + \Delta t(\ddot{d}_{i-1} + \ddot{d}_i)/2 \tag{2.11}$$

Boggs [5] has shown that iteration with the trapezoidal rule in a predictor-corrector form is an efficient procedure for solving nonlinear problems. Nickell [52] has also recommended use of the trapezoidal rule with limited iterations. Weaver [55] has carried out numerical experiments for limited iteration with the trapezoidal rule and reported satisfactory accuracy for this approach. He recommends that the number of iterations be limited to two.

2.6 Runge-Kutta Methods

Runge-Kutta methods are well-known in the classical literature on numerical analysis [17,29,47,49]. They are designed to approximate Taylor's series. The general fourth-order Runge-Kutta formulas for an equation in the form $\ddot{d} = f(t,d)$ reduces to [17]

$$d_i = d_{i-1} + \dot{d}_{i-1}\Delta t + \frac{\Delta t}{6}(\psi_1 + \psi_2 + \psi_3) + 0(\Delta t)^5 \qquad (2.12)$$

$$\dot{d}_i = \dot{d}_{i-1} + \frac{1}{6}(\psi_1 + 2\psi_2 + 2\psi_3 + \psi_4) + 0(\Delta t)^5 \qquad (2.13)$$

where

$$\psi_1 = \Delta t\ f\left(t_{i-1}\ ,\ d_{i-1}\right) \qquad (2.14)$$

$$\psi_2 = \Delta t\ f\left(t_{i-1} + \Delta t/2\ ,\ d_{i-1} + \frac{\Delta t}{2}\dot{d}_{i-1}\right) \qquad (2.15)$$

$$\psi_3 = \Delta t\ f\left(t_{i-1} + \Delta t/2\ ,\ d_{i-1} + \frac{\Delta t}{2}\dot{d}_{i-1} + \frac{\Delta t}{4}\psi_1\right) \qquad (2.16)$$

$$\psi_4 = \Delta t\ f\left(t_{i-1} + \Delta t\ ,\ d_{i-1} + \Delta t\ \dot{d}_{i-1} + \frac{\Delta t}{2}\psi_2\right) \qquad (2.17)$$

This explicit single-step method is self-starting and highly accurate (its truncation error is of the order of $(\Delta t)^5$). However, the four function evaluations makes this approach rather time consuming. Gupta [16] has used this method successfully in a nodewise manner in solving geometrically nonlinear problems. Weaver [55] also reported good results for this procedure, considering both speed and accuracy. In addition, he recommended that (for the purpose of making more extensive comparisons) the two-cycle iteration with the trapezoidal rule and the fourth-order Runge-Kutta method be implemented in a finite element program, using a diagonal mass matrix and a nodewise solution procedure.

2.7 Stiffly-Stable Methods

For high frequencies of vibration, the accuracy of the solution is not as important as the stability. In order to guarantee stability, one may be forced to use a very small time step that will unduly increase the cost of computations. For the purpose of producing A-stability of higher vibrational modes (or stiff components), Gear [12,13] presented a new class of time integrators, called stiffly-stable methods. Jensen [21] modified Gear's third-order formula and presented a new formula that has a larger stable time step. However, numerical experiments carried out by Weaver [55] indicate that the second-order Gear method and the third-order Gear method with Jensen's modifications do not appear to be competitive with the other methods studied. Recently, Park [44,45] has developed an improved stiffly-stable method by combining Gear's second-order and third-order formulas. He obtained the following multi-step implicit difference equation for velocity:

$$\dot{d}_i = (20d_i - 30d_{i-1} + 12d_{i-2} - 2d_{i-3}) / (12\Delta t) \qquad (2.18)$$

He proved that this formula is unconditionally stable for linear systems and applied it for both linear and nonlinear oscillators. He concluded that it is second best to the Newmark-Beta method for linear systems and is better than the Newmark-Beta as well as the Houbolt procedure for nonlinear problems. This method seems to be promising, and it warrants more investigation by comparing it with other good possibilities.

A multitude of other methods have been proposed by various investigators. However, based on the information available in the literature, they do not appear to be as good as those already mentioned. A literature survey of numerous solution techniques is presented in Reference 55.

From the study of the literature, some contradictory results are seen regarding the choice of the best solution technique for linear and (especially) nonlinear analysis of structures. These apparent contradictions are due to the interaction of many factors involved. For certain problems some specific procedure appears to be the optimum technique. For example, in wave propagation problems one must use a small time step in order to trace the response properly. Consequently, an explicit technique like the central difference procedure appears to be the most suitable one. More generally, however, the choice should be based upon accuracy, stability, storage requirements, and computational efficiency.

CHAPTER 3

FINITE-ELEMENT FORMULATION OF EQUATIONS OF MOTION

3.1 Introduction

Not very many practical problems, especially in a field such as earthquake engineering, have so-called exact or analytical solutions. Even for those which have analytical solutions, many simplifying assumptions must be introduced in order that they may be amenable to analytical solution procedures like similarity solutions and Fourier and Laplace transformations. Thus, applying some sort of numerical or approximate solutions to many practical problems is essential. Among the common approximate methods, perturbation methods, power series, method of weighted residuals, the finite difference technique and the finite element method can be mentioned. With the advent of high speed digital computers, the finite difference method and the finite element method appear to be the most successful of all. For problems with complicated configurations and boundary conditions, the finite element method has advantages over the method of finite differences. In this study, the method of finite elements has been used.

There are different ways of formulating the problem by finite elements. From a structural point of view, the finite element formulation can be divided into three categories:

1. Displacement formulation (stiffness or potential energy approach).
2. Stress formulation (flexibility or complementary potential energy approach).

3. Mixed formulation (mixed-energy principles or Reissner principle).

Up to this time, the majority of finite element theory has been based on the displacement formulation, and in this work a displacement approach has been used.

The type of element used in this study to model the plane stress and plane strain problem is the isoparametric quadrilateral element. The approach of isoparametric formulation has been used with great success in many different areas [6,62]. Recently, it has been shown that the isoparametric formulation is highly versatile and efficient for nonlinear structural problems [1,63].

In Sections 3.2 and 3.3, equations of motion will be formulated for an isoparametric quadrilateral element.

3.2 Linear Analysis

3.2.1 Finite Element Formulation

The principle of virtual work for an elastic structure in equilibrium and subjected to a system of virtual displacements can be stated by the following equation [46]:

$$\delta W = \delta U \tag{3.1}$$

where δW is the virtual work of external loads and δU is the virtual strain energy of internal stresses. It is noted that the virtual displacements must be a set that do not violate the constraints (geometric boundary conditions) of the structure.

When we apply this principle to a typical element, the virtual work of external loads is given by the expression

$$\delta W = \underset{\sim}{p}' \, \delta \underset{\sim}{q} + \int_A \underset{\sim}{w}_s' \, \delta \underset{\sim}{u} \, dA + \int_V \underset{\sim}{w}_v' \, \delta \underset{\sim}{u} \, dV - \int_V \rho \, \underset{\sim}{\ddot{u}}' \, \delta \underset{\sim}{u} \, dV \qquad (3.2)$$

where $\underset{\sim}{q}$, $\underset{\sim}{u}$, $\underset{\sim}{p}$, $\underset{\sim}{w}_s$ and $\underset{\sim}{w}_v$ are vectors defining the nodal displacements, generic displacements, concentrated nodal loads, distributed surface loads, and distributed body forces. In addition, A , V and ρ define area, volume and mass density of the element. The prime sign indicates the transpose of a vector.

Similarly, the virtual strain energy of the stresses is given by the expression

$$\delta U = \int_V \underset{\sim}{\sigma}' \, \delta \underset{\sim}{\epsilon} \, dV \qquad (3.3)$$

where $\underset{\sim}{\sigma}$ and $\delta \underset{\sim}{\epsilon}$ are vectors of stresses and virtual strains, respectively.

Generic displacements and nodal displacements are related to each other by the matrix of displacement shape functions $\underset{\sim}{T}_1$ as follows:

$$\underset{\sim}{u} = \underset{\sim}{T}_1 \, \underset{\sim}{q} \qquad (3.4)$$

Also, the strain-displacement relationships can be written as

$$\underset{\sim}{\epsilon} = \underset{\sim}{T}_2 \, \underset{\sim}{u} = \underset{\sim}{T}_2 \, \underset{\sim}{T}_1 \, \underset{\sim}{q} = \underset{\sim}{B} \, \underset{\sim}{q} \qquad (3.5)$$

where $\underset{\sim}{T}_2$ is a matrix containing derivatives with respect to the coordinates and $\underset{\sim}{B}$ is the strain-displacement matrix. In addition, the following relation holds among the stresses and strains

$$\underset{\sim}{\sigma} = \underset{\sim}{T}_3 \, (\underset{\sim}{\epsilon} - \underset{\sim}{\epsilon}_T - \underset{\sim}{\epsilon}_P) \qquad (3.6)$$

where $\underset{\sim}{\epsilon}_T$ and $\underset{\sim}{\epsilon}_P$ are vectors of temperature strains and prestrains, and $\underset{\sim}{T}_3$ is the stress-strain matrix (to be derived for plane problems in the following article).

Substituting Eqs. (3.2) and (3.3) into Eq. (3.1) and replacing $\underset{\sim}{u}$, $\underset{\sim}{\sigma}$, and $\underset{\sim}{\epsilon}$ from Eqs. (3.4) through (3.6) (and noting that the virtual displacement $\delta\underset{\sim}{q}$ is arbitrary and therefore can be cancelled from both sides of the equation), we obtain the following equations of motion:

$$\underset{\sim}{m}\,\ddot{\underset{\sim}{q}} + \underset{\sim}{K}\,\underset{\sim}{q} = \underset{\sim}{p} + \underset{\sim}{p}_S + \underset{\sim}{p}_V + \underset{\sim}{p}_T + \underset{\sim}{p}_P \tag{3.7}$$

in which

$$\underset{\sim}{m} = \int_V \rho\, \underset{\sim}{T}_1'\, \underset{\sim}{T}_1\, dV \tag{3.8}$$

$$\underset{\sim}{K} = \int_V \underset{\sim}{T}_1'\, \underset{\sim}{T}_2'\, \underset{\sim}{T}_3\, \underset{\sim}{T}_2\, \underset{\sim}{T}_1\, dV = \int_V \underset{\sim}{B}'\, \underset{\sim}{T}_3\, \underset{\sim}{B}\, dV \tag{3.9}$$

$$\underset{\sim}{p}_S = \int_A \underset{\sim}{T}_1'\, \underset{\sim}{w}_s\, dA \tag{3.10}$$

$$\underset{\sim}{p}_V = \int_V \underset{\sim}{T}_1'\, \underset{\sim}{w}_v\, dV \tag{3.11}$$

$$\underset{\sim}{p}_T = \int_V \underset{\sim}{T}_1'\, \underset{\sim}{T}_2'\, \underset{\sim}{T}_3\, \underset{\sim}{\epsilon}_T\, dV \tag{3.12}$$

$$\underset{\sim}{p}_P = \int_V \underset{\sim}{T}_1'\, \underset{\sim}{T}_2'\, \underset{\sim}{T}_3\, \underset{\sim}{\epsilon}_P\, dV \tag{3.13}$$

It is clear that $\underset{\sim}{p}_S$, $\underset{\sim}{p}_V$, $\underset{\sim}{p}_T$ and $\underset{\sim}{p}_P$ are equivalent nodal loads due to surface loads, volume loads, temperature strains, and prestrains.

3.2.2 Stress-Strain Relationships

Plane Stress

In the case of plane stress, by definition we have [52]

$$\sigma_{zz} = \sigma_{xz} = \sigma_{yz} = 0 \qquad (3.14)$$

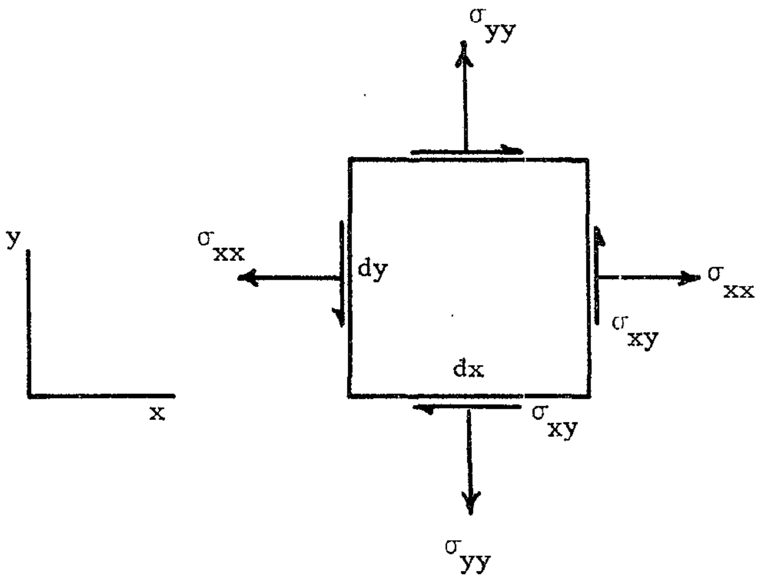


Fig. 3.1 Stresses at a Point in a Two Dimensional Problem

Employing Hooke's law, one determines strains in terms of stresses in an isotropic material as follows:

$$\epsilon_{xx} = \frac{1}{E} (\sigma_{xx} - \nu \sigma_{yy}) \qquad (3.15)$$

$$\epsilon_{yy} = \frac{1}{E} (\sigma_{yy} - \nu \sigma_{xx}) \qquad (3.16)$$

$$\gamma_{xy} = \frac{2(1 + \nu)}{E} \sigma_{xy} \qquad (3.17)$$

where E and ν are Young's modulus and Poisson's ratio.

Solving the above equations for stresses, one obtains the stress-strain matrix $\underset{\sim}{T}_3$. Thus,

$$\underset{\sim}{\sigma} = \underset{\sim}{T}_3 \underset{\sim}{\epsilon} \tag{3.18}$$

where

$$\underset{\sim}{T}_3 = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{3.19}$$

and $\underset{\sim}{\sigma} = \{\sigma_{xx} , \sigma_{yy} , \sigma_{xy}\}$; $\underset{\sim}{\epsilon} = \{\epsilon_{xx} , \epsilon_{yy} , \gamma_{xy}\}$

Plane Strain

In the case of plane strain, we assume that:

$$\epsilon_{zz} = \gamma_{xz} = \gamma_{yz} = 0 \tag{3.20}$$

For an isotropic material the strains in terms of the stresses are as follows:

$$\epsilon_{xx} = \frac{1}{E} (\sigma_{xx} - \nu \sigma_{yy} - \nu \sigma_{zz}) \tag{3.21}$$

$$\epsilon_{yy} = \frac{1}{E} (\sigma_{yy} - \nu \sigma_{xx} - \nu \sigma_{zz}) \tag{3.22}$$

$$\epsilon_{zz} = \frac{1}{E} (\sigma_{zz} - \nu \sigma_{xx} - \nu \sigma_{yy}) = 0 \tag{3.23}$$

$$\gamma_{xy} = \frac{2(1 + \nu)}{E} \sigma_{xy} \tag{3.24}$$

Solving the above equations for stresses, we obtain matrix $\underset{\sim}{T}_3$

[defined by Eq. (3.18)] as

$$\underset{\sim}{T}_3 = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \qquad (3.25)$$

Also, from Eq. (3.23), the stress in the z-direction will be found as

$$\sigma_{zz} = \nu\,(\sigma_{xx} + \sigma_{yy}) \qquad (3.26)$$

3.2.3 Evaluation of Element Matrices

In this section, we formulate and derive the element matrices for a 4-node isoparametric quadrilateral element. This element is displayed in Fig. 3.2.

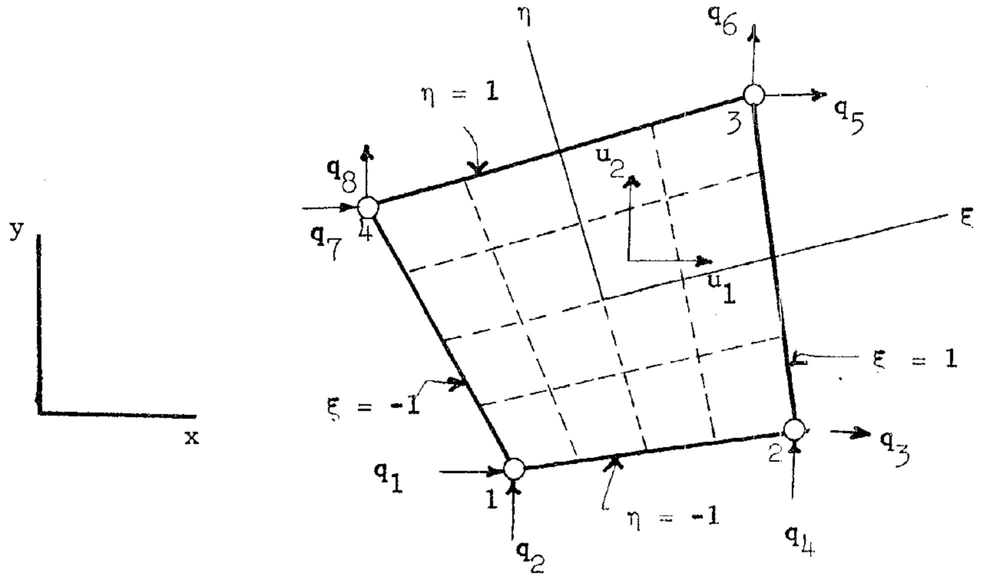


Fig. 3.2 An Isoparametric Quadrilateral Element

For this element, the same interpolation formulas define both the geometric and displacement shape functions, as given by the following expressions:

$$x = \frac{1}{4}\,[(1-\xi)(1-\eta)x_1 + (1+\xi)(1-\eta)x_2 + (1+\xi)(1+\eta)x_3 + (1-\xi)(1+\eta)x_4]$$

$$y = \frac{1}{4}\,[(1-\xi)(1-\eta)y_1 + (1+\xi)(1-\eta)y_2 + (1+\xi)(1+\eta)y_3 + (1-\xi)(1+\eta)y_4]$$

$$(3.27)$$

and

$$u_1 = \frac{1}{4}\,[(1-\xi)(1-\eta)q_1 + (1+\xi)(1-\eta)q_3 + (1+\xi)(1+\eta)q_5 + (1-\xi)(1+\eta)q_7]$$

$$u_2 = \frac{1}{4}\,[(1-\xi)(1-\eta)q_2 + (1+\xi)(1-\eta)q_4 + (1+\xi)(1+\eta)q_6 + (1-\xi)(1+\eta)q_8]$$

$$(3.28)$$

or in matrix form

$$\begin{bmatrix} u_1 \\ \\ u_2 \end{bmatrix} = \begin{bmatrix} f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 & 0 \\ \\ 0 & f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \cdot\cdot \\ q_8 \end{bmatrix} \qquad (3.29)$$

where

$$\begin{aligned} f_1(\xi,\eta) &= (1-\xi)(1-\eta)/4 \\ f_2(\xi,\eta) &= (1+\xi)(1-\eta)/4 \\ f_3(\xi,\eta) &= (1+\xi)(1+\eta)/4 \\ f_4(\xi,\eta) &= (1-\xi)(1+\eta)/4 \end{aligned} \qquad (3.30)$$

Concisely,

$$\underset{\sim}{u}(\xi,\eta) = \underset{\sim}{T}_1(\xi,\eta)\ \underset{\sim}{q} \tag{3.31}$$

Generic displacements are in terms of local natural coordinates, whereas differentiations with respect to the global coordinates x and y are required in the strain-displacement relationships. Thus, we must use the chain rule for differentiation. In matrix form we can write

$$\begin{bmatrix} \frac{\partial f}{\partial \xi} \\ \\ \frac{\partial f}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix} = \underset{\sim}{J} \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{3.32}$$

and

$$\begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial \xi} \\ \\ \frac{\partial f}{\partial \eta} \end{bmatrix} = \underset{\sim}{J}^{-1} \begin{bmatrix} \frac{\partial f}{\partial \xi} \\ \\ \frac{\partial f}{\partial \eta} \end{bmatrix} \tag{3.33}$$

The array $\underset{\sim}{J}$ is called the Jacobian matrix. Terms in this matrix can be evaluated by differentiating Eqs. (3.27), as follows:

$$J_{11} = \frac{\partial x}{\partial \xi} = \sum_{i=1}^{4} \frac{\partial f_i}{\partial \xi} x_i$$

$$J_{12} = \frac{\partial y}{\partial \xi} = \sum_{i=1}^{4} \frac{\partial f_i}{\partial \xi} y_i$$

$$J_{21} = \frac{\partial x}{\partial \eta} = \sum_{i=1}^{4} \frac{\partial f_i}{\partial \eta} x_i \qquad \text{(Cont'd)}$$

$$J_{22} = \frac{\partial y}{\partial \eta} = \sum_{i=1}^{4} \frac{\partial f_i}{\partial \eta} y_i \qquad (3.34)$$

In matrix form,

$$\underset{\sim}{J} = \underset{\sim}{D}_L \underset{\sim}{C} \qquad (3.35)$$

where

$$\underset{\sim}{D}_L = \begin{bmatrix} \frac{\partial f_1}{\partial \xi} & \frac{\partial f_2}{\partial \xi} & \frac{\partial f_3}{\partial \xi} & \frac{\partial f_4}{\partial \xi} \\ \frac{\partial f_1}{\partial \eta} & \frac{\partial f_2}{\partial \eta} & \frac{\partial f_3}{\partial \eta} & \frac{\partial f_4}{\partial \eta} \end{bmatrix}$$

$$= \frac{1}{4} \begin{bmatrix} -(1-\eta) & (1-\eta) & (1+\eta) & -(1+\eta) \\ -(1-\xi) & -(1+\xi) & (1+\xi) & (1-\xi) \end{bmatrix} \qquad (3.36)$$

and

$$\underset{\sim}{C} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} \qquad (3.37)$$

The inverse of the Jacobian matrix is:

$$\underset{\sim}{J}^{-1} = \frac{\underset{\sim}{J}^a}{|\underset{\sim}{J}|} = \frac{1}{|\underset{\sim}{J}|} \begin{bmatrix} J_{22} & -J_{12} \\ -J_{21} & J_{11} \end{bmatrix} = \frac{1}{|\underset{\sim}{J}|} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} \end{bmatrix} \qquad (3.38)$$

where

$$|\underset{\sim}{J}| = J_{11} J_{22} - J_{21} J_{12} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}$$

To determine all of the derivatives required for the strain-displacement matrix, we apply Eq. (3.33) repeatedly, as follows:

$$\begin{bmatrix} \dfrac{\partial f_i}{\partial x} \\ \\ \dfrac{\partial f_i}{\partial y} \end{bmatrix} = \underset{\sim}{J}^{-1} \begin{bmatrix} \dfrac{\partial f_i}{\partial \xi} \\ \\ \dfrac{\partial f_i}{\partial \eta} \end{bmatrix} \qquad i = 1,2,3,4 \qquad (3.39)$$

Altogether,

$$\underset{\sim}{D}_G = \underset{\sim}{J}^{-1} \underset{\sim}{D}_L \qquad (3.40)$$

where

$$\underset{\sim}{D}_G = \begin{bmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_2}{\partial x} & \dfrac{\partial f_3}{\partial x} & \dfrac{\partial f_4}{\partial x} \\ \\ \dfrac{\partial f_1}{\partial y} & \dfrac{\partial f_2}{\partial y} & \dfrac{\partial f_3}{\partial y} & \dfrac{\partial f_4}{\partial y} \end{bmatrix} \qquad (3.41)$$

Now, the strain-displacement matrix $\underset{\sim}{B}(\xi,\eta) = \underset{\sim}{T}_2 \underset{\sim}{T}_1(\xi,\eta)$ can be derived in terms of elements of matrix $\underset{\sim}{D}_G$ as follows:

$$\underset{\sim}{B} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 & 0 \\ 0 & f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial f_1}{\partial x} & 0 & \frac{\partial f_2}{\partial x} & 0 & \frac{\partial f_3}{\partial x} & 0 & \frac{\partial f_4}{\partial x} & 0 \\ 0 & \frac{\partial f_1}{\partial y} & 0 & \frac{\partial f_2}{\partial y} & 0 & \frac{\partial f_3}{\partial y} & 0 & \frac{\partial f_4}{\partial y} \\ \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial x} & \frac{\partial f_4}{\partial y} & \frac{\partial f_4}{\partial x} \end{bmatrix}$$

$$= \begin{bmatrix} D_{G11} & 0 & D_{G12} & 0 & D_{G13} & 0 & D_{G14} & 0 \\ 0 & D_{G21} & 0 & D_{G22} & 0 & D_{G23} & 0 & D_{G24} \\ D_{G21} & D_{G11} & D_{G22} & D_{G12} & D_{G23} & D_{G13} & D_{G24} & D_{G14} \end{bmatrix} \qquad (3.42)$$

Having derived the matrices $\underset{\sim}{T}_1$, $\underset{\sim}{T}_3$ and $\underset{\sim}{B}$, we can also evaluate the element stiffness matrix, mass matrix, and equivalent nodal loads by numerical integrations, as follows:

$$\underset{\sim}{m} = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{T}_1'(\xi,\eta)\ \underset{\sim}{T}_1(\xi,\eta)\ |\underset{\sim}{J}|\ d\xi d\eta \qquad (3.43)$$

$$\underset{\sim}{K} = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{B}'(\xi,\eta)\ \underset{\sim}{T}_3\ \underset{\sim}{B}(\xi,\eta)\ |\underset{\sim}{J}|\ d\xi d\eta \qquad (3.44)$$

$$\underset{\sim}{p}_S = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{T}_1'(\xi,\eta)\ \underset{\sim}{w}_s\ |\underset{\sim}{J}|\ d\xi d\eta \qquad (3.45)$$

$$\underset{\sim}{p}_V = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{T}_1'(\xi,\eta)\ \underset{\sim}{w}_v\ |\underset{\sim}{J}|\ d\xi d\eta \qquad (3.46)$$

$$\underset{\sim}{p}_T = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{B}'(\xi,\eta)\ \underset{\sim}{T}_3\ \underset{\sim}{\epsilon}_T\ |\underset{\sim}{J}|\ d\xi d\eta \qquad (3.47)$$

$$\underset{\sim}{p}_P = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{B}'(\xi,\eta)\ \underset{\sim}{T}_3\ \underset{\sim}{\epsilon}_p\ |\underset{\sim}{J}|\ d\xi d\eta \qquad (3.48)$$

In these equations, h is equal to the thickness of the element for plane stress problems; and it is taken equal to unity for plane strain problems.

Detailed formulations of the element stiffnesses and equivalent nodal loads are given in Appendix A. It should be pointed out that there are two ways of calculating the element mass matrix. The approach based on Eq. (3.43) results in a non-diagonal (consistent) mass matrix. The second way of calculating the mass matrix is based upon physical intuition. In this case the mass of the element is assumed to be lumped at the nodes, resulting in a diagonal mass matrix. This approach is more efficient because of the fact that the equations of motion will be uncoupled in the accleration terms. In this study a lumped mass approach has been used, and this type of mass matrix for a quadrilateral element is derived in Appendix B.

3.3 Nonlinear Analysis

In this section, equations of motion will be derived for the very general large dispacement and large strain type of analysis. For complete generality, use of a continuum approach and tensor notation is essential for finite strain formulations. Thus, a continuum approach with tensor notation along with matrix formulation has been employed in this presentation. Two different approaches have been in common use for formulating nonlinear problems. They are the Lagrangian and the Eulerian formulations [34]. In the Lagrangian formulation, all the variables are referred to the undeformed configuration. However, in the Eulerian formulation, all the variables are referred to the deformed configuration. When coordinates are introduced into the equations, the initial coordinates for the undeformed configuration are used in the Lagrangian formulation, while updated coordinates for the deformed configuration are used in the Eulerian formulation. Some applications of these formulations for nonlinear static and dynamic problems are given in References 1, 36, and 63. But, Zienkiewicz and Nayak [63] and Powell and Mondkar [36] have preferred the Lagrangian formulation over the Eulerian formulation. They have pointed out the following advantages for the Lagrangian approach:

1. The effects of large displacements are implicit in the formulation, and no transformation of element matrices is needed in order to take into account updating of the nodal coordinates due to the change of geometry.

2. In the Lagrangian formulation, the increments of stress and strain can be related simply by the equation

$$\underset{\sim}{d\sigma} = \underset{\sim}{T}_3 \, \underset{\sim}{d\epsilon} \tag{3.49}$$

so that stresses are evaluated by simple additions. However, in the Eulerian formulation, where stresses are referred to in the deformed configuration, this cannot be accomplished because the changes may occur due to pure rigid body rotation. In this case, introduction of another measure of stress called Jaumann stress [63] is necessary, and stresses are obtained by transformation and addition.

3. For anisotropic material, the Lagrangian formulation is more advantageous. This is due to the fact that anisotropy is referred to the axes in the undeformed configuration.

In this section, a Lagrangian approach for nonlinear analysis will be presented. In the Lagrangian formulation, Green-Lagrangian strain tensor and Piola-Kirchhoff stress tensors are used as measures of strain and stress. These tensors will first be defined briefly; then incremental equations of motion will be derived from the principle of virtual displacements.

3.3.1 Kinematic Definitions

Before defining the Green-Lagrangian strain tensor and the Piola-Kirchhoff stress tensors, we need to define three other tensors. These are the deformation gradient, the spatial deformation gradient, and the Green deformation tensor [30,34].

Deformation Gradient

Suppose that a body has a particular configuration at the reference time t_o and another configuration at time t ; and consider two neighboring points P_1 and P_2 , as shown in Fig. 3.3. Let the position of the point P_1 before and after deformation be located by the vectors $\underset{\sim}{x} = \{x_1 , x_2 , x_3\}$ and $\underset{\sim}{X} = \{X_1 , X_2 , X_3\}$, respectively. The deformation gradient is denoted by $\underset{\sim}{g}$ and defined as follows:

$$d\underset{\sim}{X} = \underset{\sim}{g}\, d\underset{\sim}{x} \qquad (3.50)$$

where

$$d\underset{\sim}{X} = \{dX_1 , dX_2 , dX_3\}$$

$$dx = \{dx_1 , dx_2 , dx_3\}$$

and

$$\underset{\sim}{g} = \begin{bmatrix} \frac{\partial X_1}{\partial x_1} & \frac{\partial X_1}{\partial x_2} & \frac{\partial X_1}{\partial x_3} \\ \frac{\partial X_2}{\partial x_1} & \frac{\partial X_2}{\partial x_2} & \frac{\partial X_2}{\partial x_3} \\ \frac{\partial X_3}{\partial x_1} & \frac{\partial X_3}{\partial x_2} & \frac{\partial X_3}{\partial x_3} \end{bmatrix} \qquad (3.51)$$

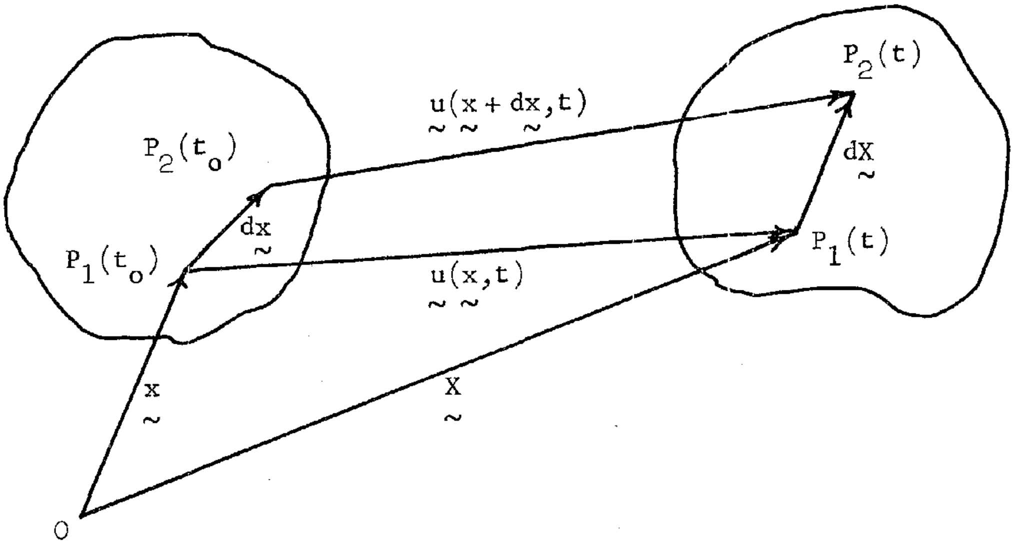
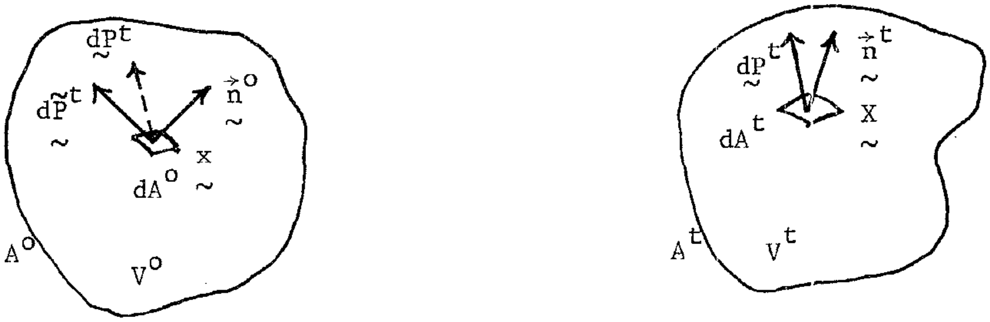


Fig. 3.3 Body Before and After Deformations



(a) Body at Time t = 0

(b) Body at Time t

Fig. 3.4 Force Vectors for Definitions of Piola-Kirchhoff Stress Tensors

Equation (3.50) can be written in indicial-summation form as:

$$dX_i = \sum_{j=1}^{3} \frac{\partial X_i}{\partial x_j} dx_j \qquad (i = 1,2,3) \tag{3.52}$$

Hereafter, the summation part will be dropped and just the indicial (short form) notation will be used. Thus, the deformation gradient can be written in indicial form as

$$\mathscr{g}_{ij} = \frac{\partial X_i}{\partial x_j} \tag{3.53}$$

It is noted that the deformation gradient $\underset{\sim}{\mathscr{g}}$ refers to the undeformed configuration.

Spatial Deformation Gradient

The spatial deformation gradient tensor is similar to the deformation gradient tensor, but with reference to the deformed configuration. We denote it by $\underset{\sim}{\mathscr{g}}^{-1}$ (the reason for this notation will be clear later). It is defined as follows:

$$\underset{\sim}{dx} = \underset{\sim}{\mathscr{g}}^{-1} \underset{\sim}{dX} \tag{3.54}$$

where

$$\underset{\sim}{\mathscr{g}}^{-1} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{bmatrix} \tag{3.55}$$

Or in indicial form

$$dx_i = \frac{\partial x_i}{\partial X_j} dX_j \tag{3.56}$$

and

$$\mathscr{g}^{-1}_{ij} = \frac{\partial x_i}{\partial X_j}$$

Now it can easily be shown that $\underset{\sim}{\mathscr{g}}\,\underset{\sim}{\mathscr{g}}^{-1} = \underset{\sim}{I}$ because a typical inner product of a row of $\underset{\sim}{\mathscr{g}}$ and a column of $\underset{\sim}{\mathscr{g}}^{-1}$ yields

$$\frac{\partial X_i}{\partial x_1}\frac{\partial x_1}{\partial X_j} + \frac{\partial X_i}{\partial x_2}\frac{\partial x_2}{\partial X_j} + \frac{\partial X_i}{\partial x_3}\frac{\partial x_3}{\partial X_j} = \delta_{ij}$$

where δ_{ij} is the Kronecker delta (1 if $i = j$; 0 if $i \neq j$).

Green Deformation Tensor

The Green deformation tensor is denoted by $\underset{\sim}{E}$ and refers to the undeformed configuration. It gives the current squared length of the vector at time t . In matrix notation we have

$$\left(ds^t\right)^2 = \underset{\sim}{dx}' \underset{\sim}{E} \underset{\sim}{dx} \tag{3.58}$$

In indicial notation this becomes

$$\left(ds^t\right)^2 = dx_i E_{ij} dx_j \tag{3.59}$$

The Green deformation tensor $\underset{\sim}{E}$ can be related to the deformation gradient $\underset{\sim}{g}$ as follows:

$$\left(ds^{t}\right)^{2} = \underset{\sim}{dX}' \, \underset{\sim}{dX} = (\underset{\sim}{g} \, \underset{\sim}{dx})' \, (\underset{\sim}{g} \, \underset{\sim}{dx}) = \underset{\sim}{dx}' \, \underset{\sim}{g}' \, \underset{\sim}{g} \, \underset{\sim}{dx}$$

Comparing this expression with Eq. (3.58), we see that

$$\underset{\sim}{E} = \underset{\sim}{g}' \, \underset{\sim}{g} \tag{3.60}$$

Or in indicial form

$$E_{ij} = \frac{\partial X_k}{\partial x_i} \frac{\partial X_k}{\partial x_j} \tag{3.61}$$

3.3.2 Green-Lagrangian Strain Tensor

By definition, the Green-Lagrangian strain tensor gives the change in the squared length of the original vector $\underset{\sim}{dx}$ as follows [34]:

In matrix notation:

$$\left(ds^{t}\right)^{2} - \left(ds^{o}\right.^{,2} = 2 \, \underset{\sim}{dx}' \, \underset{\sim}{\epsilon} \, \underset{\sim}{dx} \tag{3.62}$$

In indicial notation:

$$\left(ds^{t}\right)^{2} - \left(ds^{o}\right)^{2} = 2 \, dx_i \, \epsilon_{ij} \, dx_j \tag{3.63}$$

where ds^{o} is the length of the vector at time t_{o} .

From the relations (3.59) and (3.63) and noting that

$$\left(ds^{o}\right)^{2} = dx_i \, dx_i = dx_i \, \delta_{ij} \, dx_j \quad ,$$

we find the following relationship between the Green-Lagrangian strain tensor $\underset{\sim}{\epsilon}$ and the Green deformation tensor $\underset{\sim}{E}$:

$$2 \, \epsilon_{ij} = E_{ij} - \delta_{ij} \tag{3.64}$$

or in matrix notation

$$2 \underset{\sim}{\epsilon} = \underset{\sim}{E} - \underset{\sim}{I} \qquad (3.65)$$

After substituting E_{ij} from Eq. (3.61) into Eq. (3.64), we obtain

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial X_k}{\partial x_i}\frac{\partial X_k}{\partial x_j} - \delta_{ij}\right) \qquad (3.66)$$

We can express the Green-Lagrangian strain tensor in terms of displacements by writing out ϵ_{ij} and using the following relation:

$$\underset{\sim}{X} = \underset{\sim}{x} + \underset{\sim}{u} \qquad (3.67)$$

or

$$X_i = x_i + u_i \qquad (3.68)$$

where $\underset{\sim}{u}$ is the vector of displacements. The result is as follows:

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}\right) \qquad (3.69)$$

3.3.3 Piola-Kirchhoff Stress Tensors

In the Lagrangian formulation, where strains are referred to the original position $(\underset{\sim}{x})$ rather than the current position $(\underset{\sim}{X})$, we also need to define the stresses with respect to the original configuration. We now make two different definitions of stresses that have been used in the Lagrangian formulation. These are the so-called first and second Piola-Kirchhoff stress tensors [10,34]. They can be defined in terms of the force vectors illustrated in Fig. 3.4. The force vector $d\underset{\sim}{P}^t$ acts on a point of the surface at time t with coordinates $\underset{\sim}{X}$

whose unit outward normal is $\underset{\sim}{\vec{n}}^{t}$. Furthermore, let the corresponding unit outward normal at the reference time be $\underset{\sim}{\vec{n}}^{o}$. The first Piola-Kirchhoff stress tensor (sometimes called the Lagrangian stress tensor) gives the actual force $d\underset{\sim}{P}^{t}$ at time t on the deformed area dA^{t} . However, it is computed on the basis of the undeformed area dA^{o} as follows:

$$d\underset{\sim}{P}^{t} = \left(\underset{\sim}{\vec{n}}^{o} \cdot \underset{\sim}{T}\right) dA^{o} = \left(\underset{\sim}{\vec{n}}^{t} \cdot \underset{\sim}{\tau}\right) dA^{t} \qquad (3.70)$$

where $\underset{\sim}{\tau}$ and $\underset{\sim}{T}$ are Cauchy and first Piola-Kirchhoff stresses, respectively. It should be pointed out that the Cauchy stress $\underset{\sim}{\tau}$ (referring to the deformed configuration) is the true stress. On the other hand, the Piola-Kirchhoff stress tensors are only pseudo-stress tensors.

For the second Piola-Kirchhoff stress tensor, we first define a pseudo-force $d\underset{\sim}{\tilde{P}}^{t}$ that is related to the actual force $d\underset{\sim}{P}^{t}$ by the same transformation rule as that by which the vector $d\underset{\sim}{x}$ for the original coordinates $\underset{\sim}{x}$ is related to the vector $d\underset{\sim}{X}$ for the deformed coordinates $\underset{\sim}{X}$. Thus,

$$d\underset{\sim}{\tilde{P}}^{t} = \underset{\sim}{g}^{-1}\, d\underset{\sim}{P}^{t} \qquad (3.71)$$

or in indicial form

$$d\tilde{P}^{t}_{i} = \frac{\partial x_{i}}{\partial X_{j}}\, dP^{t}_{j} \qquad (3.72)$$

Now, the second Piola-Kirchhoff stress tensor $\underset{\sim}{\sigma}$ appears within the following expression:

$$d\underset{\sim}{\tilde{P}}^{t} = \underset{\sim}{\mathit{g}}^{-1}\, d\underset{\sim}{P}^{t} = \left(\underset{\sim}{\vec{n}}^{o} \cdot \underset{\sim}{\sigma} \right) dA^{o} \qquad (3.73)$$

It can be proven that the following relations hold between the Piola-Kirchhoff and Cauchy stress tensors [10,34]:

$$\underset{\sim}{\sigma} = \frac{\rho^{o}}{\rho^{t}} \underset{\sim}{\mathit{g}}^{-1} \underset{\sim}{\tau} (\mathit{g}^{-1})' = \underset{\sim}{T} \underset{\sim}{(\mathit{g}^{-1})'} \qquad (3.74)$$

$$\underset{\sim}{T} = \frac{\rho^{o}}{\rho^{t}} \underset{\sim}{\mathit{g}}^{-1} \underset{\sim}{\tau} \qquad (3.75)$$

Also, in indicial form

$$\sigma_{ij} = \frac{\rho^{o}}{\rho^{t}} \frac{\partial x_{i}}{\partial X_{m}} \frac{\partial x_{j}}{\partial X_{n}} \tau_{mn} \qquad (3.76)$$

$$T_{ij} = \frac{\rho^{o}}{\rho^{t}} \frac{\partial x_{i}}{\partial X_{m}} \tau_{mj} \qquad (3.77)$$

where ρ^{o} and ρ^{t} are mass densities in the undeformed and deformed configurations.

Also, the expression for the Cauchy stress in terms of the second Piola-Kirchhoff stress tensor is as follows:

$$\underset{\sim}{\tau} = \frac{\rho^{t}}{\rho^{o}} \underset{\sim}{\mathit{g}}\, \underset{\sim}{\sigma}\, \underset{\sim}{\mathit{g}}' \qquad (3.78)$$

or in indicial form

$$\tau_{ij} = \frac{\rho^{t}}{\rho^{o}} \frac{\partial X_{i}}{\partial x_{m}} \frac{\partial X_{j}}{\partial x_{n}} \sigma_{mn} \qquad (3.79)$$

It should be noted that the first Piola-Kirchhoff stress tensor is simpler. However, from Eq. (3.77) it is seen that, in general, it is a nonsymmetric matrix and, consequently, is cumbersome to use in conjunction with a symmetric strain tensor. On the other hand, we see from Eq. (3.76) that the second Piola-Kirchhoff stress tensor is symmetric whenever the Cauchy stress tensor is symmetric. As a sresult, the second Piola-Kirchhoff stress tensor is usually preferred in finite strain formulations.

Finally, one point should be added. While the Piola-Kirchhoff stress tensors are only pseudo-stresses when large strains are used, they become real stresses whenever strains are small. In such a case, the partial derivatives of the displacements with respect to the original coordinates are all small compared to unity

$$\left(\left| \frac{\partial u_i}{\partial x_j} \right| \ll 1 \right)$$

and the deformation gradient $\underset{\sim}{g}$ approaches the identity matrix $\underset{\sim}{I}$. Consequently, the Piola-Kirchhoff stress tensors approach the Cauchy stress tensor. Furthermore, the product of the partial derivatives of the displacements will be negligible in comparison to the linear terms in Eq. (3.69), and the Green-Lagrangian strain tensor reduces to the well-known linear strain tensor.

3.3.4 Finite Element Formulation

In this section we first write the incremental equations of motion by using the principle of virtual displacements and then discretize them by the method of finite elements. The principle of virtual displacements in terms of the second Piola-Kirchhoff stress tensor and the Green-Lagrangian strain tensor can be written as follows [34]:

$$\int_{V^o} \sigma_{ij}^{t+\Delta t} \delta\epsilon_{ij}^{t+\Delta t} \, dV^o = \delta W_{ext}^{t+\Delta t} - \int_{V^o} \rho^o \ddot{u}_k^{t+\Delta t} \delta(\Delta u_k) \, dV^o \qquad (3.80)$$

In this relation, which leads to dynamic equilibrium equations for the body at time $t+\Delta t$, the symbols $\sigma_{ij}^{t+\Delta t}$ and $\epsilon_{ij}^{t+\Delta t}$ represent the second Piola-Kirchhoff stress and Green-Lagrangian strain tensors at time $t+\Delta t$. The term $\delta W_{ext}^{t+\Delta t}$ denotes the virtual work of external loads and V^o is the original volume of the body. The symbol $u_k^{t+\Delta t}$ represents the displacement at time $t+\Delta t$, and Δu_k is the incremental displacement. That is,

$$\Delta u_k = u_k^{t+\Delta t} - u_k^t \qquad (3.81)$$

We wish to obtain the equations of motion between two neighboring positions at times t and $t+\Delta t$. Therefore, the virtual displacements in Eq. (3.80) are taken to be the variations of incremental displacements between time t and $t+\Delta t$ for a compatible state of deformation.

We exploit the following decomposition of stresses and strains

$$\sigma_{ij}^{t+\Delta t} = \sigma_{ij}^{t} + \Delta\sigma_{ij} \qquad (3.82)$$

$$\epsilon_{ij}^{t+\Delta t} = \epsilon_{ij}^{t} + \Delta\epsilon_{ij} \qquad (3.83)$$

where superscripts refer to time and Δ indicates the increments of stress or strain between time t and $t+\Delta t$.

From Eq. (3.83) we can write virtual strains as

$$\delta\epsilon_{ij}^{t+\Delta t} = \delta(\Delta\epsilon_{ij}) \qquad (3.84)$$

Also, Eq. (3.69) can be written as

$$\epsilon_{ij}^{t+\Delta t} = \frac{1}{2}\left(u_{i,j}^{t+\Delta t} + u_{j,i}^{t+\Delta t} + u_{k,i}^{t+\Delta t}\, u_{k,j}^{t+\Delta t}\right) \qquad (3.85)$$

where commas indicate spatial derivatives with respect to the initial configuration. Furthermore, we can write

$$u_i^{t+\Delta t} = u_i^t + \Delta u_i \qquad (3.86)$$

After substituting Eq. (3.86) into Eq. (3.85) and comparing the results with Eq. (3.83), we find the following relations for the incremental strains:

$$\Delta\epsilon_{ij} = e_{ij} + \zeta_{ij} \qquad (3.87)$$

where

$$e_{ij} = \frac{1}{2}\left(\Delta u_{i,j} + \Delta u_{j,i} + u_{k,i}^t\, \Delta u_{k,j} + u_{k,j}^t\, \Delta u_{k,i}\right) \qquad (3.88)$$

and

$$\zeta_{ij} = \frac{1}{2}\,\Delta u_{k,i}\,\Delta u_{k,j} \quad . \qquad (3.89)$$

We assume that the incremental second Piola-Kirchhoff stress tensor is linearly related to the incremental Green-Lagrangian strain tensor. Therefore,

$$\Delta\sigma_{ij} = C_{ijmn}\,\Delta\epsilon_{mn} \qquad (3.90)$$

where C_{ijmn} is the constitutive tensor. After substituting Eqs. (3.82), (3.84) and (3.90) into Eq. (3.80), we obtain

$$\int_{V^o} \left(\sigma^t_{ij} + C_{ijmn}\,\Delta\epsilon_{mn}\right) \delta\left(\Delta\epsilon_{ij}\right) dV^o = \delta W^{t+\Delta t}_{ext} - \int_{V^o} \rho^o\, \ddot{u}^{t+\Delta t}_k\, \delta\left(\Delta u_k\right) dV^o \tag{3.91}$$

or

$$\int_{V^o} C_{ijmn}\,\Delta\epsilon_{mn}\,\delta\left(\Delta\epsilon_{ij}\right) dV^o + \int_{V^o} \sigma^t_{ij}\,\delta(\Delta\epsilon_{ij} \quad dV^o = \delta W^{t+\Delta t}_{ext} - \int_{V^o} \rho^o\, \ddot{u}^{t+\Delta t}_k\, \delta\left(\Delta u_k\right) dV^o \tag{3.92}$$

After substituting Eq. (3.87) into Eq. (3.92), we finally obtain the following variational form of the equation of motion in the Lagrangian description:

$$\begin{aligned}
&\int_{V^o} C_{ijmn}\, e_{mn}\, \delta e_{ij}\, dV^o + \int_{V^o} C_{ijmn}\, e_{mn}\, \delta\zeta_{ij}\, dV^o + \int_{V^o} C_{ijmn}\, \zeta_{mn}\, \delta e_{ij}\, dV^o \\
&+ \int_{V^o} C_{ijmn}\, \zeta_{mn}\, \delta\zeta_{ij}\, dV^o + \int_{V^o} \sigma^t_{ij}\, \delta\zeta_{ij}\, dV^o = \delta W^{t+\Delta t}_{ext} \\
&- \int_{V^o} \sigma^t_{ij}\, \delta e_{ij}\, dV^o - \int_{V^o} \rho^o\, \ddot{u}^{t+\ t}_k\, \delta\left(\Delta u_k\right) dV^o
\end{aligned} \tag{3.93}$$

Now we can discretize this variational equation by finite elements. Nodal and generic displacements are related to each other through shape function $\underset{\sim}{T}_1$

$$\underset{\sim}{u}^t = \underset{\sim}{T}_1\, \underset{\sim}{q}^t \tag{3.94}$$

we assume the same shape function to approximate the increments of nodal and generic displacements between configurations at times t and $t+\Delta t$

$$\Delta \underset{\sim}{u} = \underset{\sim}{T}_1 \, \Delta \underset{\sim}{q} \tag{3.95}$$

where

$$\Delta \underset{\sim}{u} = \underset{\sim}{u}^{t+\Delta t} - \underset{\sim}{u}^{t} \tag{3.96}$$

and

$$\Delta \underset{\sim}{q} = \underset{\sim}{q}^{t+\Delta t} - \underset{\sim}{q}^{t} \tag{3.97}$$

By using the above relations, we can transform Eq. (3.93) into the following discretized form for a single finite element:

$$\left(\underset{\sim}{K}_L + \underset{\sim}{K}_1 + \underset{\sim}{K}_2 + \underset{\sim}{K}_3 + \underset{\sim}{K}_G \right) \Delta \underset{\sim}{q} = \underset{\sim}{p}^{t+\Delta t} - \underset{\sim}{f}^{t} - \underset{\sim}{m} \, \ddot{\underset{\sim}{q}}^{t+\Delta t} \tag{3.98}$$

where the arrays $\underset{\sim}{K}_L$, $\underset{\sim}{K}_1$, $\underset{\sim}{K}_2$, $\underset{\sim}{K}_3$, $\underset{\sim}{K}_G$, $\underset{\sim}{f}^{t}$ and $\underset{\sim}{m}$ are derived by equating the virtual work of the continuum to the virtual work of the equivalent discretized finite element, from the following expressions:

$$\delta \left(\Delta \underset{\sim}{q} \right)' \underset{\sim}{K}_L \, \Delta \underset{\sim}{q} = \int_{V^o} C_{ijmn} \, e_{mn} \, \delta e_{ij} \, dV^o \tag{3.99}$$

$$\delta \left(\Delta \underset{\sim}{q} \right)' \underset{\sim}{K}_1 \, \Delta \underset{\sim}{q} = \int_{V^o} C_{ijmn} \, e_{mn} \, \delta \zeta_{ij} \, dV^o \tag{3.100}$$

$$\delta \left(\Delta \underset{\sim}{q} \right)' \underset{\sim}{K}_2 \, \Delta \underset{\sim}{q} = \int_{V^o} C_{ijmn} \, \zeta_{mn} \, \delta e_{ij} \, dV^o \tag{3.101}$$

$$\delta \left(\Delta \underset{\sim}{q} \right)' \underset{\sim}{K}_3 \, \Delta \underset{\sim}{q} = \int_{V^o} C_{ijmn} \, \zeta_{mn} \, \delta \zeta_{ij} \, dV^o \tag{3.102}$$

and

$$\delta\left(\Delta \underset{\sim}{q}\right)' \underset{\sim}{K}_G \Delta \underset{\sim}{q} = \int_{V^o} \sigma_{ij}^t \, \delta\zeta_{ij} \, dV^o \qquad (3.103)$$

$$\delta\left(\Delta \underset{\sim}{q}\right)' \underset{\sim}{f}^t = \int_{V^o} \sigma_{ij}^t \, \delta e_{ij} \, dV^o \qquad (3.104)$$

$$\delta\left(\Delta \underset{\sim}{q}\right)' \underset{\sim}{m} \, \ddot{\underset{\sim}{q}}^{t+\Delta t} = \int_{V^o} \rho^o \, \ddot{u}_k^{t+\Delta t} \, \delta\left(\Delta u_k\right) dV^o \qquad (3.105)$$

Also, $\underset{\sim}{p}^{t+\Delta t}$ is the vector of external nodal forces at time $t+\Delta t$. If we assume that loads are independent of the deformation of the body, we can find the equivalent nodal loads due to surface and volume loads by the same formulas we found for the linear case in Eqs. (3.10) and (3.11) of Section 3.2.1. In this study, only these so-called conservative loads have been considered. One case of nonconservative loading has been formulated in Reference 1.

