# CHAPTER 1 INTRODUCTION 

by Ted Belytschko
Northwestern University
Copyright 1996

### 1.1 NONLINEAR FINITE ELEMENTS IN DESIGN

Nonlinear finite element analysis is an essential component of computeraided design. Testing of prototypes is increasingly being replaced by simulation with nonlinear finite element methods because this provides a more rapid and less expensive way to evaluate design concepts and design details. For example, in the field of automotive design, simulation of crashes is replacing full scale tests, both for the evaluation of early design concepts and details of the final design, such as accelerometer placement for airbag deployment, padding of the interior, and selection of materials and component cross-sections for meeting crashworthiness criteria. In many fields of manufacturing, simulation is speeding the design process by allowing simulation of processes such as sheet-metal forming, extrusion of parts, and casting. In the electronics industries, simulation is replacing drop-tests for the evaluation of product durability.

For both users and developers of nonlinear finite element programs, an understanding of the fundamental concepts of nonlinear finite element analysis is essential. Without an understanding of the fundamentals, a user must treat the finite element program as a black box that provides simulations. However, even more so than linear finite element analysis, nonlinear finite element analysis confronts the user with many choices and pitfalls. Without an understanding of the implication and meaning of these choices and difficulties, a user is at a severe disadvantage.

The purpose of this book is to describe the methods of nonlinear finite element analysis for solid mechanics. Our intent is to provide an integrated treatment so that the reader can gain an understanding of the fundamental methods, a feeling for the comparative usefulness of different approaches and an appreciation of the difficulties which lurk in the nonlinear world. At the same time, enough detail about the implementation of various techniques is given so that they can be programmed.

Nonlinear analysis consists of the following steps:

1. development of a model;
2. formulation of the governing equations;
3. discretization of the equations;
4. solution of the equations;
5. interpretation of the results.

Modeling is a term that tends to be used for two distinct tasks in engineering. The older definition emphasizes the extraction of the essential elements of mechanical behavior. The objective in this approach is to identify the simplest model which can replicate the behavior of interest. In this approach, model development is the process of identifying the ingredients of the model which can provide the qualitative and quantitative predictions.

A second approach to modeling, which is becoming more common in industry, is to develop a detailed, single model of a design and to use it to examine all of the engineering criteria which are of interest. The impetus for this approach to modeling is that it costs far more to make a model or mesh for an engineering product than can be saved through reduction of the model by specializing it for each application. For example, the same finite element model of a laptop computer can be used for a drop-test simulation, a linear static analysis and a thermal analysis. By using the same model for all of these analyses, a significant amount of engineering time can be saved. While this approach is not recommended in all situations, it is becoming commonplace in industry. In the near future the finite element model may serve as a prototype that can be used for checking many aspects of a design's performance. The decreasing cost of computer time and the increasing speed of computers make this approach highly cost-effective. However the user of finite element software must still able to evaluate the suitability of a model for a particular analysis and understand its limitations.

The formulation of the governing equations and their discretization is largely in the hands of the software developers today. However, a user who does not understand the fundamentals of the software faces many perils, for some approaches and software may be unsuitable. Furthermore, to convert experimental data to input, the user must be aware of the stress and strain measures used in the program and by the experimentalist who provided material data. The user must understand the sensitivity of response to the data and how to asses it. An effective user must be aware of the likely sources of error, how to check for these errors and estimate their magnitudes, and the limitations and strengths of various algorithms.

The solution of the discrete equations also presents a user with many choices. An inappropriate choice will result in very long run-times which can prevent him from obtaining the results within the time schedule. An understanding of the advantages and disadvantages and the approximate computer time required for various solution procedures are invaluable in the selection of a good strategy for developing a reasonable model and selecting the solution procedure.

The user's role is most crucial in the interpretation of results. In addition to the approximations inherent even in linear finite element models, nonlinear analyses are often sensitive to many factors that can make a single simulation quite misleading. Nonlinear solids can undergo instabilities, their response can be sensitive to imperfections, and the results can depend dramatically on material parameters. Unless the user is aware of these phenomena, the possibility of a misinterpretation of simulation results is quite possible.

In spite of these many pitfalls, our views on the usefulness and potential of nonlinear finite element analyses are very sanguine. In many industries, nonlinear finite element analysis have shortened design cycles and dramatically reduced the need for prototype tests. Simulations, because of the wide variety of output they produce and the ease of doing what-ifs, can lead to tremendous improvements of the engineer's understanding of the basic physics of a product's behavior under various environments. While tests give the gross but important result of whether the product withstands a certain environment, they usually provide little of the detail of the behavior of the product on which a redesign can be based if the product does not meet a test. Computer simulations, on the other hand, give detailed histories of stress and strain and other state variables, which in the hands of a good engineer give valuable insight into how to redesign the product.

Like many finite element books, this book presents a large variety of methods and recipes for the solution of engineering and scientific problems by the finite element method. However, in order to preserve a pedagogic character, we have interwoven several themes into the book which we feel are of central importance in nonlinear analysis. These include the following:

1. the selection of appropriate methods for the problem at hand;
2. the selection of a suitable mesh description and kinematic and kinetic descriptions for a given problem;
3. the examination of stability of the solution and the solution procedure;
4. an awareness of the smoothness of the response of the model and its implication on the quality and cost of the solution;
5. the role of major assumptions and the likely sources of error.