From Eq. (3.88), it is seen that e_{ij} is a linear function of displacements at time t and the incremental displacements between times t and $t+\Delta t$. However, Eq. (3.89) shows that ζ_{ij} is a quadratic function of the incremental displacements. As a result, the stiffness matrix $\underset{\sim}{K}_L$ is a function of displacements at time t , but it is not a function of the incremental displacements. In contrast, the stiffness matrices $\underset{\sim}{K}_1$ and $\underset{\sim}{K}_2$ are linear functions of the incremental displacements, and the stiffness matrix $\underset{\sim}{K}_3$ is a quadratic function of the incremental displacements. In other words, the terms of the stiffness matrices $\underset{\sim}{K}_1$ and $\underset{\sim}{K}_2$ are one order of magnitude and the terms of matrix $\underset{\sim}{K}_3$ are two orders of magnitude smaller than the terms of stiffness matrix $\underset{\sim}{K}_L$ and consequently can be neglected in most practical cases. In this case, Eq. (3.98) will reduce to the following equation:

$$\left(\underset{\sim}{K}_L + \underset{\sim}{K}_G\right) \Delta\underset{\sim}{q} = \underset{\sim}{p}^{t+\Delta t} - \underset{\sim}{f}^{t} - \underset{\sim}{m}\,\ddot{\underset{\sim}{q}}^{t+\Delta t} \qquad (3.106)$$

where $\underset{\sim}{K}_G$ is the so-called geometric stiffness matrix. In the following section, this matrix, as well as $\underset{\sim}{K}_L$, $\underset{\sim}{f}^t$ and $\underset{\sim}{m}$, will be evaluated.

3.3.5 Evaluation of Element Matrices

Stiffness Matrix $\underset{\sim}{K}_L$

First, we note that the linear incremental strain matrix $\underset{\sim}{e} = \left\{e_{11} , e_{22} , 2e_{12}\right\}$ is related to the increment of the generic displacements by a transformation matrix $\underset{\sim}{T}_4$ as follows:

$$\underset{\sim}{e} = \underset{\sim}{T}_4 \, \Delta\underset{\sim}{u} \qquad (3.107)$$

Substituting $\Delta\underset{\sim}{u}$ from Eq. (3.95) into Eq. (3.107), we find

$$\underset{\sim}{e} = \underset{\sim}{T}_4 \, \underset{\sim}{T}_1 \, \Delta\underset{\sim}{q} = \underset{\sim}{B}_L \, \Delta\underset{\sim}{q} \qquad (3.108)$$

Thus,

$$\underset{\sim}{B}_L = \underset{\sim}{T}_4 \, \underset{\sim}{T}_1 \qquad (3.109)$$

The matrix of shape functions $\underset{\sim}{T}_1$ is given in Section 3.2.3. From Eqs. (3.88) and (3.107) we obtain the matrix $\underset{\sim}{T}_4$ for plane problems as follows:

$$\underset{\sim}{T}_4 = \begin{bmatrix} 1 + u^t_{1,1}\dfrac{\partial}{\partial x} & u^t_{2,1}\dfrac{\partial}{\partial x} \\ u^t_{1,2}\dfrac{\partial}{\partial y} & 1 + u^t_{2,2}\dfrac{\partial}{\partial y} \\ u^t_{1,2}\dfrac{\partial}{\partial x} + \left(1 + u^t_{1,1}\right)\dfrac{\partial}{\partial y} & \left(1 + u^t_{2,2}\right)\dfrac{\partial}{\partial x} + u^t_{2,1}\dfrac{\partial}{\partial y} \end{bmatrix} \qquad (3.110)$$

Now the matrix $\underset{\sim}{B}_L$ can be obtained from Eq. (3.109). Noting that necessary derivatives are available from $\underset{\sim}{D}_G$ [see Eq. (3.41)], we obtain

$$\underset{\sim}{B}_L = \left[\begin{array}{cccccccc} \theta_1\, D_{G11} & \theta_2\, D_{G11} & \theta_1\, D_{G12} & \theta_2\, D_{G12} & \theta_1\, D_{G13} & \theta_2\, D_{G13} & \theta_1\, D_{G14} & \theta_2\, D_{G14} \\ \theta_3\, D_{G21} & \theta_4\, D_{G21} & \theta_3\, D_{G22} & \theta_4\, D_{G22} & \theta_3\, D_{G23} & \theta_4\, D_{G23} & \theta_3\, D_{G24} & \theta_4\, D_{G24} \\ \theta_3\, D_{G11} + \theta_1\, D_{G21} & \theta_4\, D_{G11} + \theta_2\, D_{G21} & \theta_3\, D_{G12} + \theta_1\, D_{G22} & \theta_4\, D_{G12} + \theta_2\, D_{G22} & \theta_3\, D_{G13} + \theta_1\, D_{G23} & \theta_4\, D_{G13} + \theta_2\, D_{G23} & \theta_3\, D_{G14} + \theta_1\, D_{G24} & \theta_4\, D_{G14} + \theta_2\, D_{G24} \end{array}\right] \qquad (3.111)$$

where

$$\theta_1 = 1 + {}^{t}u_{1,1} = 1 + {}^{t}q_1\, D_{G11} + {}^{t}q_3\, D_{G12} + {}^{t}q_5\, D_{G13} + {}^{t}q_7\, D_{G14}$$

$$\theta_2 = {}^{t}u_{2,1} = {}^{t}q_2\, D_{G11} + {}^{t}q_4\, D_{G12} + {}^{t}q_6\, D_{G13} + {}^{t}q_8\, D_{G14}$$

$$\theta_3 = {}^{t}u_{1,2} = {}^{t}q_1\, D_{G21} + {}^{t}q_3\, D_{G22} + {}^{t}q_5\, D_{G23} + {}^{t}q_7\, D_{G24}$$

$$\theta_4 = 1 + {}^{t}u_{2,2} = 1 + {}^{t}q_2\, D_{G21} + {}^{t}q_4\, D_{G22} + {}^{t}q_6\, D_{G23} + {}^{t}q_8\, D_{G24} \,.$$

Introducing Eq. (3.108) into Eq. (3.99) and manipulating the results, we obtain the stiffness matrix $\underset{\sim}{K}_L$ as

$$\underset{\sim}{K}_L = \int_{V^o} \underset{\sim}{B}_L' \, \underset{\sim}{T}_3 \, \underset{\sim}{B}_L \, dV^o \qquad (3.112)$$

where $\underset{\sim}{T}_3$ is the constitutive matrix $(\underset{\sim}{T}_3 \equiv C_{ijmn})$. For an isoparametric quadrilateral element, $\underset{\sim}{K}_L$ can be written in terms of the natural coordinates ξ and η as follows:

$$\underset{\sim}{K}_L = h \int_{-1}^{1}\int_{-1}^{1} \underset{\sim}{B}_L' \, \underset{\sim}{T}_3 \, \underset{\sim}{B}_L \, |\underset{\sim}{J}| \, d\xi d\eta \qquad (3.113)$$

<u>Geometric Stiffness Matrix</u> $\underset{\sim}{K}_G$

As a preliminary step in evaluating the geometric stiffness matrix for plane stress (or plane strain) problem, we decompose the incremental strains $(\Delta\underset{\sim}{\epsilon})$ into linear $(\underset{\sim}{e})$ and nonlinear (ζ) terms

$$\Delta\underset{\sim}{\epsilon} = \underset{\sim}{e} + \underset{\sim}{\zeta} \qquad (3.114)$$

or

$$\begin{bmatrix} \Delta\epsilon_{11} \\ \Delta\epsilon_{22} \\ 2\,\Delta\epsilon_{12} \end{bmatrix} = \begin{bmatrix} e_{11} \\ e_{22} \\ 2e_{12} \end{bmatrix} + \begin{bmatrix} \zeta_{11} \\ \zeta_{22} \\ 2\zeta_{12} \end{bmatrix} \qquad (3.115)$$

Terms of matrices $\underset{\sim}{e}$ and $\underset{\sim}{\zeta}$ can be found from Eqs. (3.88) and (3.89). The geometric stiffness matrix is derived from the following virtual work equation:

$$\delta\left(\Delta\underset{\sim}{q}\right)' \underset{\sim}{K}_G \, \Delta\underset{\sim}{q} = \int_{V^o} \sigma^t_{ij} \, \delta\zeta_{ij} \, dV^o \qquad (3.103) \text{ repeated}$$

It can be shown that

$$\begin{aligned} \sigma^t_{ij}\,\zeta_{ij} &= \frac{1}{2}\,\sigma^t_{11}\left[\left(\Delta u_{1,1}\right)^2 + \left(\Delta u_{2,1}\right)^2\right] + \frac{1}{2}\,\sigma^t_{12}\left[\Delta u_{1,1}\,\Delta u_{1,2} + \Delta u_{2,1}\,\Delta u_{2,2}\right] \\ &\quad + \frac{1}{2}\,\sigma^t_{21}\left[\Delta u_{1,1}\,\Delta u_{1,2} + \Delta u_{2,1}\,\Delta u_{2,2}\right] + \frac{1}{2}\,\sigma^t_{22}\left[\left(\Delta u_{1,2}\right)^2 + \left(\Delta u_{2,2}\right)^2\right] \\ &= \frac{1}{2}\,\underset{\sim}{T}_7'\,\underset{\sim}{T}_6'\,\underset{\sim}{T}_5\,\underset{\sim}{T}_6\,\underset{\sim}{T}_7 \end{aligned} \qquad (3.116)$$

where

$$\underset{\sim}{T}_5 = \begin{bmatrix} \sigma^t_{11} & \sigma^t_{12} & 0 & 0 \\ \sigma^t_{21} & \sigma^t_{22} & 0 & 0 \\ 0 & 0 & \sigma^t_{11} & \sigma^t_{12} \\ 0 & 0 & \sigma^t_{21} & \sigma^t_{22} \end{bmatrix} \qquad (3.117)$$

$$\underset{\sim}{T}_6 = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 & 0 \\ 0 & 0 & \frac{\partial}{\partial x} & 0 \\ 0 & 0 & 0 & \frac{\partial}{\partial y} \end{bmatrix} \qquad (3.118)$$

$$\underset{\sim}{T}_7 = \begin{bmatrix} \Delta u_1 \\ \Delta u_1 \\ \Delta u_2 \\ \Delta u_2 \end{bmatrix} \qquad (3.119)$$

We can relate $\underset{\sim}{T}_7$ to incremental nodal displacements by a transformation matrix $\underset{\sim}{T}_8$

$$\underset{\sim}{T}_7 = \underset{\sim}{T}_8 \, \underset{\sim}{\Delta q} \qquad (3.120)$$

where

$$\underset{\sim}{T}_8 = \begin{bmatrix} f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 & 0 \\ f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 & 0 \\ 0 & f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 \\ 0 & f_1 & 0 & f_2 & 0 & f_3 & 0 & f_4 \end{bmatrix} \qquad (3.121)$$

The functions f_1 , f_2 , f_3 and f_4 are defined in Eqs. (3.30). After substituting Eq. (3.120) into Eq. (3.116), we find

$$\sigma^t_{ij}\, \zeta_{ij} = \frac{1}{2} \left(\underset{\sim}{\Delta q}\right)' \underset{\sim}{T}'_8\, \underset{\sim}{T}'_6\, \underset{\sim}{T}_5\, \underset{\sim}{T}_6\, \underset{\sim}{T}_8 \left(\underset{\sim}{\Delta q}\right) \qquad (3.122)$$

And taking the variation of this expression, we obtain

$$\sigma^t_{ij}\, \delta\zeta_{ij} = \delta\left(\underset{\sim}{\Delta q}\right)' \underset{\sim}{T}'_8\, \underset{\sim}{T}'_6\, \underset{\sim}{T}_5\, \underset{\sim}{T}_6\, \underset{\sim}{T}_8 \left(\underset{\sim}{\Delta q}\right) \qquad (3.123)$$

Finally, by comparing Eq. (3.103) with Eq. (3.123), we see that the formula for the geometric stiffness matrix is

$$\underset{\sim}{K}_G = \int_{V^o} \underset{\sim}{T}'_8\, \underset{\sim}{T}'_6\, \underset{\sim}{T}_5\, \underset{\sim}{T}_6\, \underset{\sim}{T}_8\, dV^o \qquad (3.124)$$

And for an isoparametric quadrilateral element in terms of local coordinates, we have

$$\underset{\sim}{K}_G = h \int_{-1}^{1}\int_{-1}^{1} \underset{\sim}{T}'_8\, \underset{\sim}{T}'_6\, \underset{\sim}{T}_5\, \underset{\sim}{T}_6\, \underset{\sim}{T}_8\, \left|\underset{\sim}{J}\right|\, d\xi\, d\eta \qquad (3.125)$$

Terms in the matrix $\underset{\sim}{W} = \underset{\sim}{T}'_8\, \underset{\sim}{T}'_6\, \underset{\sim}{T}_5\, \underset{\sim}{T}_6\, \underset{\sim}{T}_8$ have been derived explicitly in terms of stresses and the terms of matrix $\underset{\sim}{D}_G$. They are given in Table 3.1. It is interesting to note that only ten terms of this matrix are nonzero and unique. The use of such explicit formulas results in efficient computer programs.

Nodal Force Vector $\underset{\sim}{f}^t$

If we let

$$\underset{\sim}{\sigma}^t = \begin{bmatrix} \sigma_{11}^t \\ \sigma_{22}^t \\ \sigma_{12}^t \end{bmatrix} \qquad (3.126)$$

and use Eq. (3.108) in Eq. (3.104), we obtain

$$\underset{\sim}{f}^t = \int_{V^o} \underset{\sim}{B}_L' \, \underset{\sim}{\sigma}^t \, dV^o \qquad (3.127)$$

For an isoparametric quadrilateral element in terms of local coordinates we have

$$\underset{\sim}{f}^t = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{B}_L' \, \underset{\sim}{\sigma}^t \, |\underset{\sim}{J}| \, d\xi \, d\eta \qquad (3.128)$$

Element Mass Matrix $\underset{\sim}{m}$

From the integral expression of Eq. (3.105) and the relation

$$\underset{\sim}{\ddot{u}}^{t+\Delta t} = \underset{\sim}{T}_1 \, \underset{\sim}{\ddot{D}}^{t+\Delta t}$$

we obtain

$$\underset{\sim}{m} = \int_{V^o} \rho^o \, \underset{\sim}{T}_1' \, \underset{\sim}{T}_1 \, dV^o \qquad (3.129)$$

TABLE 3.1 Terms in Matrix $\underset{\sim}{W}$

$$\underset{\sim}{W} = \begin{bmatrix} w_1 & 0 & w_2 & 0 & w_3 & 0 & w_4 & 0 \\ & w_1 & 0 & w_2 & 0 & w_3 & 0 & w_4 \\ & & w_5 & 0 & w_6 & 0 & w_7 & 0 \\ & & & w_5 & 0 & w_6 & 0 & w_7 \\ \text{symmetric} & & & & w_8 & 0 & w_9 & 0 \\ & & & & & w_8 & 0 & w_9 \\ & & & & & & w_{10} & 0 \\ & & & & & & & w_{10} \end{bmatrix}$$

where

$$w_1 = D^2_{G11}\ \sigma^t_{11} + 2D_{G11}\ D_{G21}\ \sigma^t_{12} + D^2_{G21}\ \sigma^t_{22}$$

$$w_2 = D_{G11}\ D_{G12}\ \sigma^t_{11} + \left(D_{G11}\ D_{G22} + D_{G21}\ D_{G12}\right)\sigma^t_{12} + D_{G21}\ D_{G22}\ \sigma^t_{22}$$

$$w_3 = D_{G11}\ D_{G13}\ \sigma^t_{11} + \left(D_{G11}\ D_{G23} + D_{G21}\ D_{G13}\right)\sigma^t_{12} + D_{G21}\ D_{G23}\ \sigma^t_{22}$$

$$w_4 = D_{G11}\ D_{G14}\ \sigma^t_{11} + \left(D_{G11}\ D_{G24} + D_{G14}\ D_{G21}\right)\sigma^t_{12} + D_{G21}\ D_{G24}\ \sigma^t_{22}$$

$$w_5 = D^2_{G12}\ \sigma^t_{11} + 2\ D_{G12}\ D_{G22}\ \sigma^t_{12} + D^2_{G22}\ \sigma^t_{22}$$

$$w_6 = D_{G12}\ D_{G13}\ \sigma^t_{11} + \left(D_{G12}\ D_{G23} + D_{G13}\ D_{G22}\right)\sigma^t_{12} + D_{G22}\ D_{G23}\ \sigma^t_{22}$$

$$w_7 = D_{G12}\ D_{G14}\ \sigma^t_{11} + \left(D_{G12}\ D_{G24} + D_{G14}\ D_{G22}\right)\sigma^t_{12} + D_{G22}\ D_{G24}\ \sigma^t_{22}$$

$$w_8 = D^2_{G13}\ \sigma^t_{11} + 2\ D_{G13}\ D_{G23}\ \sigma^t_{12} + D^2_{G23}\ \sigma^t_{22}$$

$$w_9 = D_{G13}\ D_{G14}\ \sigma^t_{11} + \left(D_{G13}\ D_{G24} + D_{G14}\ D_{G23}\right)\sigma^t_{12} + D_{G23}\ D_{G24}\ \sigma^t_{22}$$

$$w_{10} = D^2_{G14}\ \sigma^t_{11} + 2\ D_{G14}\ D_{G24}\ \sigma^t_{12} + D^2_{G24}\ \sigma^t_{22}$$

In terms of local coordinates for an isoparametric element we have

$$\underset{\sim}{m} = h \int_{-1}^{1}\int_{-1}^{1} \rho^{o}\, \underset{\sim}{T}_1' \,\underset{\sim}{T}_1 \,|\underset{\sim}{J}|\, d\xi\, d\eta \qquad (3.130)$$

However, as was mentioned in Section 3.2.3, there is another way of calculating the element mass matrix (lumped mass method) resulting in a diagonal mass matrix (see Appendix B).

3.3.6 Constitutive Equations of Plasticity

In Sections 3.3.4 and 3.3.5 finite element equations of motion were derived for very general large displacement and large strain cases. No restriction was imposed on the constitutive matrix $\underset{\sim}{T}_3$; and any material law can be used, as long as it relates the increment of second Piola-Kirchhoff stress to the increment of Green-Lagrangian strain. However, in this section we restrict ourselves to the case of small strain plasticity. The question of proper constitutive equations for large strain plasticity has not been quite settled and is being actively investigated by many researchers. A great deal of additional research is needed to formulate the constitutive law for the problem of large strain flow [1,36].

In plasticity theory, it is usually assumed that the plastic deformations are independent of time and that the material is inviscid (athermal plasticity). Time-dependent plastic deformations are studied in the theory of creep, the theory of viscoplasticity, and the new science of rheology. In this section we first discuss briefly the basic principles of the theory of plasticity, after which Yamada's plastic stress-strain matrix is presented.

Criteria for Yielding

A yield criterion is a postulate which characterizes the limit of elasticity under any possible combination of stresses. For an isotropic material the yield condition must be a symmetric function of the principal stresses

$$\Phi(\sigma_{ij}) = \Phi(\sigma_1 , \sigma_2 , \sigma_3) = \kappa \qquad (3.131)$$

where σ_1 , σ_2 and σ_3 are the three principal normal stresses and κ is a constant connected with the yield limit of the material. Equation (3.131) can also be written as a function of the stress invariants I_1 , I_2 and I_3 as follows:

$$\Phi = \Phi(I_1 , I_2 , I_3) = \kappa \qquad (3.132)$$

where

$$\begin{aligned} I_1 &= \sigma_1 + \sigma_2 + \sigma_3 \\ I_2 &= - (\sigma_1\sigma_2 + \sigma_1\sigma_3 + \sigma_2\sigma_3) \\ I_3 &= \sigma_1\sigma_2\sigma_3 \end{aligned}$$

Furthermore, experiments have shown that the hydrostatic (or mean) pressure has negligible influence on yielding. Therefore, the yield function can be written as

$$\Phi = \Phi(I_2 , I_3) = \kappa \qquad (3.133)$$

Several yield criteria have been proposed, but two of them have shown good agreements with experiments. They are the Tresca and von Mises yield criteria [23,35]. In the Tresca yield condition, yielding occurs whenever the maximum shear stress reaches the maximum shear stress in the uniaxial tension test. If we assume $\sigma_1 \geq \sigma_2 \geq \sigma_3$, the Tresca yield condition can be written in the form

$$\Phi = \frac{\sigma_1 - \sigma_3}{2} = \kappa = \frac{1}{2} \sigma_s \qquad (3.134)$$

where σ_s is the yield stress in the uniaxial test. The Tresca yield function in the stress space is a prism whose trace on the so-called π-plane (the plane passing through the origin with the equation $\sigma_1 + \sigma_2 + \sigma_3 = 0$) is a right hexagon as shown in Fig. 3.5.

The von Mises criterion is postulated in terms of the second invariant of stresses

$$\Phi = I_2 = \kappa = 2\sigma_s^2 \qquad (3.135)$$

or

$$(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2 = 2\sigma_s^2$$

This function is a cylinder in the stress space, and its projection on the π-plane is a circle that circumscribes the Tresca hexagon. The von Mises yield criterion generally fits the experimental data rather better than the Tresca yield condition. In addition, von Mises' criterion is mathematically easier to use, and no information is needed concerning the magnitude of principla stresses. However, it should be noted that the von Mises and Tresca yield criteria are not considerably different. The discrepancy can be further decreased if one takes the circle which lies midway between the circumscribed and inscribed circles to the Tresca hexagon (see Fig. 3.5).

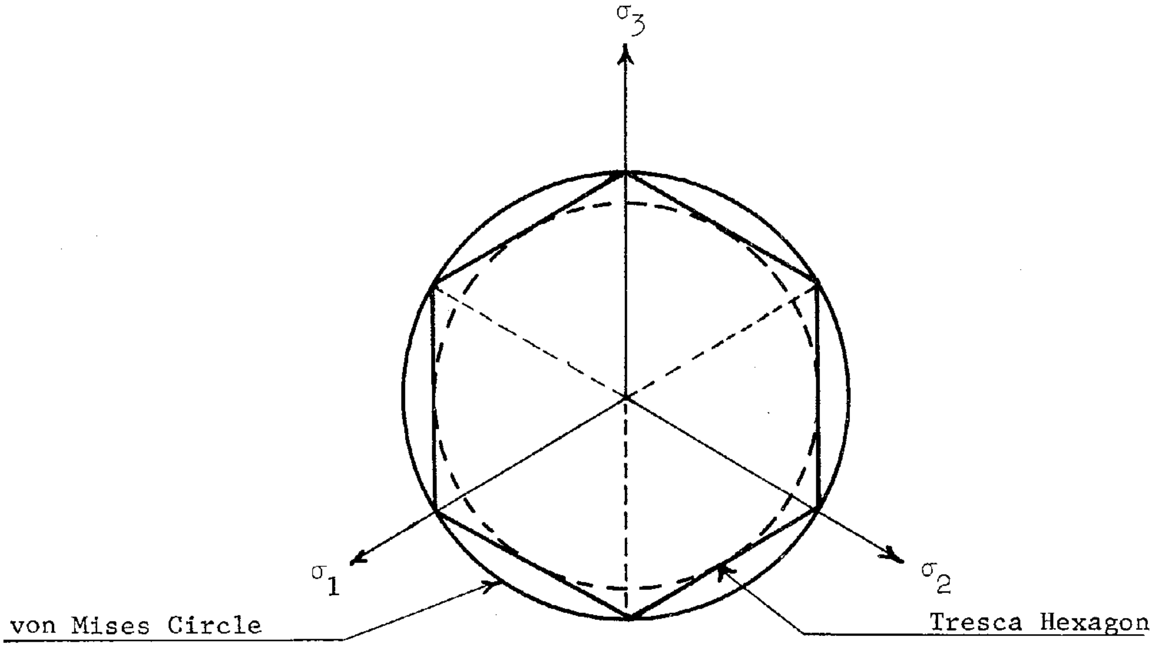


Fig. 3.5 Representation of yield criteria on π-plane

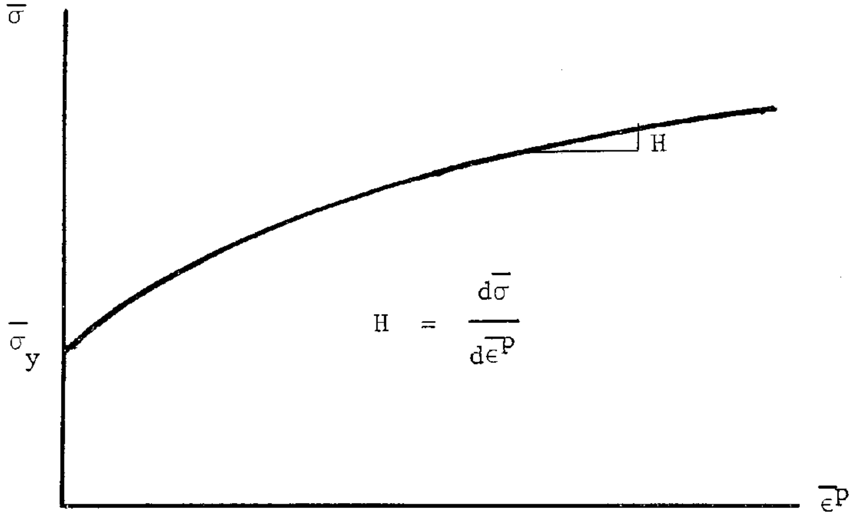


Fig. 3.6 Effective stress-effective plastic strain curve

Strain Hardening, Loading and Unloading

The yield conditions discussed above indicate the hypothetical surface at which the material first starts yielding. However, many materials (especially metals) show strain hardening, which means that plastic deformation increases the elastic limit of the material (for the uniaxial test, this means a positive slope in the plastic range of the stress-strain curve). Consequently, the yield surface changes for continued straining beyond the initial yield. The strain hardening of the material can be taken into account if one generalizes Eqs. (3.131), (3.134) and (3.135) by changing the constant κ to be a strain hardening parameter that varies as yielding occurs. Thus, from Eq. (3.131), continued plastic loading and unloading from a plastic state to an elastic state as well as neutral loading can be stipulated as follows:

$$\text{For plastic loading} \qquad \Phi = \kappa \;, \quad d\Phi = \frac{\partial\Phi}{\partial\sigma_{ij}} d\sigma_{ij} > 0$$

$$\text{For unloading} \qquad \Phi = \kappa \;, \quad d\Phi = \frac{\partial\Phi}{\partial\sigma_{ij}} d\sigma_{ij} < 0$$

$$\text{For neutral loading} \qquad \Phi = \kappa \;, \quad d\Phi = \frac{\partial\Phi}{\partial\sigma_{ij}} d\sigma_{ij} = 0$$

The rule of isotropic hardening constitutes the simplest behavior of this type. Under this rule it is implied that as the yield surface expands, it preserves its initial shape without any side translations. On the π-plane, the yield surfaces for the Tresca and von Mises yield conditions can be visualized as a series of concentric regular hexagons and circles. This isotropic hardening rule is mathematically easy to handle, but it ignores the Bauschinger effect. This effect refers to a metal that is first strained under uniform tension and is then reloaded in compression. It is observed that the compression yielding occurs at a significantly reduced stress. In order to take into account the Bauschinger effect, several other hardening rules have been proposed [18,23,35] which are more complicated, and their use in specific problems involves mathematical difficulties. In a recent investigation [50], several hardening rules including the isotropic hardening rule and the kinematic hardening rule (in which the Bauschinger effect is accounted for) were compared with experimental results. It was found that among all hardening rules considered, the isotropic hardening was in the best agreement with the experimental results.

Flow Theory of Plasticity

In the deformation theory of plasticity, equations of plastic deformations are constructed in the form of relations between finite stress and strain. In contrast, in the flow theory of plasticity, increments of stress and strain are related to each other. Equations of the deformation theory of plasticity are simpler to use, but they suffer from certain basic deficiencies. Use of the deformation theory in the case of nonproportional loading can lead to unsatisfactory results. Consequently, flow theory has been generally favored in solving plasticity problems. In the flow theory of plasticity for isotropic material, the total strain increment $(d\epsilon_{ij})$ is decomposed into an elastic strain increment $(d\epsilon^e_{ij})$ and a plastic strain increment $(d\epsilon^p_{ij})$

$$d\epsilon_{ij} = d\epsilon^e_{ij} + d\epsilon^p_{ij} \tag{3.136}$$

The elastic component of the total strain increment is related to the increment of stress by Hooke's law

$$d\sigma_{ij} = C_{ijkl}\, d\epsilon^e_{kl} \tag{3.137}$$

where C_{ijkl} is the elastic stress-strain tensor. Equation (3.137) can be written in a matrix form as follows:

$$\underset{\sim}{d\sigma} = \underset{\sim}{T_3}\, \underset{\sim}{d\epsilon}^e \tag{3.138}$$

where $\underset{\sim}{d\sigma}$ and $\underset{\sim}{d\epsilon}^e$ are the vectors of incremental stress and elastic strain. Matrix $\underset{\sim}{T_3}$ for plane stress and plane strain cases are given in Section 3.2.2.

In the flow theory, it is also assumed that the plastic strain increment is proportional to the deviatoric stress. Thus,

$$d\epsilon^p_{ij} = \sigma'_{ij}\, d\lambda \tag{3.139}$$

where $d\lambda$ is some scalar factor of proportionality. In this expression σ'_{ij} is the deviatoric stress, defined as

$$\sigma'_{ij} = \sigma_{ij} - \sigma\, \delta_{ij} \tag{3.140}$$

in which $\sigma = \frac{1}{3}\sigma_{ii}$ is the mean pressure. Equation (3.138), attributed to Prandtl and Reuss, is known as the Prandtl-Reuss equation [18].

Finally, it is assumed that in the plastic range of the material the density (and consequently the volume) does not change. Therefore, the material can be assumed to be incompressible.

Yamada's Incremental Plastic Stress-Strain Matrix

By using Eq. (3.136) and linear stress-strain relationships, we can separate the deviatoric and the volumetric strain increments. Then the Prandtl-Reuss equation, including the von Mises yield condition, can be written as

$$d\epsilon'_{ij} = \sigma'_{ij}\, d\lambda + \frac{d\sigma'_{ij}}{2G} \tag{3.141}$$

$$d\epsilon_{ii} = \frac{1 - 2\nu}{E}\, d\sigma_{ii} \tag{3.142}$$

$$\sigma'_{ij}\, \sigma'_{ij} = \frac{2}{3}\, \bar{\sigma}^2 \tag{3.143}$$

where E is Young's modulus, G is the shear modulus and ν is Poisson's ratio. A prime indicates deviatoric components of stress or strain. The scalar multiplier $d\lambda$ is given by the following relation [18]

$$d\lambda = \frac{3}{2}\, \frac{d\bar{\epsilon}^P}{\bar{\sigma}} = \frac{3}{2}\, \frac{d\bar{\sigma}}{\bar{\sigma} H} \tag{3.144}$$

where $H = d\bar{\sigma}/d\bar{\epsilon}^P$, and $\bar{\sigma}$ and $d\bar{\epsilon}^P$ are the equivalent (or effective) stress and the plastic strain increment. These terms are defined as

$$\bar{\sigma} = \sqrt{\frac{3}{2}\, \sigma'_{ij}\, \sigma'_{ij}} \tag{3.145}$$

$$d\bar{\epsilon}^P = \sqrt{\frac{2}{3}\, d\epsilon^P_{ij}\, d\epsilon^P_{ij}} \tag{3.146}$$

Note that H is the slope of the effective stress-effective plastic strain curve, as indicated in Fig. 3.6.

By using Eqs. (3.141), (3.143) and (3.144) and after certain manipulations, one can obtain the following relation between incremental stress and strain:

$$d\sigma_{ij} = 2G \left[d\epsilon_{ij} + \frac{\nu}{1 - 2\nu} \delta_{ij} \, d\epsilon_{ii} - \sigma'_{ij} \, \frac{\sigma'_{kl} \, d\epsilon_{kl}}{\frac{2}{3} \bar{\sigma}^2 \left(1 + \frac{H}{3G} \right)} \right] \qquad (3.147)$$

This equation can be represented in matrix form as

$$\underset{\sim}{d\sigma} = \underset{\sim}{D^p} \, \underset{\sim}{d\epsilon} \qquad (3.148)$$

where $\underset{\sim}{d\sigma}$ and $\underset{\sim}{d\epsilon}$ are the vectors of incremental stress and strain. The explicit formulation of this plastic stress-strain matrix $\underset{\sim}{D^p}$ was first introduced by Yamada, et al. [59,60].

In summary, Yamada's explicit formulation of the plastic stress-strain matrix is based on the Prandtl-Reuss incremental equations of plasticity with their associated differential form of the von Mises yield criterion. In this formulation, elastic compressibility as well as isotropic hardening of the material are taken into account, but the Bauschinger effect is ignored. Yamada's formulation of the plastic stress-strain matrix, specialized for plane stress and plane strain conditions (see Table 3.2) has been used in the present investigation. It should be noted that the plastic stress-strain matrices given in Table 3.2 are defined by Eq. (3.148) in which the vectors $\underset{\sim}{d\sigma}$ and $\underset{\sim}{d\epsilon}$ for plane problems are specified as

$$\underset{\sim}{d\sigma} = \begin{bmatrix} d\sigma_{xx} \\ d\sigma_{yy} \\ d\sigma_{xy} \end{bmatrix} \quad \text{and} \quad \underset{\sim}{d\epsilon} = \begin{bmatrix} d\epsilon_{xx} \\ d\epsilon_{yy} \\ d\gamma_{xy} \end{bmatrix}$$

Implementation of these concepts into computer programs will be discussed in the following chapter.

TABLE 3.2

Plastic Stress-Strain Matrix for Plane Stress and Plane Strain Conditions

A. Plane Stress

$$\underset{\sim}{D}^p = \frac{E}{\Gamma_3}\begin{bmatrix} \sigma'^2_{yy} + 2\Gamma_1 & & \text{Sym.} \\ -\sigma'_{xx}\sigma'_{yy} + 2\nu\,\Gamma_1 & \sigma'^2_{xx} + 2\Gamma_1 & \\ -\dfrac{\sigma'_{xx} + \nu\,\sigma'_{yy}}{1+\nu}\,\sigma_{xy} & -\dfrac{\sigma'_{yy} + \nu\,\sigma'_{xx}}{1+\nu}\,\sigma_{xy} & \dfrac{\Gamma_2}{2(1+\nu)} + \dfrac{2H}{9E}(1-\nu)\,\bar{\sigma}^2 \end{bmatrix}$$

where, $\Gamma_1 = \dfrac{2H}{9E}\,\bar{\sigma}^2 + \dfrac{\sigma^2_{xy}}{1+\nu}$

$$\Gamma_2 = \sigma'^2_{xx} + 2\nu\,\sigma'_{xx}\sigma'_{yy} + \sigma'^2_{yy}$$

$$\Gamma_3 = \Gamma_2 + 2(1-\nu^2)\,\Gamma_1$$

B. Plane Strain

$$\underset{\sim}{D}^p = \frac{E}{1+\nu}\begin{bmatrix} \dfrac{1-\nu}{1-2\nu} - \dfrac{\sigma'^2_{xx}}{\Gamma_4} & & \text{Sym.} \\ \dfrac{\nu}{1-2\nu} - \dfrac{\sigma'_{xx}\sigma'_{yy}}{\Gamma_4} & \dfrac{1-\nu}{1-2\nu} - \dfrac{\sigma'^2_{yy}}{\Gamma_4} & \\ -\dfrac{\sigma'_{xx}\sigma_{xy}}{\Gamma_4} & -\dfrac{\sigma'_{yy}\sigma_{xy}}{\Gamma_4} & \dfrac{1}{2} - \dfrac{\sigma^2_{xy}}{\Gamma_4} \end{bmatrix}$$

where , $\Gamma_4 = \dfrac{2}{3}\,\bar{\sigma}^2\left(1 + \dfrac{H}{3G}\right)$

CHAPTER 4

ALGORITHMS FOR DYNAMIC ANALYSIS

4.1 Linear Analysis

4.1.1 Introduction

In Section 3.2 of Chapter 3, equations of motion were derived for one finite element. After assembling the elements, we can write the undamped linear equations of motion for the whole system as

$$\underset{\sim}{M}\,\underset{\sim}{\ddot{D}} + \underset{\sim}{S}\,\underset{\sim}{D} = \underset{\sim}{A} \qquad (4.1)$$

where $\underset{\sim}{M}$ and $\underset{\sim}{S}$ are the mass and stiffness matrices of the assembled structure. $\underset{\sim}{D}$, $\underset{\sim}{\ddot{D}}$ and $\underset{\sim}{A}$ are the vectors of nodal displacements, nodal accelerations and nodal actions, respectively. In order to obtain the response of the structure, one must solve the set of linear second-order differential equations represented by Eq. (4.1). In the following sections, four different algorithms are presented for solving these equations.

4.1.2 Direct Linear Extrapolation with Trapezoidal Rule

Two versions of the direct linear extrapolation technique with the trapezoidal rule have been implemented for linear analysis. They consist of solutions for total displacements and for incremental displacements. That for total displacements calculated with uniform time steps Δt is developed as follows (subscripts indicate time stations):

The approximation formula (trapezoidal rule) for velocities is

$$\dot{\underset{\sim}{D}}_i = \dot{\underset{\sim}{D}}_{i-1} + \left(\ddot{\underset{\sim}{D}}_{i-1} + \ddot{\underset{\sim}{D}}_i\right) \Delta t/2 \tag{4.2}$$

and that for displacements is

$$\underset{\sim}{D}_i = \underset{\sim}{D}_{i-1} + \left(\dot{\underset{\sim}{D}}_{i-1} + \dot{\underset{\sim}{D}}_i\right) \Delta t/2 \tag{4.3}$$

Substitution of Eq. (4.2) into Eq. (4.3) yields

$$\underset{\sim}{D}_i = \underset{\sim}{D}_{i-1} + \dot{\underset{\sim}{D}}_{i-1}\Delta t + \left(\ddot{\underset{\sim}{D}}_{i-1} + \ddot{\underset{\sim}{D}}_i\right)(\Delta t)^2/4 \tag{4.4}$$

The linear equations of motion [see Eq. (4.1)] can be written for the i-th time station as

$$\underset{\sim}{M}\ddot{\underset{\sim}{D}}_i + \underset{\sim}{S}\,\underset{\sim}{D}_i = \underset{\sim}{A}_i \tag{4.5}$$

Solve for $\ddot{\underset{\sim}{D}}_i$ in Eq. (4.4) as follows:

$$\ddot{\underset{\sim}{D}}_i = 4\left[\underset{\sim}{D}_i - \underset{\sim}{D}_{i-1} - \dot{\underset{\sim}{D}}_{i-1}\Delta t - \ddot{\underset{\sim}{D}}_{i-1}(\Delta t)^2/4\right] \Big/ (\Delta t)^2 \tag{4.6}$$

From Eq. (4.3) we also obtain

$$\dot{\underset{\sim}{D}}_i = 2\left[\underset{\sim}{D}_i - \underset{\sim}{D}_{i-1} - \dot{\underset{\sim}{D}}_{i-1}\Delta t/2\right] \Big/ (\Delta t) \tag{4.7}$$

Substitute Eq. (4.6) into Eq. (4.5) and arrange terms into the form

$$\underset{\sim}{S}^{*}\,\underset{\sim}{D}_i = \underset{\sim}{A}_i^{*} \tag{4.8}$$

where

$$\underset{\sim}{S}^{*} = \underset{\sim}{S} + 4\underset{\sim}{M}/(\Delta t)^2 \tag{4.9}$$

and

$$\underset{\sim}{A}_i^{*} = \underset{\sim}{A}_i + 4\underset{\sim}{M}\left[\underset{\sim}{D}_{i-1} + \dot{\underset{\sim}{D}}_{i-1}\Delta t + \ddot{\underset{\sim}{D}}_{i-1}(\Delta t)^2/4\right]\Big/(\Delta t)^2 \qquad (4.10)$$

The symbol $\underset{\sim}{S}^{*}$ represents the effective stiffness matrix, and $\underset{\sim}{A}_i^{*}$ is the effective nodal load vector.

The set of linear algebraic equations (4.8) must be solved simultaneously in order to obtain the displacements $\underset{\sim}{D}_i$. In this work, Choleskey's decomposition method [54] has been employed. For linear analysis the stiffness matrix is constant with time, and only one decomposition of the effective stiffness matrix is needed at the outset of calculations. Thus, we have

$$\underset{\sim}{S}^{*} = \underset{\sim}{U}'\underset{\sim}{U} \qquad (4.11)$$

where $\underset{\sim}{U}$ is an upper triangular matrix.

The following recurrence expressions are used in each time step for direct linear extrapolation by the trapezoidal rule with solution for total displacements:

$$\underset{\sim}{Q}_{i-1} = 4\left[\underset{\sim}{D}_{i-1}/(\Delta t)^2 + \dot{\underset{\sim}{D}}_{i-1}/\Delta t + \ddot{\underset{\sim}{D}}_{i-1}/4\right] \qquad (4.12a)$$

$$\underset{\sim}{A}_i^{*} = \underset{\sim}{A}_i + \underset{\sim}{M}\,\underset{\sim}{Q}_{i-1} \qquad (4.12b)$$

$$\underset{\sim}{U}'\underset{\sim}{D}_i^{*} = \underset{\sim}{A}_i^{*} \quad \text{(forward solution)} \qquad (4.12c)$$

$$\underset{\sim}{U}\,\underset{\sim}{D}_i = \underset{\sim}{D}_i^{*} \quad \text{(backward solution)} \qquad (4.12d)$$

$$\ddot{\underset{\sim}{D}}_i = -\underset{\sim}{Q}_{i-1} + 4\,\underset{\sim}{D}_i/(\Delta t)^2 \quad \text{[from Eq. (4.6)]} \qquad (4.12e)$$

$$\dot{\underset{\sim}{D}}_i = 2(\underset{\sim}{D}_i - \underset{\sim}{D}_{i-1})/\Delta t - \dot{\underset{\sim}{D}}_{i-1} \quad \text{[from Eq. (4.7)]} \qquad (4.12f)$$

In a similar manner, we can develop expressions for the solution of incremental displacements. The equations of motion in incremental form are

$$\underset{\sim}{M}\,\Delta\underset{\sim}{\ddot{D}}_i + \underset{\sim}{S}\,\Delta\underset{\sim}{D}_i = \Delta\underset{\sim}{A}_i \tag{4.13}$$

where $\Delta\underset{\sim}{\ddot{D}}_i = \underset{\sim}{\ddot{D}}_i - \underset{\sim}{\ddot{D}}_{i-1}$, $\Delta\underset{\sim}{D}_i = \underset{\sim}{D}_i - \underset{\sim}{D}_{i-1}$ and $\Delta\underset{\sim}{A}_i = \underset{\sim}{A}_i - \underset{\sim}{A}_{i-1}$. From Eq. (4.6) we obtain the incremental accelerations as

$$\Delta\underset{\sim}{\ddot{D}}_i = 4\,\Delta\underset{\sim}{D}_i/(\Delta t)^2 - 2(\underset{\sim}{\ddot{D}}_{i-1} + 2\underset{\sim}{\dot{D}}_{i-1}/\Delta t) \tag{4.14}$$

Also, the incremental velocities are obtained from Eq. (4.7) as

$$\Delta\underset{\sim}{\dot{D}}_i = \underset{\sim}{\dot{D}}_i - \underset{\sim}{\dot{D}}_{i-1} = 2\Delta\underset{\sim}{D}_i/\Delta t - 2\underset{\sim}{\dot{D}}_{i-1} \tag{4.15}$$

Substituting Eq. (4.14) into Eq. (4.13), we find

$$\underset{\sim}{S}^{*}\Delta\underset{\sim}{D}_i = \Delta\underset{\sim}{A}_i^{*} \tag{4.16}$$

where

$$\underset{\sim}{S}^{*} = \underset{\sim}{S} + 4\underset{\sim}{M}/(\Delta t)^2 \tag{4.17}$$

and

$$\Delta\underset{\sim}{A}_i^{*} = \Delta\underset{\sim}{A}_i + 2\underset{\sim}{M}(\underset{\sim}{\ddot{D}}_{i-1} + 2\underset{\sim}{\dot{D}}_{i-1}/\Delta t) \tag{4.18}$$

In order to solve Eq. (4.16) for the incremental displacements, the effective stiffness matrix $\underset{\sim}{S}^{*}$ will be decomposed as indicated by Eq. (4.11).

The following recurrence expressions are used in each time step for direct linear extrapolation by the trapezoidal rule with solution for incremental displacements:

$$\underset{\sim}{Q}_{i-1} = 2\left(\ddot{\underset{\sim}{D}}_{i-1} + 2\dot{\underset{\sim}{D}}_{i-1}/\Delta t\right) \tag{4.19a}$$

$$\Delta\underset{\sim}{A}^{*}_{i} = \Delta\underset{\sim}{A}_{i} + \underset{\sim}{M}\,\underset{\sim}{Q}_{i-1} \tag{4.19b}$$

$$\underset{\sim}{U}'\ \Delta\underset{\sim}{D}^{*}_{i} = \Delta\underset{\sim}{A}^{*}_{i} \quad \text{(forward solution)} \tag{4.19c}$$

$$\underset{\sim}{U}\ \Delta\underset{\sim}{D}_{i} = \Delta\underset{\sim}{D}^{*}_{i} \quad \text{(backward solution)} \tag{4.19d}$$

$$\Delta\ddot{\underset{\sim}{D}}_{i} = -\underset{\sim}{Q}_{i-1} + 4\ \Delta\underset{\sim}{D}_{i}/(\Delta t)^{2} \quad \text{[from Eq. (4.14)]} \tag{4.19e}$$

$$\Delta\dot{\underset{\sim}{D}}_{i} = 2\Delta\underset{\sim}{D}_{i}/\Delta t - 2\dot{\underset{\sim}{D}}_{i-1} \quad \text{[from Eq. (4.15)]} \tag{4.19f}$$

It should be pointed out that if the mass matrix is diagonal, the matrix-vector multiplications in Eqs. (4.12b) and (4.19b) simplify to mere scaling operations. Thus, solving for displacements (either total or incremental) is more efficient than the alternative possibilities of solving for velocities or accelerations.

4.1.3 Central Difference Predictor

The second central difference of displacements with respect to time for a multi-degree-of-freedom system can be written as

$$\ddot{\underset{\sim}{D}}_{i-1} = \left(\underset{\sim}{D}_{i-2} - 2\underset{\sim}{D}_{i-1} + \underset{\sim}{D}_{i}\right)\Big/(\Delta t)^{2} \tag{4.20}$$

Solving this expression for displacements at time t_i , one obtains

$$\underset{\sim}{D}_{i} = -\underset{\sim}{D}_{i-2} + 2\ \underset{\sim}{D}_{i-1} + \ddot{\underset{\sim}{D}}_{i-1}(\Delta t)^{2} \tag{4.21}$$

From Eq. (4.5) we have

$$\ddot{\underset{\sim}{D}}_{i} = \underset{\sim}{M}^{-1}\ (\underset{\sim}{A}_{i} - \underset{\sim}{S}\ \underset{\sim}{D}_{i}) \tag{4.22}$$

By repetitive use of Eqs. (4.21) and (4.22) in each time step, one will obtain the response of the structure. However, the central difference predictor [Eq. (4.21)] is not a self-starting formula and cannot be used for the first time step. In order to start the procedure, we use a truncated Taylor series for the first time step. That is,

$$\underset{\sim}{D}_1 = \underset{\sim}{D}_o + \dot{\underset{\sim}{D}}_o \Delta t + \frac{1}{2} \ddot{\underset{\sim}{D}}_o (\Delta t)^2 \qquad (4.23)$$

4.1.4 Two-Cycle Iteration with Trapezoidal Rule

Using the trapezoidal rule for velocities and displacements [Eqs. (4.2) and (4.3)], we can develop an iterative algorithm in a predictor-corrector form. In the prediction phase, we use Euler's formula as a predictor for the velocities at the end of the first time step, as follows:

$$\left(\dot{\underset{\sim}{D}}_1 \right)_1 = \dot{\underset{\sim}{D}}_o + \ddot{\underset{\sim}{D}}_o \Delta t \qquad (4.23a)$$

where $\dot{\underset{\sim}{D}}_o$ and $\ddot{\underset{\sim}{D}}_o$ are the initial velocities and accelerations, respectively. Euler's extrapolation formula could also be used to start the iteration (as a predictor) in each subsequent time step. However, to improve the accuracy of the results, we will use the following formula (as a predictor for the velocities) to start the iteration after the first time step:

$$\left(\dot{\underset{\sim}{D}}_i \right)_1 = \dot{\underset{\sim}{D}}_{i-2} + 2\ddot{\underset{\sim}{D}}_{i-1} \Delta t \qquad (i \geq 2) \qquad (4.23b)$$

It can be shown that this formula has less local truncation error than Euler's formula [53]. Based on the predicted values of velocities, the displacements are estimated from the trapezoidal rule.

For the correction step, the trapezoidal rule for the velocities and displacements can be written as the following recurrence expressions for the j-th iteration of the i-th time step

$$\left(\dot{\underset{\sim}{D}}_i\right)_j = \underset{\sim}{Q}_{i-1} + \left(\ddot{\underset{\sim}{D}}_i\right)_{j-1} \Delta t/2 \qquad (j > 1) \qquad (4.24a)$$

$$\left(\underset{\sim}{D}_i\right)_j = \underset{\sim}{R}_{i-1} + \left(\dot{\underset{\sim}{D}}_i\right)_j \Delta t/2 \qquad (4.24b)$$

where

$$\underset{\sim}{Q}_{i-1} = \dot{\underset{\sim}{D}}_{i-1} + \ddot{\underset{\sim}{D}}_{i-1} \Delta t/2 \qquad (4.24c)$$

and

$$\underset{\sim}{R}_{i-1} = \underset{\sim}{D}_{i-1} + \dot{\underset{\sim}{D}}_{i-1} \Delta t/2 \qquad (4.24d)$$

In addition, Eq. (4.22) can be written for the j-th iteration of the i-th time step as

$$\left(\ddot{\underset{\sim}{D}}_i\right)_j = \underset{\sim}{M}^{-1}\left[\underset{\sim}{A}_i - \underset{\sim}{S}\left(\underset{\sim}{D}_i\right)_j\right] \qquad (4.24e)$$

This iterative algorithm can be used repeatedly until the displacements are calculated with the desired accuracy. However, in order to make this procedure more efficient, the number of iterations should be limited. In this study, the number of iterations is restricted to only two. In other words, a PECE algorithm (predict-evaluate-correct-evaluate) has been used. It has been shown [31] that if one uses the Euler predictor and the trapezoidal rule corrector, the characteristic equation of the PEC algorithm (predict-evaluate-correct) has one parasitic root, whereas the PECE algorithm has none. Consequently, the stability of the PECE algorithm is better than that of the PEC algorithm. Therefore, we use the PECE algorithm in this study, even though the extra evaluation is rather time consuming.

4.1.5 Normal Mode Method

The normal mode method of dynamic analysis is probably the most widely used procedure for solving linear problems. One advantage of the normal mode method is the fact that natural frequencies and mode shapes of the structure are found as well as displacements and stresses. This additional information about the characteristic behavior of the structure may also be of value to the analyst. However, this method requires solving an algebraic eigenvalue problem for free vibrations of the structure. Even though considerable progress has been made in developing efficient algorithms for the solution of the eigenvalue problem, this process is still quite time consuming.

In the normal mode method of dynamic analysis, the principal modes of vibration are used as generalized coordinates. In these coordinates the equations of motion are uncoupled, and each equation can be solved as a one-degree-of-freedom problem [53]. The assumption of normal (or principal) modes may be expressed as

$$\underset{\sim}{D}(t) = \sum_{j=1}^{N} \underset{\sim}{Z}_j \sin(\omega_j t + \varphi_j) \qquad (4.25)$$

where ω_j and φ_j are the angular frequency and phase angle of the j-th mode. The symbol $\underset{\sim}{Z}_j$ denotes a column matrix of amplitudes for the same mode.

Equations of undamped free vibrations can be written as

$$\underset{\sim}{M}\,\ddot{\underset{\sim}{D}} + \underset{\sim}{S}\,\underset{\sim}{D} = \underset{\sim}{0} \tag{4.26}$$

Substitution of the j-th term of Eq. (4.25) into Eq. (4.26) produces the following set of algebraic equations:

$$\underset{\sim}{S}\,\underset{\sim}{Z}_j = \omega_j^2\,\underset{\sim}{M}\,\underset{\sim}{Z}_j \tag{4.27}$$

This equation represents the so-called nonstandard form of the eigenvalue problem. By solving this equation, one obtains the eigenvalues (ω_j) and the eigenvectors $(\underset{\sim}{Z}_j)$. We place all of the eigenvectors column-wise into a modal matrix $\underset{\sim}{Z}$, as follows:

$$\underset{\sim}{Z} = \left[\underset{\sim}{Z}_1, \underset{\sim}{Z}_2, \ldots\ldots\ldots\ldots, \underset{\sim}{Z}_N\right] \tag{4.28}$$

In order to transform the equations of motion [Eq. (4.1)] to principal coordinates, we premultiply it by $\underset{\sim}{Z}'$ and insert the identity matrix $\underset{\sim}{I} = \underset{\sim}{Z}\,\underset{\sim}{Z}^{-1}$ after $\underset{\sim}{M}$ and $\underset{\sim}{S}$ to obtain

$$\underset{\sim}{Z}'\,\underset{\sim}{M}\,\underset{\sim}{Z}\,\underset{\sim}{Z}^{-1}\,\ddot{\underset{\sim}{D}} + \underset{\sim}{Z}'\,\underset{\sim}{S}\,\underset{\sim}{Z}\,\underset{\sim}{Z}^{-1}\,\underset{\sim}{D} = \underset{\sim}{Z}'\,\underset{\sim}{A} \tag{4.29}$$

This equation can be rewritten as

$$\underset{\sim}{M}_p\,\ddot{\underset{\sim}{D}}_p + \underset{\sim}{S}_p\,\underset{\sim}{D}_p = \underset{\sim}{A}_p \tag{4.30}$$

where the arrays in principal coordinates are

$$\underset{\sim}{M}_p = \underset{\sim}{Z}'\,\underset{\sim}{M}\,\underset{\sim}{Z} \tag{4.31}$$

$$\underset{\sim}{S}_p = \underset{\sim}{Z}'\,\underset{\sim}{S}\,\underset{\sim}{Z} \tag{4.32}$$

$$\underset{\sim}{D}_p = \underset{\sim}{Z}^{-1} \underset{\sim}{D} \qquad (4.33)$$

$$\underset{\sim}{A}_p = \underset{\sim}{Z}' \underset{\sim}{A} \qquad (4.34)$$

Taking advantage of orthogonality characteristics of eigenvectors, one can show that matrices $\underset{\sim}{M}_p$ and $\underset{\sim}{S}_p$ are diagonal arrays [53]. Consequently, Eq. (4.30) represents a set of uncoupled differential equations in principal occordinates. Each equation in these coordinates is solved by Duhamel's integral in order to obtain $\underset{\sim}{D}_p$. Finally, we will find the displacements in the original coordinates by the back-transformation

$$\underset{\sim}{D} = \underset{\sim}{Z} \underset{\sim}{D}_p \qquad (4.35)$$

Due to the high cost of finding the eigenvalues and eigenvectors, selection of an efficient algorithm is imperative in the normal mode method of dynamic analysis. Several procedures have been developed and are in common use for solving the algebraic eigenvalue problem [57,61]. If all of the eigenvalues and eigenvectors are desired, the Householder-QR method generally appears to be the most efficient procedure. In this algorithm, Householder transformations are first used to reduce the original equations to tridiagonal form, after which the QR algorithm is used to find the eigenvalues and eigenvectors [57]. The Householder-QR method has been employed in this investigation.

Some investigators have attempted to extend this procedure to nonlinear analysis [9 ,42]. However, because of the high cost of repetitive solution of the eigenvalue problem, this approach does not seem to be competitive with direct integration methods in nonlinear problems.

4.2 Implementation of Material and Geometric Nonlinearities into Dynamic Algorithms

One important consideration in nonlinear analysis is that of approximating stresses within an element. This has a great effect on the speed of the algorithms. One approach is to take the stresses as constant within an element, based on their values at the geometric center. This assumption simplifies the analysis and results in considerable saving in the overall computation time. In the isoparametric finite element formulation, one may evaluate the stresses at the numerical integration points to improve the accuracy of the results. However, evaluation of the stresses at the integration points significantly increases the cost of analysis as well as the amount of storage required. For a method to be efficient and practical, both cost and accuracy obtained must be considered. Therefore, in this investigation, these two different procedures for stress approximation have been incorporated in all nonlinear dynamic algorithms in order to study their effectiveness for different methods.

When both displacements and strains developed in the structure are small, we can ignore geometric nonlinearities. In this case, we can write the equations of motion for the original (undeformed) configuration. The formulation developed in Section 3.2 for linear analysis can be used with some modifications in order to incorporate and satisfy the constitutive law for elastoplastic analysis. When the stresses are assumed to be constant throughout the element, we calculate the effective stress in each time step (or iteration cycle) from the following relation:

$$\bar{\sigma} = \frac{1}{\sqrt{2}} \left[(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2 + 6(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2) \right]^{1/2} \tag{4.36}$$

If $\bar{\sigma}$ is less than σ_s, the element is elastic, and the elastic stress-strain relations derived in Section 3.2.2 are used to evaluate the tangential stiffness matrix in Eq. (3.44). However, if $\bar{\sigma} > \sigma_s$, the stresses are modified by an averaging scheme, as follows:

$$\sigma_{\text{modified}} = \sigma_{\text{previous}} + \frac{1}{2} \Delta\sigma \tag{4.37}$$

The next time stresses are evaluated, the plastic stress-strain matrix $\underset{\sim}{D}^p$ given in Section 3.3.6 is used to calculate the incremental stresses

$$\underset{\sim}{\Delta\sigma} = \underset{\sim}{D}^p \underset{\sim}{\Delta\epsilon} \qquad \begin{matrix}(3.148)\\ \text{(repeated)}\end{matrix}$$

In addition, $\underset{\sim}{D}^p$ is used instead of the elastic stress-strain matrix $\underset{\sim}{T}_3$ in Eq. (3.44) to evaluate the element stiffness matrix.

When plastification of the element continues beyond the yield limit, we must take into account the strain hardening behavior of the material. To do so, we save the maximum previous effective stress for each element and compare it against the current effective stress. In this case, if $\bar{\sigma} > \bar{\sigma}_{max}$, plastic loading is indicated. An updated plastic stress-strain matrix $\underset{\sim}{D}^p$ must be used in the subsequent time step (or iteration cycle) in place of $\underset{\sim}{T}_3$ to evaluate both the element stiffness matrix and the incremental stresses. However, if $\bar{\sigma} < \bar{\sigma}_{max}$, elastic unloading has occurred. In this case the elastic stress-strain matrix of Section 3.2.2 must be used to evaluate the stiffness matrix and incremental stresses.

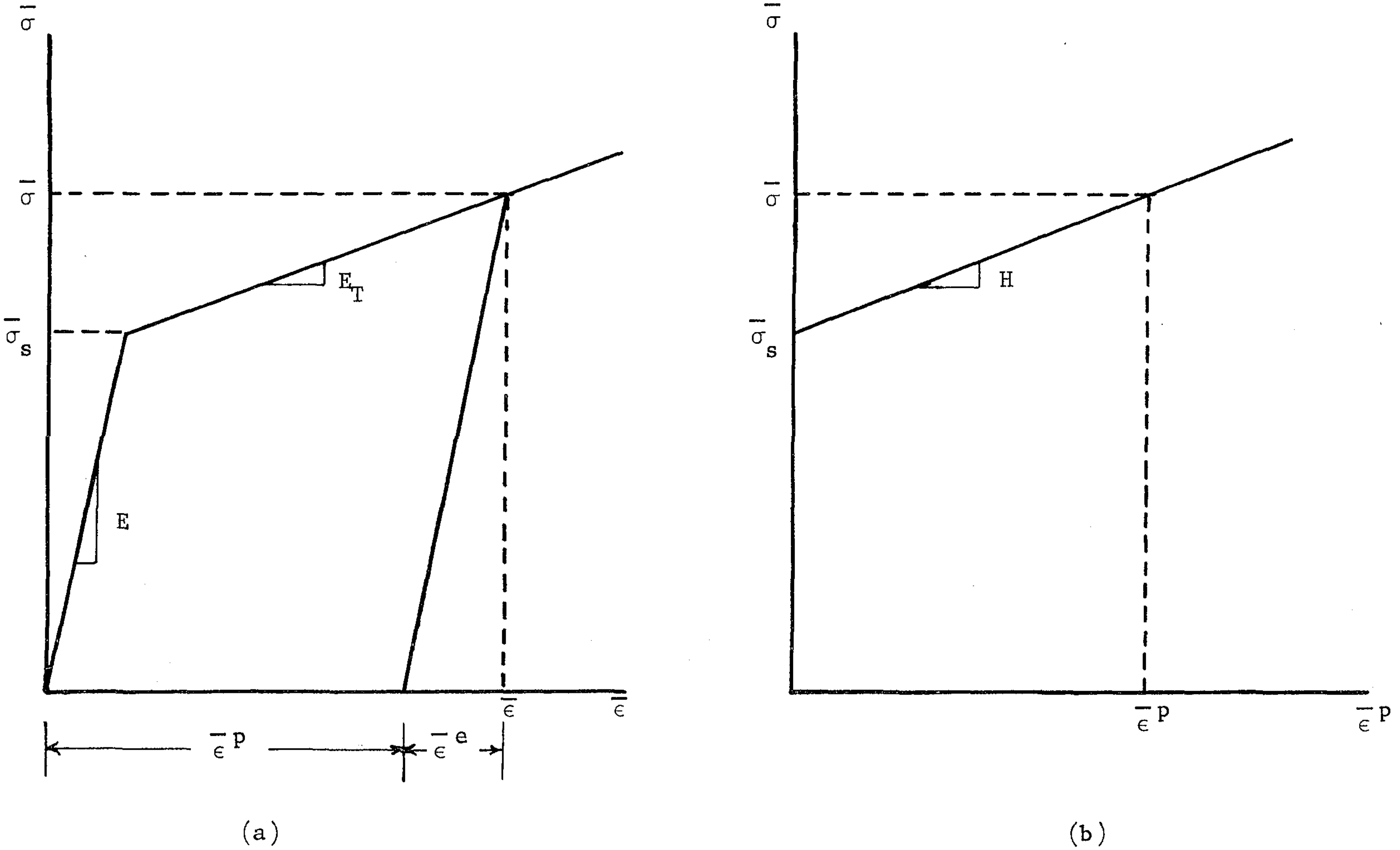
A similar procedure is adopted whenever stresses are evaluated at the four spatial integration points. In this case, the stresses $\underset{\sim}{\sigma}$, the effective stress $\bar{\sigma}$, and the maximum effective stress $\bar{\sigma}_{max}$ are calculated and saved at each of the integration points. Based on these values, the plastic stress-strain matrix $\underset{\sim}{D}^p$ is also evaluated at each of the integration points. The storage required will increase substantially because several stress arrays must be saved at each integration point.

In this investigation, a bilinear effective stress-effective strain diagram has been assumed, as shown in Fig. 4.1(a). This simple model can adequately approximate the elastoplastic behavior of most metals and can be determined experimentally. Figure 4.1(b) shows the effective stress-effective plastic strain diagram. We can find a simple relationship between the tangent modulus E_T in Fig. 4.1(a) and the plastic modulus H in Fig. 4.1(b). From Fig. 4.1(a) we can write

$$\bar{\sigma} = \bar{\sigma}_s + E_T\left(\bar{\epsilon} - \frac{\bar{\sigma}_s}{E}\right) \qquad (4.38)$$

and

$$\bar{\epsilon} = \bar{\epsilon}^p + \bar{\epsilon}^e = \bar{\epsilon}^p + \frac{\bar{\sigma}}{E} \qquad (4.39)$$



(a)

(b)

Fig. 4.1 Bilinear effective stress-effective strain diagrams

where the superscripts e and p indicate the elastic and plastic components of strain. In a uniaxial tension test, $\bar{\sigma}_s$ is equal to the yield stress ($\bar{\sigma}_s = \sigma_s$) . Similarly, from Fig. 4.1(b), we obtain

$$\bar{\sigma} = \bar{\sigma}_s + H \bar{\epsilon}^p \qquad (4.40)$$

By substituting Eq. (4.39) into Eq. (4.38) and solving for $\bar{\sigma} - \bar{\sigma}_s$, we find

$$\bar{\sigma} - \bar{\sigma}_s = \frac{E_T \bar{\epsilon}^p}{1 - \frac{E_T}{E}} \qquad (4.41)$$

Substituting for $\bar{\sigma}$ in Eq. (4.41) from Eq. (4.40), we get the following relationship between the plastic modulus H and the tangent modulus E_T :

$$H = \frac{E_T}{1 - \frac{E_T}{E}} \qquad (4.42)$$

The inverse relationship is

$$E_T = \frac{H}{1 + \frac{H}{E}} \qquad (4.43)$$

In the computer program for nonlinear analysis, only the tangent modulus E_T need be given as input. The plastic modulus H , which is required in the plastic stress-strain matrix $\underset{\sim}{D}^p$ (see Table 3.2), is computed from Eq. (4.42).

As in elastic analysis, matrix multiplications have been avoided in calculating the element stiffness matrix and the equivalent nodal loads. This has been accomplished by the efficient formulation described in Appendix A.

When geometric nonlinearities are to be taken into account, the equations of motion must be written with respect to the deformed configuration; and we no longer can use the linear equations derived in Section 3.2. A more rigorous formulation of the equations of motion is necessary (see Section 3.3) for the very general large displacement and large strain analysis. In this case, the strain-displacement relationships are complicated, but the implementation of material nonlinearity is no more difficult than that described for small-displacement elastoplastic analysis in the previous paragraphs. Matrix multiplications have been avoided in calculating the geometric stiffness matrix $\underset{\sim}{K}_G$ by the formulation developed in Section 3.3.5. However, due to the complexity of the strain-displacement relationships, the matrix multiplications for calculating the element stiffness matrix $\underset{\sim}{K}_L$ [see Eq. (3.111)] cannot be carried out explicitly. Nevertheless, when the stresses are calculated only at the geometric centers of the elements, it is possible to improve the efficiency of the stiffness calculations. For this purpose, we decompose the matrix $\underset{\sim}{T}_3$ into the product of a lower triangular matrix and an upper triangular matrix, each of which is the transpose of the other [54],

$$\underset{\sim}{T}_3 = \underset{\sim}{U}' \underset{\sim}{U} \qquad (4.44)$$

where $\underset{\sim}{U}$ is the upper triangular matrix. Substitution of Eq. (4.44) into Eq. (3.111) yields

$$\underset{\sim}{K}_L = h \int_{-1}^{1} \int_{-1}^{1} (\underset{\sim}{U}\,\underset{\sim}{B}_L)' \, (\underset{\sim}{U}\,\underset{\sim}{B}_L) \, |\underset{\sim}{J}| \, d\xi d\eta \tag{4.45}$$

Thus, the matrix multiplication in Eq. (3.111) is replaced by decomposition of matrix $\underset{\sim}{T}_3$, scaling and combining rows of $\underset{\sim}{B}_L$, and calculating only the upper triangular part of the product $\underset{\sim}{B}_L' \, \underset{\sim}{T}_3 \, \underset{\sim}{B}_L$. Whenever stresses are assumed constant within the element, only one decomposition is needed to evaluate the element stiffness matrix $\underset{\sim}{K}_L$. As a result, this technique reduces the cost of stiffness computations. However, when stresses are computed at the four spatial integration points, four decompositions of matrix $\underset{\sim}{T}_3$ are needed for each element. This offsets the above savings in computations. Consequently, this technique has been implemented only when the stresses are assumed constant within the element. In such a case, the elastic stress-strain matrix derived in Section 3.2.2 is decomposed and saved at the outset of the calculations. This is expedient in order to avoid repetitive decomposition of this matrix when the loading is elastic or when unloading occurs.

4.3 Nonlinear Analysis by Implicit Methods

4.3.1 Introduction

In the following sections we present three algorithms for nonlinear analysis by implicit procedures. They are the Newmark-Beta method, the Houbolt procedure, and Park's stiffly-stable method. The common feature of all three algorithms is that in each time step the stiffness matrix must be formed, and the incremental displacements are found by solving a set of linear algebraic equations. Therefore, an efficient method for solving such equations is crucial for the implicit algorithms. In this investigation, Cholesky's square root method has been used, as described in Section 4.1.2.

In Chapter 3, the equations of motion for a single element were derived using a superscript t to denote time. However, for analyzing the whole structure, in this chapter we use a subscript i to denote the time stations. From Eq. (3.106) we can write the incremental equations of motion for the structure as

$$\underset{\sim}{S}_{i-1} \, \underset{\sim}{\Delta D}_{i} = \underset{\sim}{A}_{i} - \underset{\sim}{F}_{i-1} - \underset{\sim}{M} \, \underset{\sim}{\ddot{D}}_{i} \tag{4.46}$$

where $\underset{\sim}{F}_{i-1}$ is the vector of internal loads for the assembled structure (to be defined in the following paragraph) and $\underset{\sim}{S}_{i-1}$ is called the tangential stiffness matrix.

When geometric nonlinearities are to be taken into account, the tangential stiffness matrix and the internal force vector are evaluated by the following relations:

$$\underset{\sim}{S}_{i-1} = \sum_{j=1}^{n_e} \left(\underset{\sim}{K}_L + \underset{\sim}{K}_G \right)_j \tag{4.47}$$

$$\underset{\sim}{F}_{i-1} = \sum_{j=1}^{n_e} \left(\int_{V^o} \underset{\sim}{B}_L^{\prime} \, \underset{\sim}{\sigma}_{i-1} dV^o \right)_j \tag{4.48}$$

where n_e is the number of elements. The arrays $\underset{\sim}{K}_L$, $\underset{\sim}{K}_G$ and $\underset{\sim}{B}_L$ were defined in Section 3.3.5.

However, when only the nonlinearities due to elasto-plastic behavior of the material are to be considered, we use the following relations developed in Section 3.2 for linear analysis:

$$\underset{\sim}{S}_{i-1} = \sum_{j=1}^{n_e} (K)_j \tag{4.49}$$

$$\underset{\sim}{F}_{i-1} = \sum_{j=1}^{n_e} \left(\int_V \underset{\sim}{B}' \, \underset{\sim}{\sigma}_{i-1} dV \right)_j \tag{4.50}$$

where $\underset{\sim}{B}$ is the linear strain-displacement matrix given in Section 3.2.3. The matrix $\underset{\sim}{K}$ is also given in Section 3.2.3 [Eq. (3.43)], but now the elasto-plastic behavior of the materials will be accounted for through the constitutive matrix $\underset{\sim}{T}_3$, as described in Section 4.2.

The nonlinear equation (4.46) can be solved in a variety of ways, as follows:

1) Newton-Raphson iterative procedure.

The Newton-Raphson procedure is a well-known method for the solution of nonlinear algebraic equations [17]. In applying this method, we form the tangential stiffness matrix in each cycle of iteration. Based on the updated stiffness matrix, the incremental displacements are computed from Eq. (4.46). They are added to the displacements at the end of the previous time step in order to find the new displacements. This process is repeated until the desired convergence is obtained.

2) Constant stiffness iterative procedure.

In the constant stiffness iterative procedure, the stiffness matrix is formed only once in each time step. This may be considered as a special case of the Newton-Raphson method.

3) Incremental procedure with load correction

In the incremental procedure with load correction, Eq. (4.46) is used only once in each time step, and no iterations are carried out.

4) Incremental procedure

Equation (4.46) can be written in the following form:

$$\underset{\sim}{M}\,\Delta\ddot{\underset{\sim}{D}}_i + \underset{\sim}{S}_{i-1}\,\Delta\underset{\sim}{D}_i = \Delta\underset{\sim}{A}_i + \left(\underset{\sim}{A}_{i-1} - \underset{\sim}{F}_{i-1} - \underset{\sim}{M}\,\ddot{\underset{\sim}{D}}_{i-1}\right) \qquad (4.51)$$

In the incremental procedure, the terms inside the parentheses are omitted. Thus, Eq. (4.51) reduces to

$$\underset{\sim}{M}\,\Delta\ddot{\underset{\sim}{D}}_i + \underset{\sim}{S}_{i-1}\,\Delta\underset{\sim}{D}_i = \Delta\underset{\sim}{A}_i \qquad (4.52)$$

Evaluation of the tangential stiffness matrix and its decomposition is the most time consuming part of the nonlinear analysis. Also, stresses need to be updated in each cycle, which requires many computations. Because of this costly process, the Newton-Raphson iterative procedure does not seem to be promising for multi-degree systems [6,24]. Therefore, in this study, we have utilized only the second and third approaches discussed above.

4.3.2 Newmark-Beta Method

In the Newmark-Beta method, the velocities and displacements at the end of each time step are obtained by the following relations (see Section 2.3):

$$\dot{\underset{\sim}{D}}_i = \dot{\underset{\sim}{D}}_{i-1} + \gamma\Delta t\,\ddot{\underset{\sim}{D}}_i + (1-\gamma)\Delta t\,\ddot{\underset{\sim}{D}}_{i-1} \qquad (4.53)$$

$$\underset{\sim}{D}_i = \underset{\sim}{D}_{i-1} + \beta(\Delta t)^2\ddot{\underset{\sim}{D}}_i + (0.5-\beta)(\Delta t)^2\ddot{\underset{\sim}{D}}_{i-1} + \Delta t\,\dot{\underset{\sim}{D}}_{i-1} \qquad (4.54)$$

From Eq. (4.54), we find the incremental displacements

$$\Delta \underset{\sim}{D}_i = \underset{\sim}{D}_i - \underset{\sim}{D}_{i-1} = \beta(\Delta t)^2 \ddot{\underset{\sim}{D}}_i + (0.5 - \beta)(\Delta t)^2 \ddot{\underset{\sim}{D}}_{i-1} + \Delta t \, \dot{\underset{\sim}{D}}_{i-1} \quad (4.55)$$

Solve for $\ddot{\underset{\sim}{D}}_i$ from Eq. (4.55)

$$\ddot{\underset{\sim}{D}}_i = \frac{1}{\beta(\Delta t)^2} \Delta \underset{\sim}{D}_i - \frac{1}{\beta \Delta t} \dot{\underset{\sim}{D}}_{i-1} + \left(1 - \frac{1}{2\beta}\right) \ddot{\underset{\sim}{D}}_{i-1} \quad (4.56)$$

Substitute Eq. (4.56) into Eq. (4.46) to obtain

$$\left[\underset{\sim}{S}_{i-1} + \frac{1}{\beta(\Delta t)^2} \underset{\sim}{M}\right] \Delta \underset{\sim}{D}_i = \underset{\sim}{A}_i - \underset{\sim}{F}_{i-1} + \underset{\sim}{M}\left[\frac{1}{\beta \Delta t} \dot{\underset{\sim}{D}}_{i-1} + \left(\frac{1}{2\beta} - 1\right) \ddot{\underset{\sim}{D}}_{i-1}\right]$$

(4.57)

which can be solved for $\Delta \underset{\sim}{D}_i$. Equation (4.57) can be written as

$$\underset{\sim}{S}^* \Delta \underset{\sim}{D}_i = \Delta \underset{\sim}{A}_i^* \quad (4.58)$$

where $\underset{\sim}{S}^*$ and $\Delta \underset{\sim}{A}_i^*$ are given in Table 4.1. In order to find velocities, substitute Eq. (4.56) into Eq. (4.53)

$$\dot{\underset{\sim}{D}}_i = \left(1 - \frac{\gamma}{\beta}\right) \dot{\underset{\sim}{D}}_{i-1} + \frac{\gamma}{\beta \Delta t} \Delta \underset{\sim}{D}_i + \Delta t \left(1 - \frac{\gamma}{2\beta}\right) \ddot{\underset{\sim}{D}}_{i-1} \quad (4.59)$$

By employing Eqs. (4.56), (4.58) and (4.59), we write an algorithm for nonlinear dynamic analysis. This algorithm (with equilibrium iterations) is presented in Table 4.1.

4.3.3 Houbolt Method

In the Houbolt method, the nodal accelerations are given by the following expression (see Section 2.4):

TABLE 4.1 Algorithm for Newmark-Beta Method with Equilibrium Iterations for Nonlinear Analysis

For each time step do the following:

1. Calculate the effective stiffness matrix

$$\underset{\sim}{S}^{*} = \underset{\sim}{S}_{i-1} + \frac{1}{\beta(\Delta t)^{2}} \underset{\sim}{M}$$

2. Calculate the effective incremental load vector

$$\underset{\sim}{\Delta A}_{i}^{*} = \underset{\sim}{A}_{i} - \underset{\sim}{F}_{i-1} + \underset{\sim}{M}\,\underset{\sim}{Q}_{i-1}$$

where $\underset{\sim}{Q}_{i-1}$ is computed in the previous step.

(For the first time step $\underset{\sim}{F}_{o} = \underset{\sim}{0}$ and $\underset{\sim}{Q}_{o} = \frac{1}{\beta\Delta t} \dot{\underset{\sim}{D}}_{o} + (\frac{1}{2\beta} - 1)\ddot{\underset{\sim}{D}}_{o}$)

3. Decompose the effective stiffness matrix

$$\underset{\sim}{S}^{*} = \underset{\sim}{U}' \underset{\sim}{U}$$

4. Solve for the incremental displacements

$$\underset{\sim}{U}' \underset{\sim}{\Delta D}_{i}^{*} = \underset{\sim}{\Delta A}_{i}^{*} \qquad \text{(forward solution)}$$

$$\underset{\sim}{U} \underset{\sim}{\Delta D}_{i} = \underset{\sim}{\Delta D}_{i}^{*} \qquad \text{(backward solution)}$$

5. Compute displacements and accelerations

$$\underset{\sim}{D}_{i} = \underset{\sim}{D}_{i-1} + \underset{\sim}{\Delta D}_{i}$$

$$\ddot{\underset{\sim}{D}}_{i} = \frac{1}{\beta(\Delta t)^{2}} \underset{\sim}{\Delta D}_{i} - \underset{\sim}{Q}_{i-1}$$

6. If equilibrium iteration is not considered, go to step 12; otherwise, set $j = 0$ and continue.

7. Start the jth iteration

$$j \leftarrow j + 1$$

TABLE 4.1 (Continued)

8. Compute the vector of residual (or out-of-balance) loads

$$\hat{\underset{\sim}{A}}{}_i^j = \underset{\sim}{A}_i - \underset{\sim}{M}\,\ddot{\underset{\sim}{D}}{}_i^{j-1} - \underset{\sim}{F}{}_i^{j-1}$$

9. Solve for the jth correction to the displacement increments

$$\underset{\sim}{U}'\,\Delta(\Delta\underset{\sim}{D}_i)^* = \hat{\underset{\sim}{A}}{}_i^j$$

$$\underset{\sim}{U}\,\Delta(\Delta\underset{\sim}{D}_i)^j = \Delta(\Delta\underset{\sim}{D}_i)^*$$

10. Calculate new displacements and accelerations

$$\Delta\underset{\sim}{D}{}_i^j = \Delta\underset{\sim}{D}{}_i^{j-1} + \Delta(\Delta\underset{\sim}{D}_i)^j$$

$$\underset{\sim}{D}{}_i^j = \underset{\sim}{D}_{i-1} + \Delta\underset{\sim}{D}{}_i^j = \underset{\sim}{D}{}_i^{j-1} + \Delta(\Delta\underset{\sim}{D}_i)^j$$

$$\ddot{\underset{\sim}{D}}{}_i^j = -\underset{\sim}{Q}_{i-1} + \frac{1}{\beta(\Delta t)^2}\,\Delta\underset{\sim}{D}{}_i^j$$

11. Check iteration convergence

If $\dfrac{\|\Delta(\Delta\underset{\sim}{D}_i)^j\|}{\|\underset{\sim}{D}{}_i^j\|} \leq$ tolerance, go to step 12

Otherwise, if $\begin{cases} j \geq \text{MNIT} & \text{, go to step 12} \\ j < \text{MNIT} & \text{, go to step 7} \end{cases}$

where, MNIT = maximum number of iterations

12. Calculate velocities and the array $\underset{\sim}{Q}_i$

$$\dot{\underset{\sim}{D}}_i = \left(1 - \frac{\gamma}{\beta}\right)\dot{\underset{\sim}{D}}_{i-1} + \Delta t\left(1 - \frac{\gamma}{2\beta}\right)\ddot{\underset{\sim}{D}}_{i-1} + \frac{\gamma}{\beta\Delta t}\,\Delta\underset{\sim}{D}{}_i^j$$

$$\underset{\sim}{Q}_i = \frac{1}{\beta\Delta t}\,\dot{\underset{\sim}{D}}_i + \left(\frac{1}{2\beta} - 1\right)\ddot{\underset{\sim}{D}}_i$$

$$\ddot{\underset{\sim}{D}}_i = (-\underset{\sim}{D}_{i-3} + 4\underset{\sim}{D}_{i-2} - 5\underset{\sim}{D}_{i-1} + 2\underset{\sim}{D}_i)/(\Delta t)^2 \qquad (4.60)$$

Substitute Eq. (4.60) into Eq. (4.46) and solve for the incremental displacements in the resulting expression

$$\left[\underset{\sim}{S}_{i-1} + \frac{2}{(\Delta t)^2}\underset{\sim}{M}\right]\Delta\underset{\sim}{D}_i = \underset{\sim}{A}_i - \underset{\sim}{F}_{i-1} + \frac{1}{(\Delta t)^2}\underset{\sim}{M}\,(\underset{\sim}{D}_{i-3} - 4\underset{\sim}{D}_{i-2} + 3\underset{\sim}{D}_{i-1}) \qquad (4.61)$$

This equation can be written in the form of Eq. (4.58), for which $\underset{\sim}{S}^*$ and $\Delta\underset{\sim}{A}_i^*$ are given in Table 4.2. By repetitive use of Eqs. (4.60) and (4.61) we can find the displacements in each time step (see Table 4.2).

The Houbolt method is not self-starting, so a special procedure must be used to find the displacements for the first two time steps. For this purpose, we use the Newmark-Beta method with $\beta = 0.25$ and $\gamma = 0.5$.

4.3.4 Park's Stiffly-Stable Method

Park presented the following difference equations for velocities (see Section 2.7):

$$\dot{\underset{\sim}{D}}_i = (20\underset{\sim}{D}_i - 30\underset{\sim}{D}_{i-1} + 12\underset{\sim}{D}_{i-2} - 2\underset{\sim}{D}_{i-3})/(12\Delta t) \qquad (4.62)$$

We use the same formula to find the accelerations, as follows:

$$\ddot{\underset{\sim}{D}}_i = (20\dot{\underset{\sim}{D}}_i - 30\dot{\underset{\sim}{D}}_{i-1} + 12\dot{\underset{\sim}{D}}_{i-2} - 2\dot{\underset{\sim}{D}}_{i-3})/(12\Delta t) \qquad (4.63)$$

Equation (4.62) can also be written in terms of the incremental displacements $\Delta\underset{\sim}{D}_i$ as

$$\dot{\underset{\sim}{D}}_i = (20\Delta\underset{\sim}{D}_i - 10\underset{\sim}{D}_{i-1} + 12\underset{\sim}{D}_{i-2} - 2\underset{\sim}{D}_{i-3})/(12\Delta t) \qquad (4.64)$$

TABLE 4.2 Algorithm for Houbolt Method with Equilibrium Iterations for Nonlinear Analysis

For each time step (after the first two steps), do the following:

1. Calculate the effective stiffness matrix

$$\underset{\sim}{S}^{*} = \underset{\sim}{S} + \frac{2}{(\Delta t)^2}\underset{\sim}{M}$$

2. Calculate the effective incremental load vector

$$\Delta\underset{\sim}{A}_i^{*} = \underset{\sim}{A}_i - \underset{\sim}{F}_{i-1} + \frac{1}{(\Delta t)^2}\underset{\sim}{M}\,(\underset{\sim}{D}_{i-3} - 4\underset{\sim}{D}_{i-2} + 3\underset{\sim}{D}_{i-1})$$

3. Decompose the effective stiffness matrix

$$\underset{\sim}{S}^{*} = \underset{\sim}{U}'\,\underset{\sim}{U}$$

4. Solve for incremental displacements

$$\underset{\sim}{U}'\,\Delta\underset{\sim}{D}_i^{*} = \Delta\underset{\sim}{A}_i^{*} \qquad \text{(forward solution)}$$

$$\underset{\sim}{U}\,\Delta\underset{\sim}{D}_i = \Delta\underset{\sim}{D}_i^{*} \qquad \text{(backward solution)}$$

5. Compute displacements and accelerations

$$\underset{\sim}{D}_i = \underset{\sim}{D}_{i-1} + \Delta\underset{\sim}{D}_i$$

$$\ddot{\underset{\sim}{D}}_i = \frac{2}{(\Delta t)^2}\underset{\sim}{D}_i + \underset{\sim}{Q}_{i-1}$$

where $\underset{\sim}{Q}_{i-1}$ is computed in the previous step.

6. If equilibrium iteration is not considered, go to step 12; otherwise, set $j = 0$ and continue.

TABLE 4.2 (Continued)

7. Start the jth iteration

$$j \leftarrow j + 1$$

8. Compute the vector of residual (or out-of-balance) loads

$$\underset{\sim}{\hat{A}}{}_i^j = \underset{\sim}{A}_i - \underset{\sim}{M}\,\underset{\sim}{\ddot{D}}{}_i^{j-1} - \underset{\sim}{F}{}_i^{j-1}$$

9. Solve for the jth correction to the displacement increments

$$\underset{\sim}{U}'\,\Delta(\Delta\underset{\sim}{D}_i)^* = \underset{\sim}{\hat{A}}{}_i^j$$

$$\underset{\sim}{U}\,\Delta(\Delta\underset{\sim}{D}_i)^j = \Delta(\Delta\underset{\sim}{D}_i)^*$$

10. Calculate new displacements and accelerations

$$\Delta\underset{\sim}{D}{}_i^j = \Delta\underset{\sim}{D}{}_i^{j-1} + \Delta(\Delta\underset{\sim}{D}_i)^j$$

$$\underset{\sim}{D}{}_i^j = \underset{\sim}{D}_{i-1} + \Delta\underset{\sim}{D}{}_i^j = \underset{\sim}{D}{}_i^{j-1} + \Delta(\Delta\underset{\sim}{D}_i)^j$$

$$\underset{\sim}{\ddot{D}}{}_i^j = \frac{2}{(\Delta t)^2}\,\underset{\sim}{D}{}_i^j + \underset{\sim}{Q}_{i-1}$$

11. Check iteration convergence

$$\text{if} \quad \frac{\|\Delta(\Delta\underset{\sim}{D}_i)^j\|}{\|\underset{\sim}{D}{}_i^j\|} \leq \text{tolerance, go to step 12}$$

$$\text{Otherwise, if} \quad \begin{cases} j \geq \text{MNIT} & \text{, go to step 12} \\ j < \text{MNIT} & \text{, go to step 7} \end{cases}$$

12. Calculate the array $\underset{\sim}{Q}_i$

$$\underset{\sim}{Q}_i = (-\,\underset{\sim}{D}_{i-2} + 4\underset{\sim}{D}_{i-1} - 5\underset{\sim}{D}_i)/(\Delta t)^2 \; .$$

Substitution of this expression for $\underset{\sim}{\dot{D}}_i$ into Eq. (4.63) yields

$$\underset{\sim}{\ddot{D}}_i = \frac{5}{36(\Delta t)^2}\left(20\Delta\underset{\sim}{D}_i - 10\underset{\sim}{D}_{i-1} + 12\underset{\sim}{D}_{i-2} - 2\underset{\sim}{D}_{i-3}\right) + \frac{1}{12\Delta t}\left(-30\underset{\sim}{\dot{D}}_{i-1} + 12\underset{\sim}{\dot{D}}_{i-2} - 2\underset{\sim}{\dot{D}}_{i-3}\right) \qquad (4.65)$$

Substitute $\underset{\sim}{\ddot{D}}_i$ from Eq. (4.65) into Eq. (4.46) to obtain

$$\left[\underset{\sim}{S}_{i-1} + \left(\frac{5}{3\Delta t}\right)^2 \underset{\sim}{M}\right]\Delta\underset{\sim}{D}_i = \underset{\sim}{A}_i - \underset{\sim}{F}_{i-1} + \underset{\sim}{M}\left[\frac{5}{18(\Delta t)^2}\left(5\underset{\sim}{D}_{i-1} - 6\underset{\sim}{D}_{i-2} + \underset{\sim}{D}_{i-3}\right) + \frac{1}{6\Delta t}\left(15\underset{\sim}{\dot{D}}_{i-1} - 6\underset{\sim}{\dot{D}}_{i-2} + \underset{\sim}{\dot{D}}_{i-3}\right)\right] \qquad (4.66)$$

From this equation, the incremental displacements $\Delta\underset{\sim}{D}_i$ can be found. Equation (4.66) is of the same form as Eq. (4.58) for which $\underset{\sim}{S}^*$ and $\Delta\underset{\sim}{A}_i^*$ are given in Table 4.3. Like the Houbolt method, this procedure is not self-starting. Therefore, we use the Newmark-Beta method for the first two time steps.

4.4 Nonlinear Analysis by Explicit Methods

4.4.1 Introduction

In the following sections, we present three algorithms for nonlinear analysis by explicit procedures. They are the central difference method, two-cycle iteration with the trapezoidal rule, and the fourth-order Runge-Kutta method. For all of these procedures, we adopt a nodewise solution technique. In this approach only two rows of the

TABLE 4.3 Algorithm for Park Stiffly-Stable Method with Equilibrium Iterations for Nonlinear Analysis

For each time step (after the first two steps), do the following:

1. Calculate the effective stiffness matrix

$$\underset{\sim}{S}^{*} = \underset{\sim}{S} + \left(\frac{5}{3\Delta t}\right)^{2} \underset{\sim}{M}$$

2. Calculate the effective incremental load vector

$$\underset{\sim}{\Delta A}_{i}^{*} = \underset{\sim}{A}_{i} - \underset{\sim}{F}_{i-1} + \underset{\sim}{M}\left[\frac{5}{18(\Delta t)^{2}}\left(5\underset{\sim}{D}_{i-1} - 6\underset{\sim}{D}_{i-2} + \underset{\sim}{D}_{i-3}\right) + \frac{1}{6\Delta t}\left(15\underset{\sim}{\dot{D}}_{i-1} - 6\underset{\sim}{\dot{D}}_{i-2} + \underset{\sim}{\dot{D}}_{i-3}\right)\right]$$

3. Decompose the effective stiffness matrix

$$\underset{\sim}{S}^{*} = \underset{\sim}{U}'\underset{\sim}{U}$$

4. Solve for the incremental displacements

$$\underset{\sim}{U}' \, \underset{\sim}{\Delta D}_{i}^{*} = \underset{\sim}{\Delta A}_{i}^{*} \qquad \text{(forward solution)}$$

$$\underset{\sim}{U} \, \underset{\sim}{\Delta D}_{i} = \underset{\sim}{\Delta D}_{i}^{*} \qquad \text{(backward solution)}$$

5. Compute displacements, velocities, and accelerations

$$\underset{\sim}{D}_{i} = \underset{\sim}{D}_{i-1} + \underset{\sim}{\Delta D}_{i}$$

$$\underset{\sim}{\dot{D}}_{i} = \left(\frac{5}{3\Delta t}\right)\underset{\sim}{\Delta D}_{i} + \underset{\sim}{R}_{i-1}$$

$$\underset{\sim}{\ddot{D}}_{i} = \left(\frac{5}{3\Delta t}\right)\underset{\sim}{\dot{D}}_{i} + \underset{\sim}{Q}_{i-1}$$

where $\underset{\sim}{R}_{i-1}$ and $\underset{\sim}{Q}_{i-1}$ are computed in the previous step.

6. If equilibrium iteration is not considered, go to step 12; otherwise, set $j = 0$ and continue.

7. Start the jth iteration

$$j \leftarrow j + 1$$

TABLE 4.3 (Continued)

8. Compute the vector of residual (or out-of-balance) loads

$$\underset{\sim}{\hat{A}}{}_i^j = \underset{\sim}{A}{}_i - \underset{\sim}{M}\,\underset{\sim}{\ddot{D}}{}_i^{j-1} - \underset{\sim}{F}{}_i^{j-1}$$

9. Solve for the jth correction to the displacement increments

$$\underset{\sim}{U}'\,\Delta(\Delta\underset{\sim}{D}{}_i)^* = \underset{\sim}{\hat{A}}{}_i^j$$

$$\underset{\sim}{U}\,\Delta(\Delta\underset{\sim}{D}{}_i)^j = \Delta(\Delta\underset{\sim}{D}{}_i)^*$$

10. Calculate new displacements, velocities, and accelerations

$$\Delta\underset{\sim}{D}{}_i^j = \Delta\underset{\sim}{D}{}_i^{j-1} + \Delta(\Delta\underset{\sim}{D}{}_i)^j$$

$$\underset{\sim}{D}{}_i^j = \underset{\sim}{D}{}_{i-1} + \Delta\underset{\sim}{D}{}_i^j = \underset{\sim}{D}{}_i^{j-1} + \Delta(\Delta\underset{\sim}{D}{}_i)^j$$

$$\underset{\sim}{\dot{D}}{}_i^j = \frac{5}{3\Delta t}(\Delta\underset{\sim}{D}{}_i)^j + \underset{\sim}{R}{}_{i-1}$$

$$\underset{\sim}{\ddot{D}}{}_i^j = \frac{5}{3\Delta t}\underset{\sim}{\dot{D}}{}_i^j + \underset{\sim}{Q}{}_{i-1}$$

11. Check iteration convergence

$$\text{if} \quad \frac{\|\Delta(\Delta\underset{\sim}{D}{}_i)^j\|}{\|\underset{\sim}{D}{}_i^j\|} \leq \text{tolerance} \quad , \quad \text{go to step 12}$$

$$\text{Otherwise if} \quad \begin{cases} j \geq \text{MNIT} \quad , & \text{go to step 12} \\ j < \text{MNIT} \quad , & \text{go to step 7} \end{cases}$$

12. Calculate the arrays $\underset{\sim}{Q}{}_i$ and $\underset{\sim}{R}{}_i$

$$\underset{\sim}{Q}{}_i = (-\,30\,\underset{\sim}{\dot{D}}{}_i + 12\underset{\sim}{\dot{D}}{}_{i-1} - 2\underset{\sim}{\dot{D}}{}_{i-2})/(12\Delta t)$$

$$\underset{\sim}{R}{}_i = (-\,10\underset{\sim}{D}{}_i + 12\underset{\sim}{D}{}_{i-1} - 2\underset{\sim}{D}{}_{i-2})/(12\Delta t)$$

tangential stiffness matrix need be generated at any stage in the analysis. Consequently, large problems can be analyzed within the fast core of the computer because it is not necessary to solve simultaneous equations. This fact makes the explicit methods more attractive than the implic procedures. In all three explicit procedures, the mass matrix must be inverted. Thus, the use of a lumped mass matrix makes these methods more efficient. Implementation of the nodewise approach into a computer program will be discussed in Chapter 5.

4.4.2 Central Difference Predictor

The central difference formula for a multi-degree system was formulated in Section 4.1.3, and an algorithm for applying it nodewise is given in Table 4.4. At each node of the structure, two rows of the tangential stiffness matrix (corresponding to two degrees of freedom at that node) are generated. Then the incremental accelerations pertaining to the two degrees of freedom are calculated from Eq. (4.52). This process is repeated for all nodes.

4.4.3 Two-Cycle Iteration with Trapezoidal Rule

An algorithm for two-cycle iteration with the trapezoidal rule is given in Table 4.5. The incremental accelerations are computed in a nodewise manner, as described in Section 4.4.2 for the central difference method. However, in the first evaluation phase, we evaluate $\underset{\sim}{S}_{i-\frac{1}{2}}$ (instead of $\underset{\sim}{S}_{i-1}$) for which half of the incremental displacments during the time step i are used in calculation of the incremental stresses. This will improve the accuracy of the results.

TABLE 4.4 Algorithm for Central Difference Predictor for Nonlinear Analysis

For each time step after the first,* do the following:

1. Evaluate the incremental accelerations in a nodewise manner

$$\Delta\ddot{\utilde{D}}_i = \utilde{M}^{-1}(\Delta\utilde{A}_i - \utilde{S}_{i-1}\,\Delta\utilde{D}_i)$$

2. Calculate the total accelerations

$$\ddot{\utilde{D}}_i = \ddot{\utilde{D}}_{i-1} + \Delta\ddot{\utilde{D}}_i$$

3. Compute the total and incremental displacements

$$\utilde{D}_{i+1} = 2\utilde{D}_i - \utilde{D}_{i-1} + \ddot{\utilde{D}}_i(\Delta t)^2$$

$$\Delta\utilde{D}_{i+1} = \utilde{D}_{i+1} - \utilde{D}_i$$

*Displacements at the end of the first time step are calculated from the Taylor series as follows

$$\utilde{D}_1 = \utilde{D}_o + \dot{\utilde{D}}_o\,\Delta t + \frac{1}{2}\,\ddot{\utilde{D}}_o(\Delta t)^2$$

TABLE 4.5 Algorithm for Two-Cycle Iteration with Trapezoidal Rule for Nonlinear Analysis

For each time step, do the following:

1. Prediction

$$\begin{aligned}\underset{\sim}{\dot{D}}_i &= \underset{\sim}{\dot{D}}_{i-1} + \underset{\sim}{\ddot{D}}_{i-1}\,\Delta t & &(\text{for } i = 1)\\ \underset{\sim}{\dot{D}}_i &= \underset{\sim}{\dot{D}}_{i-2} + 2\underset{\sim}{\ddot{D}}_{i-1}\,\Delta t & &(\text{for } i > 1)\\ \underset{\sim}{D}_i &= \underset{\sim}{Q}_{i-1} + \underset{\sim}{\dot{D}}_i\,\Delta t/2\\ \Delta\underset{\sim}{D}_i &= \underset{\sim}{D}_i - \underset{\sim}{D}_{i-1}\end{aligned}$$

where

$$\underset{\sim}{Q}_{i-1} = \underset{\sim}{D}_{i-1} + \underset{\sim}{\dot{D}}_{i-1}\,\Delta t/2$$

2. Evaluation

$$\begin{aligned}\Delta\underset{\sim}{\ddot{D}}_i &= \underset{\sim}{M}^{-1}(\Delta\underset{\sim}{A}_i - \underset{\sim}{S}_{i-\frac{1}{2}}\,\Delta\underset{\sim}{D}_i)\\ \underset{\sim}{\ddot{D}}_i &= \underset{\sim}{\ddot{D}}_{i-1} + \Delta\underset{\sim}{\ddot{D}}_i\end{aligned}$$

3. Correction

$$\begin{aligned}\underset{\sim}{\dot{D}}_i &= \underset{\sim}{R}_{i-1} + \underset{\sim}{\ddot{D}}_i\,\Delta t/2\\ \underset{\sim}{D}_i &= \underset{\sim}{Q}_{i-1} + \underset{\sim}{\dot{D}}_i\,\Delta t/2\\ \Delta\underset{\sim}{D}_i &= \underset{\sim}{D}_i - \underset{\sim}{D}_{i-1}\end{aligned}$$

where

$$\underset{\sim}{R}_{i-1} = \underset{\sim}{\dot{D}}_{i-1} + \underset{\sim}{\ddot{D}}_{i-1}\,\Delta t/2$$

4. Evaluation

$$\begin{aligned}\Delta\underset{\sim}{\ddot{D}}_i &= \underset{\sim}{M}^{-1}(\Delta\underset{\sim}{A}_i - \underset{\sim}{S}_i\,\Delta\underset{\sim}{D}_i)\\ \underset{\sim}{\ddot{D}}_i &= \underset{\sim}{\ddot{D}}_{i-1} + \Delta\underset{\sim}{\ddot{D}}_i\end{aligned}$$

4.4.4 Fourth-Order Runge-Kutta Method

The fourth-order Runge-Kutta method for a single equation in the form $\ddot{d} = f(t,d)$ was given in Section 2.6 of Chapter 2. By applying this method to the incremental equations of motion [Eq. (4.52)], we can write an algorithm for nonlinear analysis that is given in Table 4.6. Evaluation of the functions $\underset{\sim}{\Psi}_1$, $\underset{\sim}{\Psi}_2$, $\underset{\sim}{\Psi}_3$ and $\underset{\sim}{\Psi}_4$ is done in a nodewise manner. In this approach the tangential stiffness matrix is revised only once in each time step, just before the last evaluation (where $\underset{\sim}{S}_{i-\frac{1}{2}}$ is needed).

TABLE 4.6 Algorithm for Fourth-Order Runge-Kutta Method

For each time step, do the following:

$$\underset{\sim}{\Delta D}_i = \underset{\sim}{\Delta D}_{i-1} + \underset{\sim}{\Delta \dot{D}}_{i-1} \Delta t + \frac{\Delta t}{6}(\underset{\sim}{\Psi}_1 + \underset{\sim}{\Psi}_2 + \underset{\sim}{\Psi}_3)$$

$$\underset{\sim}{\Delta \dot{D}}_i = \underset{\sim}{\Delta \dot{D}}_{i-1} + \frac{1}{6}(\underset{\sim}{\Psi}_1 + 2\underset{\sim}{\Psi}_2 + 2\underset{\sim}{\Psi}_3 + \underset{\sim}{\Psi}_4)$$

$$\underset{\sim}{D}_i = \underset{\sim}{D}_{i-1} + \underset{\sim}{\Delta D}_i$$

$$\underset{\sim}{\dot{D}}_i = \underset{\sim}{\dot{D}}_{i-1} + \underset{\sim}{\Delta \dot{D}}_i$$

where

$$\underset{\sim}{\Psi}_1 = \Delta t\, \underset{\sim}{M}^{-1} (\underset{\sim}{\Delta A}_{i-1} - \underset{\sim}{S}_{i-1} \underset{\sim}{\Delta D}_{i-1})$$

$$\underset{\sim}{\Psi}_2 = \Delta t\, \underset{\sim}{M}^{-1} \left[\frac{1}{2} \underset{\sim}{\Delta A}_i - \underset{\sim}{S}_{i-1} (\underset{\sim}{\Delta D}_{i-1} + \frac{\Delta t}{2} \underset{\sim}{\Delta \dot{D}}_{i-1})\right]$$

$$\underset{\sim}{\Psi}_3 = \Delta t\, \underset{\sim}{M}^{-1} \left[\frac{1}{2} \underset{\sim}{\Delta A}_i - \underset{\sim}{S}_{i-1} (\underset{\sim}{\Delta D}_{i-1} + \frac{\Delta t}{2} \underset{\sim}{\Delta \dot{D}}_{i-1} + \frac{\Delta t}{4} \underset{\sim}{\Psi}_1)\right]$$

$$\underset{\sim}{\Psi}_4 = \Delta t\, \underset{\sim}{M}^{-1} \left[\underset{\sim}{\Delta A}_i - \underset{\sim}{S}_{i-\frac{1}{2}} (\underset{\sim}{\Delta D}_{i-1} + \Delta t\, \underset{\sim}{\Delta \dot{D}}_{i-1} + \frac{\Delta t}{2} \underset{\sim}{\Psi}_2)\right]$$

CHAPTER 5

COMPUTER APPLICATION

5.1 Introduction

Three Computer Programs, prepared exclusively for this study, will be described in this chapter. Program RESPPSQ4 is for linear analysis. Program NODIMP is for nonlinear analysis by implicit schemes and program NODEXP is for nonlinear analysis by explicit schemes and the nodewise solution technique. The three programs require no auxiliary storage and all the calculations are done within the main core of the digital computer. All three programs were written in the FORTRAN IV language and were run under the WATFIV and FORTRAN-H Compilers. In program RESPPSQ4, for linear analysis, all real variables are declared single precision. However, in programs NODIMP and NODEXP, for nonlinear analysis, all real variables are declared long (double) precision. This is due to the fact that round-off error for nonlinear analysis can be quite significant and may produce erroneous results.

A description of each program, including a macroflow chart, program notations, and required input data, is included in the following sections. Actual program listings, with descriptive comment statements and sample outputs are presented in Appendices.

In order to evaluate the efficiency of different methods, the library program PCLOCK provided by Stanford University Computation Center was employed. PCLOCK computes the number of centi-seconds elapsed between two calls. This computation time measurement was the basis for the comparisons of efficiency of different procedures in this study.

5.2 Program for Linear Analysis

5.2.1 Description of Program RESPPSQ4

Program RESPPSQ4 calculates the response for a plane stress (or plane strain) problem by the following methods:

1. Central difference predictor
2. Direct extrapolation with trapezoidal rule
3. Two cycle iteration with trapezoidal rule
4. Normal mode method

For the direct linear extrapolation, two approaches have been coded:

--using total displacement, velocity and acceleration with solution for displacement,

--Using incremental displacement, velocity and acceleration, with solution for incremental displacement.

In this program five different kinds of loading have been considered, namely, nodal loads, surface loads, volume loads and loads due to temperature strain and prestrains. For nodal loading, five types of loading can be applied through the use of LTYPE parameter as follows:

LTYPE=1	Indicates a constant load, but differemt at each node,
LTYPE=2	Indicates a piecewise linear load,
LTYPE=3	Indicates an equal sinusoidal forcing function at each node in the x direction,
LTYPE=4	Like LTYPE=3, but in the y direction.
LTYPE=5	Loads are given as data for all loaded nodes.

The program RESPPSQ4 consists of a main program that uses nineteen subroutines. These subroutines are described briefly in the following article.

5.2.2 Subroutine Synopsis and Macroflow Chart

A macroflow chart of the program RESPPSQ4 is given in Table 5.1. The double boxes in the flow chart represent subroutines. A brief description of the function of each subroutine follows:

SDATA: Reads and prints the structure data.

LDATA: Compiles the load data and prints pertinent information.

SURLOD: Calculates the equivalent nodal loads due to surface loads. Linear variation of components of surface loading between two adjacent nodes has been considered. This is demonstrated in Fig. 5.1. WS1, WS2, WS3, and WS4 are force per unit inclined length.

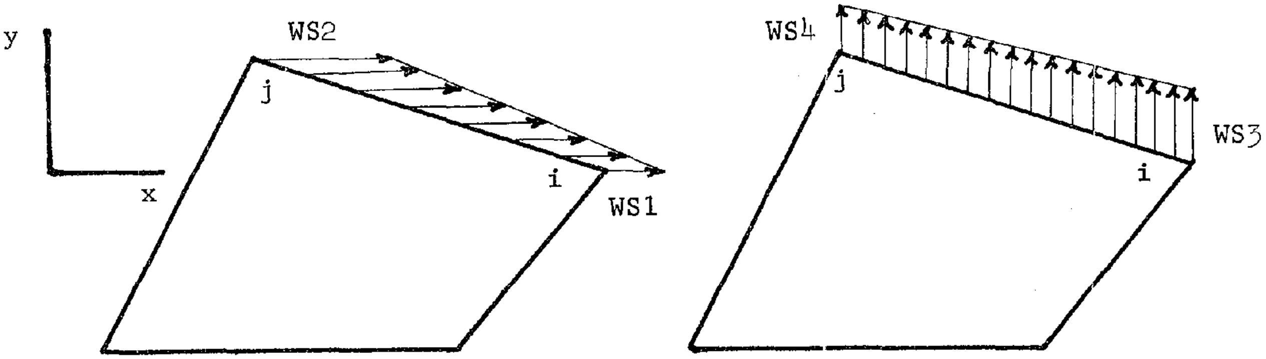


Fig. 5.1

VOLLOD: Calculates the equivalent nodal loads due to volume load WV1 (uniform distribution in the X direction), and WV2 (uniform distribution in the Y direction).

TPSTRN: Calculates the equivalent nodal loads due to uniform temperature strain.

PRSTRN: Computes the equivalent nodal loads due to the following prestrains:
PS1 = uniform expansion in the X direction
PS2 = uniform expansion in the Y direction
PS3 = uniform positive shear strain

TABLE 5.1 Macroflow Chart for RESPPSQ4

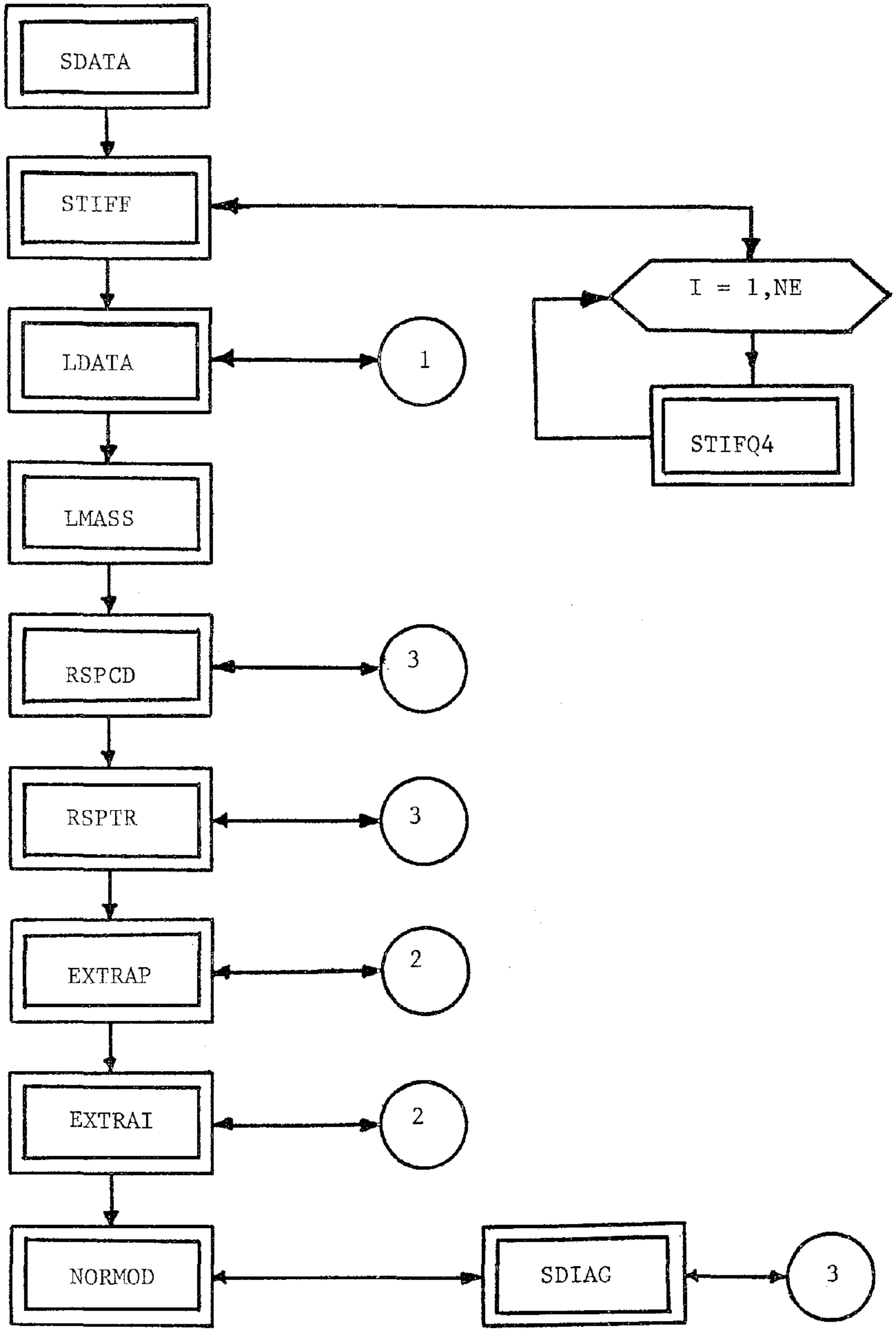
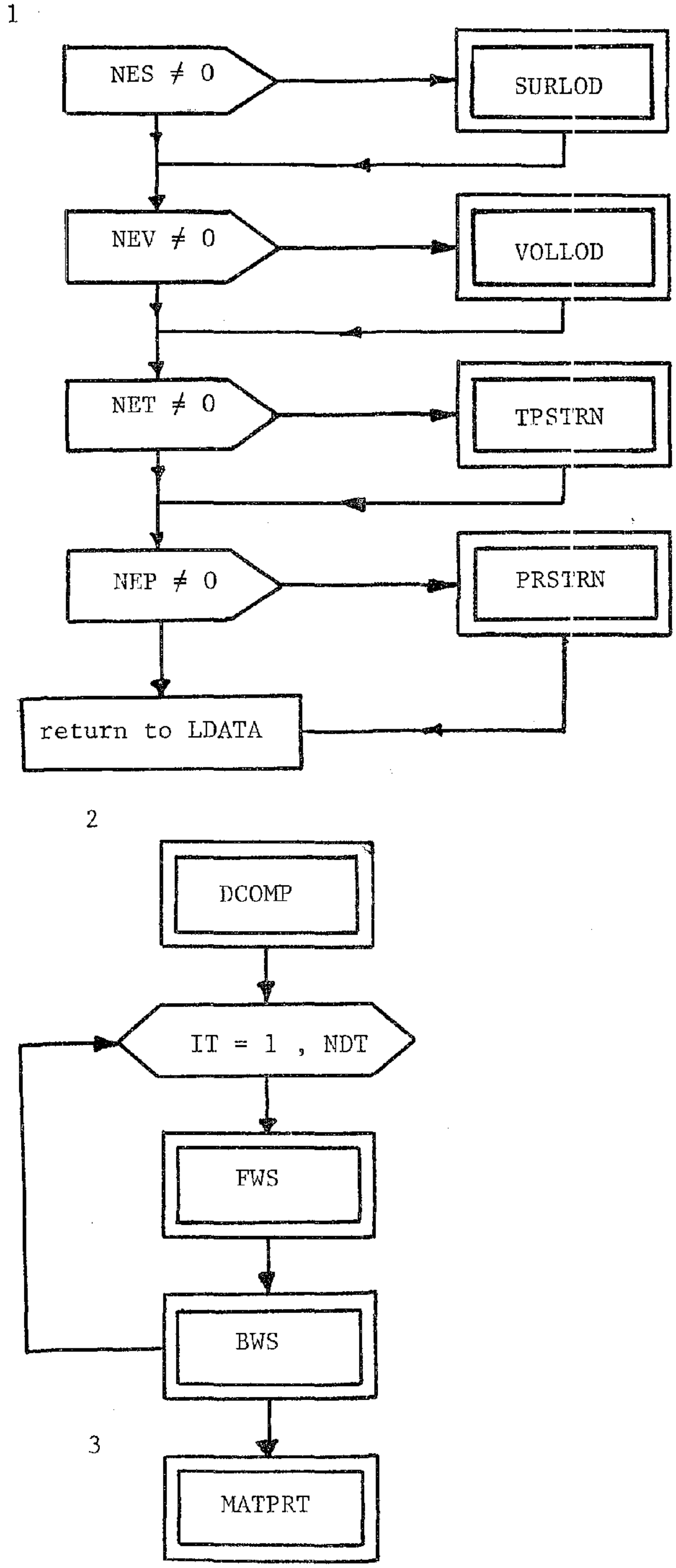


TABLE 5.1 - continued



STIFQ4: Generates the stiffness matrix for an isoparametric quadrilateral element.

STIFF: Assembles the element stiffness matrix into the structure stiffness matrix.

LMASS: Calculates the lumped mass matrix.

RRSPCD: Calculates the response by the central difference procedure.

RSPTR: Conducts the two cycle iteration with trapezoidal rule.

EXTRAP: Calculates the response by direct linear extrapolation with the trapezoidal rule and solution for total displacements.

EXTRAI: Computes the response by direct linear extrapolation with the trapezoidal rule but with solution for incremental displacements.

NORMOD: Performs the normal mode procedure.

SDIAG: Finds the eigenvalues and eigenvectors needed in the normal mode procedure by the Householder-QR method.

DCOMP: Decomposes a positive-definite matrix (Cholesky decomposition).

FWS: Finds the forward solution in the Cholesky decomposition.

BWS: Finds the backward and final solution in the Cholesky decomposition.

MATPRT: Prints the output.

5.2.3 Program Notation

Table 5.2 defines the important simple and subscripted variables that are used in the program RESPPSQ4. Real and Integer variables are designated by letters R and I accordingly.

5.2.4 Preparation of Data

Required input data for RESPPSQ4 are summarized in Table 5.3. Two standard FORTRAN number fields have been used and are identified for each card by F for real variables and I for integer variables.

TABLE 5.2 Notations for Program RESPPSQ4

I. Simple Variables

Variable(s)	Type	Definition
ALPHA	R	Thermal expansion coefficient
DELTAT	R	Temperature strain
DT	R	Incremental time step
E1,E2,E3	R	Stress-strain constants
ETA	R	Variable of spatial integrals

IPS	I	Indicator for plane stress or plane strain
		IPS = 0 for plane stress
		IPS = 1 for plane strain

LTYPE	I	Type of nodal loading
MAXT	R	Last time for response calculation
NDF	I	Number of degrees of freedom
NDISP	I	Number of possible displacements
NDT	I	Number of time steps
NE	I	Number of elements
NEP	I	Number of elements with prestrain
NES	I	Number of elements with surface load
NET	I	Number of elements with temperature strain
NEV	I	Number of elements with volume load
NLN	I	Number of loaded nodes
NN	I	Number of nodes
NRN	I	Number of restrained nodes
NSEG	I	Number of loading segments
PS1	R	Prestrain in x direction

TABLE 5.2 - continued

Variable(s)	Type	Definition
PS2	R	Prestrain in y direction
PS3	R	Shear prestrain
WS1,WS2,WS3,WS4	R	Intensities of surface load
WV1,WV2	R	Intensities of volume load
XI	R	Variable of spatial integrals

II. Subscripted Variables

A. Vectors

Variable(s)	Type	Definition
A( )	R	Acceleration
AO( )	R	Initial acceleration
DO( )	R	Initial displacements
DON,DD,DQ,DOLD( )	R	Nodal displacements (any time step)
EIG( )	R	Eigenvalues
EE( )	R	Eigenvectors
FO( )	R	Initial loads
F1,FON,F1N( )	R	Nodal force (any time step)
GAUSS( )	R	Coordinates of Gaussian quadrature
H( )	R	Young modulus
ID,IR( )	I	Rearrangement index
J1,J2,J3,J4( )	I	Element node numbers
M( )	R	Lumped mass matrix
MI( )	R	Scaled mass matrix

TABLE 5.2 - continued

Variable(s)	Type	Definition
NCRL( )	I	Cumulative restraint list
PR( )	R	Poisson ratio
RHO( )	R	Mass per unit volume
RL( )	I	Restrained code
TH( )	R	Thickness of the elements
VO( )	R	Initial velocities
V1,VOO,VON,VOLD( )	R	Nodal velocity (any time step)
X( )	R	Nodal X-coordinates
Y( )	R	Nodal Y-coordinates

B. Doubly-Subscripted Variables

Variable	Type	Definition
C(,)	R	Matrix of nodal coordinates for an element
D(,)	R	Array of nodal displacements
F(,)	R	Applied nodal forces
G(,)	R	Elements of strain-displacement matrix
SE(,)	R	Element stiffness matrix
SN(,)	R	Structure stiffness matrix
STM(,)	R	Scaled stiffness matrix
XX(,)	R	Eigenvectors

TABLE 5.3 Input Data for RESPPSQ4

I. Structure data and control parameter

Variables	Fields	No. of cards
1. NE,NN,NRN	3I3	1
2. IPS	I3	1
3. J,X(J),Y(J)	I3,2F10.	NN
4. I,J1(I),J2(I),J3(I),J4(I), TH(I),H(I),PR(I),RHO(I)	5I3,4F10.	NE
5. K,RL(2K-1),RL(2K)	3I2	NRN

II. Load Data

Variables	Fields	No. of cards
1. LTYPE,NLN,NES,NEV, NET,NEP,NDT,DT	7I5,F10.	1
2. K,FO(2K-1),FO(2K),DO(2K-1), DO(2K),VO(2K-1),VO(2K)	I5,6F10.	NLN
3. NSEG (if LTYPE = 2,3 or 4)	I5	1
4. K (number of loaded node)	I5	1
CX,TX1,CY,TY1	4F10.	1
SLOPEX,TX2,SLOPEY,TY2 (if LTYPE = 2, repeat this NLN times)	4F10.	1
5. AMP,OMEGA,C,T1,T2 (if LTYPE = 3 or 4)	5F10.	NSEG
6. K,F(IT,2K-1),F(IT,2K) (if LTYPE = 5)	I5,2F10.	NLN times NDT
7. IS,JS,WS1,WS2,WS3,WS4 (if NES not equal to zero)	2I5,4F10.	NES
8. K,WV1,WV2 (if NEV not equal to zero)	I5,2F10.	NEV
9. K,DELTAT,ALPHA (if NET not equal to zero)	I5,2F10.	NET
10. K,PS1,PS2,PS3 (if NEP not equal to zero)	I5,3F10.	NEP

5.3 Program for Nonlinear Analysis by Implicit Methods

5.3.1 Description of program NODIMP

NODIMP is a program for nonlinear dynamic analysis of plane stress (or plane strain) problems by three implicit procedures as follows:

1. Newmark-Beta method
2. Houbolt procedure
3. Park stiffly-stable method

Both material and geometric nonlinearities have been implemented. They can be controlled by a parameter named IGN. When geometric nonlinearity is ignored, IGN must be set equal to zero. If geometric nonlinearity is to be taken into account, IGN must be set equal to one.

Two different types of stress approximation have been incorporated in the program. Parameter ISTRES controls the stress approximation. It takes a value of zero when the stresses are taken constant within the element based on their values at the geometric center. It is assigned a value of one when the stresses are evaluated at the four element integration points. In addition, equilibrium iterations for implicit schemes have been implemented. In this case, when stresses are assumed constant throughout the element based on their values at the geometric center of the element, the internal force vector is calculated by adding the incremental internal force changes based on the stiffness matrix. However, when stresses are calculated at the integration points, in order to obtain more accurate results, the internal force vector is calculated directly from the stresses.

In this program three different types of loadings have been considered, namely, nodal loads, surface loads, and volume loads. For nodal loads two types of loading can be applied through the use of the parameter LTYPE. When LTYPE is assigned a value of one, a constant nodal load is applied. If LTYPE is assigned a value of two, a piecewise-linear nodal load is applied. In this case, the time and the value of load at the beginning and end of each segment are given as input data; and the piecewise-linear load is approximated by constant average values, as is shown in Fig. 5.2. It should be noted that nodal loads are given only for a selected number of degrees of freedom through an array of load indexes. Similarly, the output, that is, the displacements and stresses are printed only for a selected number of nodal degrees of freedom through arrays of response and stress indexes. Furthermore, the output

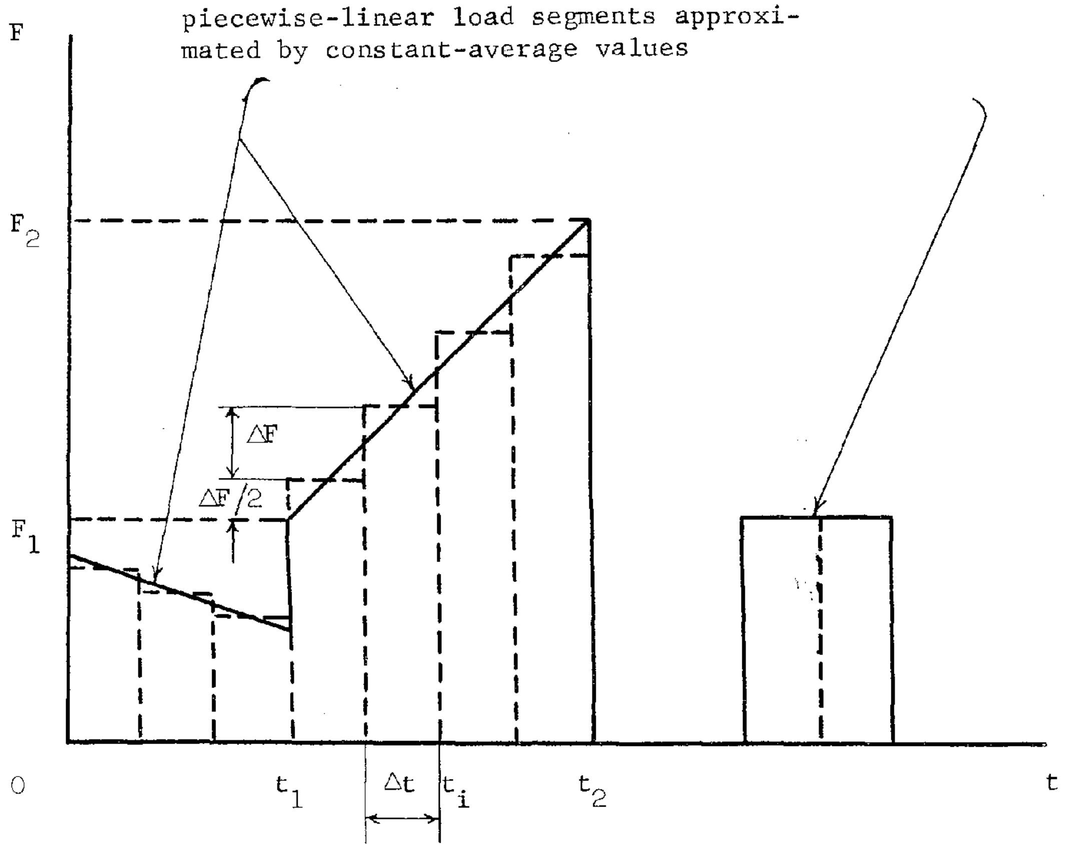

F
piecewise-linear load segments approximated by constant-average values
F_2
ΔF
$\Delta F/2$
F_1
0
t_1
Δt
t_i
t_2
t

Fig. 5.2

need not be saved and printed for every time step, but it can be stored and printed for every chosen interval of time. The interval for printing the output is given as data by parameter NIOUT that indicates the number of time steps for which the output is to be stored and printed. These features are essential for saving storage, especially when the nodal loading and output are to be plotted.

The program NODIMP consists of a main program that uses twenty eight subroutines. These subroutines are described briefly in the following article.

5.3.2 Subroutine Synopsis and Macroflow Chart

A macroflow chart of the program NODIMP is given in Table 5.4. Subroutines SURLOD, VOLLOD, LMASS, FWS, BWS and MATPRT are the same as described for the program RESPPSQ4 in Art. 5.2.2. Moreover, subroutines SDATA, LDATA and STIFF perform the same functions as in the program RESPPSQ4, but they are modified for the nonlinear analysis. Subroutines DCOMP and DCOMPO decompose a positive-definite matrix by Cholesky decomposition. However, in DCOMP, the reciprocals of diagonal elements are stored in the diagonal positions. This is expedient whenever solution of a linear system of equations by forward and backward sweeps is desired [54]. Concise descriptions of functions of the remaining subroutines are given in the following paragraphs:

ELAST: Creates the elastic stress-strain matrix.

CLEAR: Clears and initializes the required values for each method at the outset of computations.

PLAST: Creates the incremental plastic stress-strain matrix at the geometric center of the element or at four spatial integration points.

STIFQ4: Generates the stiffness matrix for an element when geometric nonlinearity is not taken into account.

TABLE 5.4 Macroflow chart for NODIMP

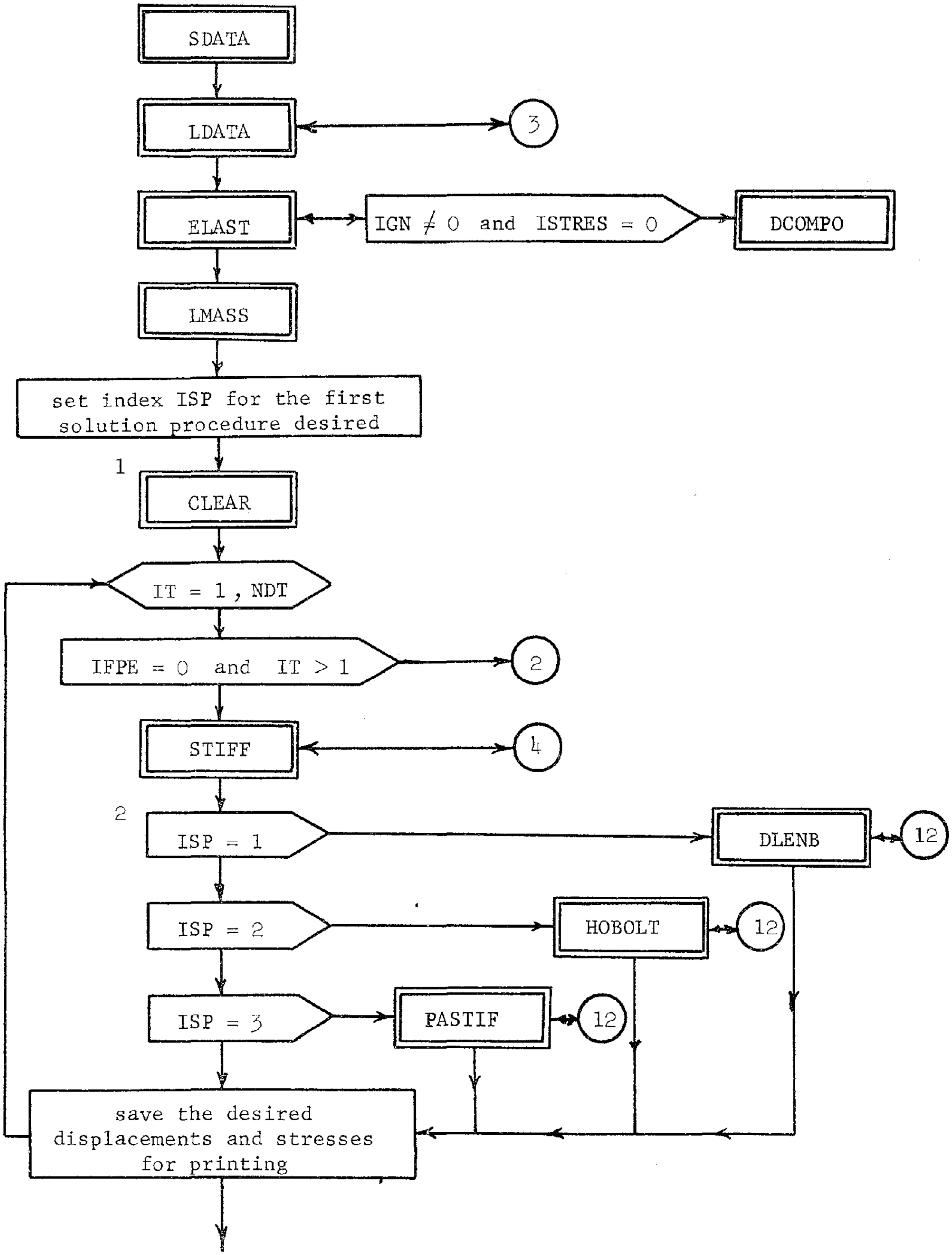
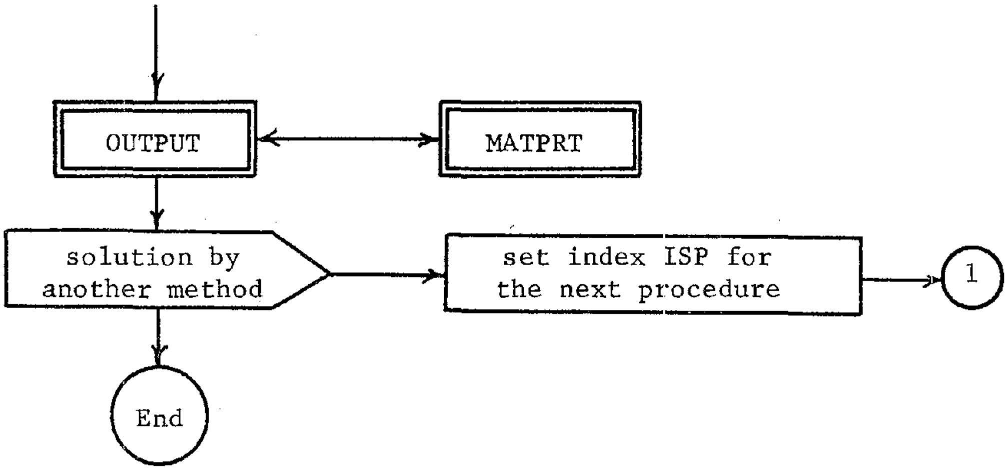
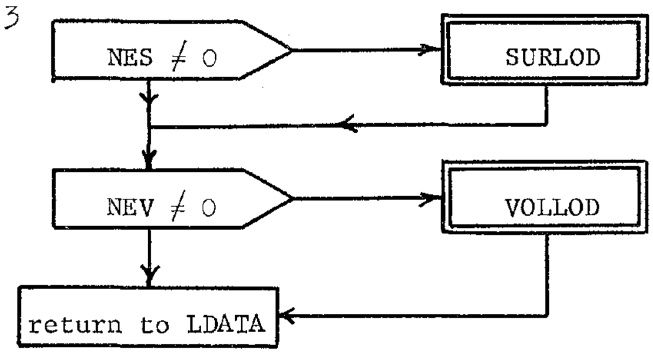


TABLE 5.4 - continued


OUTPUT
MATPRT
solution by another method
set index ISP for the next procedure
1
End



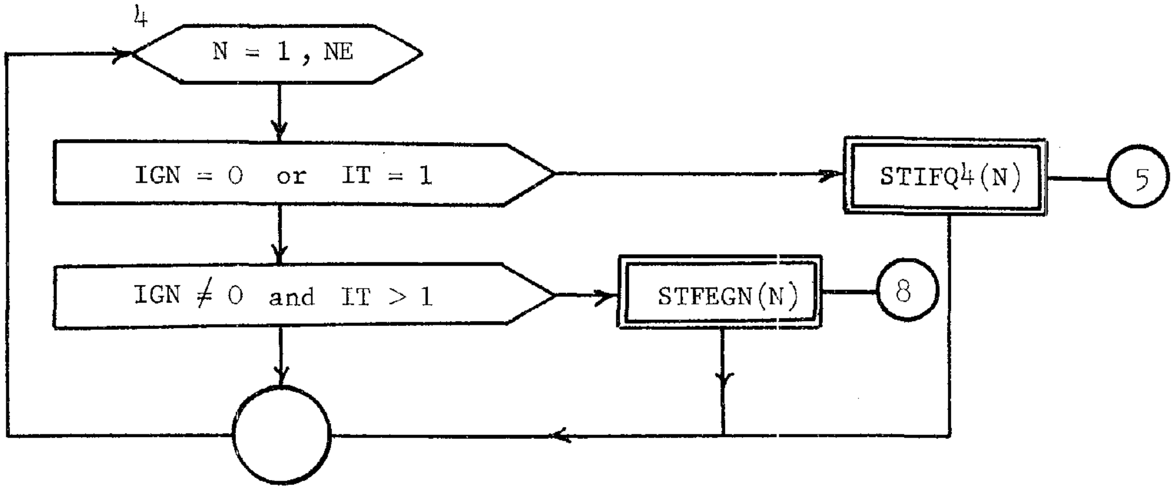


TABLE 5.4 - continued

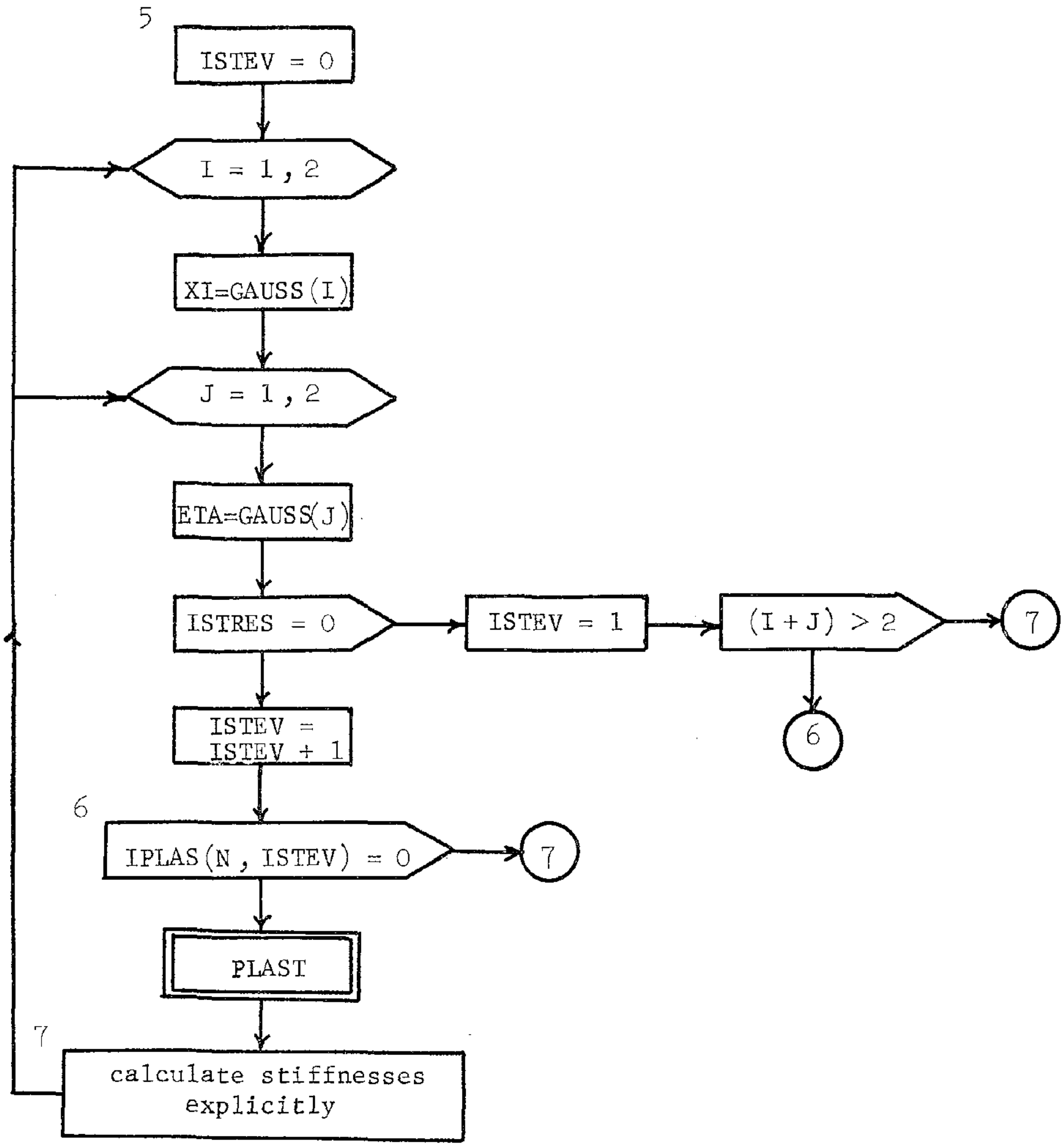


TABLE 5.4 - continued

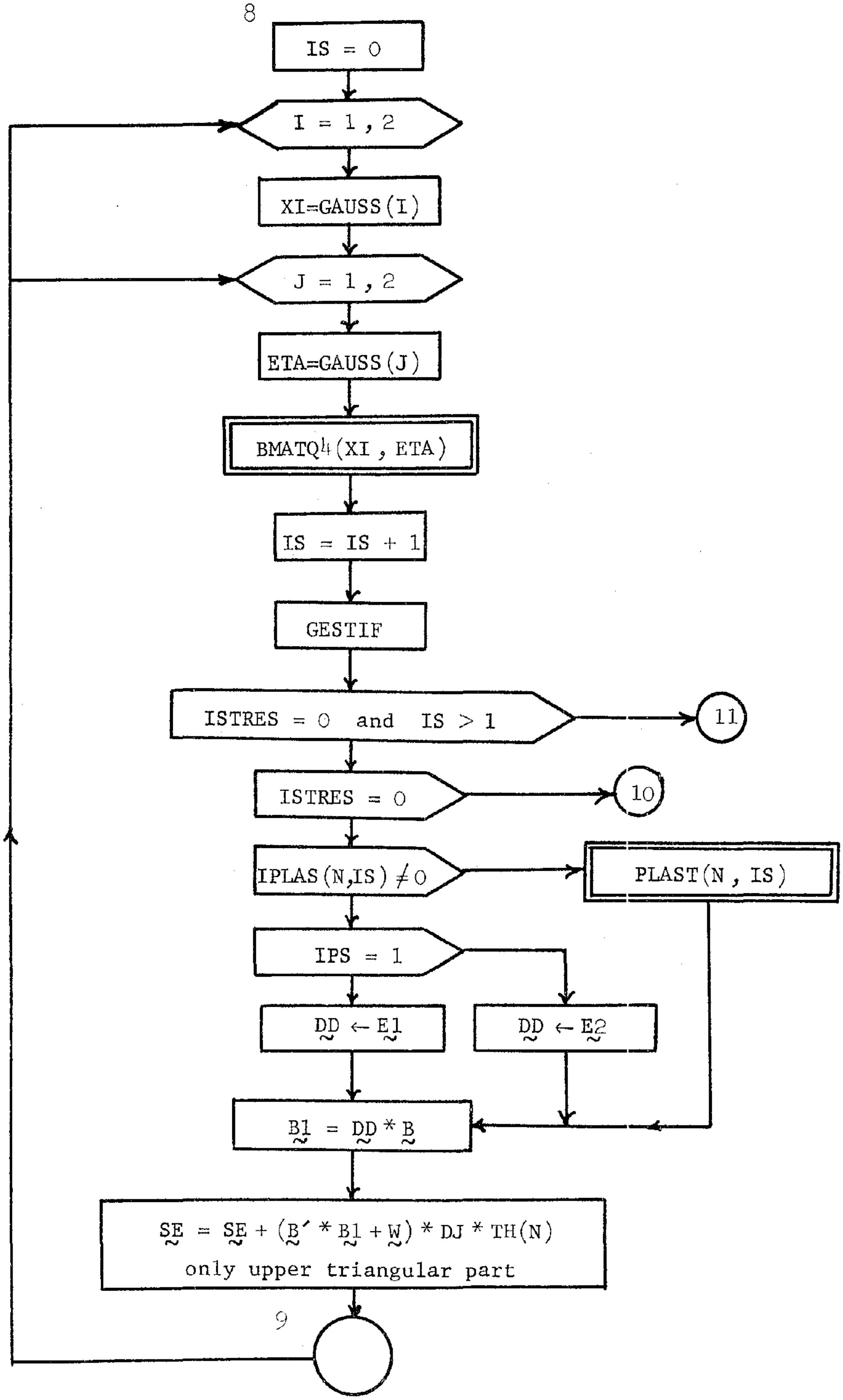
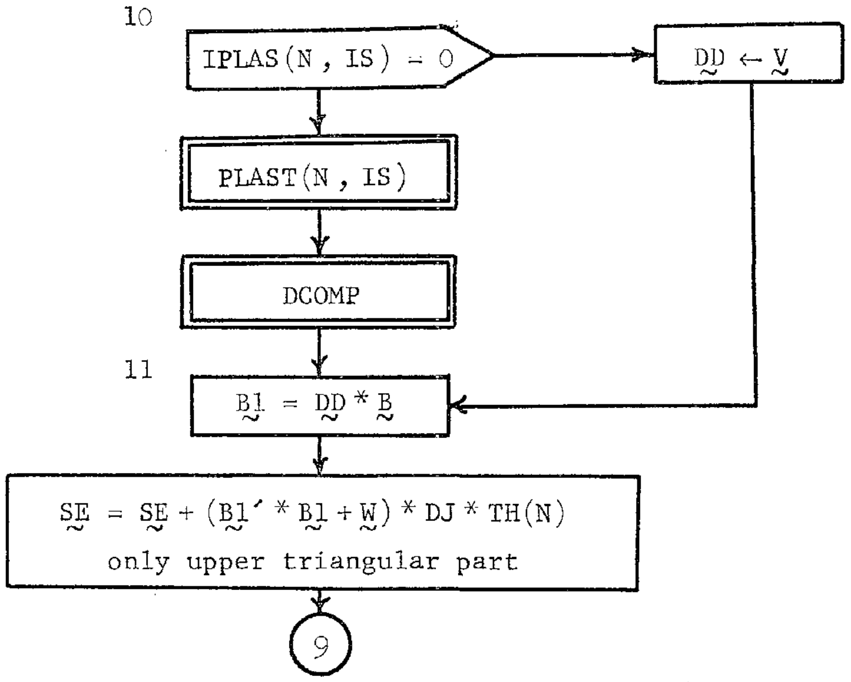


TABLE 5.4 - continued


10
IPLAS(N , IS) = 0
DD ← V
PLAST(N , IS)
DCOMP
11
B1 = DD * B
SE = SE + (B1' * B1 + W) * DJ * TH(N)
only upper triangular part
9

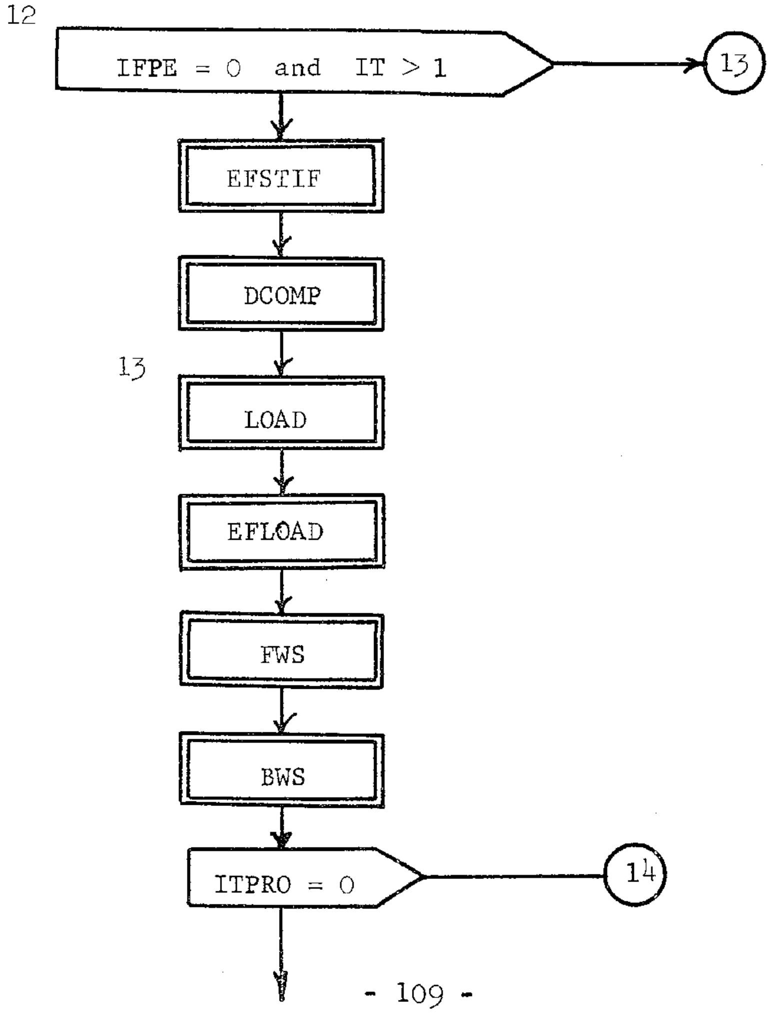


TABLE 5.4 - continued

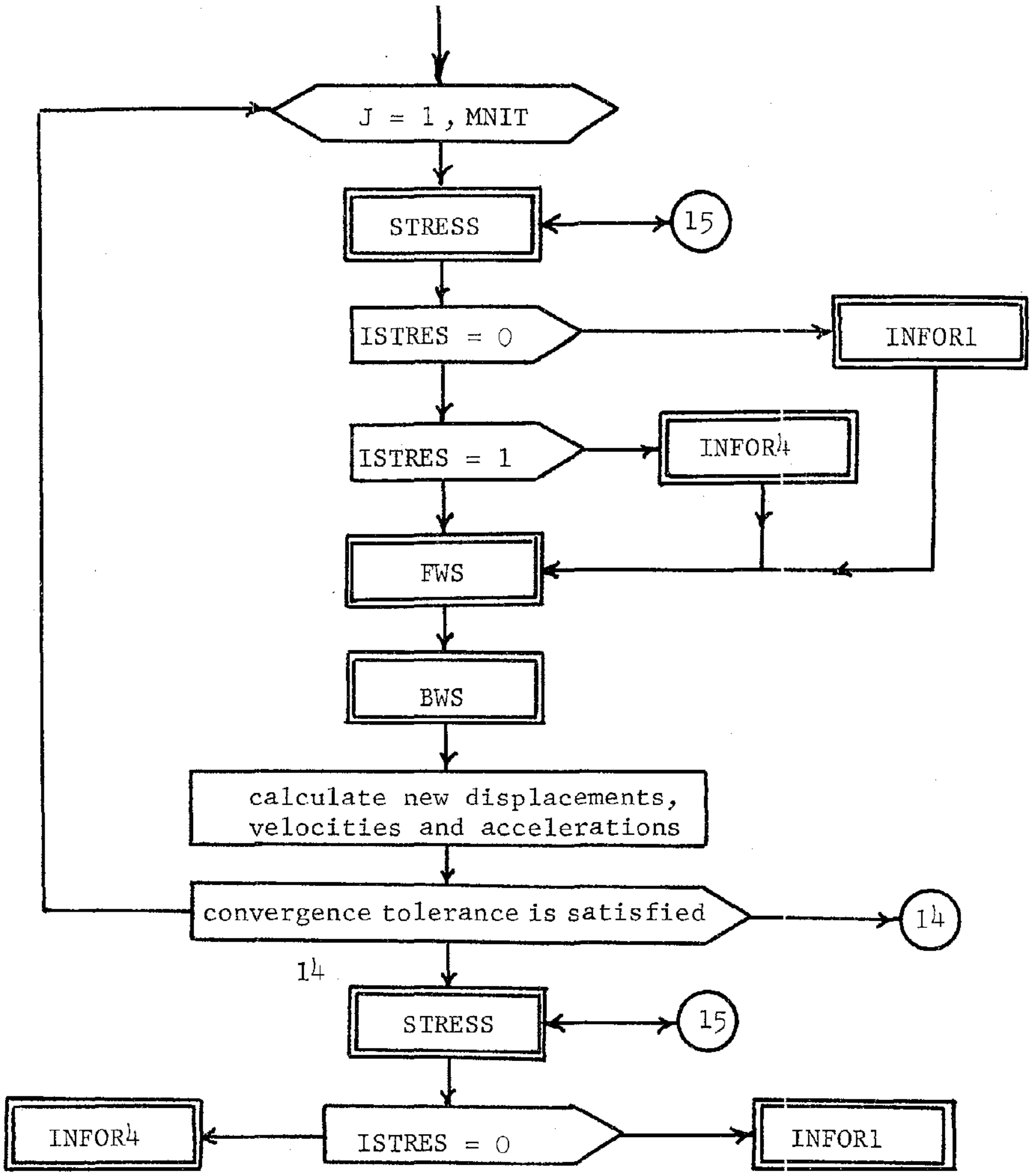
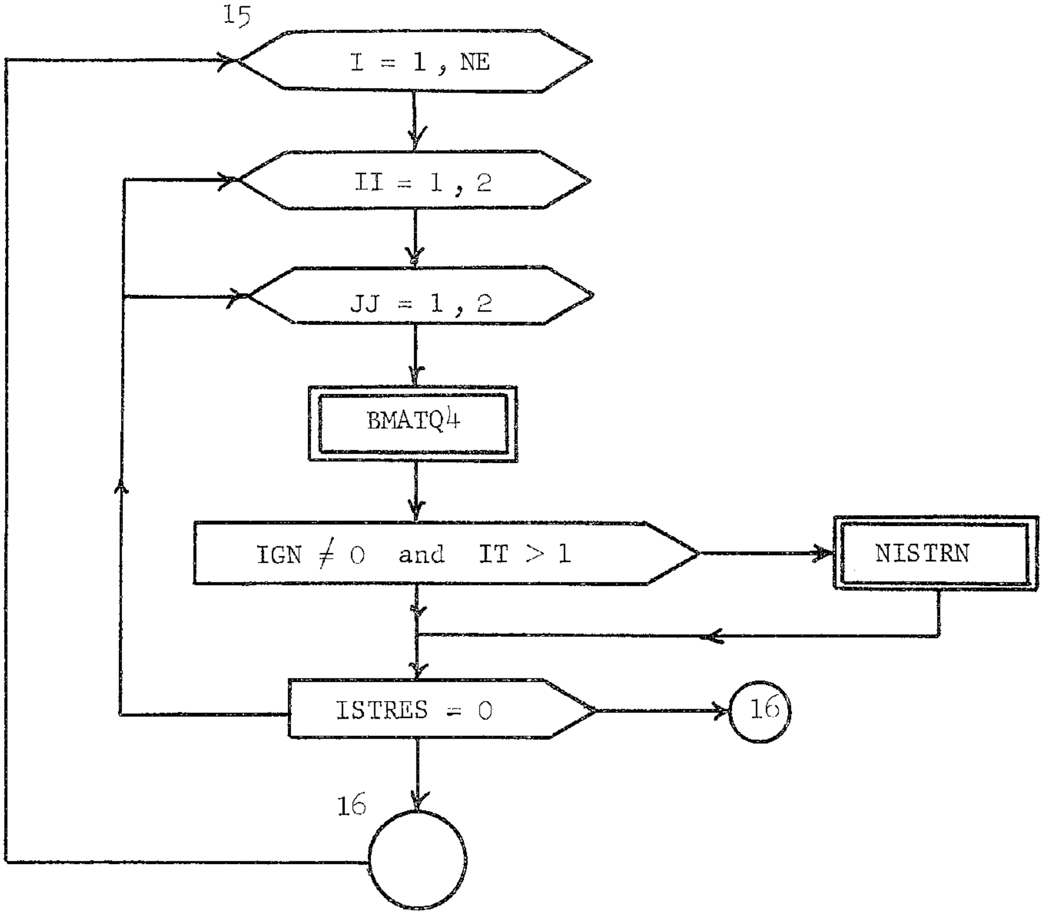


TABLE 5.4 - continued



STFEGN: Generates the stiffness matrix for an element when geometric nonlinearity is taken into account.

GESTIF: Calculates the contribution of geometric stiffness matrix to tangential stiffness matrix.

DLENB: Computes the response by Newmark-Beta method.

HOBOLT: Calculates the response by Houbolt method.

PASTIF: Performs Park stiffly-stable procedure.

EFSTIF: Calculates the effective stiffness matrix.

STRESS: Finds the stresses at the geometric center of the element or four integration points.

NISTRN: Computes the nonlinear components of incremental strains and adds them to linear incremental strains.

INFORC: Calculates the vector of internal nodal point forces.

LOAD: Calculates the vector of external nodal point loads.

EFLOAD: Evaluates the effective loads.

BMATQ4: Creates the strain-displacement matrix for a quadrilateral isoparametric element in a local point.

OUTPUT: Prints the heading and desired output for each method.

5.3.3 Program Notations

Table 5.5 defines the important simple and subscripted variables that are used in program NODIMP. Variables which begin with the letters I to N inclusive are declared as integer variables. Otherwise, they are declared long (double) precision real variables.

5.3.4 Preparation of Data

Required input data for NODIMP are summarized in Table 5.6. Two standard FORTRAN number fields have been used and are identified for each card by F for long precision real variables and I for integer variables.

TABLE 5.5 Notations for Program NODIMP

I. Simple Variables

Variable	Definition
BETA	Parameter of Newmark-Beta method
DELTA	Parameter of Newmark-Beta method
DT	Incremental time step
E	Young Modulus
ET	Tangent modulus
ETA	Variable of spatial integrals
F1	Value of load at the beginning of a loading segment
F2	Value of load at the end of a loading segment
GAMA	Parameter of Newmark-Beta method
H	Plastic modulus
IFPE	Indicater for first plastic element IFPE=0 no plastic element IFPE=1 at least one element is plastified
IGN	Index for geometric nonlinearity IGN=0 geometric nonlinearity is ignored IGN=1 geometric nonlinearity is to be taken into account

IPS	Indicator for plane stress or plane strain
	IPS=0 for plane stress
	IPS=1 for plane strain

IPU	Indicator for printing plastification and unloading If plastification and unloading are to be printed out, IPU=1, otherwise IPU=0

ISP	Index for solution procedure
	ISP=1 for Newmark-Beta method
	ISP=2 for Houbolt method
	ISP=3 for Park stiffly-stable mrthod

ISTOUT	Indicator for printing (ISTOUT=1) or not printing (ISTOUT=0) the stresses

TABLE 5.5 - continued

Variable	Definition
ISTRES	Control parameter for stress evaluation ISTRES=1 stresses are evaluated at the four integration points ISTRES=0 stresses are evaluated only at the geometric center of the elements

ITPRO	Control parameter for equiblirium iterations
	ITPRO=0 no iterations
	ITPRO=1 Constant stiffness iterations

LTYPE	Type of nodal loading
METOD1	Indicator for Newmark-Beta method (METOD1=1)
METOD2	Indicator for Houbolt method (METOD2=2)
METOD3	Indicator for Park method (METOD3=3)
MNIT	Maximum number of iterations
ND1	Number of displacements with nonzero initial displacements or velocities
ND2	Number of displacements with nonzero loads
ND3	Number of displacements where response is desired
NDF	Number of degrees of freedom
NDISP	Number of possible displacements
NDT	Number of time steps
NE	Number of elements
NE1	Number of elements with stresses desired
NES	Number of elements with surface load
NEV	Number of elements with volume load
NIOUT	Interval for printing the output
NN	Number of nodes
NRN	Number of restrained nodes
NSEG	Number of loading segments

TABLE 5.5 - continued

Variable	Definition
NTPO	Number of time steps for printing the output
PR	Poisson ratio
SIGH	Hydrostatic stress value
T1	Time at the biginning of a loading segment
T2	Time at the end of a loading segment
TOL	Convergence tolerance
WS1,WS2,WS3,WS4	Intensities of surface load
WV1,WV2	Intensities of volume load
XI	Variable of spatial integrals
YS	Yield stress

II. Subscripted Variables

A. Vectors

Variables	Definition
A( )	Accelerations
AO( )	Initial accelerations
CDI( )	Correction to incremental displacements
DI( )	Incremental displacements(at any time)
DN( )	Nodal displacements(at any time)
DN1,DN2,DN3( )	Displacements at the end of previous time steps (for Park method)
DO( )	Initial displacements
FE( )	Equivalent nodal loads due to surface or volume load
FI( )	Internal nodal forces

TABLE 5.5 - continued

Variable	Definition
FN( )	External nodal forces
FO( )	Initial loads
GAUSS( )	Coordinates of Gaussian quadrature
ID,IR( )	Rearrangement index
J1,J2,J3,J4( )	Element node numbers
LL( )	List of indexes for displacements with nonzero loads
LR( )	List of indexes for response-time histories
LS( )	List of indexes for stresses in elements
NCRL( )	Cumulative restraint list
NRL( )	Restraint list
RHO( )	Mass per unit volume
RM( )	Lumped mass matrix
STRAIN( )	Strains
TH( )	Thickness of the elements
VO( )	Initial velocities
VN1,VN2,VN3( )	Velocities at the end of previous time steps (for Park method)
X( )	Nodal X-coordinates
Y( )	Nodal Y-coordinates

B. Doubly-Subscripted Variables

Variable	Definition
B(,)	Strain-displacement matrix
C(,)	Matrix of nodal coordinates for an element
D(,)	Array of nodal displacements for printing
DD(,)	Incremental plastic stress-strain matix

TABLE 5.5 - continued

Variable	Definition
DE(,)	Nodal displacements for an element
DIE(,)	Nodal incremental displacements for an element
E1(,)	Elastic stress-strain matrix for plane stress
E2(,)	Elastic stress-strain matrix for plane strain
ESN(,)	Effective stiffness matrix
F(,)	Applied nodal forces
G(,)	Elements of strain-displacement matrix
IPLAS(,)	Indicator for plastification(=1) and unloading(=0)
SE(,)	Element stiffness matrix
SIGDX(,)	X-component of deviatoric stress
SIGDY(,)	Y-component of deviatoric stress
SIGEM(,)	Maximum effective stress
SIGET(,)	Effective stress
SN(,)	Structure stiffness matrix
SX(,)	Normal stresses in the X direction to be printed
SXY(,)	Shear stresses to be printed
SY(,)	Normal stresses in the Y direction to be printed
V(,)	Decomposed of elastic stress-strain matrix
W(,)	Contribution of geometric stiffnesses to tangent stiffness matrix

Note: For variables SIGDX, SIGDY, SIGET, SIGEM and IPLAS the first subscript refers to the element number and second subscript indicates either the integration points(when ISTRES=1) or geometric center of the element(when ISTRES=0). In the latter case only one column of these arrays is needed. For variables SX, SY, and SXY, first subscript refers to time increment while the second subscript refers to the elements with stresses desired.

TABLE 5.5 - continued

C. Multiply-Subscripted Variables

Variable	Definition
SIGT(, ,)	Total stresses
SIGT1(, ,)	Total stresses at the end of previous time step

Note: The first subscript of these variables refers to the element number. Second subscript indicates type of the stress and can take a value of one to four indicating normal stresses in the X and Y directions, shear stress in the XY plane and normal stress in the Z direction. Finally, third subscript indicates either the integration points (when ISTRES=1) or geometric center of the element(when ISTRES=0).

TABLE 5.6 Input Data for NODIMP

I. Structure data and control parameters

Variables	Fields	No. of cards
1. METOD1,METOD2,METOD3	3I3	1
2. ISTOUT,IPU,IGN	3I3	1
3. ITPRO,ISTRES,NIOUT,NTPO,MNIT, TOL	5I3,F10.	1
4. NE,NN,NRN	3I3	1
5. IPS	I3	1
6. J,X(J),Y(J)	I3,2F10.	NN
7. E,PR,ET,YS	4F12.	1
8. I,J1(I),J2(I),J3(I),J4(I), TH(I),RHO(I)	5I3,2F10.	NE
9. K,NRL(2K-1),NRL(2K)	3I2	NRN

II. Load and Output Data

Variables	Fields	No. of cards
1. LTYPE,ND1,ND2,ND3,NE1, NES,NEV,NDT,DT	8I5,F10.	1
2. J,DO(J),VO(J) (if ND1 not equal to zero)	I5,2F10.	ND1
3. J,LL(J) (if ND2 not equal to zero)	2I5	ND2
4. J,FO(J) (if ND2 not equal to zero and LTYPE=1)	I5,F10.	ND2

TABLE 5.6 - continued

Variables	Fields	No. of cards
5. J,NSEG	2I5	1
T1,F1,T2,F2 (if LTYPE=2, repeat this ND2 times)	4F10.	NSEG
6. IS,JS,WS1,WS2,WS3,WS4 (if NES not equal to zero)	2I5,4F10.	NES
7. K,WV1,WV2 (if NEV not equal to zero)	I5,2F10.	NEV
8. J,LR(J) (if ND3 not equal to zero)	2I5	ND3
9. J,LS(J) (if NE1 not equal to zero)	2I5	NE1

III. Parameter of Newmark-Beta Method

Variable	Fields	No. of cards
1. DELTA (if METOD1=1)	F10.	1

5.4 Program for Nonlinear Analysis by Explicit Methods

5.4.1 Description of Program NODEXP

NODEXP is a program for nonlinear dynamic analysis of plane stress (or plane strain) problems by the nodewise solution technique and three explicit methods as follows:

1. Central difference predictor
2. Two-cycle iteration with the trapezoidal rule
3. Fourth-order Runge-Kutta method

Material nonlinearity as well as geometric nonlinearity have been implemented; and two types of stress approximations have been incorporated in the program NODEXP, as described in Art. 5.3.1 for program NODIMP. In addition, the same types of loadings and limitations on the number of loaded nodes and the number of response histories have been considered as for the program NODIMP.

In the nodewise solution technique, in addition to the arrays J1, J2, J3, and J4 that contain node numbers for each element, we also need another array containing element numbers associated with each node. In order to save only nonzero terms and consequently to save storage, it is expedient to save the latter array in vector form. However, another array is also necessary to store the number of elements connected at each node. These two arrays are called NIE (node identification by the elements) and NEN (number of elements at each node). In the program NODEXP, only the arrays J1, J2, J3, and J4 are given as data and the arrays NEN and NIE are computed from them automatically by subroutine NIBE.

The program NODEXP consists of a main program that uses twenty six subroutines. These subroutines are described concisely in the following article.

5.4.2 Subroutine Synopsis and Macroflow Chart

A macroflow chart of the program NODEXP is given in Table 5.7. In this flow chart, for the sake of brevity, details of the load, stress and stiffness calculations have been omitted. These steps are done in the same way as for the program NODIMP. However, it should be noted that in this program only two rows of the stiffness matrices are generated at each stage of the computations. Subroutines LDATA, SURLOD, VOLLOD, LMASS, ELAST, PLAST, BMATQ4, NISTRN, DCOMPO, OUTPUT, and MATPRT are the same as described for the program NODIMP in Art. 5.3.2. Furthermore, subroutines SDATA, CLEAR and STRESS perform the same functions as in the program NODIMP, but they are modified for the nodewise solution technique and explicit methods. Brief descriptions of functions of the remaining subroutines follow:

NIBE: Generates the arrays NIE (node identification by the elements) and NEN (number of elements at each node).

CDP: Calculates the response by central difference predictor.

ITERTR: Computes the response by two-cycle iteration average acceleration method or trapezoidal rule.

EVAL(IA): Performs the evaluation required in the central difference predictor and two-cycle iteration with trapezoidal rule. IA is set equal to zero for the first time step and is greater than zero after the first time step.

RUNKUT: Calculates the response by the fourth-order Runge-Kutta method.

EVALON: Performs the four evaluations required in the fourth-order Runge-Kutta method.

INCLOD: Computes the incremental nodal loads.

STIF(N): Creates two rows of the structure stiffness matrix corresponding to node N.

TABLE 5.7 Macroflow chart for NODEXP

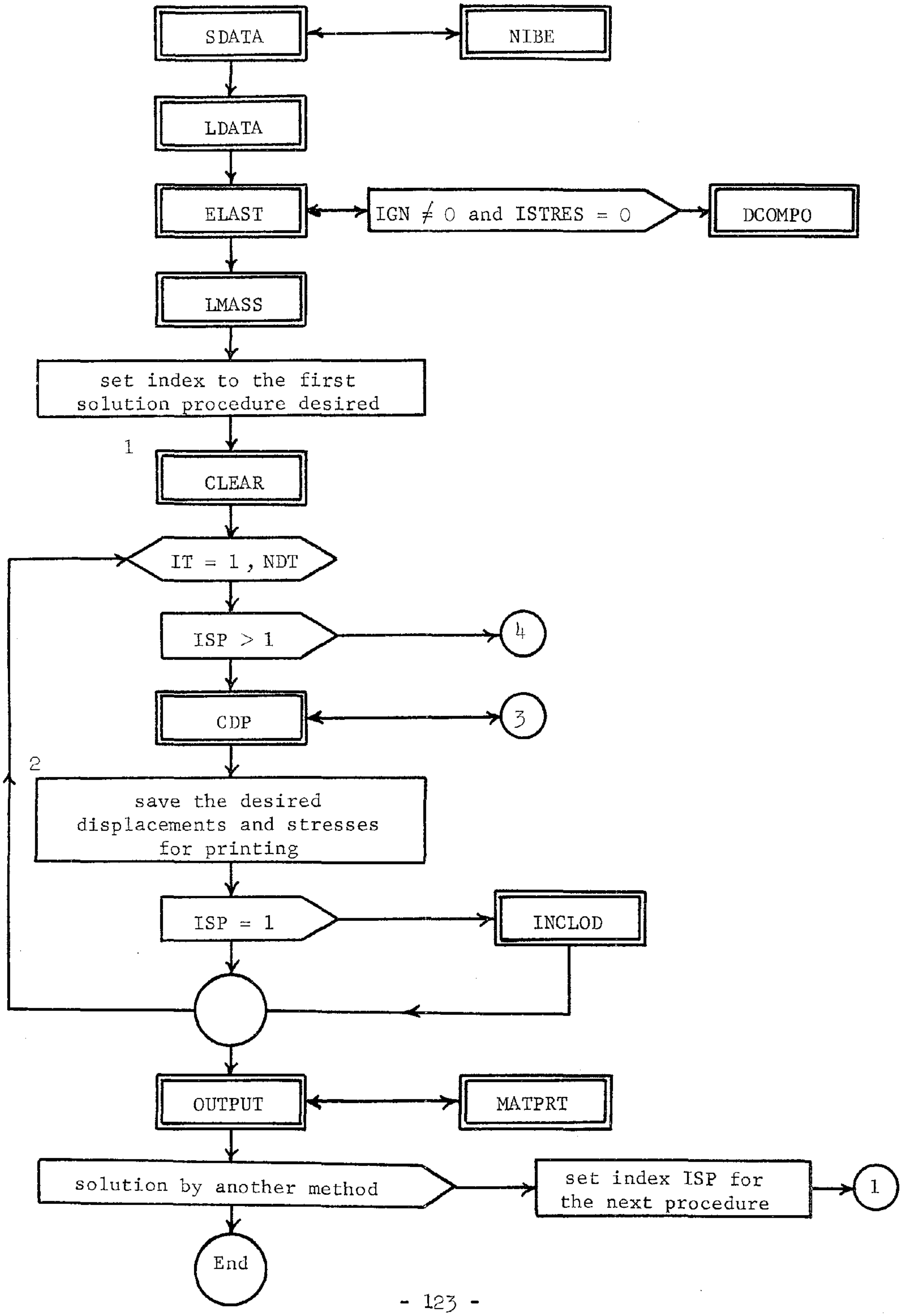
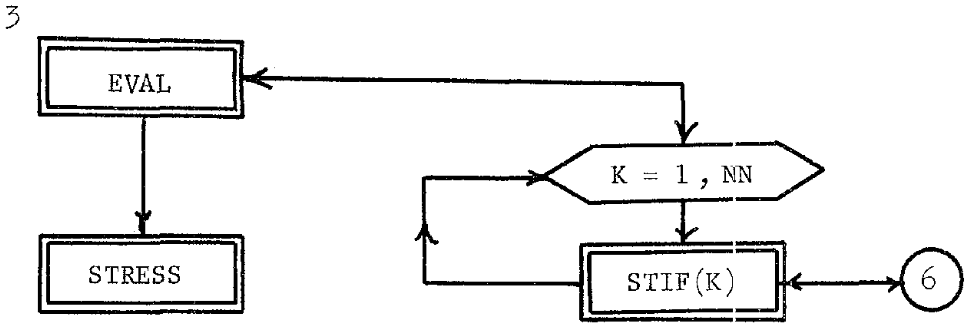

SDATA
NIBE
LDATA
ELAST
IGN ≠ 0 and ISTRES = 0
DCOMPO
LMASS
set index to the first solution procedure desired
1
CLEAR
IT = 1 , NDT
ISP > 1
4
CDP
3
2
save the desired displacements and stresses for printing
ISP = 1
INCLOD
OUTPUT
MATPRT
solution by another method
set index ISP for the next procedure
1
End

TABLE 5.7 - continued



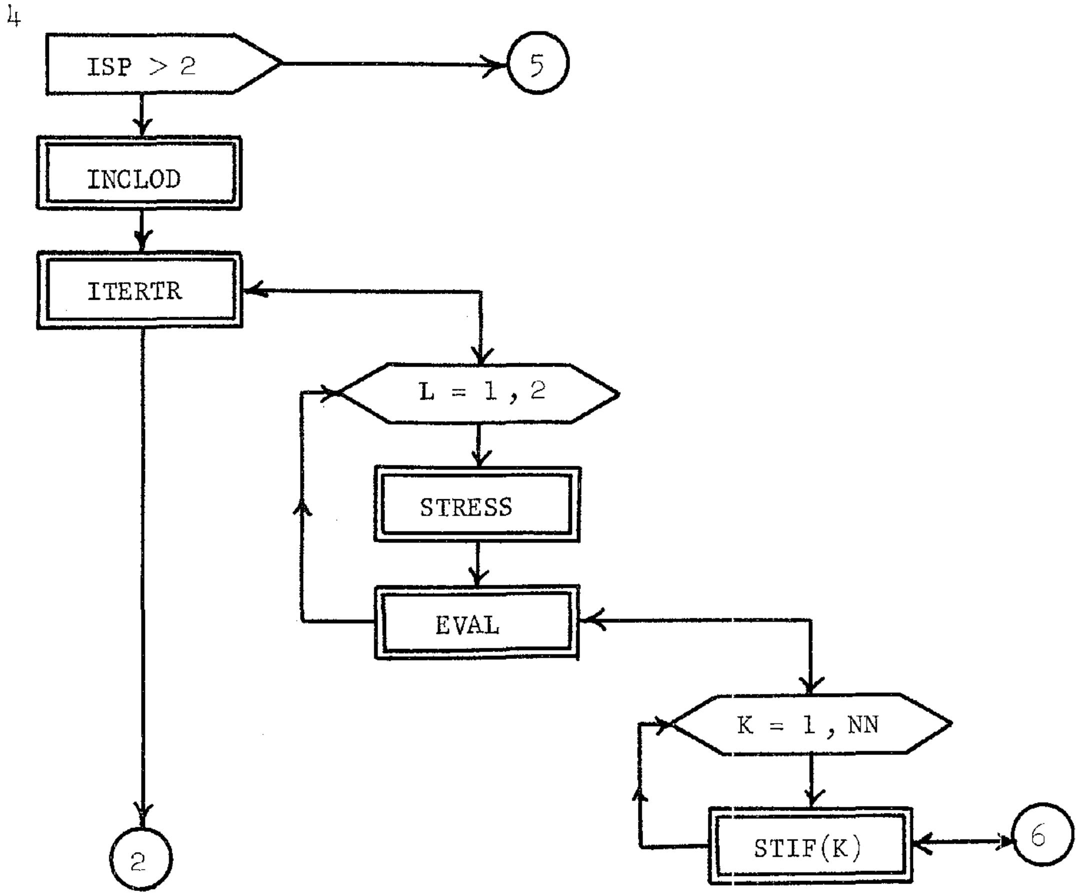
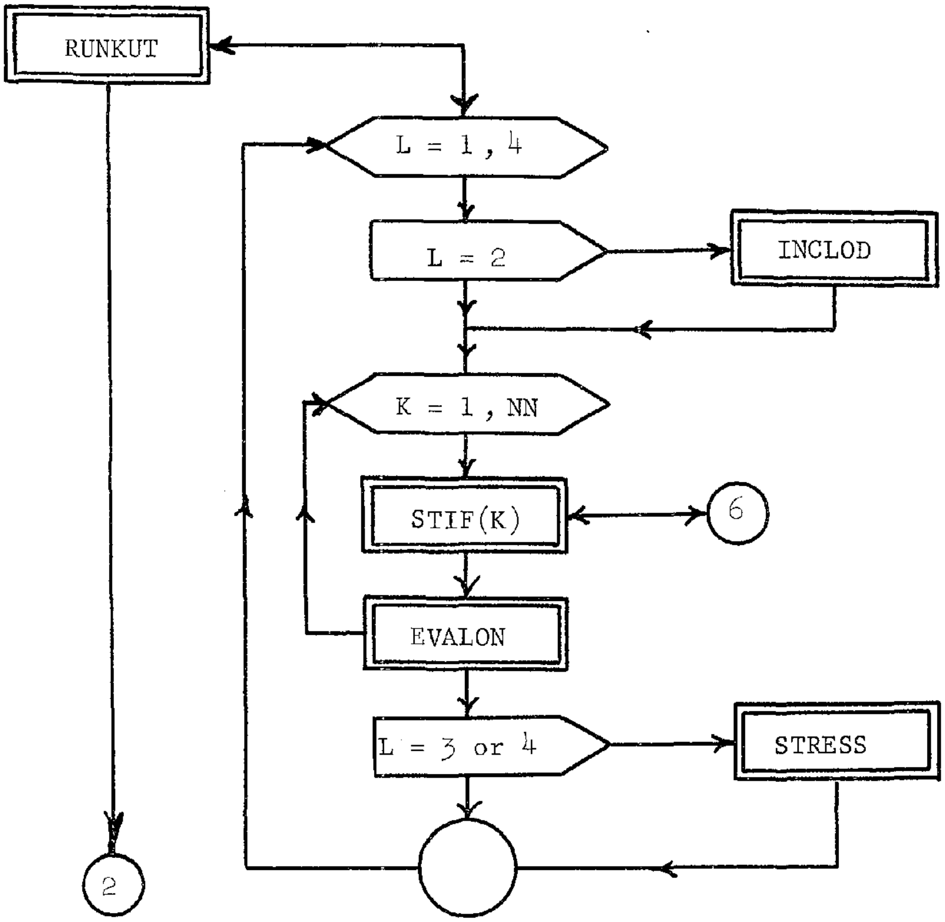
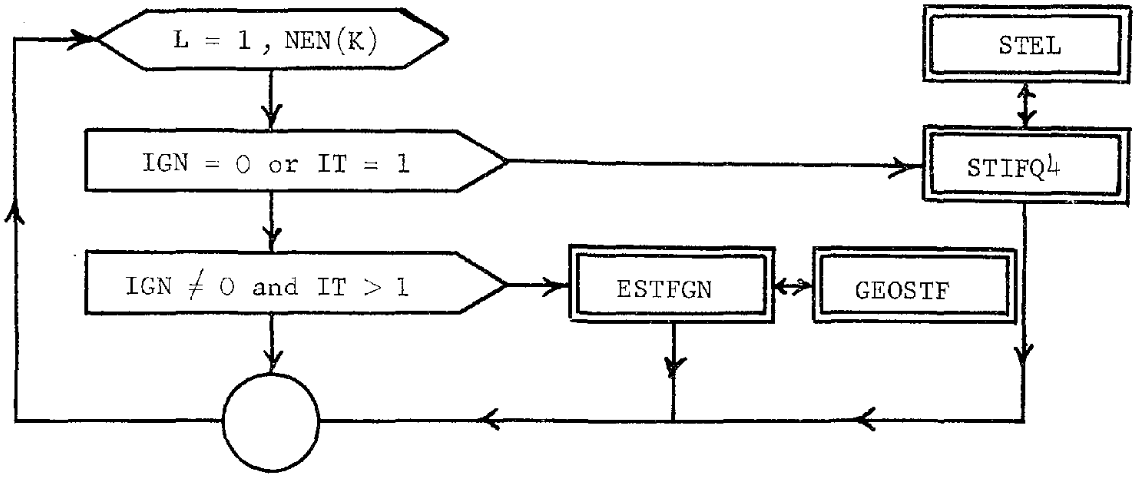


TABLE 5.7 - continued

5



6



STIFQ4(N,M): Creates two rows of the stiffness matrix for an element N corresponding to node M, when geometric nonlinearity is ignored.

STEL: Calculates stiffnesses explicitly for each element integration point, to be used by subroutine STIFQ4.

ESTFGN(N,M): Creates two rows of the stiffness matrix for an element N and corresponding to node M, when geometric nonlinearity is taken into account.

GEOSTF: Computes the contribution of the geometric stiffness matrix to the tangential stiffness matrix corresponding to a node.

5.4.3 Program Notations

Most of the notations for program NODEXP are chosen the same way as those for program NODIMP. Table 5.8 defines the important additional and renamed variables that are used in the program NODEXP. Similar to the program NODIMP, variables which begin with the letters I to N inclusive are declared as integer variables. Otherwise, they are declared long (double) precision real variables.

5.4.4 Preparation of Data

Required input data for NODEXP are almost the same as those for the program NODIMP that are summarized in Table 5.6, but with three exceptions. The differences are as follows:

--On the second card the variables ISTRES, NIOUT, NTPO, ISTOUT and IPU are read with the format 5I3.

--On the third card the variable IGN is read with the format I3.

--The last card must be omitted.

TABLE 5.8 Additional Notations List for Program NODEXP

I. Simple Variables

Variable	Definition
ISP	Index for solution procedures ISP=1 for central difference predictor ISP=2 for two-cycle iteration with trapezoidal rule ISP=3 for fourth-order Runge-Kutta method
METOD1	Indicator for central difference predictor (METOD1=1)
METOD2	Indicator for two-cycle iteration with trapezoidal rule (METOD2=2)
METOD3	Indicator for fourth-order Runge-Kutta method (METOD3=3)

II. Subscripted Variables

A. Vectors

Variables	Definition
DF( )	Nodal force increments
DI1( )	Incremental displacements at the end of previous time step
F1,F2,F3,F4( )	Functions of fourth-order Runge-Kutta method
NEN( )	Number of elements at each node
NIE( )	Node element numbers
VI1( )	Incremental velocities at the end of previous time step

B. Doubly-Subscripted Variables

Variables	Definition
SE(,)	Two rows of the element stiffness matrix
SN(,)	Two rows of the structure stiffness matrix

5.5 Example Problem

Many complex practical structures with complicated loadings can be idealized as plane stress or plane strain problems that can be solved by the methods and computer programs developed in this study. For example, the plane stress problem can be used to analyze stiffened sheet construction, box beams and arches. Practical applications of the plane strain problem occur in the stress analysis of soil systems, dams, tunnels, concrete walls and other long solid structures whose geometry and loading are constant in the long direction [11,20,62]. However, owing to the high cost of extensive comparisons of numerous solution techniques studied in this work (and in order to keep the cost of computer time at a minimum), a rather simple example was used to compare the accuracy, stability and efficiency of different procedures. The example problem is illustrated in Figs. 5.4 through 5.8. In this problem, which is a plane stress problem, a square plate 10" × 10" × 0.1" is loaded suddenly with a uniform line load at the middle line of the plate as shown in Fig. 5.4. The distributed loading on the plate is assumed to be replaced with equivalent forces at the nodes. In this case, they will be step loads as indicated in Fig. 5.3. The plate has two free edges and is restrained at the other edges as shown in the figures. Obviously, the cost of response calculations increases with the number of degrees of freedom or size of the problem. For efficiency comparisons, the plate is divided into four, nine, sixteen, twenty-five and thirty-six finite elements as shown in Figs. 5.4 through 5.8, and the response is calculated for a selected number of nodal displacements. In addition, the time step for response calculations is varied from 10^{-6} seconds to 5×10^{-5} seconds. The following material constants are used throughout this study:

Elastic modulus	:	30×10^6 psi
Poisson ratio	:	0.3
Tangent modulus	:	12×10^6 psi
Yield stress	:	36000 psi

The numbering system employed is shown for the 16-element plate in Fig. 5.9. In Figs. 5.4 through 5.8, NDF refers to the number of degrees of freedom.

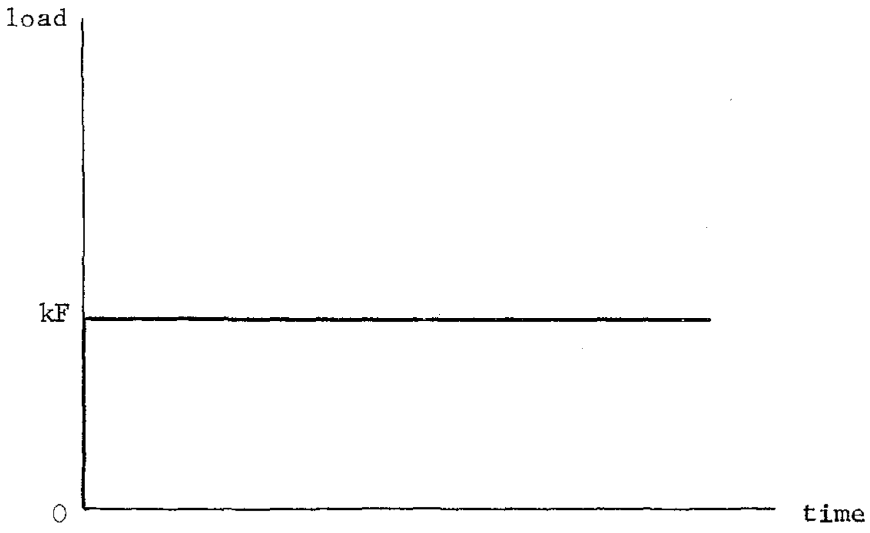


Fig. 5.3 Step Loading at the Nodes

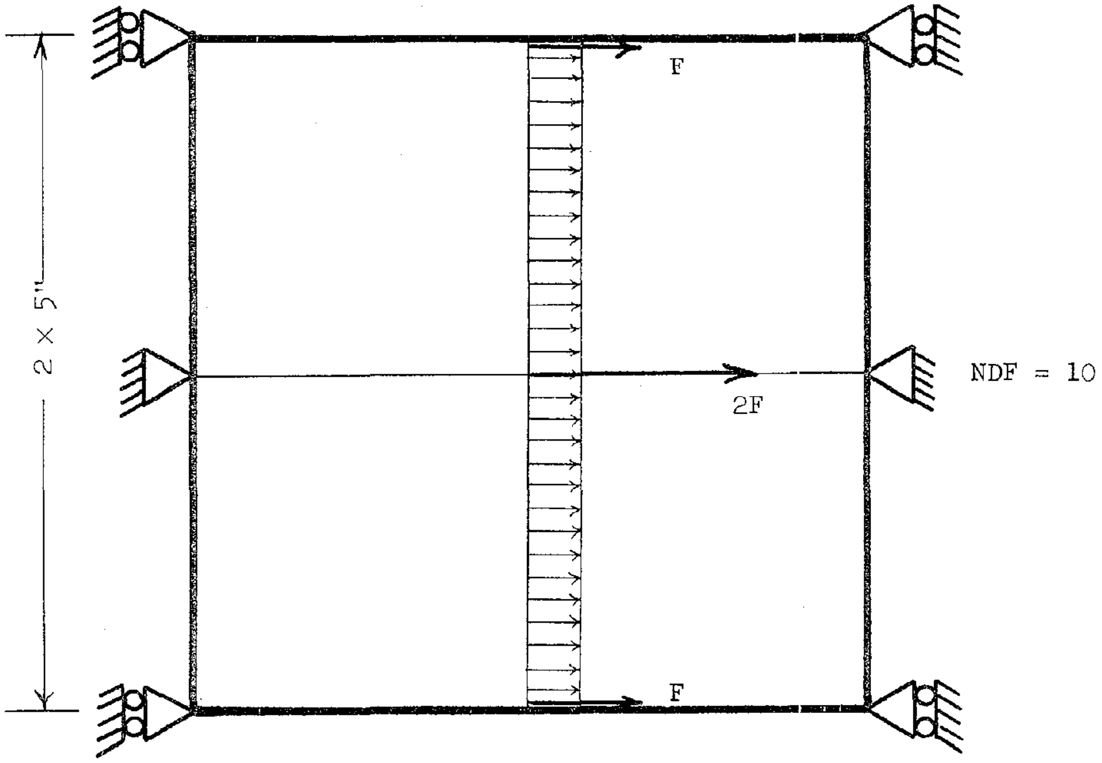


Fig. 5.4 4-Element Plate

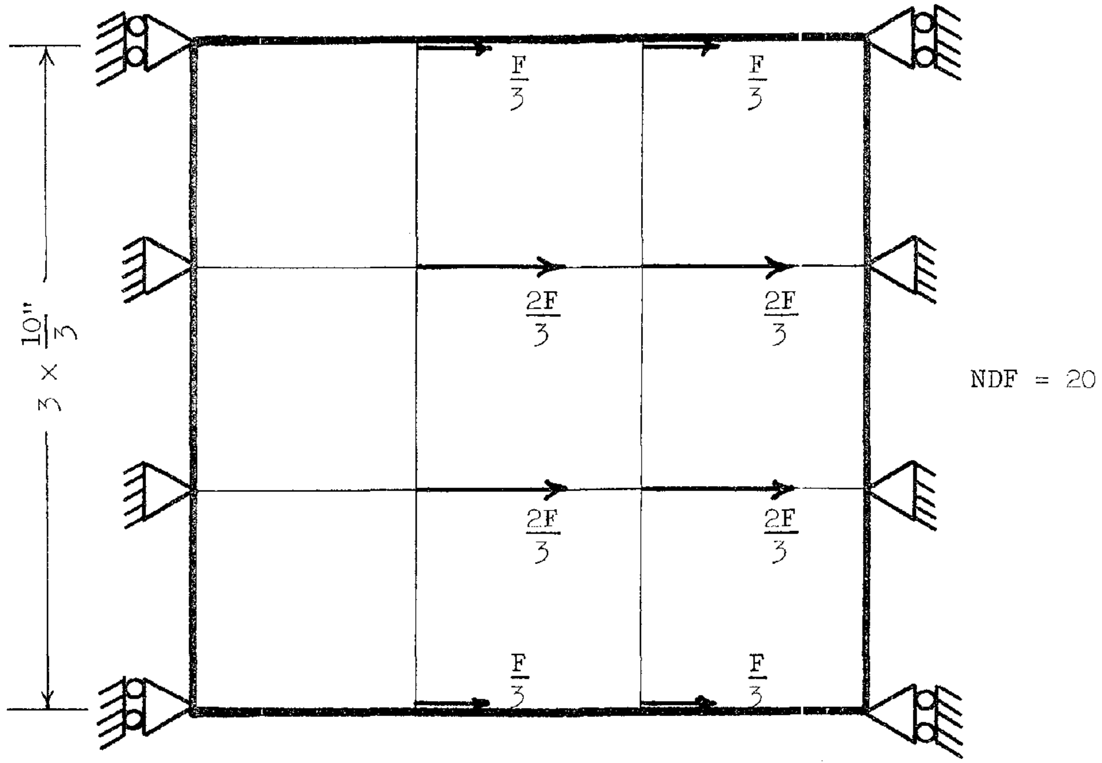


Fig. 5.5 9-Element Plate

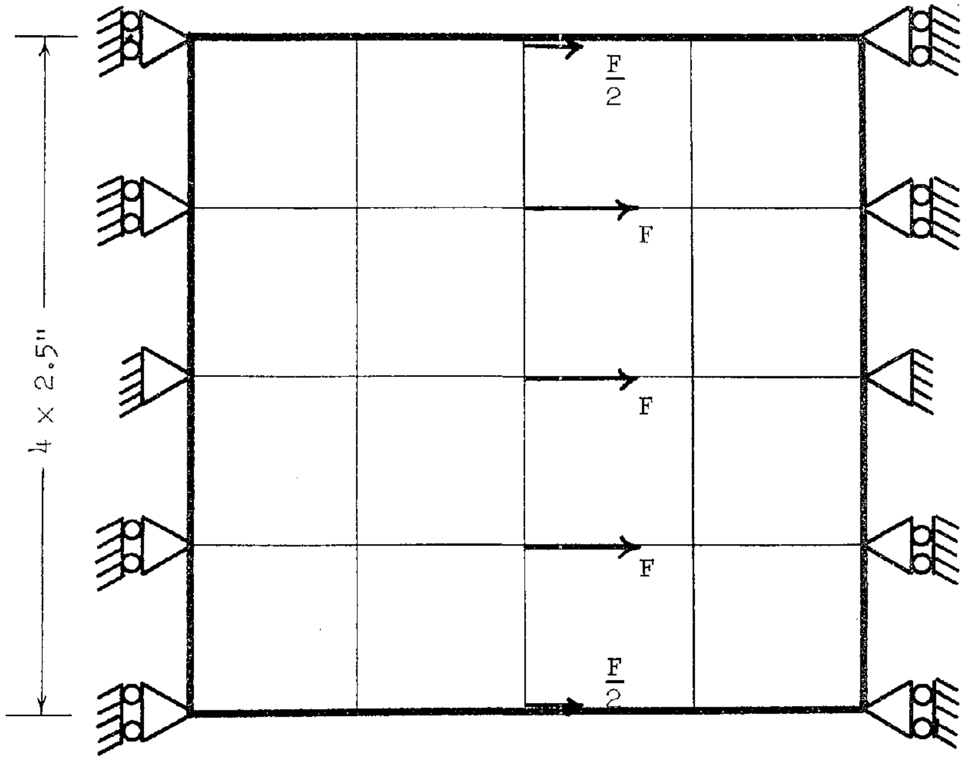


Fig. 5.6 16-Element Plate

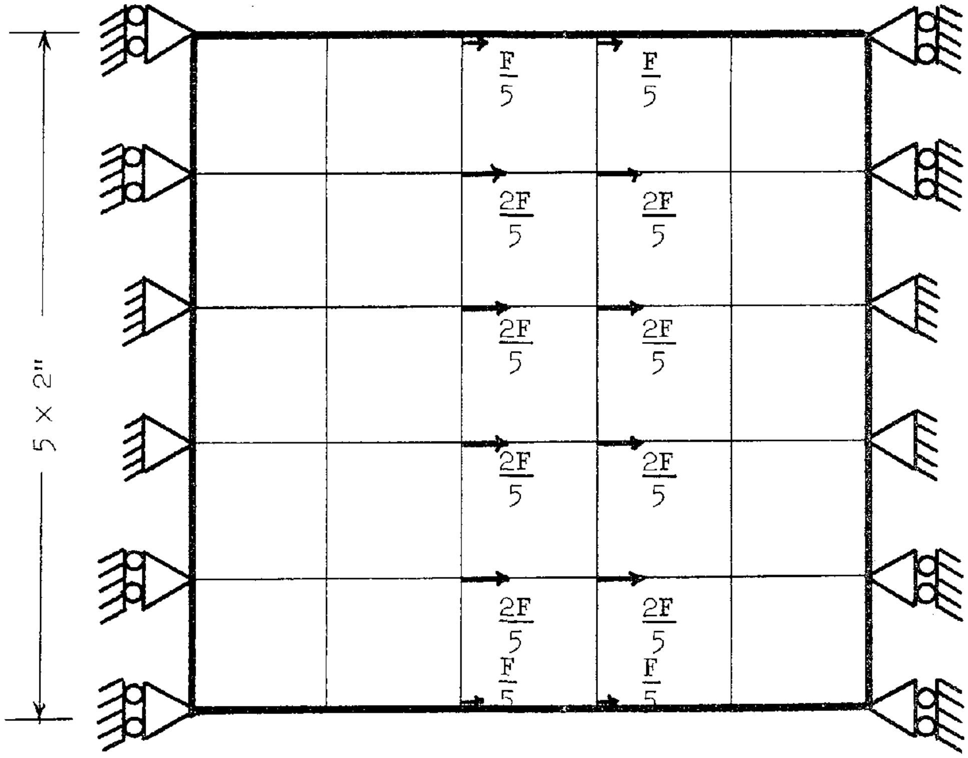


Fig. 5.7 25-Element Plate

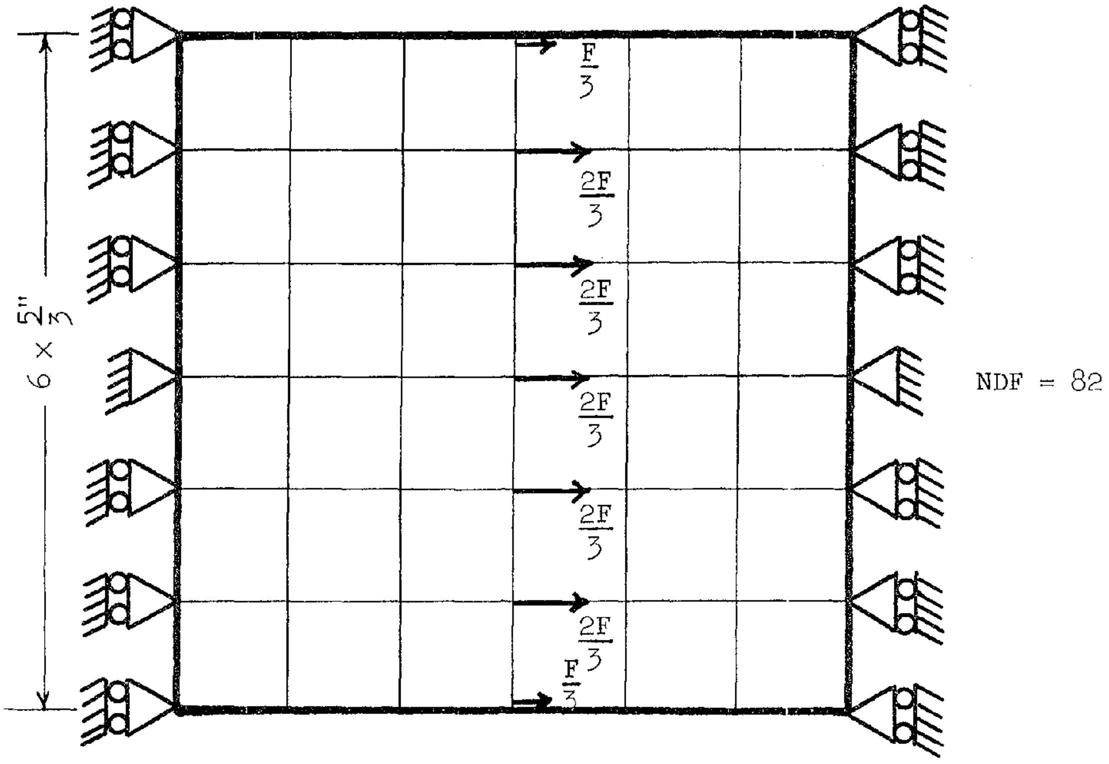


Fig. 5.8 36-Element Plate

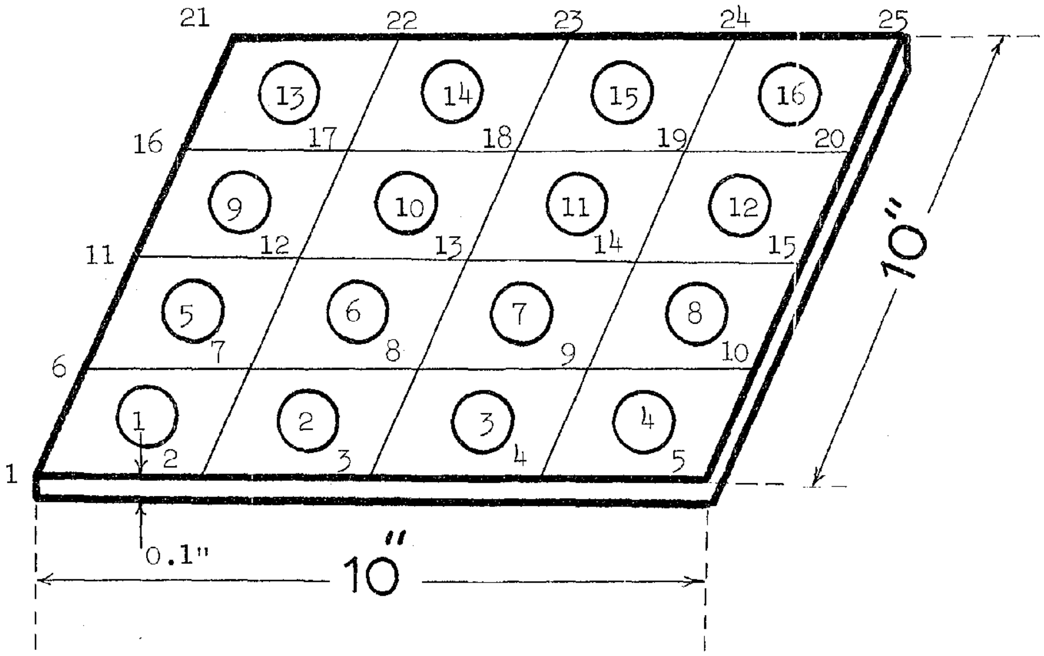


Fig. 5.9 Numbering System for the Plate

CHAPTER 6

NUMERICAL EXPERIMENTS AND RESULTS

6.1 Linear Dynamic Analysis

6.1.1 Introduction

In the following sections the four solution techniques presented in Section 4.1 are compared for efficiency, accuracy, and stability. Load F for the example is taken equal to 10 lb, and all results for linear analysis were obtained under the WATFIV compiler. Figure 6.1 shows a typical response plot of the plate in the direction of loading for the 16-element case. On this plot, where a time step of 10^{-6} seconds is used, results of all methods are indistinguishable from each other. Figure 6.2 shows the response for the normal mode method for the same case but with a larger time step of 5×10^{-6} sec. For this size of time step, the other three methods give similar results that are practically the same.

6.1.2 Efficiency

Table 6.1 shows the efficiency comparisons for the four finite element networks shown in Section 5.5 (Figs. 5.4 through 5.7), using one hundred equal time steps of 10^{-6} sec. These results are plotted in Fig. 6.3 as computation time versus number of degrees of freedom. Direct extrapolation methods and the central difference procedure

NE = 16

DT = 10^{-6} sec

NDT = 100

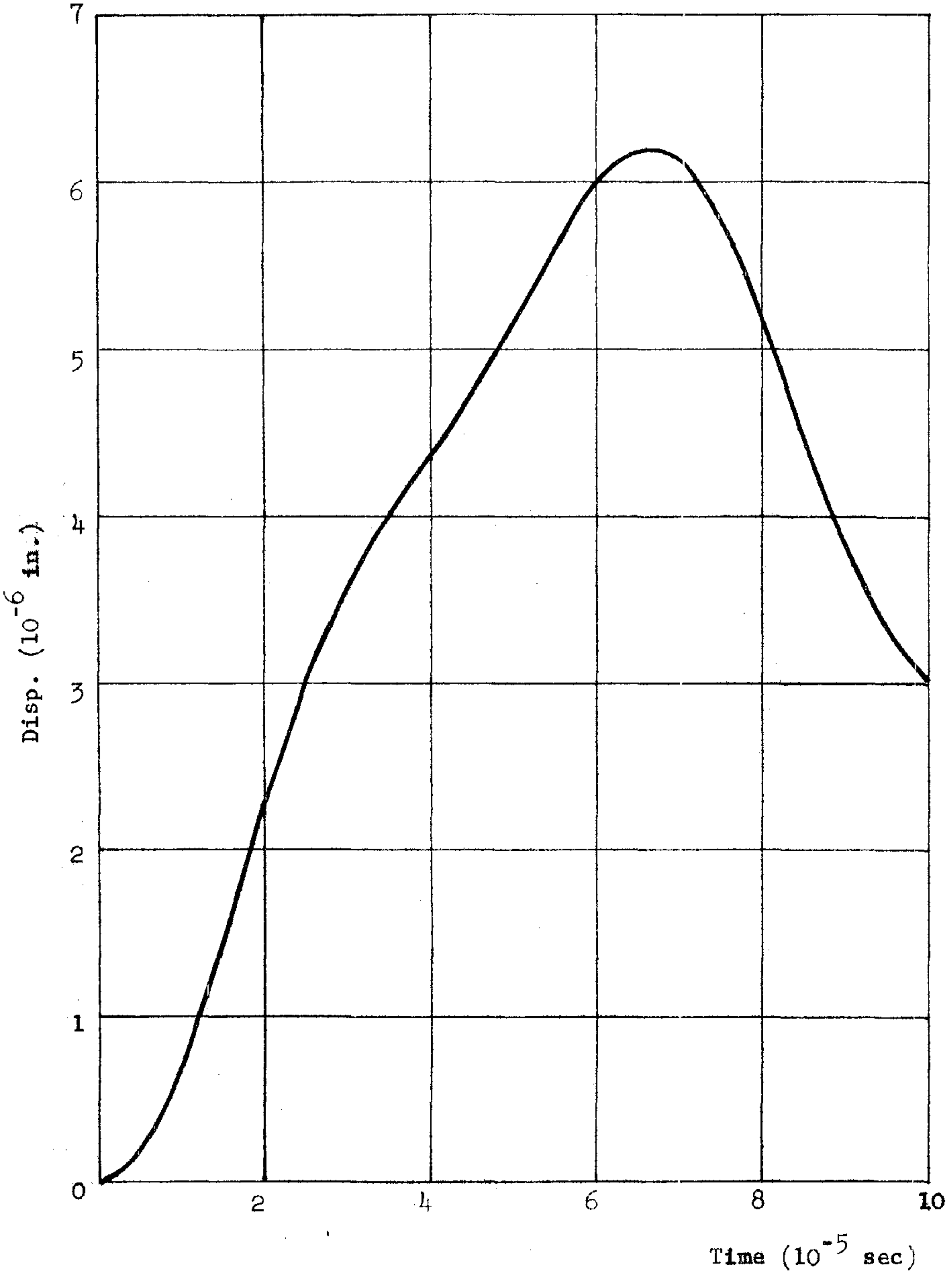


Fig. 6.1 Response of Center of the Plate in the Direction of Loading

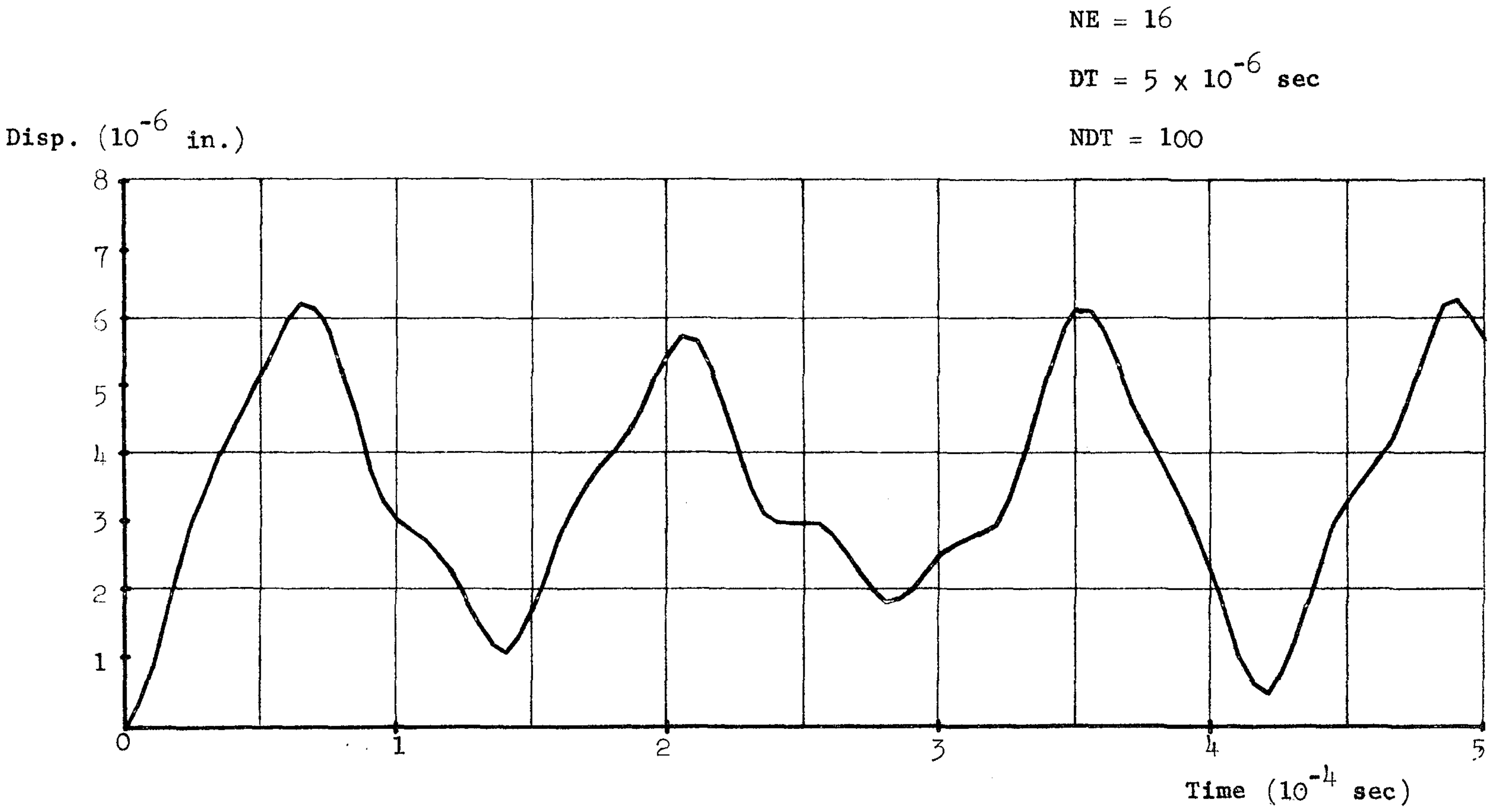


Fig. 6.2 Response of Center of the Plate in the Direction of Loading by Normal Mode Method

TABLE 6.1 Computation Times (sec) for Linear Algorithms

Solution Technique	4 Elements NDF = 10	9 Elements NDF = 20	16 Elements NDF = 38	25 Elements NDF = 56
Direct Extrapolation	2.16	6.86	22.72	49.97
Direct Extrapolation With Incremental Displacements	2.20	7.23	22.36	48.28
Central Difference	1.85	6.75	23.69	52.06
Iteration With Trapezoidal Rule	3.76	13.71	47.84	
Normal Mode	6.30	24.00	98.16	

NDF = Number of Degrees of Freedom

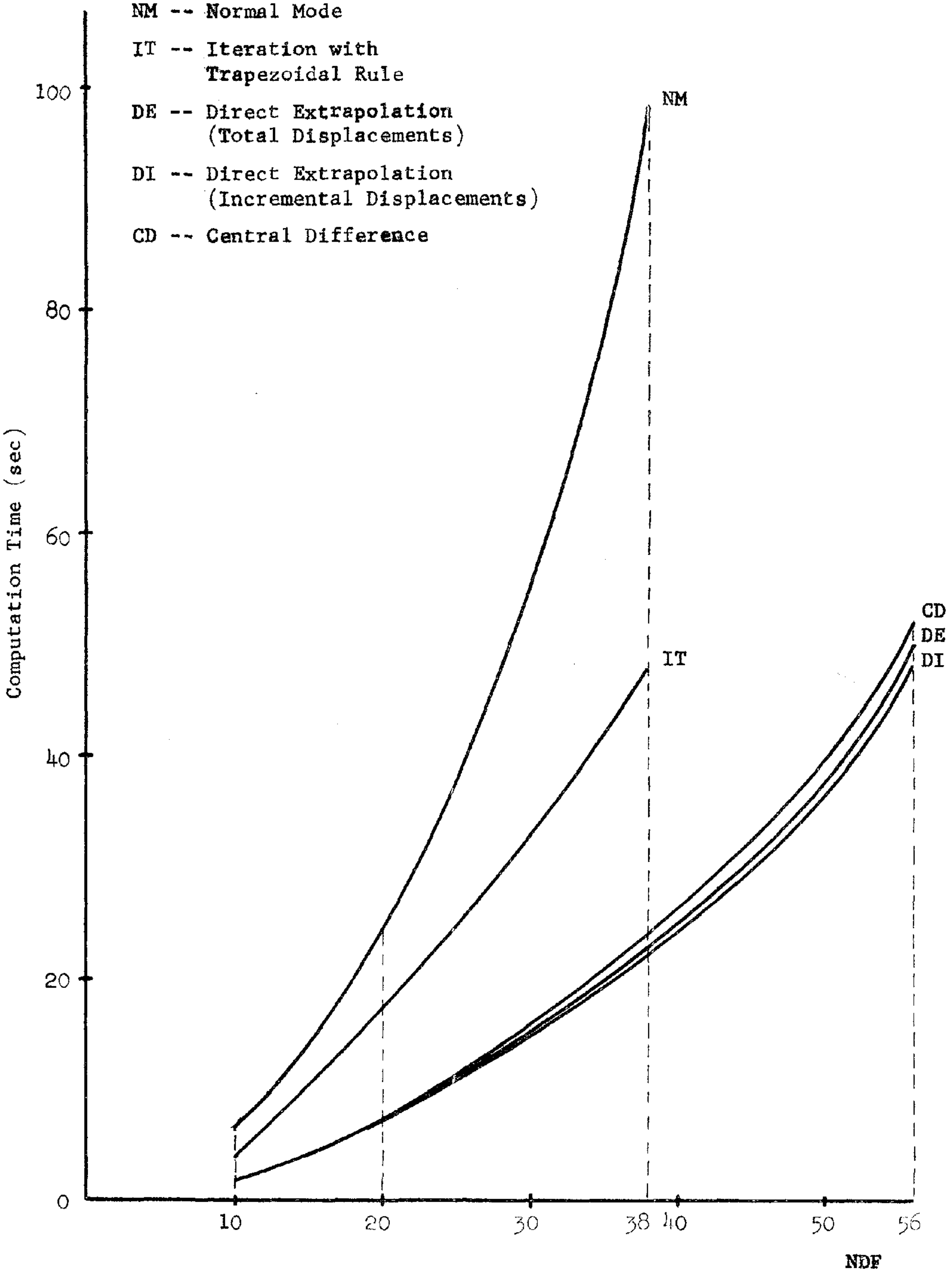

NM -- Normal Mode
IT -- Iteration with Trapezoidal Rule
DE -- Direct Extrapolation (Total Displacements)
DI -- Direct Extrapolation (Incremental Displacements)
CD -- Central Difference
NM
IT
CD
DE
DI
Computation Time (sec)
100
80
60
40
20
0
10
20
30
38
40
50
56
NDF

Figure 6.3

require approximately the same amount of computation time. As expected, the normal mode procedure is the slowest of all methods studied here.

Laswell [32] compared the same solution techniques for linear problems using a plate bending example. He found the central difference procedure to be the most efficient of all methods. However, in the present study the direct linear extrapolation methods are seen to be slightly more efficient than the central difference procedure. This discrepancy is probably due to improved coding in the present work.

6.1.3 Accuracy

For the example problem, the normal mode method provides a reliable solution against which other solutions can be compared. Table 6.2 shows the value of maximum deflection in the direction of loading at the center of the plate for different methods. These results are for 4-element and 16-element meshes. They are based on a time step of 10^{-6} sec and for a time range from zero to 10^{-4} sec. By examining the maximum deflection for different methods, we observe that for a small number of degrees of freedom all methods are of comparable accuracy. However, as the number of degrees of freedom increases, direct linear extrapolation is less accurate. One would suspect that this is due to round-off error. In order to investigate this problem, the program for linear analysis (RESPPSQ4) was changed from single precision to double precision arithmetic. Table 6.3 contains results for the 16-element mesh, using double precision arithmetic. It can be seen that the direct linear extrapolation procedure is more sensitive to round-off error than the other methods studied.

TABLE 6.2 Maximum Deflections (10^{-6} in.)

METHOD	4 Elements NDF = 10	16 Elements NDF = 38
Direct Extrapolation	6.32501	6.24371
Direct Extrapolation With Incremental Displacements	6.32390	6.23772
Central Difference	6.32937	6.27483
Iteration with Trapezoidal Rule	6.32846	6.27099
Normal Mode	6.32863	6.27391

NDF = Number of Degrees of Freedom

TABLE 6.3 Maximum Deflection and Execution Time for 16-Element Mesh Using Double Precision Arithmetic

Method	Maximum Deflection (10^{-6} in.)	Computation Time (sec)
Direct Extrapolation	6.27106	22.77
Direct Extrapolation with Incremental Displacements	6.27106	23.26
Central Difference	6.27517	23.72
Iteration with Trapezoidal Rule	6.27101	47.43
Normal Mode	6.27381	95.89

TABLE 6.4 Stability Results for 4-Element Model

Method	Size of Time Step (sec)				
	0.000001	0.00001	0.00003	0.00005	0.0001
Direct Extrapolation	STABLE	STABLE	STABLE	STABLE	STABLE
Central Difference	STABLE	STABLE	STABLE	UNSTABLE	UNSTABLE
Iteration with Trapezoidal Rule	STABLE	STABLE	UNSTABLE	UNSTABLE	UNSTABLE

6.1.4 Stability

Because of limitations of available computer time, the stability check was done only for the 4-element mesh, and the results appear in Table 6.4. The size of the time step was increased gradually from 10^{-6} sec to 10^{-4} sec for each numerical integration method in order to determine its stability limit. It is seen that the stability limit for two-cycle iteration with the trapezoidal rule is the smallest of all methods studied. Because no approximation formula is used in the normal mode method, a stability check is not applicable to this procedure.

6.2 Nonlinear Dynamic Analysis

6.2.1 Introduction

Due to the fact that the cost of nonlinear dynamic analysis is significantly higher than that of linear analysis, a prudent strategy for numerical experimentation was adopted in order not to exceed the limited computer time available. Accordingly, we first compare the performance of explicit and implicit methods separately (in Sections 6.2.2 and 6.2.3, respectively). These comparisons are based on a 4-element mesh (see Fig. 5.4) with elastoplastic material properties, neglecting geometric nonlinearity.

In Section 6.2.4, the best of the explicit methods is compared with the best of the implicit methods for larger problems (Figs. 5.5 through 5.8). Furthermore, geometric nonlinearities are also taken into account in this stage.

All of the results in Sections 6.2.2 and 6.2.3 were obtained under the WATFIV compiler. However, the FORTRAN-IVH compiler was used for the results in Section 6.2.4 because of its much faster execution time. In the diagrams on the following pages, computation times under the WATFIV and FORTRAN-IVH compilers are denoted by CTW and CTF, respectively. In order to reduce round-off error, double precision arithmetic was used throughout.

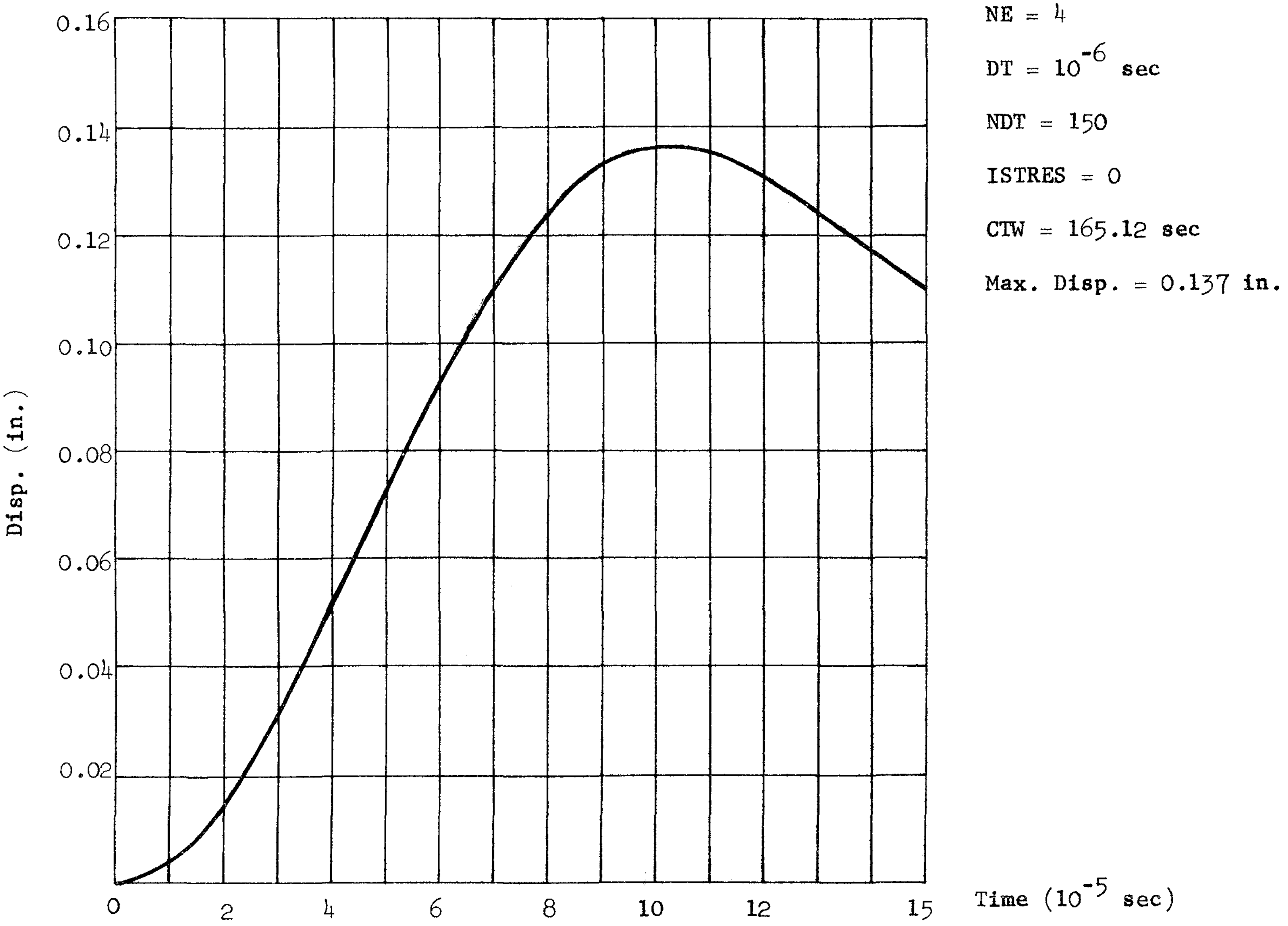
In the following sections, the load F (see Figs. 5.4 to 5.8) is taken equal to 100,000 lbs. This load is high enough to produce a considerable amount of inelastic strain. All plots are responses of the center of the plate in the direction of loading.

6.2.2 Comparison of Explicit Methods

The best method will be defined as that for which the ratio of the time step to the computation time is maximum for a given time range, while producing reasonably accurate results. At first we assume the stresses to be constant over each element, based on their values at the geometric center of the element. This approximation results in considerable saving in the overall computation time.

Figure 6.4 shows the response at the center of the plate by the central difference predictor with a time step equal to 10^{-6} seconds. For this small time step, two-cycle iteration with the trapezoidal rule and the fourth-order Runge-Kutta method produce similar results that practically coincide with the diagram in Fig. 6.4 (The maximum displacement in this figure is equal to 0.137 in.)

Figure 6.5 shows comparable results for three explicit methods for a time step of 10^{-5} sec and a time range of 3×10^{-4} sec. The first peak values for the central difference predictor, the trapezoidal rule, and the Runge-Kutta method are equal to 0.138, 0.139,



NE = 4

DT = 10^{-6} sec

NDT = 150

ISTRES = 0

CTW = 165.12 sec

Max. Disp. = 0.137 in.

Fig. 6.4 Response by Central Difference Predictor

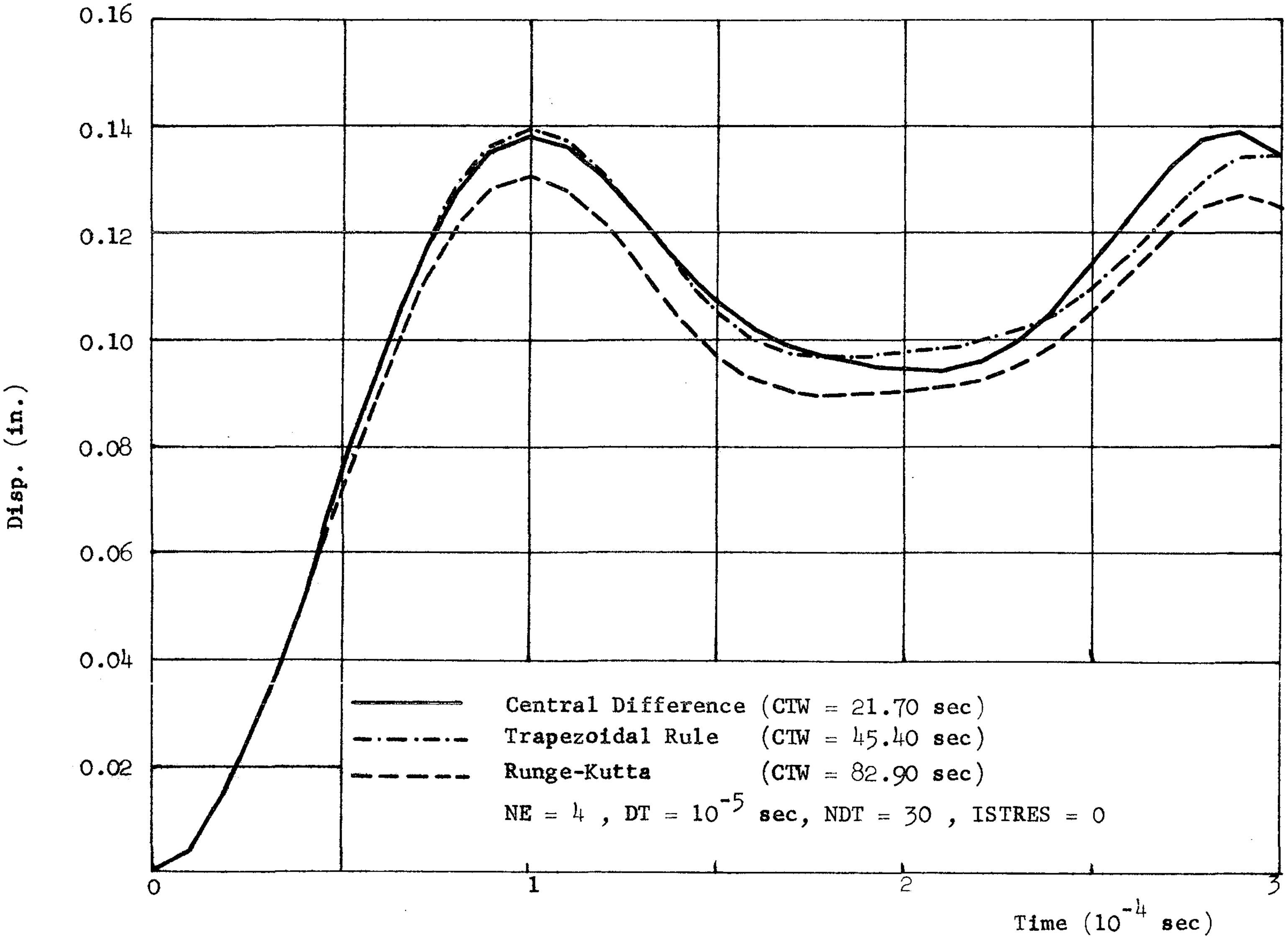


Fig. 6.5 Response by Three Explicit Methods

and 0.131 in., respectively. The computation times for these methods in the above order are approximately 22, 45, and 83 seconds. The central difference predictor apparently gives the best results in terms of accuracy and efficiency. Figure 6.6 shows the response by the central difference procedure with a time step of 10^{-5} sec. This is considered as the base case against which the results in the remainder of this section will be compared.

In order to study these methods further, we increased the time increment to 2×10^{-5} and 3×10^{-5} sec. For the first of these time steps, the response for the central difference predictor (see Fig. 6.7) becomes unstable, while the other two methods give inaccurate results. With the time increment of 3×10^{-5} sec (see Fig. 6.8), both the central difference procedure and the trapezoidal rule give unstable results, whereas the response for the Runge-Kutta method is suppressed. It is interesting to see that as we increase the time step to 4×10^{-5} and 5×10^{-5} sec, the response for the Runge-Kutta method becomes drastically suppressed (see Fig. 6.9).

Figures 6.10 and 6.11 show responses for the three explicit methods (DT = 2×10^{-5} and 3×10^{-5} sec, respectively) with the stresses evaluated at the four integration points instead of at the geometric center of each element. This change was expected to result in greater accuracy. However, it does not seem to improve the performance of either the trapezoidal rule or the Runge-Kutta method. In addition, it tends to make the results for the central difference procedure even more unstable.

From these comparisons, we conclude that among the explicit methods studied the central difference procedure is the best. With this method,

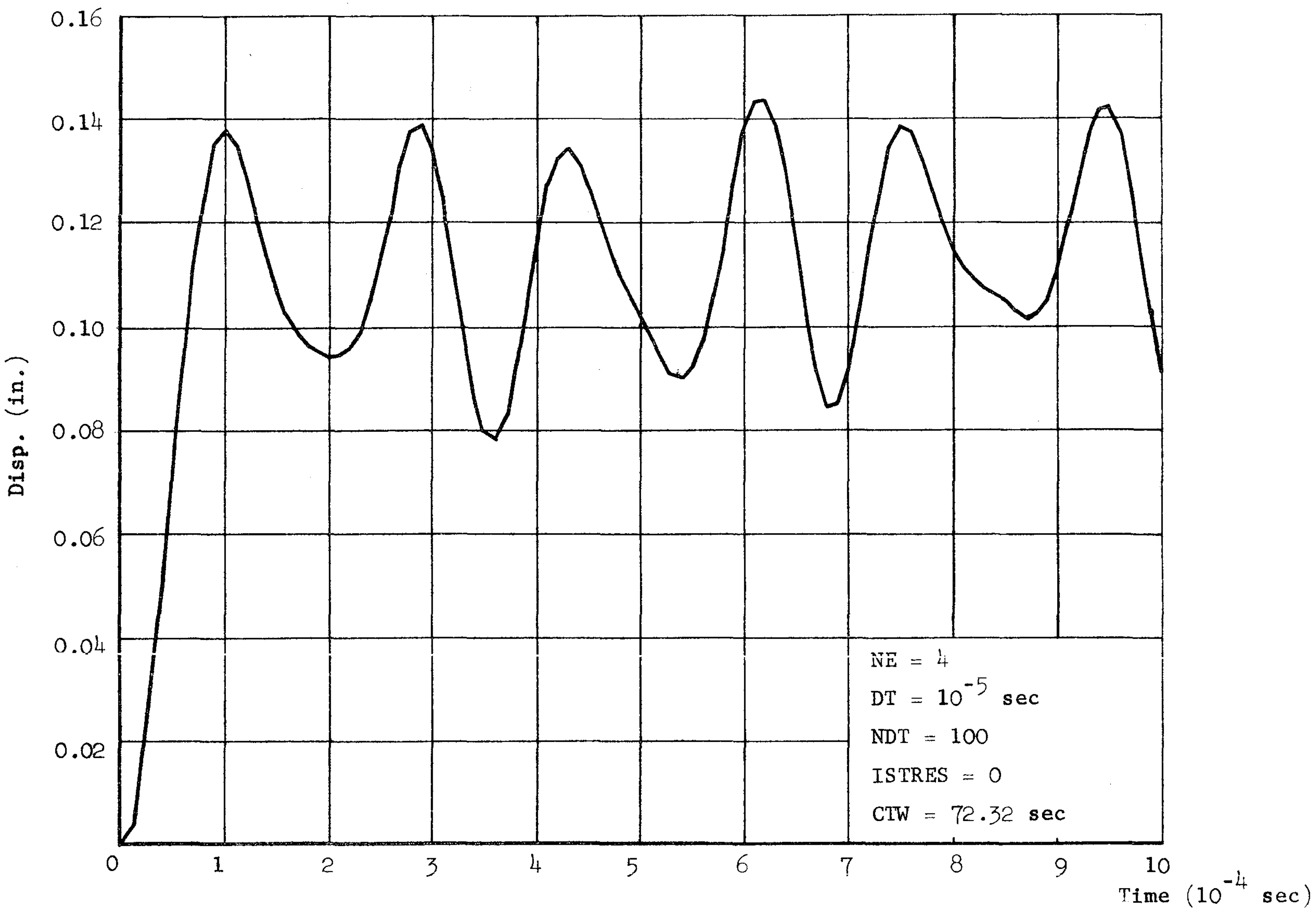


Fig. 6.6 Response by Central Difference Predictor

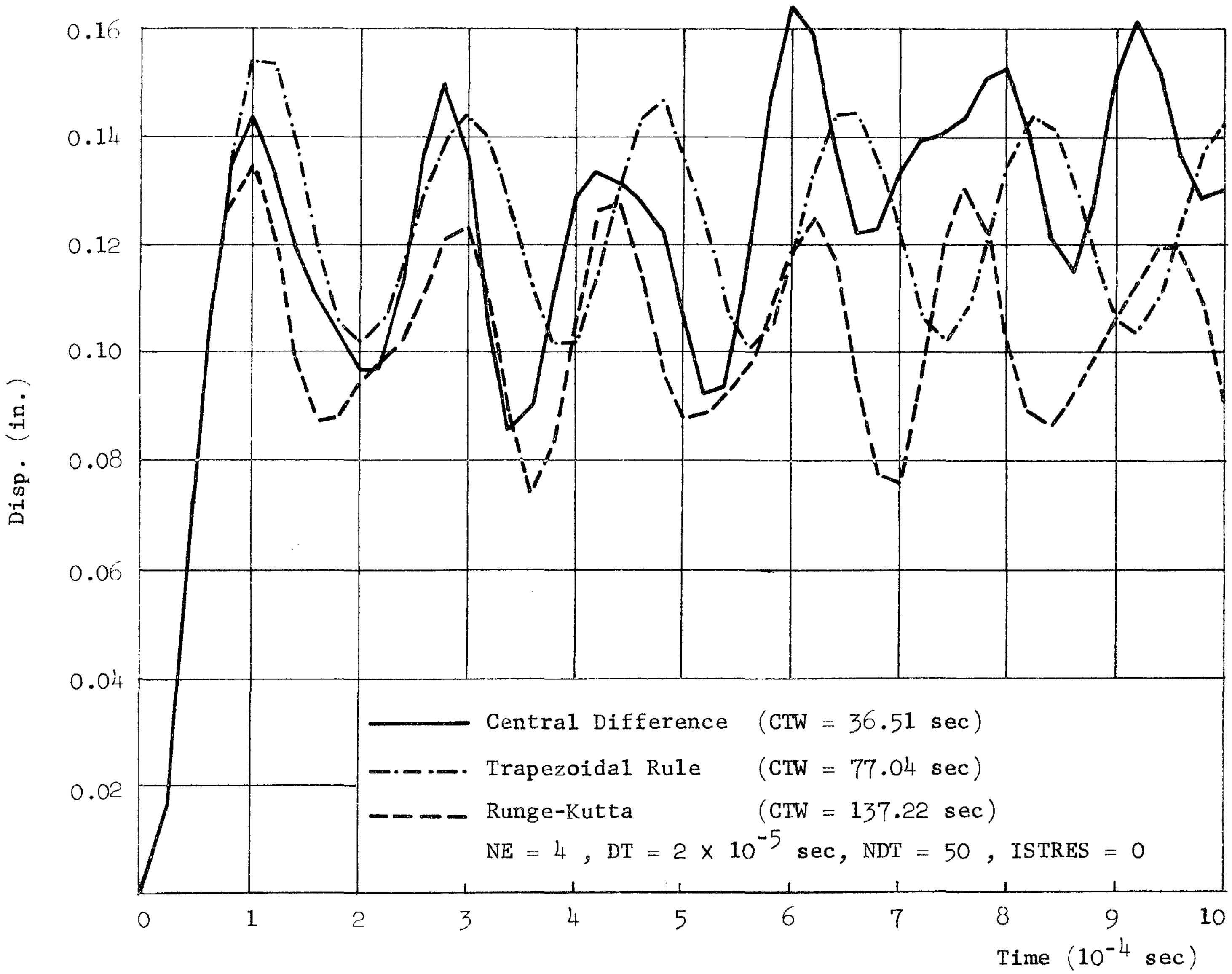


Fig. 6.7 Response by Three Explicit Methods

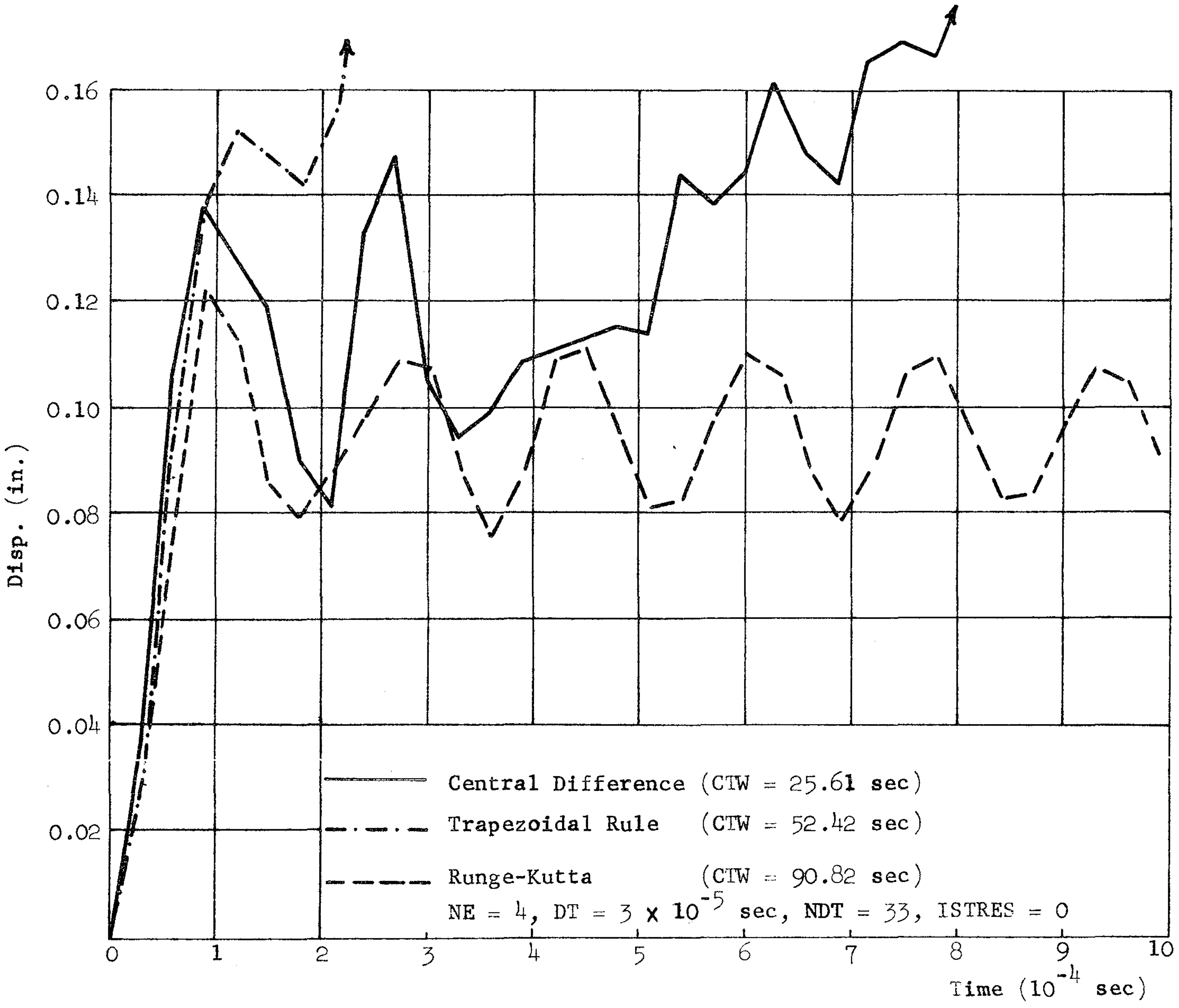


Fig. 6.8 Response by Three Explicit Methods

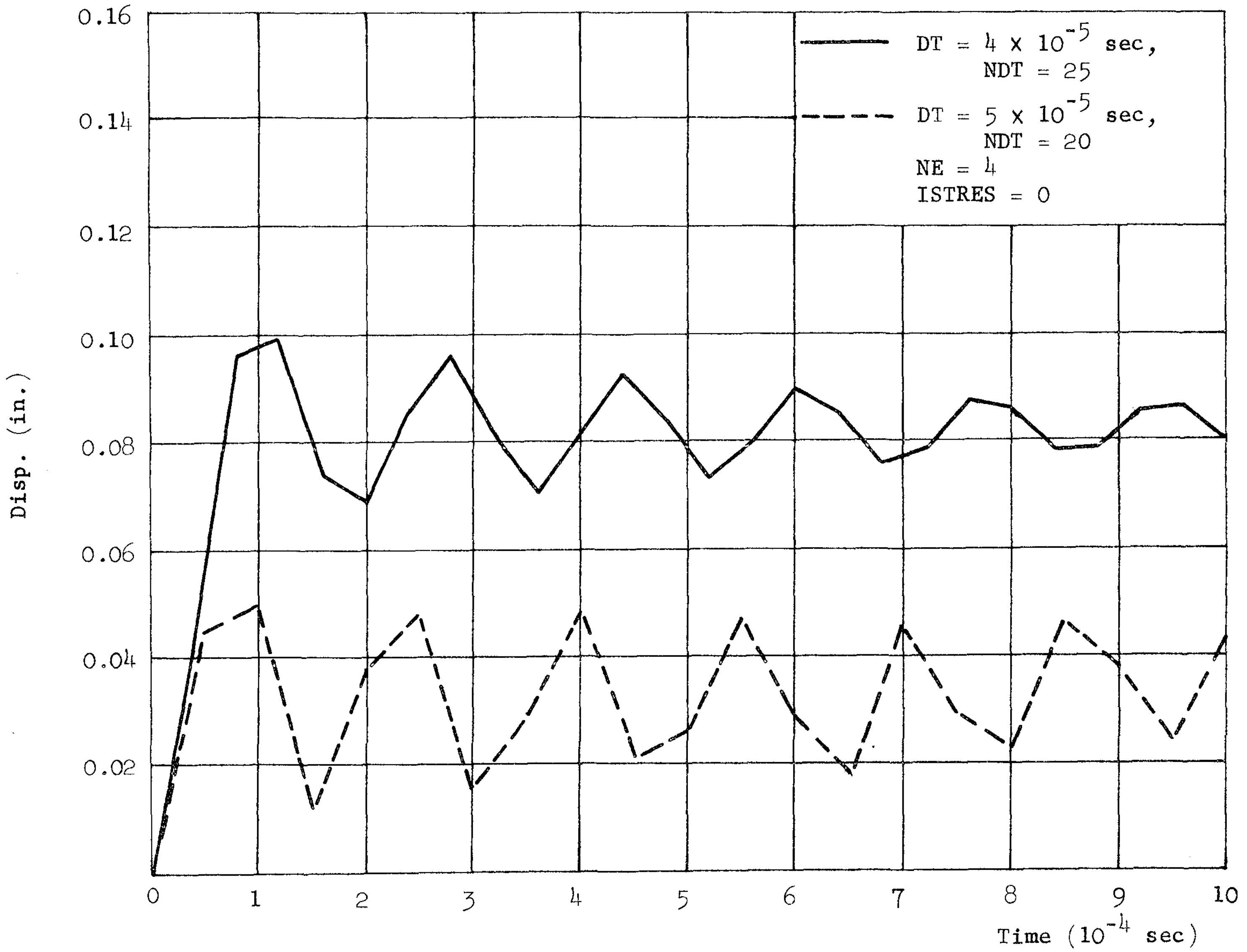


Fig. 6.9 Response by Runge-Kutta Method

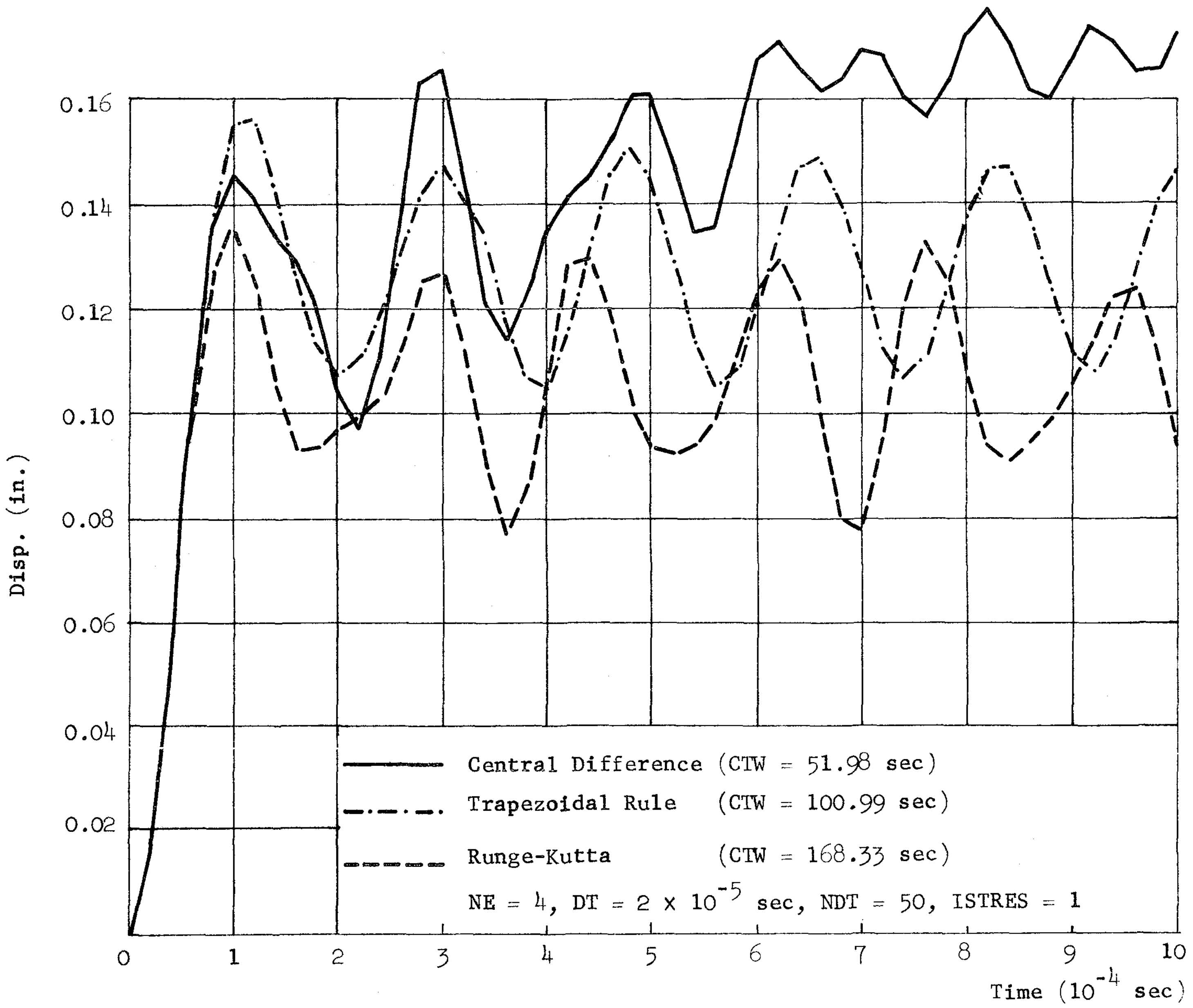


Fig. 6.10 Response by Three Explicit Methods

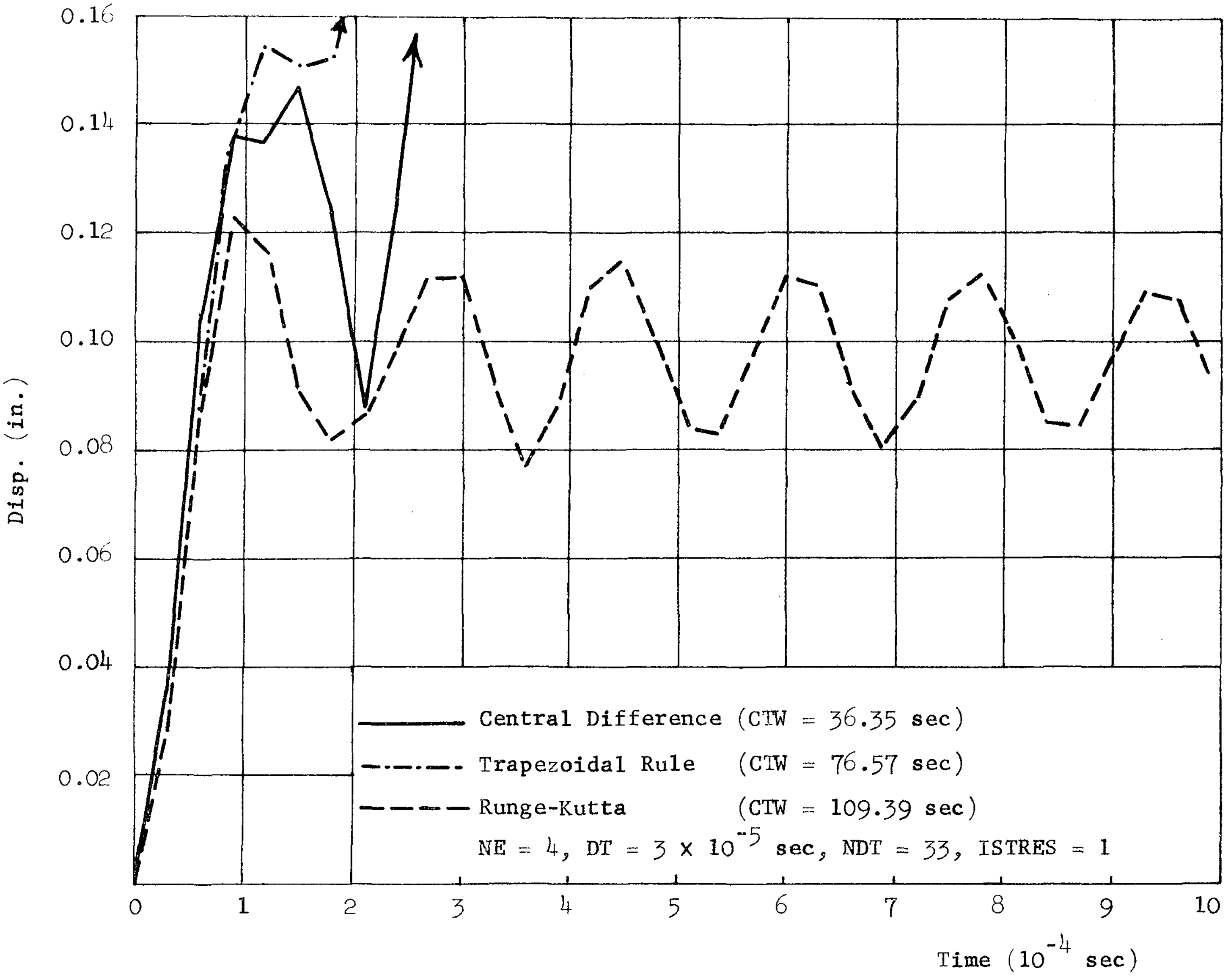


Fig. 6.11 Response by Three Explicit Methods

we should calculate the stresses only at the geometric center of the element and take them to be constant over the element.

6.2.3 Comparison of Implicit Methods

Turning now to implicit methods, we first assume the stresses to be constant throughout each element. The results for the Newmark-Beta method (with $\beta = 0.25$ and $\gamma = 0.5$), the Houbolt procedure, and Park's stiffly-stable method are plotted in Fig. 6.12. The time step selected is 10^{-5} sec, and the range is from zero to 10^{-3} sec. Comparing these results with those of the central difference procedure in Figs. 6.4 and 6.5, we observe that the responses are significantly suppressed for all three implicit methods. In order to improve the accuracy of the response, the time step was reduced to the value of 10^{-6} sec. In this case, the responses of all three methods coincide with each other, as shown in Fig. 6.13. This diagram corresponds to that for the central difference procedure in Fig. 6.4. However, the maximum displacement in Fig. 6.13 is 0.133 in., compared with 0.137 in. in Fig. 6.4. It is likely that the inaccuracy in Fig. 6.13 is due to the way the stresses are approximated within each element. To investigate this matter, the stresses were evaluated at the four integration points (parameter ISTRES = 1) instead of being evaluated only at the geometric center of the element. In this case, with a time step of 10^{-6} sec, the results of the three implicit methods coincide identically with that for the central difference procedure in Fig. 6.4.

Figure 6.14 shows the response curves with a time step of 10^{-5} sec and ISTRES = 1 . These plots are comparable to that for the central difference predictor in Fig. 6.6. The first maximum for the Newmark (with

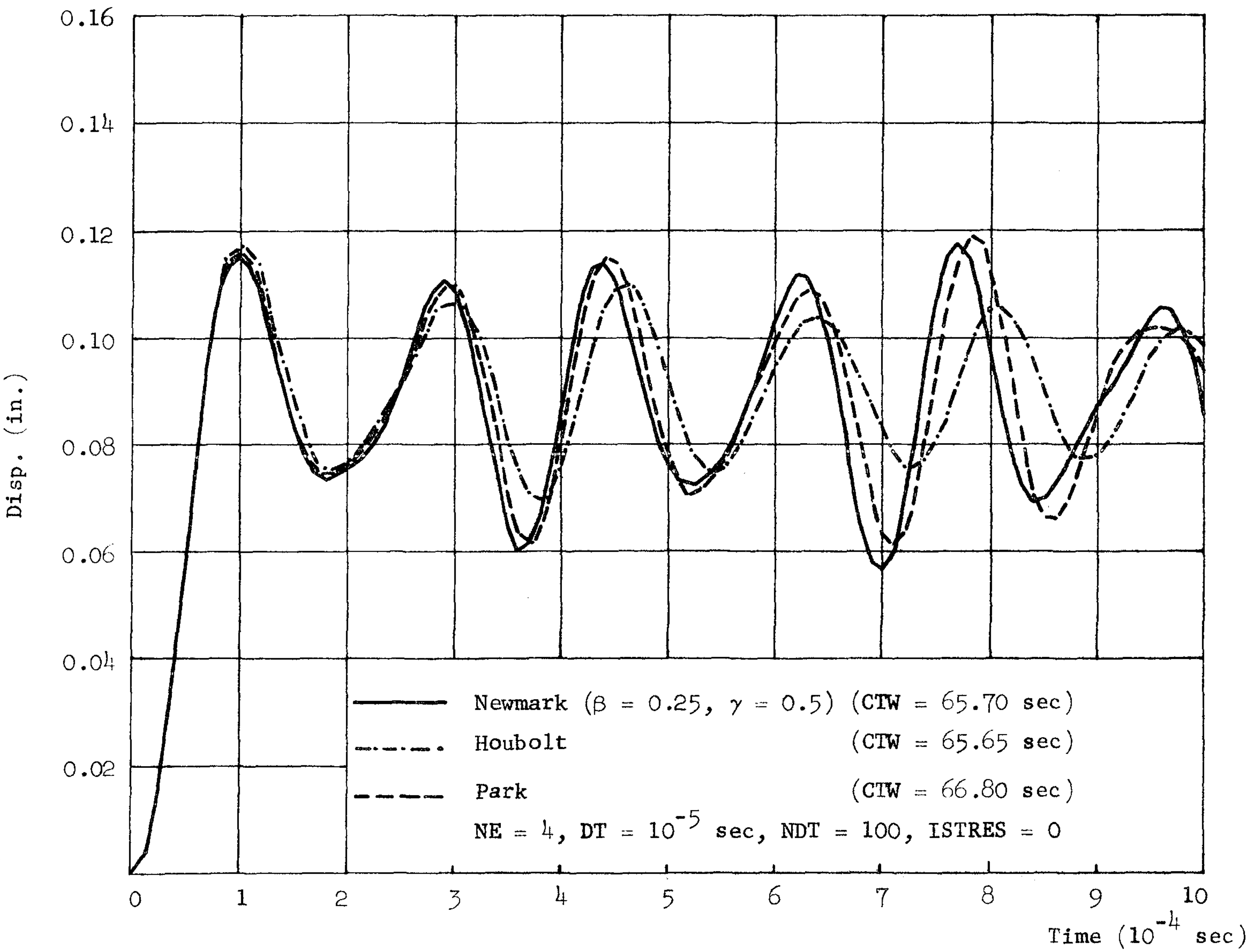
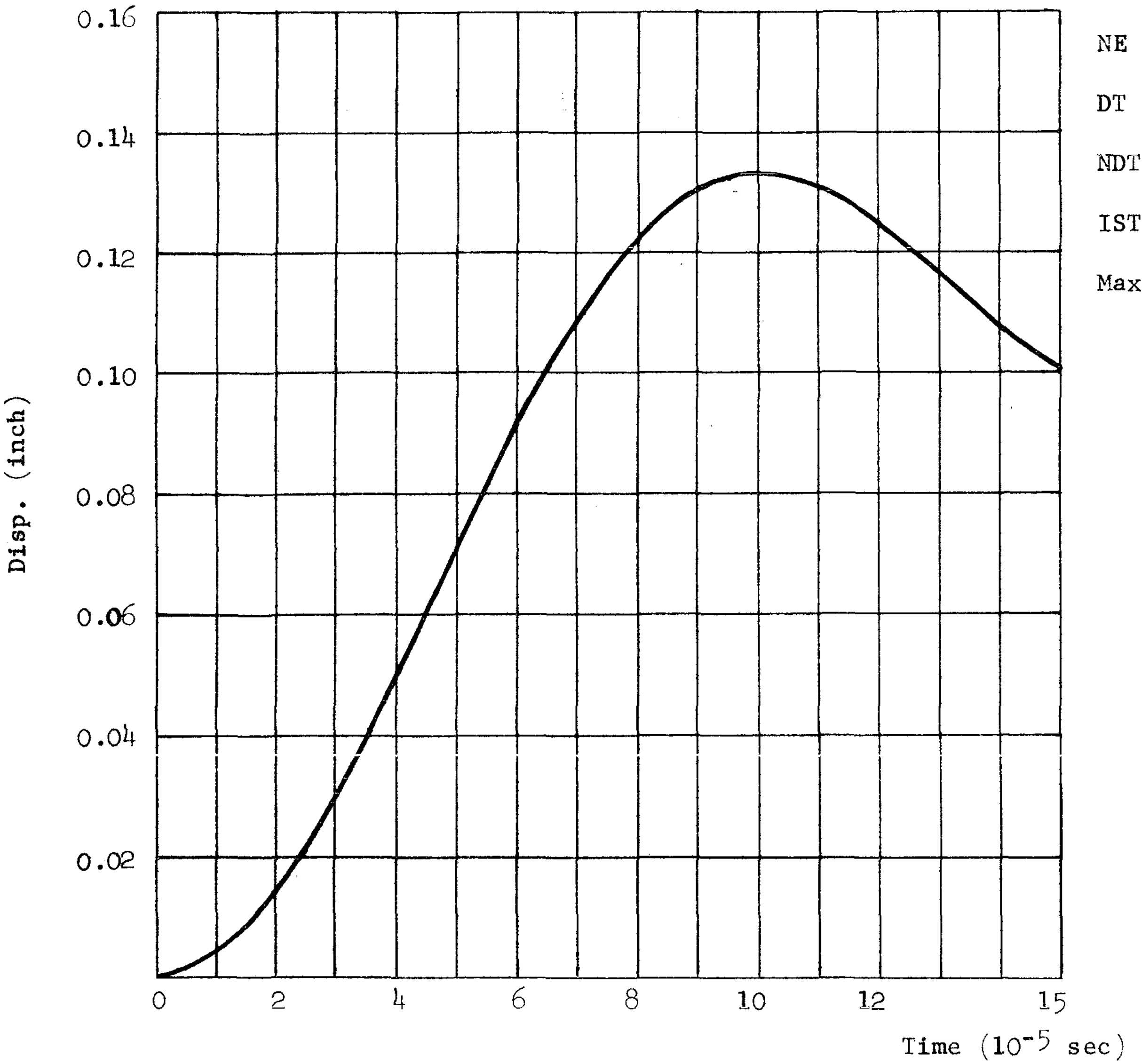


Fig. 6.12 Response by Three Implicit Methods



NE = 4

DT = 10^{-6} sec

NDT = 150

ISTRES = 0

Max. Disp. = 0.133 in.

Fig. 6.13 Response by **Implicit** Methods

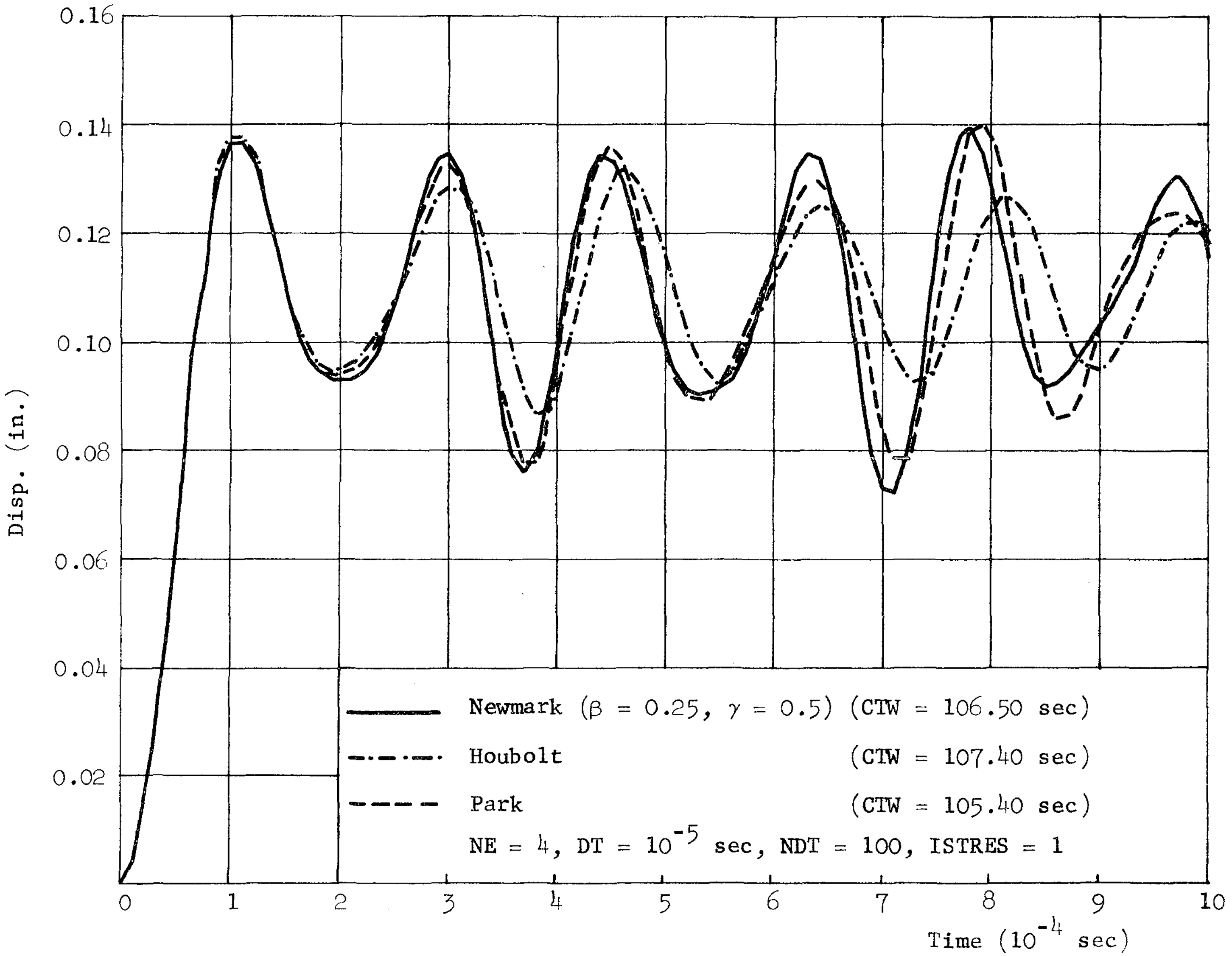


Fig. 6.14 Response by Three Implicit Methods

$\beta = 0.25$ and $\gamma = 0.5$), Houbolt, and Park methods are 0.137, 0.138, and 0.137 in., respectively. Consequently, it is concluded that for the implicit procedures, the stresses should be evaluated at the spatial integration points. Otherwise, extremely small time steps must be used in order to obtain accurate results. Therefore, in the remainder of this work (for implicit methods) we evaluate the stresses at the integration points.

Computation times for the Newmark (with $\beta = 0.25$ and $\gamma = 0.5$), Houbolt, and Park methods in Fig. 6.14 are 106, 107, and 105 sec, respectively, which are close to each other. However, some damping is observed in the response for the Houbolt procedure, and the Newmark average acceleration method tends to be unstable. Compared with the central difference procedure, all three methods apparently show an increased time interval between successive peaks. This increase is largest for the Houbolt procedure, and smallest for the Newmark method. When the time step is increased, these characteristics are magnified. Figure 6.15 shows the responses for the three implicit methods with a time step of 3×10^{-5} sec, for which the Park method appears to give marginally better results.

As discussed in Section 2.3, artificial damping can be introduced in the Newmark-Beta method by the δ control parameter. The dashed curves in Figs. 6.16 and 6.17 show the results by the Newmark method with $\delta = 0.05$ ($\beta = 0.276$ and $\gamma = 0.55$). It is seen that the stability is improved at the expense of some loss of accuracy due to damping (or suppression) of the response. The first maximum in Fig. 6.16 for the Newmark method with $\beta = 0.276$ and $\gamma = 0.55$ is 0.135 in. compared with 0.137 in. for the previous solution in Fig. 6.4.

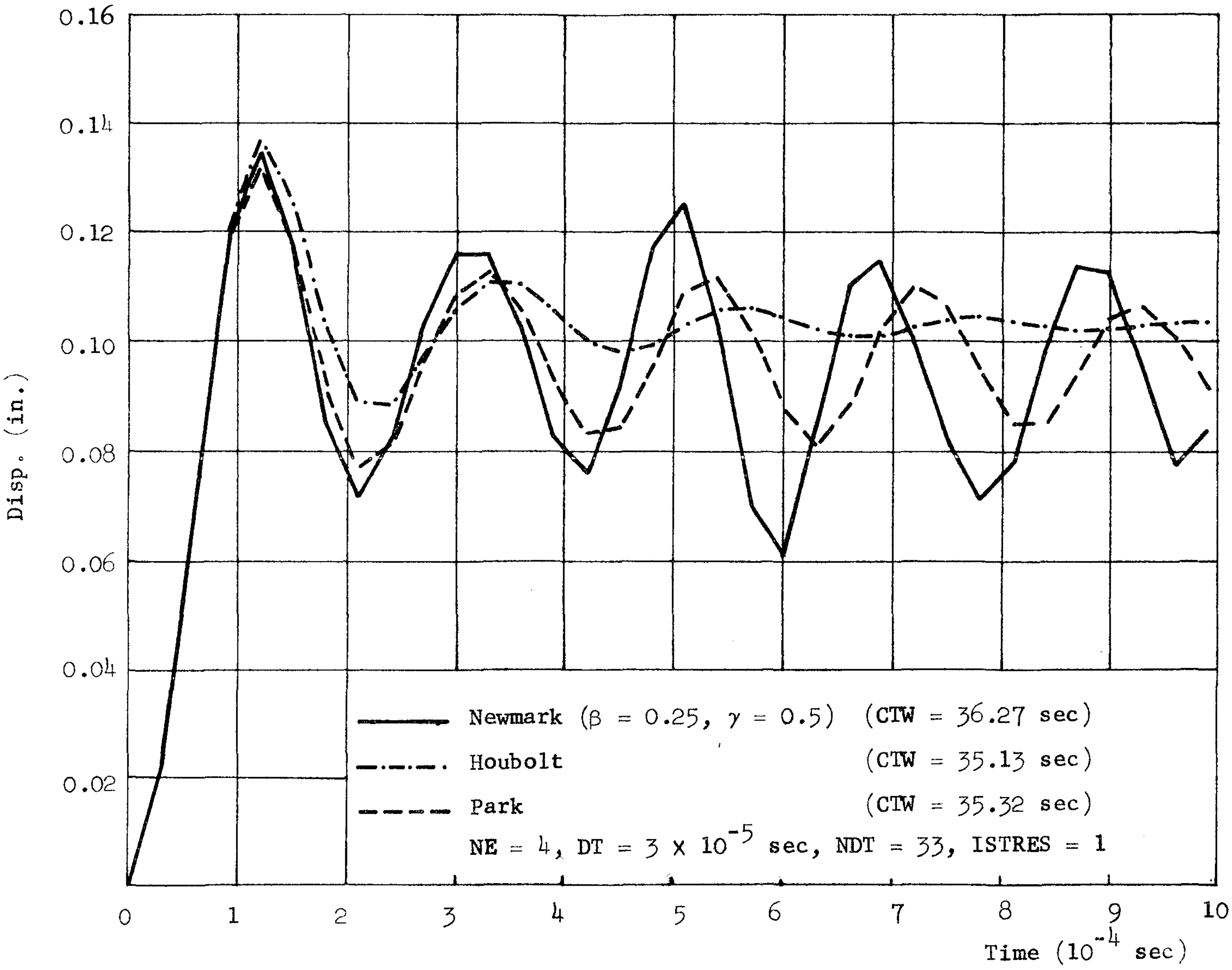


Fig. 6.15 Response by Three Implicit Methods

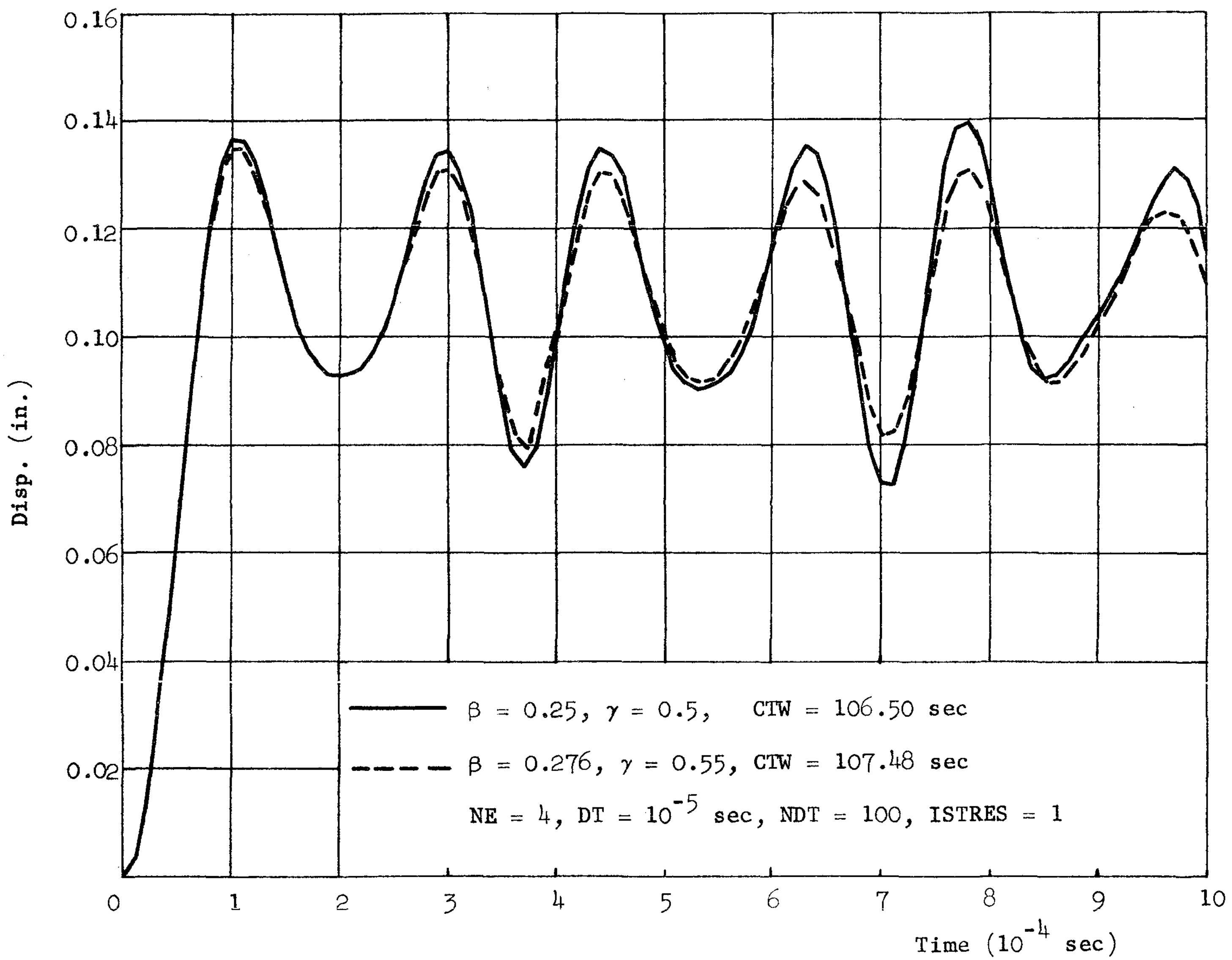


Fig. 6.16 Response by Newmark Method

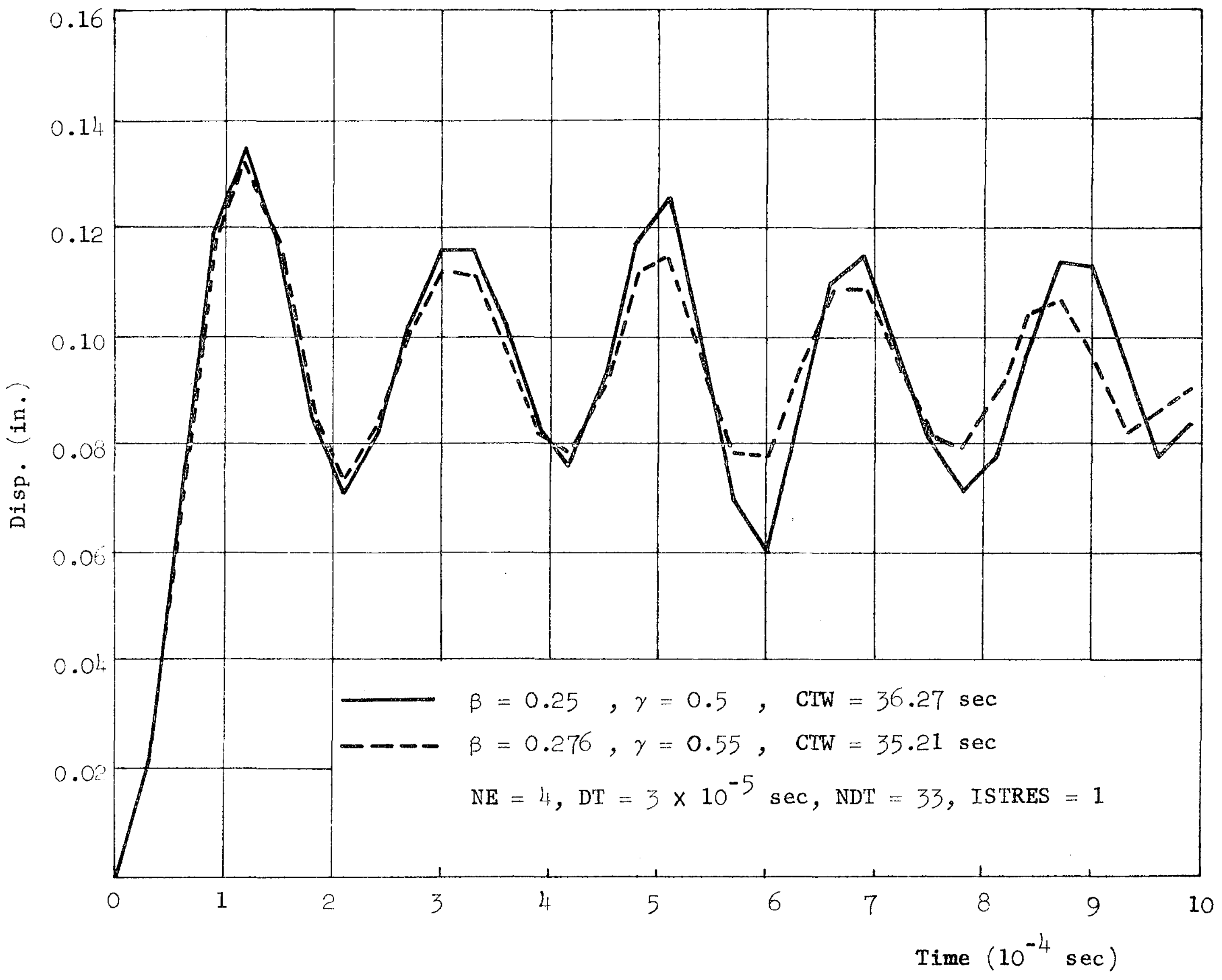


Fig. 6.17 **Response by Newmark Method**

Figures 6.18 and 6.19 show the results for the three implicit methods with equilibrium iterations for a time step of 2×10^{-5} sec. The maximum number of iterations is limited to five, and a convergence tolerance is taken equal to 0.001. By inspecting these plots, we observe that Park's method gives somewhat better results. The Newmark method with $\delta = 0$ ($\beta = 0.25$ and $\gamma = 0.5$) is barely stable (see Fig. 6.18). The result is improved by choosing a value of δ equal to 0.05 ($\beta = 0.276$ and $\gamma = 0.55$). However, the computation time is increased by approximately a factor of two (see Fig. 6.19) in order to satisfy the tolerance of 0.001.

Evidently, for large time steps, the process of constant stiffness iterations does not improve the results drastically. In such cases, one may modify the stiffnesses in each cycle of iteration (Newton-Ralphson method). This would probably improve the results to some extent, but the computation time would increase significantly.

Based on the above comparisons, we can conclude that the performance of Park's stiffly-stable method is the best of the three implicit methods studied in this investigation. In the following section, Park's method will be compared with the explicit central difference procedure.

6.2.4 Comparison of the Best Explicit Method with the Best Implicit Method

Table 6.5 shows the computation times for the four finite element networks shown previously in Figs. 5.4, 5.5, 5.6, and 5.8 using forty equal time steps of 10^{-5} sec. These times are plotted in Fig. 6.20. It is seen that when geometric nonlinearities are neglected, the computation time for the central difference procedure is considerably less

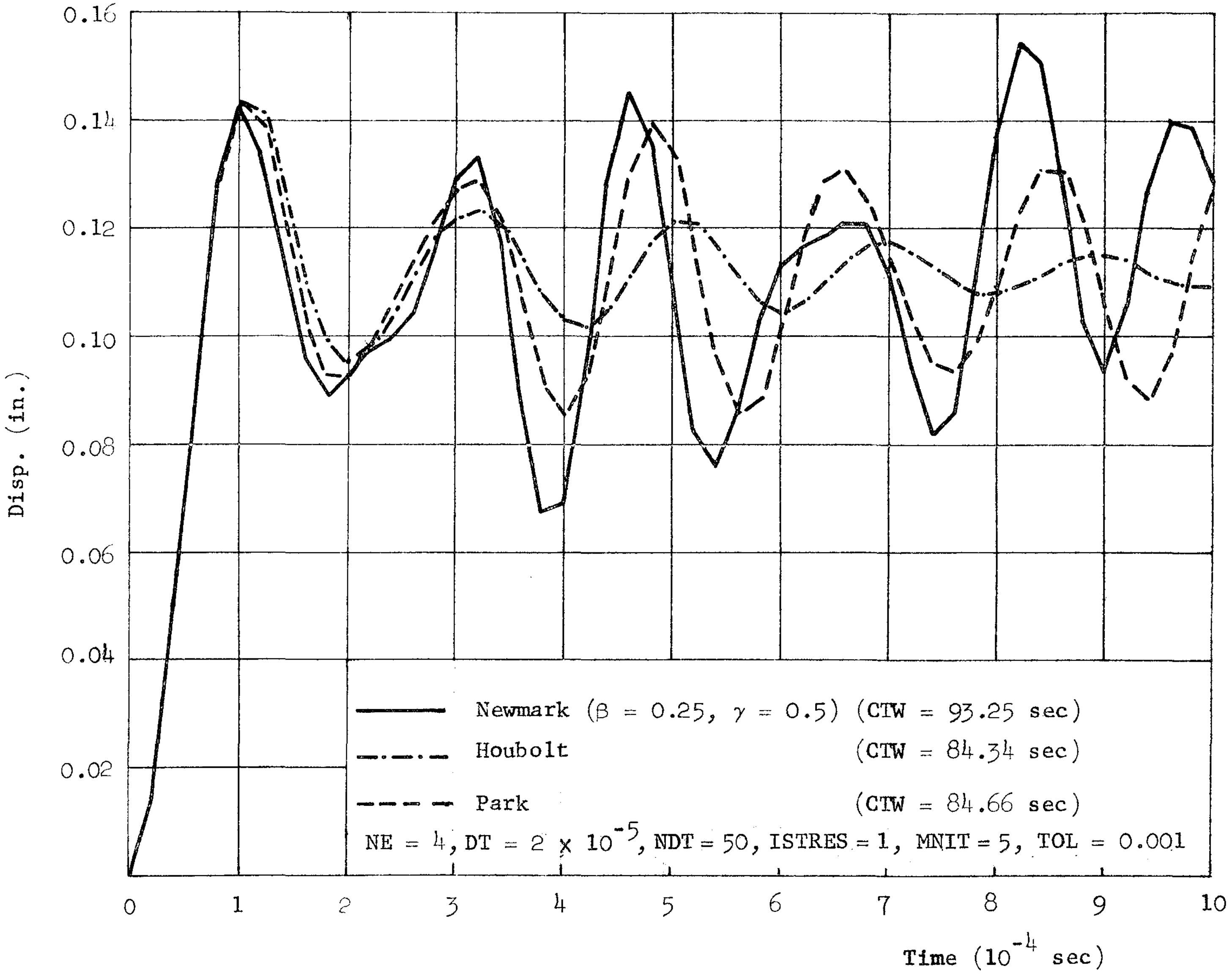


Fig. 6.18 Response by Three Implicit Methods with Equilibrium Iterations

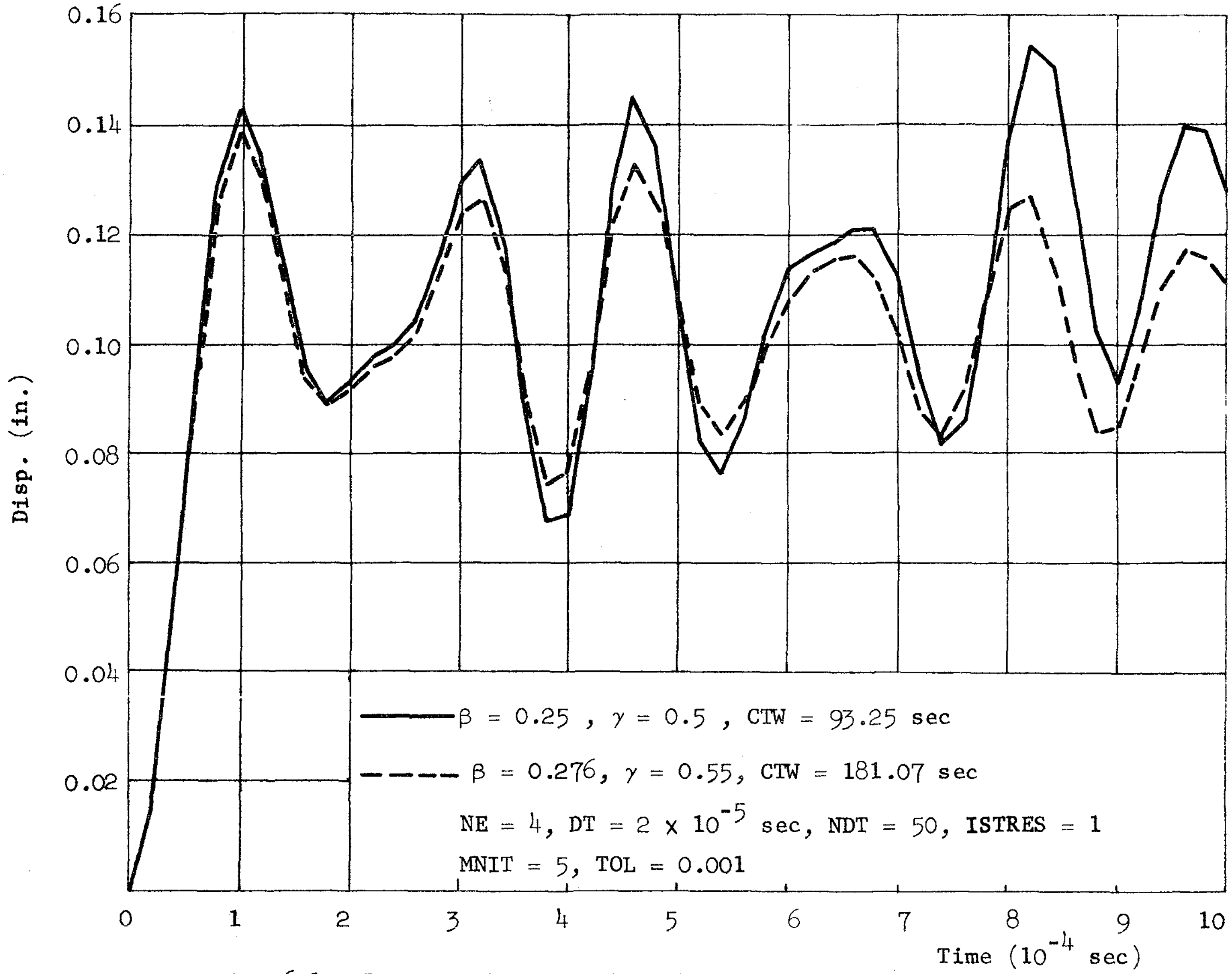


Fig. 6.19 Response by Newmark Method with Equilibrium Iterations

TABLE 6.5 Computation time for FORTRAN IVH Compiler (sec)

Method		4 Elements NDF = 10	9 Elements NDF = 20	16 Elements NDF = 38	36 Elements NDF = 82
Central Difference	IGN = 0	5.26	13.06	28.71	87.85
	IGN = 1	20.11	47.55	101.80	250.96
Park	IGN = 0	8.69	22.42	56.42	261.04
	IGN = 1	17.72	42.62	93.94	344.25

NDF = Number of Degrees of Freedom

$$\text{IGN} = \begin{cases} 0 & \text{Geometric nonlinearities are neglected} \\ 1 & \text{Geometric nonlinearities are taken into account} \end{cases}$$

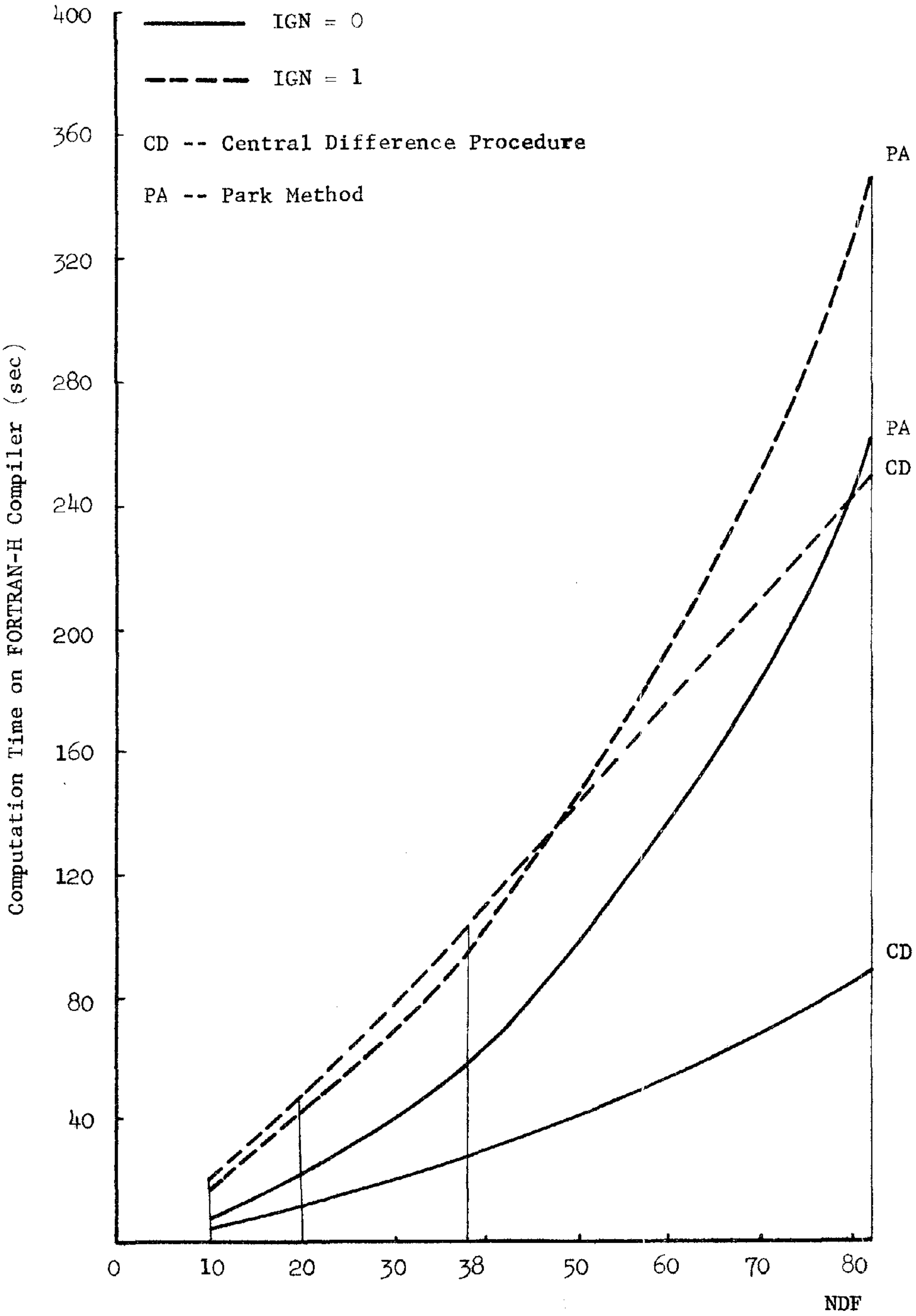

IGN = 0
IGN = 1
CD -- Central Difference Procedure
PA -- Park Method
Computation Time on FORTRAN-H Compiler (sec)
400
360
320
280
240
200
160
120
80
40
0
10
20
30
38
50
60
70
80
NDF
PA
PA
CD
CD

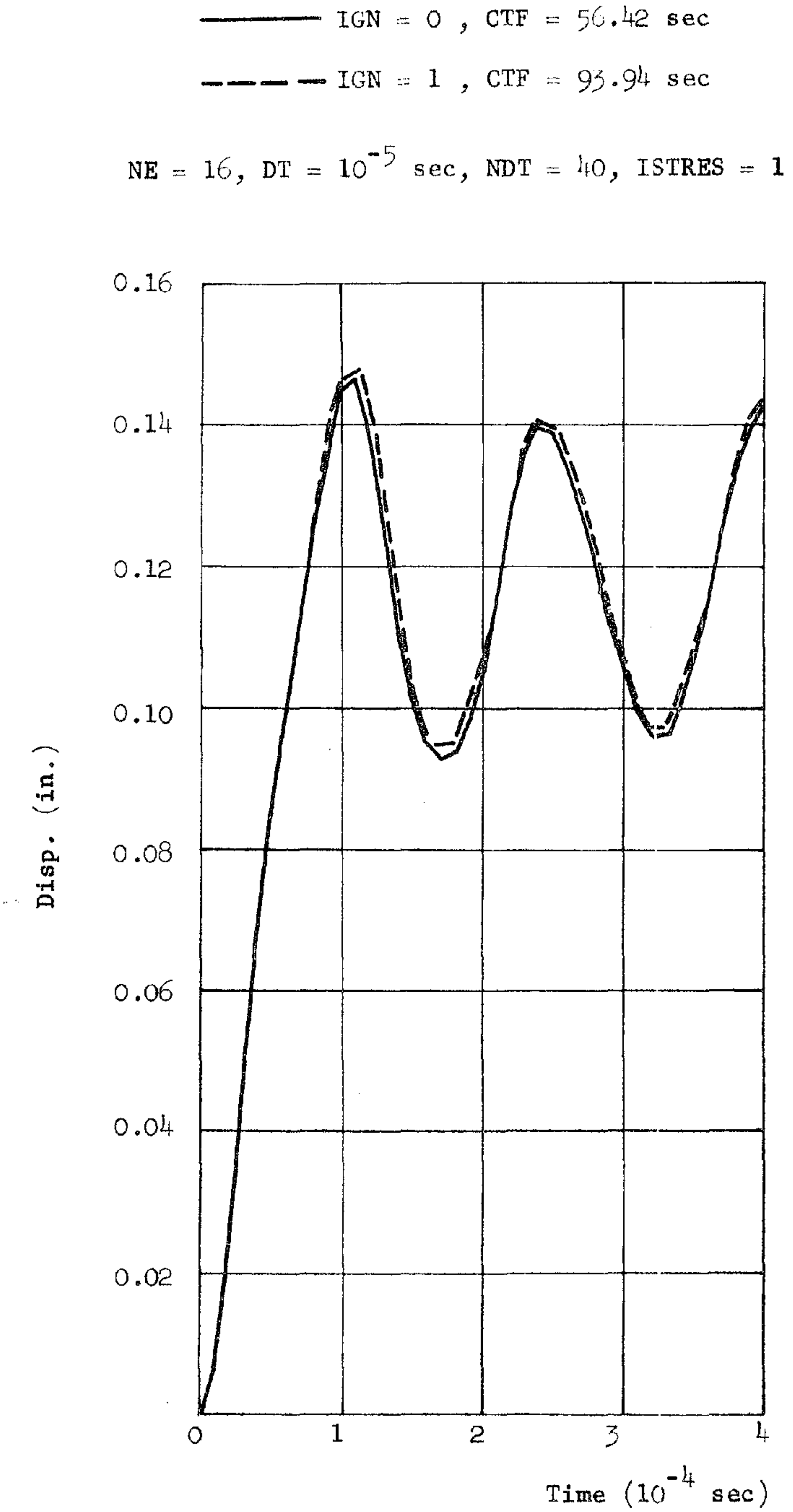
Figure 6.20

than that of the Park method. However, as observed in Section 6.2.2, the central difference procedure is prone to unstable behavior.

Solid curves in Figs. 6.21 and 6.22 show the responses of the two methods for the 16-element mesh. The Park method produces relatively good results, but the response for the central difference procedure is unstable. The computation times for these two cases (when IGN = 0) are 56 and 29 sec for the Park and central difference methods, respectively. The time step was reduced to 0.667×10^{-5} sec for the central difference procedure. Computation time was increased to 44 sec, but poor results were still obtained, as shown by the dot-dash line in Fig. 6.22. When the time step was further decreased to 0.5×10^{-5} sec, the response (see Fig. 6.23) is comparable to that of the Park method with $DT = 10^{-5}$ sec (see Fig. 6.21). However, the computation time was increased to 57 sec (for IGN = 0). Therefore, we observe that for the 16-element mesh and elastoplastic analysis the performance of each method is about the same.

When geometric nonlinearities are taken into account, the computation times for the Park method with $DT = 10^{-5}$ sec (see dashed curve in Fig. 6.21) and central difference procedure with $DT = 0.5 \times 10^{-5}$ sec (see dashed curve in Fig. 6.23) are 94 and 204 sec, respectively. In this case, Park's method is more efficient than the central difference method.

For the 36-element mesh, the results are plotted in Figs. 6.24 and 6.25. In this case, with $DT = 10^{-5}$ sec (solid curves), Park's method produces good results, but the central difference procedure is unstable. The time increment was then reduced to 0.333×10^{-5} sec for the central difference method (dot-dash curve in Fig. 6.25). The result is not as



NE = 16, DT = 10^{-5} sec, NDT = 40, ISTRES = 1

Fig. 6.21 Response by Park's Method

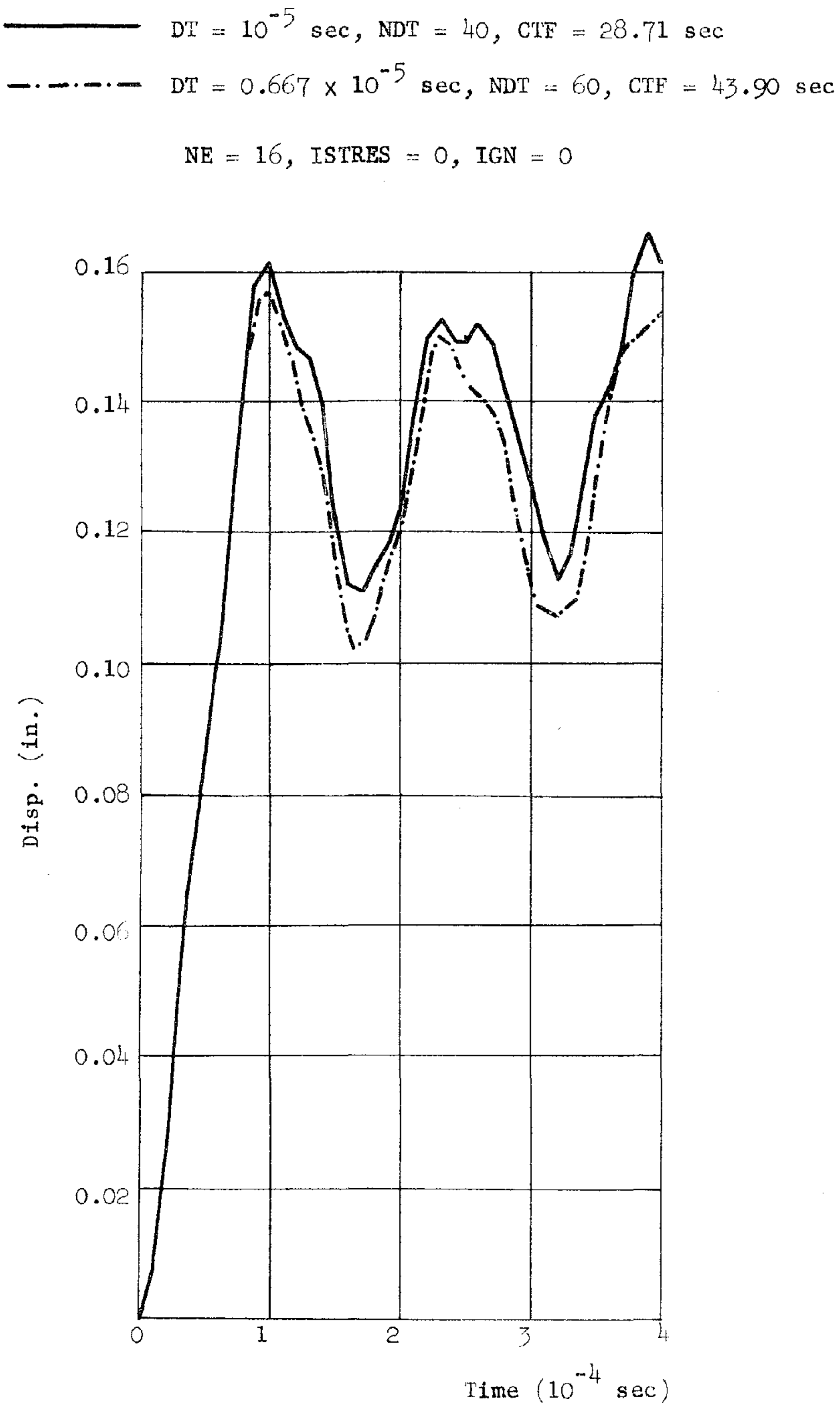


Fig. 6.22 Response by Central Difference Method

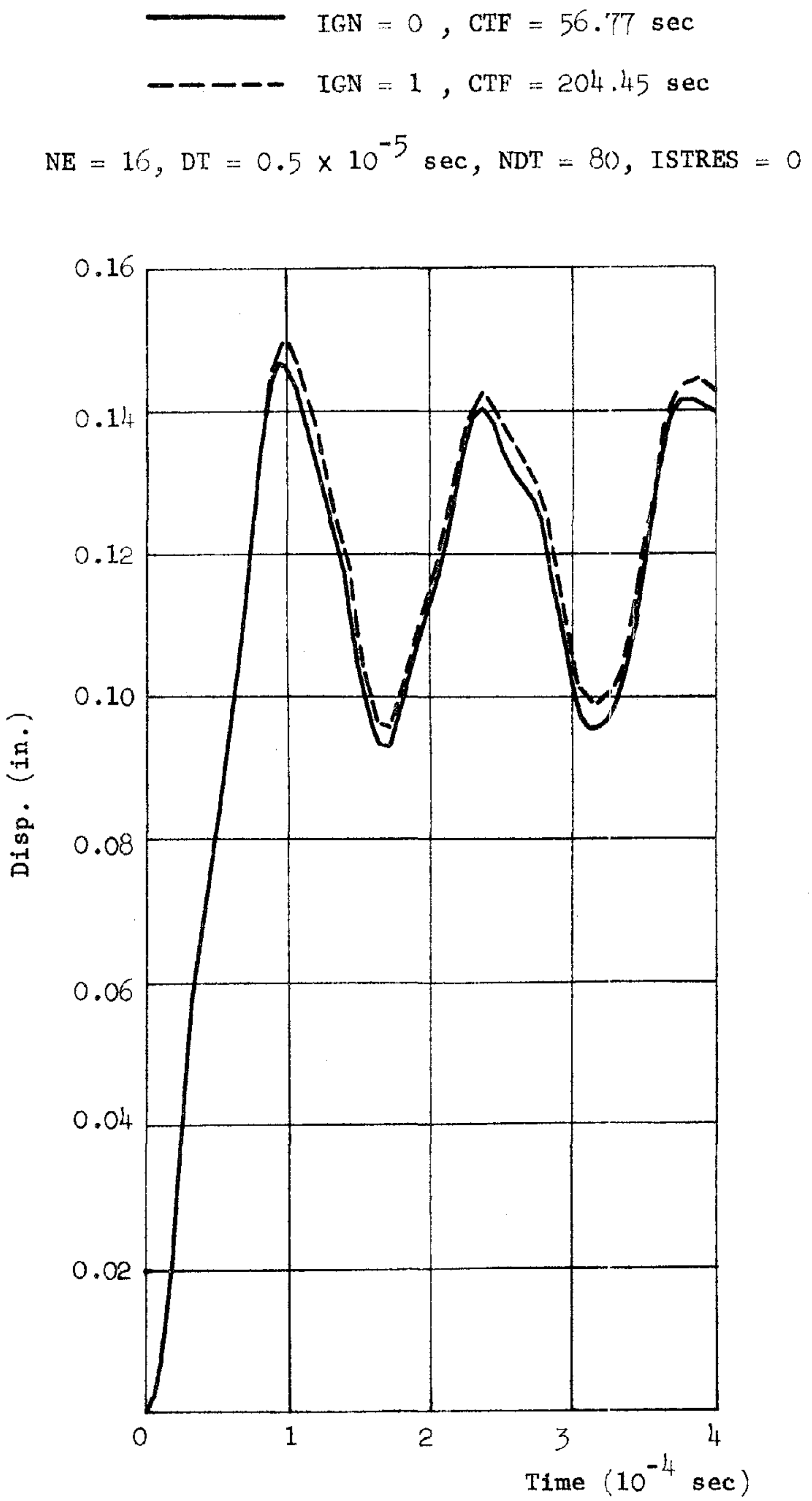


Fig. 6.23 Response by Central Difference Method

NE = 36, ISTRES = 1, IGN = 0

DT = 10^{-5} sec, NDT = 40, CTF = 261.04 sec

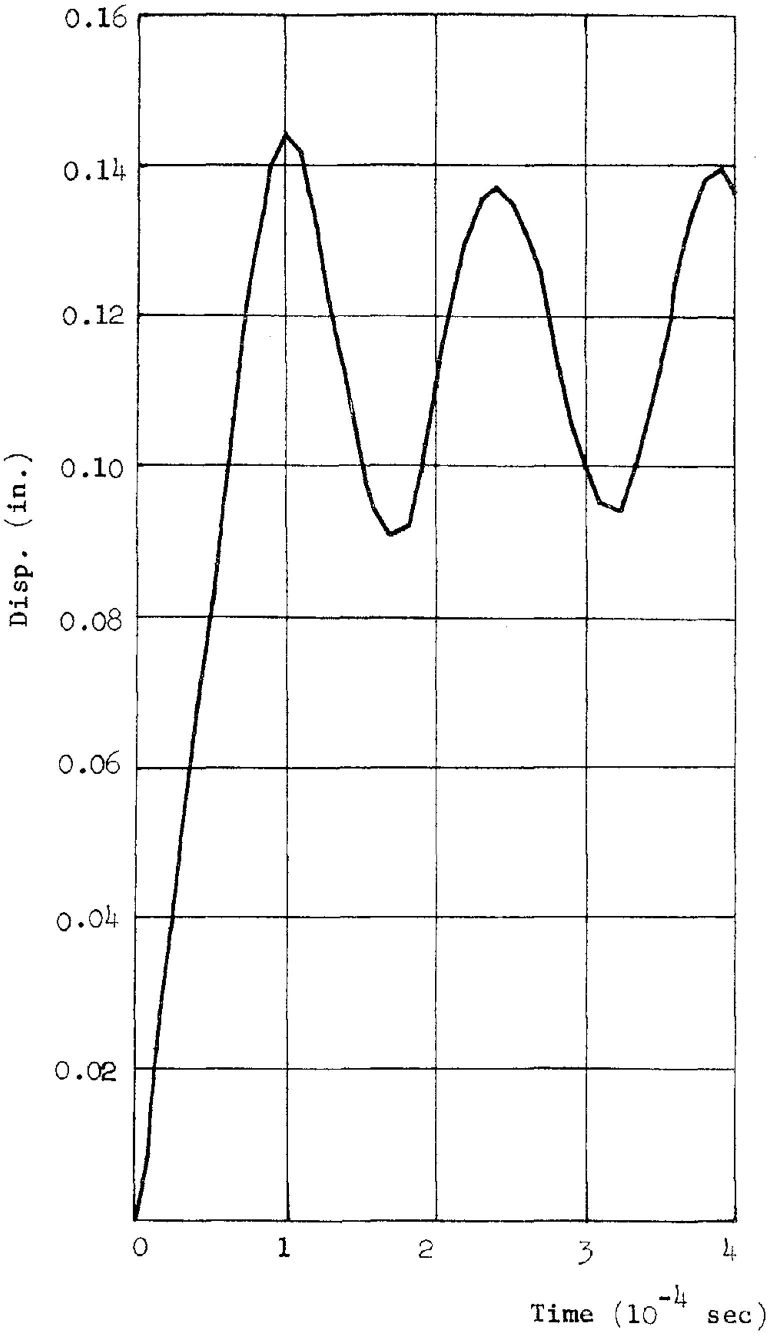


Fig. 6.24 Response by Park's Method

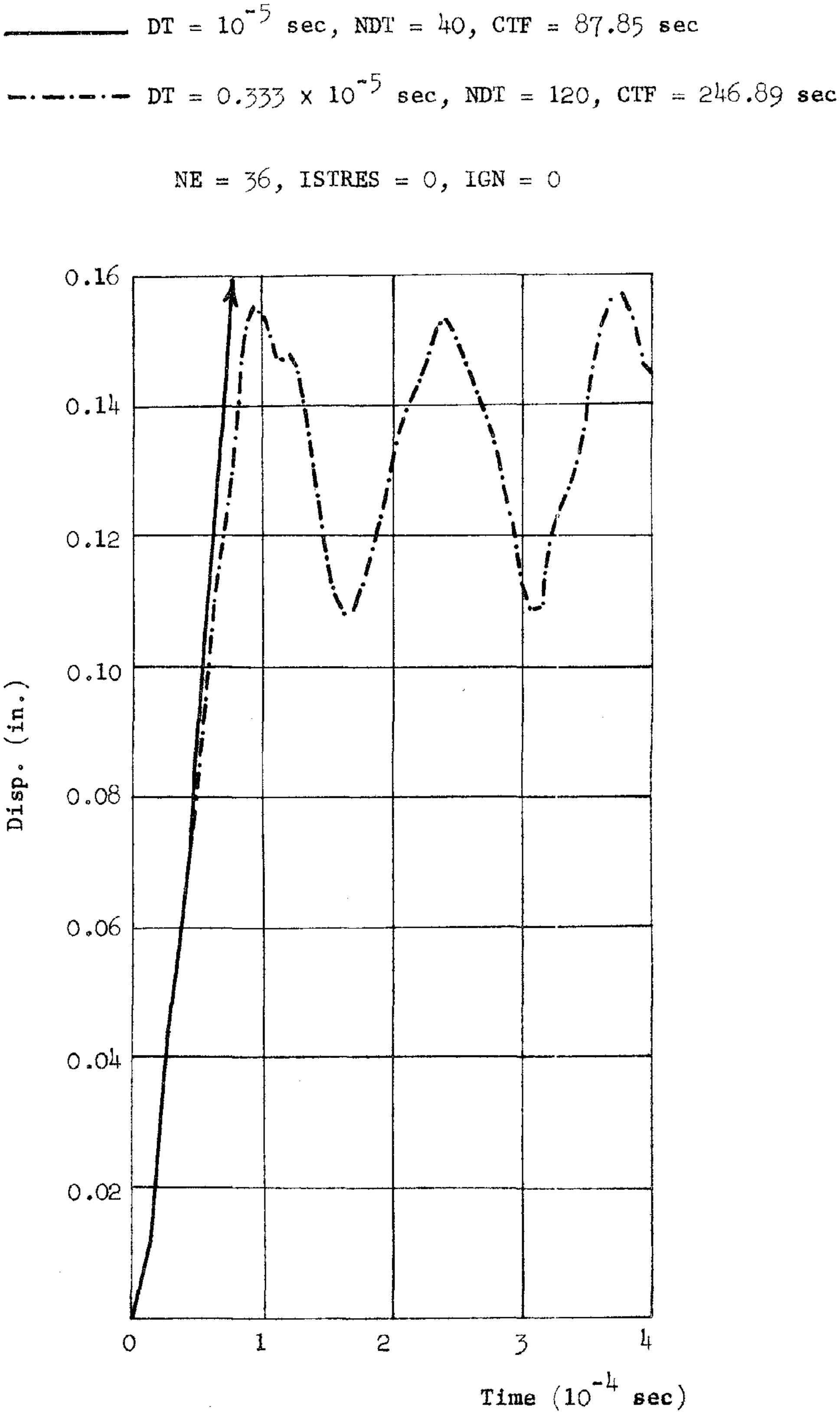


Fig. 6.25 Response by Central Difference Method

good as that for the Park method with DT = 10^{-5} sec , whereas their computation times for elastoplastic analysis (IGN = 0) are close to each other. Consequently, for the 36-element mesh and when IGN = 0 , the Park method is marginally better than the central difference procedure. The advantage of the Park method becomes more distinct when geometric nonlinearities are taken into account. This is due to the fact that incorporation of geometric nonlinearities increases the computation cost more drastically for the central difference procedure than for the Park method (see Fig. 6.20).

CHAPTER 7

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

7.1 Summary and Conclusions

Several competitive solution techniques for linear and nonlinear dynamic analysis of structures by the finite element method were studied. The accuracy, stability, and efficiency of the solution procedures were examined by comparing the results from a plane stress sample problem. The type of element used is the isoparametric quadrilateral. An efficient operational procedure was developed for this element in order to avoid matrix multiplications wherever possible. A lumped mass approach has been used, which results in a diagonal mass matrix. This approach is more efficient than the consistent (non-diagonal) mass formulation because the equations of motion are uncoupled in the acceleration terms.

7.1.1 Linear Analysis

For linear analysis four solution techniques were compared. They are direct linear extrapolation with the trapezoidal rule, the central difference predictor, two cycle iteration with the trapezoidal rule, and the normal mode method.

Among the methods studied, direct linear extrapolation with the trapezoidal rule appears to be the best technique for linear dynamic analysis. Its unconditional stability for linear systems is an important feature of this method. The central difference procedure should be rated second, and two-cycle iteration with the trapezoidal rule is third.

We also found that numerical integration methods are somewhat sensitive to round-off error, especially for large problems. Consequently, the use of double precision arithmetic is recommended as a standard procedure.

The normal mode method is competitive with the other methods studied only if modal truncation is used. In problems where the frequencies and mode shapes are desired, use of the normal mode procedure is mandatory.

7.1.2 Nonlinear Analysis

For nonlinear analysis, both material and geometric nonlinearities were included in the finite element formulation. Elastoplastic behavior was incorporated through a bilinear effective stress-effective strain curve, for which strain hardening of the material is taken into account; but the Bauschinger effect is neglected.

Three explicit methods were investigated. They are the central difference predictor, two-cycle iteration with the trapezoidal rule, and the fourth-order Runge-Kutta method. For these methods, a nodewise solution technique has been developed for which only two rows of the tangential stiffness matrix are generated at any stage in the analysis. Consequently, large problems with several thousand degrees of freedom can be analyzed within the fast core of the computer, which is advantageous.

Three implicit methods were also studied. They are the Newmark-Beta approach, the Houbolt procedure, and Park's stiffly-stable method. For small time steps, all six methods studied in this investigation gave results that are very close to each other.

Among the three explicit methods, it was concluded that the central difference procedure is the best, whereas the performances of the other two methods are about equal. For large time steps the central difference procedure and two-cycle iteration with the trapezoidal rule produce unstable results, but the response for the fourth-order Runge-Kutta method is unduly suppressed. We also found that for explicit methods the stresses should be evaluated only at the geometric center of the element.

Regarding the three implicit approaches, Park's stiffly-stable method was found to be somewhat better than the Newmark-Beta method; and Houbolt's procedure must be rated third. For large time steps, the results for the Newmark-Beta method with $\beta = 0.25$ and $\gamma = 0.5$ tends to be unstable, while the response by the Houbolt method is overdamped. It was also shown that for the implicit methods studied, the stresses should be evaluated at the spatial integration points.

Upon comparing the best explicit method against the best implicit method, we found that Park's formula is marginally better than the central difference predictor for elastoplastic analysis (when geometric nonlinearities are ignored). The advantage of the Park method becomes more distinct when geometric nonlinearities are taken into account. However, for large problems it may become necessary to use the explicit central difference method in order to obtain solutions in a reasonable amount of time.

7.2 Recommendations

Park's method should be compared against the central difference approach for large problems having hundreds or thousands of degrees of freedom. The problems studied should be highly nonlinear with regard to geometric as well as material properties. An appropriate example could be a plate or a shell in bending.

Use of a lumped mass approach is expedient for explicit methods and has a substantial effect on their efficiency. However, for implicit methods a consistent mass approach can be employed without much additional computational effort. This technique could be compared with the lumped mass approach for the implicit procedures.

Damping was neglected in the present study. One may include and study the effects of damping for different solution techniques. However, the nature of damping is poorly understood, and its incorporation into algorithms might tend to obscure the comparisons.

In Section 6.2, we observed that with large time steps the response for the central difference procedure was substantially magnified. On the contrary, the response for the fourth-order Runge-Kutta method is highly suppressed. As a result, the combination of these two methods into one algorithm might result in a more efficient procedure. This matter is worthy of further study.

REFERENCES

1. Bathe, K. J., Ramm, E. and Wilson, E. L., "Finite Element Formulations for Large Deformation Dynamic Analysis", Int. J. Num. Meth. Engng, Vol. 9, 1975, pp. 353-386.

2. Bathe, K. J., and Wilson, E. L., "Stability and Accuracy Analysis of Direct Integration Methods", Int. J. Earthq. Engng Struct. Dyn., Vol. 1, No. 3, Jan.-March 1973, pp. 283-291.

3. Bathe, K. J., and Wilson, E. L., "Solution Methods of Eigenvalue Problems in Structural Mechanics", Int. J. Num. Meth. Engng., Vol. 6, 1973, pp. 213-226.

4. Belytschko, T., "Transient Analysis", Structural Mechanics Computer Programs, editted by Pilkey, W. et al, University Press of Virginia, 1974, pp. 255-276.

5. Boggs, P. T., "The Solution of Nonlinear Systems of Equations by A-stable Integration Techniques", SIAM J. Numer. Anal., Vol. 8, No. 4, Dec. 1971, pp. 767-785.

6. Cook, R. D., Concepts and Applications of Finite Element Analysis , John Wiley and Sons Inc., New York, 1974.

7. Dahlquist, G. G., "A Special Stability Problem for Linear Multistep Methods", Nordisk Tidskrift Informations Behandling, Vol. 3, 1963, pp. 27-43.

8. Distefano, G. P., "Causes of Instabilities in Numerical Integration Techniques", Int. J. Computer Maths, Vol. 2, 1968, pp. 123-142.

9. Dunham, R. S., Nickel, R. E., and Stickler, D. C., "Integration Operators for Transient Structural Analysis", Int. J. Computers Structs, Vol. 2, No. 1/2, Feb. 1972, pp. 1-15.

10. Fung, Y. C., Foundations of Solid Mechanics , Prentice Hall Inc., Englewood Cliffs, New Jersey, 1965.

11. Gallagher, R. H., Finite Element Analysis - Fundamentals , Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1975.

12. Gear, C. W., Numerical Initial Value Problems in Ordinary Differential Equations , Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.

13. Gear, C. W., "Numerical Integration of Stiff Ordinary Differential Equations", Rpt. No. 221, Dept. Computer Sci., Univ. Ill. Urbana, Jan. 1967.

14. Gere, J. M., and Weaver, W., Jr., Analysis of Framed Structures , D. Van Nostrand Co., Inc., Princeton, New Jersey, 1965.

15. Goudreau, G. L., "Evaluation of Discrete Methods for the Linear Dynamic Response of Elastic and Viscoelastic Solids", Report No. SESM 69-15, University of California, Berkeley, 1970.

16. Gupta, K. K., "Dynamic Response Analysis of Geometrically Nonlinear Structures Subjected to High Impact", Int. J. Num. Meth. Engng , Vol. 4, No. 2, March-April 1972, pp. 163-174.

17. Hildebrand, F. B., Introduction to Numerical Analysis , McGraw Hill Book Co., New York, 1956.

18. Hill, R., The Mathematical Theory of Plastisity , Clarendon Press, Oxford, 1950.

19. Houbolt, J. C., "A Recurrence Matrix Solution for the Dynamic Response of Elastic Aircraft", J. Aero, Sci., Vol. 17, 1950, pp. 540-550.

20. Huebner, K. H., The Finite Element Method for Engineers , John Wiley and Sons, New York, 1975.

21. Jensen, P. S., "Transient Analysis of Structurs by Stiffly-Stable Methods", Int. J. Computers Structs, Vol. 4, 1974, pp. 615-626.

22. Johnson, D. E., "A Proof of the Stability of the Houbolt Method", AIAA J., Vol. 4, No. 8, August 1966, pp. 1450-1451.

23. Kachanov, L. M., Foundations of the Theory of Plasticity , American Elsevier Pub. Co., New York, 1971.

24. Kao, R., "A Comparison of Newton-Raphson Methods and Incremental Procedures for Geometrically Nonlinear Analysis", Computers and Structs, Vol. 4, 1974, pp.1091-1097.

25. Key, S. W., "A Finite Element Procedure for the Large Deformation Dynamic Response of Axisymmetric Solids", Computer Meth. In Appl. Mechs and Engng, No. 4, 1974, pp. 195-218.

26. Key, S. W. and Beisinger, Z. E., "Transient Dynamic Analysis of Thin Shells by the Finite Element Method", Proc. 3rd Conf. Mat. Meth. Struct. Engng, AFIT, Wright-Patterson AFB, Ohio, 1971.

27. Krieg, R. D., "Unconditional Stability in Numerical Time Integration Methods", J. Appl. Mech., ASME, Paper No. 73-APM-1, presented at the Appl. Mech. Summer Conf., G.I.T., Atlanta, Ga., June 20-22, 1973.

28. Krieg, R. D., and Key, S. W., "Transient Shell Response by Numerical Time Integration", Int. J. Num. Meths Engng, Vol. 7, 1973, pp. 273-286.

29. Kutta, W., "Beitrag zur näherungsweisen Integration totaler Differentialgleichungen", Z. Math. Phys., Vol. 46, 1901, pp. 435-453.

30. Lai, W. M., Rubin, D., and Krempl, E. Introduction to Continuum Mechanics , Pergamon Press Inc., New York 1974.

31. Lapidus, L., and Seinfeld, J. H., Numerical Solution of Ordinary Differential Equations , Academic Press, New York and London, 1971.

32. Laswell, B. H., "Comparison of Calculation Techniques for Dynamic Response of Plates", Engineer Thesis, Dept. of Civil Engng, Stanford University, June 1974.

33. McNamara, J. F., "Solution Schemes for Problems of Nonlinear Structural Dynamics", J. Pressure Vessel Technology, ASME, Vol. 96, 1974, pp. 96-102.

34. Malvern, L. E., Introduction to the Mechanics of a continuous Medium , Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1969.

35. Mendelson, A., Plasticity: Theory and Applications , The McMillan Co., New York, 1968.

36. Mondkar, D. P., and Powell, G. H., "Static and Dynamic Analysis of Nonlinear Structures", Report No. EERC 75-10, Dept. of Civil Engng, Univ. of California, Berkeley, California, March 1975.

37. Nagarajan, S. and Popov, E. P., "Elastic-Plastic Dynamic Analysis of Axisymmetric Solids", Int. J. Computers Structs, Vol. 4, 1974, pp. 1117-1134.

38. Nagarajan, S. and Popov, E. P., "Nonlinear Dynamic Analysis of Axisymmetric Shells", Int. J. Num. Meth. Engng, Vol. 9, 1975, pp. 535-550.

39. Newmark, N. M., "A Method of Computation for Structural Dynamics", J. Engng Mech. Div., ASCE, Vol. 85, No. EM3, July 1959, pp. 67-94.

40. Nickel, R. E., "On the Stability of Approximation Operators in Problems of Structural Dynamics", Int. J. Solids Structs., Vol. 7, 1971, pp. 301-319.

41. Nickel, R. E., "Direct Integration Methods in Structural Dynamics", J. Engng Mech. Div., ASCE, Vol. 99, No. EM2, April 1973, pp. 303-317.

42. Nickel, R. E., "Nonlinear Dynamics by Mode Superposition", AIAA-ASME-SAE 15th Structures, Structural Dynamics, and Materials Conference, Las Vegas, Nevada, April 1974.

43. Oden, J. T., Finite Elements of Nonlinear Continua , McGraw-Hill Book Co., New York, 1972.

44. Park, K. C., "An Improved Stiffly Stable Method for Direct Integration of Nonlinear Structural Dynamic Equations", presented at the Applied Mechanics Western Conf., Univ. of Hawaii,Honolulu, Hawaii, March 1975.

45. Park, K. C., "Evaluating Time Integration Methods for Nonlinear Dynamic Analysis", Finite Element Analysis of Transient Nonlinear Structural Behavior, edited by: Belytschko, T., Osias, J. R., and Marcal, P. V., Published by ASME (AMD-VOL 14), 1975.

46. Przemieniecki, J. S., Theory of Matrix Structural Analysis , McGraw-Hill Book Co., New York, 1968.

47. Ralston, A., A First Course in Numerical Analysis , McGraw-Hill Book Co., New York, 1965.

48. Richtmyer, R. D., and Morton, K. W., Difference Methods for Initial-Value Problems , 2nd ed., John Wiley and Sons, New York, 1967.

49. Runge, C., "Über die numerische Ausflösung von Differentialgleichungen", Math. Ann., Vol. 46, 1895, pp. 167-178.

50. Stricklin, J. A., Haisler, W. E., and Riesemann, W. A., "Comments on Nonlinear Transient Structural Analysis", Finite Element Analysis of Transient Nonlinear Structural Behavior, edited by: Belytschko, T., Osias, J. R., and Marcal, P. V., Published by ASME (AMD-VOL 14), 1975.

51. Stricklin, J. A., Martinez, J. E., Tillerson, J. R., Hong, J. H., and Haisler, W. E., "Nonlinear Dynamic Analysis of Shells of Revolution by Matrix Displacement Method", AIAA J., Vol. 9, No. 4, April 1971, pp. 629-636.

52. Timoshenko, S., and Goodier, J. N., Theory of Elasticity , 3rd ed., McGraw-Hill Book Co., Inc., New York, 1970.

53. Timoshenko, S., Young, D. H., and Weaver, W., Jr., Vibration Problems in Engineering , 4th ed., John Wiley and Son, New York, 1974.

54. Weaver, W., Jr., Computer Programs for Structural Analysis , D. Van Nostrand Co. Inc., Princeton, N. J., 1967.

55. Weaver, W., Jr., "Numerical Experiments for Nonlinear Dynamic Analysis", Lockheed Missiles and Space Co., Sunnyvale, Calif., June 1973.

56. Weeks, G., "Temporal Operators for Nonlinear Structural Dynamics Problems", J. Engng Mech. Div., Vol. 98, No. EM5, Oct. 1972, pp. 1087-1104.

57. Wilkinson, J. H., The Algebraic Eigenvalue Problem , Oxford University Press, London, 1965.

58. Witmer, E. A., Morino, L., and Leech, J. W., "Optimal Predictor-Corrector Method for Systems of Second-Order Differential Equations", AIAA J., Vol. 12, No. 10, Oct. 1974, pp. 1343-1347.

59. Yamada, Y., Kawai, T., Yoshimura, N. and Sakurai, T., "Analysis of the Elastic-Plastic Problems by the Matrix Displacement Method", Proc. 2nd Conf. Matrix Methods in Structural Mechanics, Wright Patterson, A.F.B., Ohio, 1968.

60. Yamada, Y., Yoshimura, N., and Sakurai, T., "Plastic Stress-strain Matrix and its Application for the Solution of Elastic-Plastic Problems by the Finite Element Method", Int. J. Mech. Sci., Vol. 10, 1968, pp. 343-354.

61. Yoshida, D. M., "Dynamic Response of Beams and Plates to Moving Loads", Doctoral Thesis, Dept. of Civil Engng, Stanford University, 1969.

62. Zienkiewicz, O.C., The Finite Element in Engineering Science , McGraw-Hill Book Co., London, 1971.

63. Zienkiewicz, O. C., and Nayak, G. C., "A General Approach to Problems of Large Deformation, and Plasticity using Isoparametric Elements", Proc. 3rd Conf. Matrix Methods in Structural Mechanics, Wright Patterson, A.F.B., Ohio, Oct. 1971.

APPENDIX A

FORMULATIONS OF STIFFNESS MATRIX AND EQUIVALENT NODAL LOADS FOR AN ISOPARAMETRIC QUADRILATERAL ELEMENT

We wish to develop an efficient operational procedure for calculating the stiffnesses and equivalent nodal loads. Using Eqs. (3.36) and (3.37) we can express the Jacobian matrix as

$$\underset{\sim}{J} = \frac{1}{4} \begin{bmatrix} -X_3 + \eta X_2 & -Y_3 - \eta Y_4 \\ X_1 - \xi X_4 & Y_1 + \xi Y_2 \end{bmatrix} \tag{A.1}$$

where

$$\begin{array}{llll} X_1 = -x_1 - x_2 + x_3 + x_4 & ; & X_2 = x_1 - x_2 + x_3 - x_4 \\ Y_1 = -y_1 - y_2 + y_3 + y_4 & ; & Y_2 = y_1 - y_2 + y_3 - y_4 \\ X_3 = x_1 - x_2 - x_3 + x_4 & ; & X_4 = -x_1 + x_2 - x_3 + x_4 \\ Y_3 = y_1 - y_2 - y_3 + y_4 & ; & Y_4 = -y_1 + y_2 - y_3 + y_4 \end{array} \tag{A.2}$$

Determinant of Jacobian matrix:

$$\begin{aligned} |\underset{\sim}{J}| = \frac{1}{8} \Big\{ & (x_1 - x_3)(y_2 - y_4) + (x_4 - x_2)(y_1 - y_3) \\ & + \eta \; [(x_4 - x_1)(y_2 - y_3) + (x_2 - x_3)(y_1 - y_4)] \\ & + \xi \; [(x_2 - x_1)(y_3 - y_4) + (x_3 - x_4)(y_1 - y_2)] \Big\} \end{aligned} \tag{A.3}$$

Inverse of Jacobian matrix:

$$\underset{\sim}{J}^{-1} = \frac{1}{4|\underset{\sim}{J}|} \begin{bmatrix} Y_1 + \xi\, Y_2 & Y_3 + \eta\, Y_4 \\ - X_1 + \xi\, X_4 & - X_3 + \eta\, X_2 \end{bmatrix} \qquad \text{(A.4)}$$

For simplicity replace $\underset{\sim}{D}_G$ by $\underset{\sim}{G}$

$$\underset{\sim}{G} \equiv \underset{\sim}{D}_G = \underset{\sim}{J}^{-1}\, \underset{\sim}{D}_L \qquad \begin{matrix}\text{(3.40)}\\ \text{repeated}\end{matrix}$$

Terms in matrix $\underset{\sim}{G}$ are as follows:

$$G_{11} = \frac{1}{16|\underset{\sim}{J}|} [(\eta - 1)(Y_1 + \xi\, Y_2) + (\xi - 1)(Y_3 + \eta\, Y_4)]$$

$$G_{12} = \frac{1}{16|\underset{\sim}{J}|} [(1 - \eta)(Y_1 + \xi\, Y_2) - (1 + \xi)(Y_3 + \eta\, Y_4)]$$

$$G_{13} = \frac{1}{16|\underset{\sim}{J}|} [(1 + \eta)(Y_1 + \xi\, Y_2) + (1 + \xi)(Y_3 + \eta\, Y_4)]$$

$$G_{14} = \frac{1}{16|\underset{\sim}{J}|} [- (1 + \eta)(Y_1 + \xi\, Y_2) + (1 - \xi)(Y_3 + \eta\, Y_4)]$$

(A.5)

$$G_{21} = \frac{1}{16|\underset{\sim}{J}|} [(\eta - 1)(- X_1 + \xi\, X_4) + (\xi - 1)(- X_3 + \eta\, X_2)]$$

$$G_{22} = \frac{1}{16|\underset{\sim}{J}|} [(1 - \eta)(- X_1 + \xi\, X_4) + (\xi + 1)(X_3 - \eta\, X_2)]$$

$$G_{23} = \frac{1}{16|\underset{\sim}{J}|} [(1 + \eta)(- X_1 + \xi\, X_4) + (1 + \xi)(- X_3 + \eta\, X_2)]$$

$$G_{24} = \frac{1}{16|\underset{\sim}{J}|} [(1 + \eta)(X_1 - \xi\, X_4) + (1 - \xi)(- X_3 + \eta\, X_2)]$$

Element Stiffness Matrix

In Chapter 3 we derived the element stiffness matrix as

$$\underset{\sim}{K} = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{B}'(\xi,\eta)\, \underset{\sim}{T_3}\, \underset{\sim}{B}(\xi,\eta)\, |\underset{\sim}{J}|\, d\xi d\eta \qquad \begin{matrix}(3.44)\\ \text{repeated}\end{matrix}$$

In the linear elastic case, the stress-strain matrix $\underset{\sim}{T_3}$ for both plane stress and plane strain can be written in the following form (see Section 3.2.2):

$$\underset{\sim}{T_3} = \begin{bmatrix} E_1 & E_2 & 0 \\ E_2 & E_1 & 0 \\ 0 & 0 & E_3 \end{bmatrix} \qquad (A.6)$$

where E_1 , E_2 and E_3 are constants. Table A.1 shows terms of the matrix $\underset{\sim}{S_E} = \underset{\sim}{B}'(\xi,\eta)\, \underset{\sim}{T_3}\, \underset{\sim}{B}(\xi,\eta)$ given explicitly in terms of the elements of matrix $\underset{\sim}{G}$ and the stress-strain constants. In the elasto-plastic case, $\underset{\sim}{T_3}$ must be replaced by the incremental elasto-plastic matrix $\underset{\sim}{D}^p$, which can be written for both plane stress and plane strain problems as follows (see Section 3.3.6):

$$\underset{\sim}{D}^p = \begin{bmatrix} D_1 & D_2 & D_3 \\ \text{sym.} & D_4 & D_5 \\ & & D_6 \end{bmatrix} \qquad (A.7)$$

where $D_1,\ldots,D_6$ are functions of stresses. Terms of the matrix $\underset{\sim}{S_E}$ for this case are given in Table A.2. In addition, terms of the matrices $\underset{\sim}{T_3}$ and $\underset{\sim}{D}^p$ for both plane stress and plane strain appear in Chapter 3.

Table A.1

Terms of matrix $\underset{\sim}{S_E} = \underset{\sim}{B}' \, (\xi,\eta) \, \underset{\sim}{T_3} \, \underset{\sim}{B} \, (\xi,\eta)$ (elastic case)

$$S_E\ (1,1) = E_1\, G_{11}^2 + E_3\, G_{21}^2$$

$$S_E\ (1,2) = (E_2 + E_3)\, G_{11}\, G_{21}$$

$$S_E\ (1,3) = E_1\, G_{11} G_{12} + E_3\, G_{21}\, G_{22}$$

$$S_E\ (1,4) = E_2\, G_{11}\, G_{22} + E_3\, G_{12}\, G_{21}$$

$$S_E\ (1,5) = E_1\, G_{11}\, G_{13} + E_3\, G_{21}\, G_{23}$$

$$S_E\ (1,6) = E_2\, G_{11}\, G_{23} + E_3\, G_{13}\, G_{21}$$

$$S_E\ (1,7) = E_1\, G_{11}\, G_{14} + E_3\, G_{21}\, G_{24}$$

$$S_E\ (1,8) = E_2\, G_{11}\, G_{24} + E_3\, G_{14}\, G_{21}$$

$$S_E\ (2,2) = E_1\, G_{21}^2 + E_3\, G_{11}^2$$

$$S_E\ (2,3) = E_2\, G_{12}\, G_{21} + E_3\, G_{22}\, G_{11}$$

$$S_E\ (2,4) = E_1\, G_{22}\, G_{21} + E_3\, G_{12}\, G_{11}$$

$$S_E\ (2,5) = E_2\, G_{13}\, G_{21} + E_3\, G_{23}\, G_{11}$$

$$S_E\ (2,6) = E_1\, G_{23}\, G_{21} + E_3\, G_{13}\, G_{11}$$

$$S_E\ (2,7) = E_2\, G_{14}\, G_{21} + E_3\, G_{24}\, G_{11}$$

$$S_E\ (2,8) = E_1\, G_{24}\, G_{21} + E_3\, G_{14}\, G_{11}$$

$$S_E\ (3,3) = E_1\, G_{12}^2 + E_3\, G_{22}^2$$

$$S_E\ (3,4) = (E_2 + E_3)\, G_{12}\, G_{22}$$

Table A.1 — Continued

$$S_E\ (3,5) = E_1\ G_{13}\ G_{12} + E_3\ G_{23}\ G_{22}$$

$$S_E\ (3,6) = E_2\ G_{23}\ G_{12} + E_3\ G_{13}\ G_{22}$$

$$S_E\ (3,7) = E_1\ G_{14}\ G_{12} + E_3\ G_{24}\ G_{22}$$

$$S_E\ (3,8) = E_2\ G_{24}\ G_{12} + E_3\ G_{14}\ G_{22}$$

$$S_E\ (4,4) = E_1\ G_{22}^2 + E_3\ G_{12}^2$$

$$S_E\ (4,5) = E_2\ G_{13}\ G_{22} + E_3\ G_{23}\ G_{12}$$

$$S_E\ (4,6) = E_1\ G_{23}\ G_{22} + E_3\ G_{13}\ G_{12}$$

$$S_E\ (4,7) = E_2\ G_{14}\ G_{22} + E_3\ G_{24}\ G_{12}$$

$$S_E\ (4,8) = E_1\ G_{24}\ G_{22} + E_3\ G_{14}\ G_{12}$$

$$S_E\ (5,5) = E_1\ G_{13}^2 + E_3\ G_{23}^2$$

$$S_E\ (5,6) = (E_2 + E_3)\ G_{23}\ G_{13}$$

$$S_E\ (5,7) = E_1\ G_{14}\ G_{13} + E_3\ G_{24}\ G_{23}$$

$$S_E\ (5,8) = E_2\ G_{24}\ G_{13} + E_3\ G_{14}\ G_{23}$$

$$S_E\ (6,6) = E_1\ G_{23}^2 + E_3\ G_{13}^2$$

$$S_E\ (6,7) = E_2\ G_{14}\ G_{23} + E_3\ G_{24}\ G_{13}$$

$$S_E\ (6,8) = E_1\ G_{24}\ G_{23} + E_3\ G_{14}\ G_{13}$$

$$S_E\ (7,7) = E_1\ G_{14}^2 + E_3\ G_{24}^2$$

$$S_E\ (7,8) = (E_2 + E_3)\ G_{14}\ G_{24}$$

$$S_E\ (8,8) = E_1\ G_{24}^2 + E_3\ G_{14}^2$$

Table A.2

Terms of matrix $\underset{\sim}{S}_E = \underset{\sim}{B}' \ (\xi,\eta) \ \underset{\sim}{D}^p \ \underset{\sim}{B}' \ (\xi,\eta)$

(elastoplastic case)

$$S_E\ (1,1) = G_{11}\ (D_1G_{11} + D_3G_{21}) + G_{21}\ (D_3G_{11} + D_6G_{21})$$

$$S_E\ (1,2) = G_{11}\ (D_2G_{21} + D_3G_{11}) + G_{21}\ (D_5G_{21} + D_6G_{11})$$

$$S_E\ (1,3) = G_{11}\ (D_1G_{12} + D_3G_{22}) + G_{21}\ (D_3G_{12} + D_6G_{22})$$

$$S_E\ (1,4) = G_{11}\ (D_2G_{22} + D_3G_{12}) + G_{21}\ (D_5G_{22} + D_6G_{12})$$

$$S_E\ (1,5) = G_{11}\ (D_1G_{13} + D_3G_{23}) + G_{21}\ (D_3G_{13} + D_6G_{23})$$

$$S_E\ (1,6) = G_{11}\ (D_2G_{23} + D_3G_{13}) + G_{21}\ (D_5G_{23} + D_6G_{13})$$

$$S_E\ (1,7) = G_{11}\ (D_1G_{14} + D_3G_{24}) + G_{21}\ (D_3G_{14} + D_6G_{24})$$

$$S_E\ (1,8) = G_{11}\ (D_2G_{24} + D_3G_{14}) + G_{21}\ (D_5G_{24} + D_6G_{14})$$

$$S_E\ (2,2) = G_{21}\ (D_4G_{21} + D_5G_{11}) + G_{11}\ (D_5G_{21} + D_6G_{11})$$

$$S_E\ (2,3) = G_{21}\ (D_2G_{12} + D_5G_{22}) + G_{11}\ (D_3G_{12} + D_6G_{22})$$

$$S_E\ (2,4) = G_{21}\ (D_4G_{22} + D_5G_{12}) + G_{11}\ (D_5G_{22} + D_6G_{12})$$

$$S_E\ (2,5) = G_{21}\ (D_2G_{13} + D_5G_{23}) + G_{11}\ (D_3G_{13} + D_6G_{23})$$

$$S_E\ (2,6) = G_{21}\ (D_4G_{23} + D_5G_{13}) + G_{11}\ (D_5G_{23} + D_6G_{13})$$

$$S_E\ (2,7) = G_{21}\ (D_2G_{14} + D_5G_{24}) + G_{11}\ (D_3G_{14} + D_6G_{24})$$

$$S_E\ (2,8) = G_{21}\ (D_4G_{24} + D_5G_{14}) + G_{11}\ (D_5G_{24} + D_6G_{14})$$

$$S_E\ (3,3) = G_{12}\ (D_1G_{12} + D_3G_{22}) + G_{22}\ (D_3G_{12} + D_6G_{22})$$

$$S_E\ (3,4) = G_{12}\ (D_2G_{22} + D_3G_{12}) + G_{22}\ (D_5G_{22} + D_6G_{12})$$

Table A.2 — Continued

$$S_E\,(3,5) = G_{12}\,(D_1G_{13} + D_3G_{23}) + G_{22}\,(D_3G_{13} + D_6G_{23})$$

$$S_E\,(3,6) = G_{12}\,(D_2G_{23} + D_3G_{13}) + G_{22}\,(D_5G_{23} + D_6G_{13})$$

$$S_E\,(3,7) = G_{12}\,(D_1G_{14} + D_3G_{24}) + G_{22}\,(D_3G_{14} + D_6G_{24})$$

$$S_E\,(3,8) = G_{12}\,(D_2G_{24} + D_3G_{14}) + G_{22}\,(D_5G_{24} + D_6G_{14})$$

$$S_E\,(4,4) = G_{22}\,(D_4G_{22} + D_5G_{12}) + G_{12}\,(D_5G_{22} + D_6G_{12})$$

$$S_E\,(4,5) = G_{22}\,(D_2G_{13} + D_5G_{23}) + G_{12}\,(D_3G_{13} + D_6G_{23})$$

$$S_E\,(4,6) = G_{22}\,(D_4G_{23} + D_5G_{13}) + G_{12}\,(D_5G_{23} + D_6G_{13})$$

$$S_E\,(4,7) = G_{22}\,(D_2G_{14} + D_5G_{24}) + G_{12}\,(D_3G_{14} + D_6G_{24})$$

$$S_E\,(4,8) = G_{22}\,(D_4G_{24} + D_5G_{14}) + G_{12}\,(D_5G_{24} + D_6G_{14})$$

$$S_E\,(5,5) = G_{13}\,(D_1G_{13} + D_3G_{23}) + G_{23}\,(D_3G_{13} + D_6G_{23})$$

$$S_E\,(5,6) = G_{13}\,(D_2G_{23} + D_3G_{13}) + G_{23}\,(D_5G_{23} + D_6G_{13})$$

$$S_E\,(5,7) = G_{13}\,(D_1G_{14} + D_3G_{24}) + G_{23}\,(D_3G_{14} + D_6G_{24})$$

$$S_E\,(5,8) = G_{13}\,(D_2G_{24} + D_3G_{14}) + G_{23}\,(D_5G_{24} + D_6G_{14})$$

$$S_E\,(6,6) = G_{23}\,(D_4G_{23} + D_5G_{13}) + G_{13}\,(D_5G_{23} + D_6G_{13})$$

$$S_E\,(6,7) = G_{23}\,(D_2G_{14} + D_5G_{24}) + G_{13}\,(D_3G_{14} + D_6G_{24})$$

$$S_E\,(6,8) = G_{23}\,(D_4G_{24} + D_5G_{14}) + G_{13}\,(D_5G_{24} + D_6G_{14})$$

$$S_E\,(7,7) = G_{14}\,(D_1G_{14} + D_3G_{24}) + G_{24}\,(D_3G_{14} + D_6G_{24})$$

$$S_E\,(7,8) = G_{14}\,(D_2G_{24} + D_3G_{14}) + G_{24}\,(D_5G_{24} + D_6G_{14})$$

$$S_E\,(8,8) = G_{24}\,(D_4G_{24} + D_5G_{14}) + G_{14}\,(D_5G_{24} + D_6G_{14})$$

Equivalent Nodal Loads Due to Surface Loads

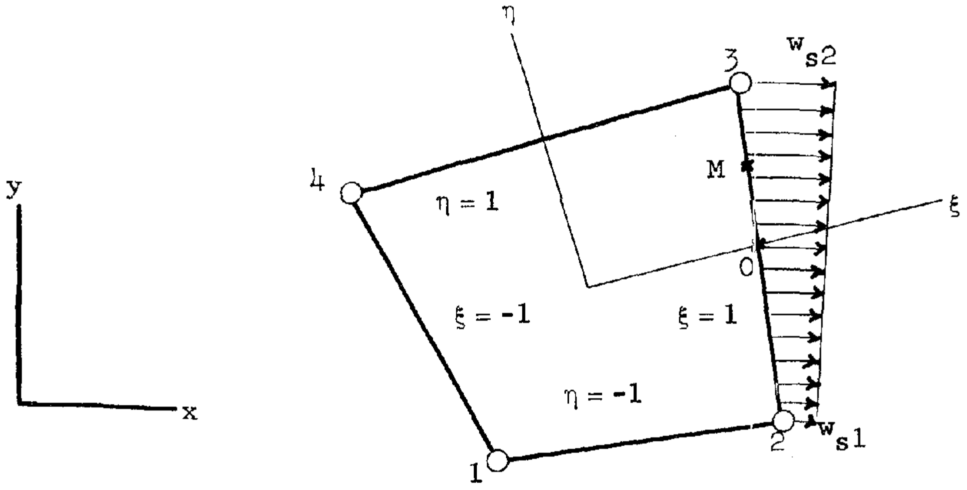


Figure A.1

Consider surface loads in the x and y directions on the edge 2-3, where $\xi = 1$ (see Fig. A.1).

$$\underset{\sim}{p}_S = \int_A \underset{\sim}{T}_1' \underset{\sim}{w}_s \, dA \qquad (3.10) \text{ repeated}$$

We assume w_{s1}, w_{s2}, w_{s3} and w_{s4} to be the force per unit inclined length (see Section 5.2.2). Consider a point M on the edge 2-3 at the distance u from the point 0, where $\eta = 0$. From Eq. (3.27) we have

$$x_M = \frac{1}{2} [(1 - \eta) x_2 + (1 + \eta) x_3] \qquad (A.8)$$

and

$$y_M = \frac{1}{2} [(1 - \eta) y_2 + (1 + \eta) y_3] \qquad (A.9)$$

If we let

$$x_0 = \frac{x_2 + x_3}{2} \quad ; \quad y_0 = \frac{y_2 + y_3}{2} \tag{A.10}$$

It can be shown that

$$u = \overline{OM} = \sqrt{(x_M - x_0)^2 + (y_M - y_0)^2} = \frac{\eta}{2} \ell \tag{A.11}$$

where ℓ = length of edge 2-3 and is given by the equation

$$\ell = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2} \tag{A.12}$$

Therefore, the infinitesimal area dA in Eq. (3.10) is obtained from the relation

$$dA = hdu = \frac{1}{2} h\ell d\eta \tag{A.13}$$

The vector of surface loads at the edge 2-3 can be written as

$$\underset{\sim}{w}_s = \left\{ w_1 , w_2 \right\} \tag{A.14}$$

where

$$w_1 = \frac{1}{2h} [(1 - \eta) w_{s1} + (1 + \eta) w_{s2}] \tag{A.15}$$

and

$$w_2 = \frac{1}{2h} [(1 - \eta) w_{s3} + (1 + \eta) w_{s4}] \tag{A.16}$$

After substituting Eqs. (A.13) to (A.16) in Eq. (3.10) and integrating along the edge 2-3, we find the terms of equivalent nodal loads $\underset{\sim}{p}_S$ explicitly as follows:

$$p_S(1) = p_S(2) = p_S(7) = p_S(8) = 0$$

$$p_S(3) = \frac{2w_{s1} + w_{s2}}{6} \ell \quad ; \quad p_S(4) = \frac{2w_{s3} + w_{s4}}{6} \ell \qquad \text{(A.17)}$$

$$p_S(5) = \frac{w_{s1} + 2w_{s2}}{6} \ell \quad ; \quad p_S(6) = \frac{w_{s3} + 2w_{s4}}{6} \ell$$

Equivalent Nodal Loads Due to Volume Loads

$$\underset{\sim}{p}_V = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{T}_1' \, \underset{\sim}{w}_v \, |\underset{\sim}{J}| \, d\xi d\eta \qquad \text{(3.46) repeated}$$

Let:

$$\underset{\sim}{w}_v = \left\{ w_{v1} \, , \, w_{v2} \right\} \qquad \text{(A.18)}$$

where

w_{v1} = uniform distribution in positive x direction

and

w_{v2} = uniform distribution in positive y direction.

Note that w_{v1} and w_{v2} are loads per unit volume. The product $\underset{\sim}{T}_1' \, \underset{\sim}{w}_v$ required for numerical integration over the area of the element is as follows:

$$\underset{\sim}{T}_1' \, \underset{\sim}{w}_v = \frac{1}{4} \begin{bmatrix} w_{v1} \; (1 - \xi)(1 - \eta) \\ w_{v2} \; (1 - \xi)(1 - \eta) \\ w_{v1} \; (1 + \xi)(1 - \eta) \\ w_{v2} \; (1 + \xi)(1 - \eta) \\ w_{v1} \; (1 + \xi)(1 + \eta) \\ w_{v2} \; (1 + \xi)(1 + \eta) \\ w_{v1} \; (1 - \xi)(1 + \eta) \\ w_{v2} \; (1 - \xi)(1 + \eta) \end{bmatrix} \qquad \text{(A.19)}$$

Equivalent Nodal Loads due to Temperature Strain

$$\underset{\sim}{p}_T = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{B}' \underset{\sim}{T}_3 \underset{\sim}{\epsilon}_T \, |\underset{\sim}{J}| \, d\xi d\eta \qquad \begin{matrix}(3.47)\\ \text{repeated}\end{matrix}$$

For uniform dilational expansion ΔT and for plane stress:

$$\underset{\sim}{\epsilon}_T = \alpha(\Delta T) \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \qquad (A.20)$$

where α is the coefficient of thermal expansion. For plane strain, α must be replaced by $\alpha(1 + \nu)$, in which ν is Poisson's ratio. The product $\underset{\sim}{B}' \underset{\sim}{T}_3 \underset{\sim}{\epsilon}_T$ required in numerical integration in Eq. (3.47) simplifies to the following:

$$\underset{\sim}{B}' \underset{\sim}{T}_3 \underset{\sim}{\epsilon}_T = \alpha \, (\Delta T)(E_1 + E_2) \begin{bmatrix} G_{11} \\ G_{21} \\ G_{12} \\ G_{22} \\ G_{13} \\ G_{23} \\ G_{14} \\ G_{24} \end{bmatrix} \qquad (A.21)$$

Equivalent Nodal Loads Due to Prestrains

$$\underset{\sim}{p}_P = h \int_{-1}^{1} \int_{-1}^{1} \underset{\sim}{B}' \, (\xi,\eta) \, \underset{\sim}{T}_3 \, \underset{\sim}{\epsilon}_P \, |\underset{\sim}{J}| \, d\xi d\eta \qquad \begin{matrix}(3.48)\\ \text{repeated}\end{matrix}$$

Let:

$$\underset{\sim}{\epsilon}_P = \begin{bmatrix} PS1 \\ PS2 \\ PS3 \end{bmatrix} \qquad (A.22)$$

where

PS1 = Uniform expansion in x direction

PS2 = Uniform expansion in y direction

PS3 = Uniform positive shear strain

The product of $\underset{\sim}{B}'(\xi,\eta)\ \underset{\sim}{T}_3\ \underset{\sim}{\epsilon}_P$ after simplification is as follows:

$$\underset{\sim}{B}'\ \underset{\sim}{T}_3\ \underset{\sim}{\epsilon}_P = \begin{bmatrix} G_{11}\ (E_1\ PS1 + E_2\ PS2) + G_{21}\ E_3\ PS3 \\ G_{21}\ (E_2\ PS1 + E_1\ PS2) + G_{11}\ E_3\ PS3 \\ G_{12}\ (E_1\ PS1 + E_2\ PS2) + G_{22}\ E_3\ PS3 \\ G_{22}\ (E_2\ PS1 + E_1\ PS2) + G_{12}\ E_3\ PS3 \\ G_{13}\ (E_1\ PS1 + E_2\ PS2) + G_{23}\ E_3\ PS3 \\ G_{23}\ (E_2\ PS1 + E_1\ PS2) + G_{13}\ E_3\ PS3 \\ G_{14}\ (E_1\ PS1 + E_2\ PS2) + G_{24}\ E_3\ PS3 \\ G_{24}\ (E_2\ PS1 + E_1\ PS2) + G_{14}\ E_3\ PS3 \end{bmatrix}$$

APPENDIX B

LUMPED MASS MATRIX FOR ISOPARAMETRIC QUADRILATERAL

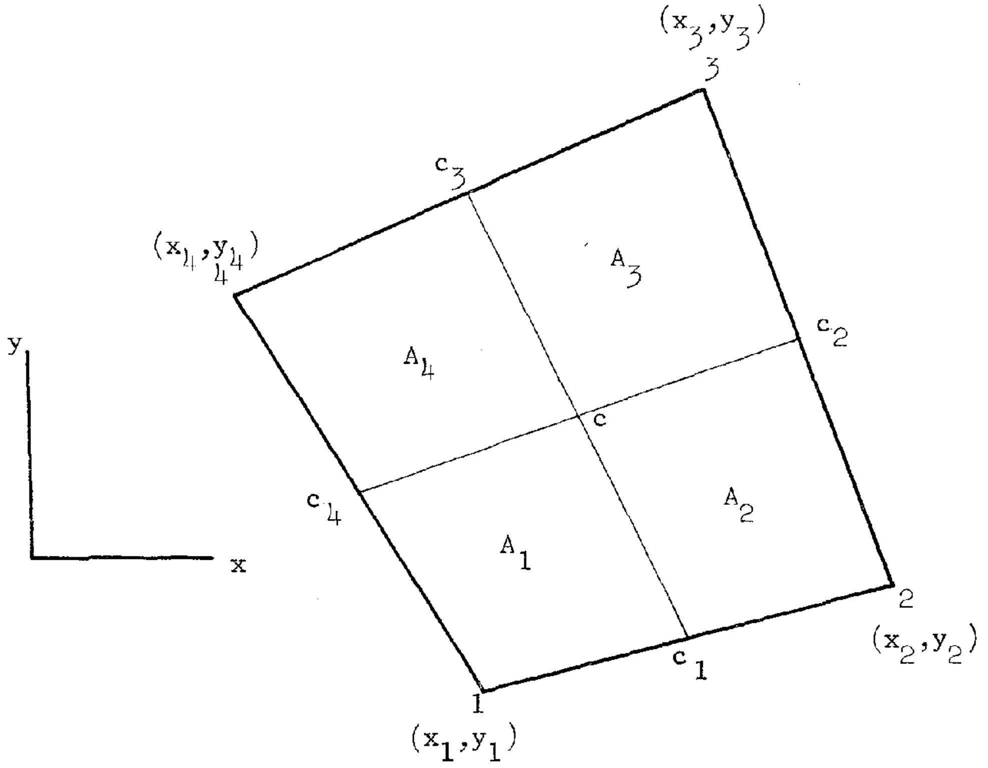


Figure B.1

The area of the quadrilateral can be determined from the following relation:

$$A = \iint dxdy = \int_{-1}^{1}\int_{-1}^{1} |\underset{\sim}{J}| \, d\xi d\eta \qquad \text{(B.1)}$$

Using the determinant of the Jacobian matrix, $|\underset{\sim}{J}|$ from Appendix A [Eq. (A.2)], integrating, and simplifying the results, we obtain

$$A = \frac{1}{2}\left[(x_1 - x_3)(y_2 - y_4) + (x_4 - x_2)(y_1 - y_3)\right] \qquad \text{(B.2)}$$

In Fig. A-1, c is the geometric center and c_1 , c_2 , c_3 , and c_4 are at the midpoints of the sides of the quadrilateral. Thus,

$$x_c = \frac{x_1 + x_2 + x_3 + x_4}{4} \quad ; \quad y_c = \frac{y_1 + y_2 + y_3 + y_4}{4} \qquad (B.3)$$

$$x_{c_1} = \frac{x_1 + x_2}{2} \quad ; \quad y_{c_1} = \frac{y_1 + y_2}{2} \qquad (B.4)$$

$$x_{c_2} = \frac{x_2 + x_3}{2} \quad ; \quad y_{c_2} = \frac{y_2 + y_3}{2} \qquad (B.5)$$

$$x_{c_3} = \frac{x_3 + x_4}{2} \quad ; \quad y_{c_3} = \frac{y_3 + y_4}{2} \qquad (B.6)$$

$$x_{c_4} = \frac{x_4 + x_1}{2} \quad ; \quad y_{c_4} = \frac{y_4 + y_1}{2} \qquad (B.7)$$

Four subsidiary quadrilaterals are labeled in Fig. B-1 as A_1 , A_2 , A_3 , and A_4 . From Eq. (B.2), we can write

$$A_1 = \frac{1}{2}\left[\left(x_1 - x_c\right)\left(y_{c_1} - y_{c_4}\right) + \left(x_{c_4} - x_{c_1}\right)\left(y_1 - y_c\right)\right] \qquad (B.8)$$

Substitute from relations (B.3), (B.4), and (B.7) into Eq. (B.8) and simplify the results to obtain

$$A_1 = \frac{1}{16}\left[(3x_1 - x_2 - x_3 - x_4)(y_2 - y_4) + (x_4 - x_2)(3y_1 - y_2 - y_3 - y_4)\right] \qquad (B.9)$$

Similarly, we find the expressions for A_2 , A_3 , and A_4 as follows:

$$A_2 = \frac{1}{16}\left[(3x_2 - x_1 - x_3 - x_4)(y_3 - y_1) + (x_1 - x_3)(3y_2 - y_1 - y_3 - y_4)\right] \qquad (B.10)$$

$$A_3 = \frac{1}{16}\left[(3x_3 - x_1 - x_2 - x_4)(y_4 - y_2) + (x_2 - x_4)(3y_3 - y_1 - y_2 - y_4)\right] \qquad (B.11)$$

$$A_4 = \frac{1}{16}\left[(3x_4 - x_1 - x_2 - x_3)(y_1 - y_3) + (x_3 - x_1)(3y_4 - y_1 - y_2 - y_3)\right] \qquad (B.12)$$

Masses at the four nodes are assigned in proportion to the above areas. Thus, the nonzero terms of the diagonal lumped mass matrix are as follows:

$$m(1,1) = m(2,2) = \rho A_1 h \tag{B.13}$$

$$m(3,3) = m(4,4) = \rho A_2 h \tag{B.14}$$

$$m(5,5) = m(6,6) = \rho A_3 h \tag{B.15}$$

$$m(7,7) = m(8,8) = \rho A_4 h \tag{B.16}$$

where ρ is the mass density and h is the thickness of the quadrilateral.

APPENDIX C

PROGRAM RESPPSQ4

(Pages 196 - 234, inclusive, consist of the program listing and sample computer output for RESPPSQ4 and are not reproduced in this report.)

APPENDIX D

PROGRAM NODIMP

(Pages 235 - 282, inclusive, consist of the program listing and sample computer output for NODIMP and are not reproduced in this report.)

pgs 197 - 234 blank

APPENDIX E

PROGRAM NODEXP

(Pages 283 -335, inclusive, consist of the program listing and sample computer output for NODEXP and are not reproduced in this report.)

pgs. 236 - 282 blank