The selection of an appropriate mesh description, i.e. whether a Lagrangian, Eulerian or arbitrary Lagrangian Eulerian mesh is used, is very important for many of the large deformation problems encountered in process simulation and failure analysis. The effects of mesh distortion need to be understood, and the advantages of different types of mesh descriptions should be borne in mind in the selection. There are many situations where a continuous remeshing or arbitrary Lagrangian Eulerian description is most suitable.

The issue of the stability of solution is central in the simulation of nonlinear processes. In numerical simulation, it is possible to obtain solutions which are not physically stable and therefore quite meaningless. Many solutions are sensitive to imperfections or material and load parameters; in some cases, there is even sensitivity to the mesh employed in the solution. A knowledgeable user of nonlinear finite element software must be aware of these characteristics and the associated pitfalls. Otherwise the results obtained by elaborate computer simulations can be quite misleading and lead to incorrect design decisions.

The issue of smoothness is also ubiquitous in nonlinear finite element analysis. Lack of smoothness degrades the robustness of most algorithms and can introduce undesirable noise into the solution. Techniques have been developed which improve the smoothness of the response; these are generally called regularization procedures. However, regularization procedures are often not
based on physical phenomena and in many cases the constants associated with the regularization are difficult to determine. Therefore, an analyst is often confronted with the dilemma of whether to choose a method which leads to smoother solutions or to deal with a discontinuous response. An understanding of the effects of regularization parameters, the presence of hidden regularizations, such as penalty methods in contact-impact, and an appreciation of the benefits of these methods is highly desirable.

The accuracy and stability of solutions is a difficult consideration in nonlinear analysis. These issues manifest themselves in many ways. For example, in the selection of an element, the analyst must be aware of stability and locking characteristics of various elements. A judicious selection of an element for a problem involves factors such as the stability of the element for the problem at hand, the expected smoothness of the solution and the magnitude of deformations expected. In addition, the analyst must be aware of the complexity of nonlinear solutions: the appearance of bifurcation points and limit points, the stability and instability of equilibrium branches. The possibility of both physical and numerical instabilities must be kept in mind and checked in a solution.

Thus the informed use of nonlinear software in both industry and research requires considerable understanding of nonlinear finite element methods. It is the objective of this book to provide this understanding and to make the reader aware of the many interesting challenges and opportunities in nonlinear finite element analysis.

### 1.2. RELATED BOOKS AND HISTORY OF NONLINEAR FINITE ELEMENTS

Several excellent texts and monographs devoted either entirely or partially to nonlinear finite element analysis have already been published. Books dealing only with nonlinear finite element analysis include Oden(1972), Crisfield(1991), Kleiber(1989), and Zhong(1993). Oden's work is particularly noteworthy since it pioneered the field of nonlinear finite element analysis of solids and structures. Some of the books which are partially devoted to nonlinear analysis are Belytschko and Hughes(1983), Zienkiewicz and Taylor(1991), Bathe(1995) and Cook, Plesha and Malkus(1989). These books provide useful introductions to nonlinear finite element analysis. As a companion book, a treatment of linear finite element analysis is also useful. The most comprehensive are Hughes (1987) and Zienkiewicz and Taylor(1991).

Nonlinear finite element methods have many roots. Not long after the linear finite element method appeared through the work of the Boeing group and the famous paper of Cough, Topp, and Martin (??), engineers in several venues began extensions of the method to nonlinear, small displacement static problems, Incidentally, it is hard to convey the excitement of the finite element community and the disdain of classical researchers for the method. For example, for many years the Journal of Applied Mechanics banned papers, either tacitly, because it was considered of no scientific substance [sentence does not finish]. The excitement in the method was fueled by Ed Wilson's liberal distribution of his first programs. In many laboratories throughout the world, engineers developed new applications by modifying and extending these early codes.

This account form those in many other books in that the focus is not on the published works, buut on the software. In nonlinear finite element analysis, as in many endeavors in this information-computer age, te software represents a more meaningful indication of the state-of-the-art than the literature since it represents what can be applied in practice.

Among the first papers on nonlinear finite element methods were Marcal and King (??) and Gallagher (??). Pedro Marcal taught at Brown in those early years of nonlinear FEM, but he soon set up a firm to market the first nonlinear finite element program in 196?; the program was called MARC and is still a major player in the commercial software scene.

At about the same time, John Swanson (??) was developing a nonlinear finite element program at Westinghouse for nuclear applications. He left Westinghouse in 19?? to market the program ANSYS, which for the period 198090 dominated the commercial nonlinear finite element scene.

Two other major players in the early commercial nonlinear finite element scene were David Hibbitt and Klaus-Jürgen Bathe. David worked with Pedro Marcal until 1972, and then co-founded HKS, which markets ABAQUS. Jürgen launched his program, ADINA, shortly after obtaining his Ph.D. at Berkeley under the tutelage of Ed Wilson while teaching at MIT.

All of these programs through the early 1990's focused on static solutions and dynamic solutions by implicit methods. There were terrific advances in these methods in the 1970's, generated mainly by the Berkeley researchers and those with Berkeley roots: Thomas J.R. Hughes, Michael Ortiz, Juan Simo, and Robert Taylor (in order of age), were the most fertile contributors, but there are many other who are referenced throughout this book.

Explicit finite element methods probably have many different origins, depending on your viewpoint. Most of us were strongly influenced by the work in the DOE laboratories, such as the work of Wilkins (??) at Lawrence Livermore and Harlow (??) at Los Alamos.

In ???, Costantino (??) developed what was probably the first explicit finite element program. It was limit to linear materials and small deformations, and computed the internal nodal forces by multiplying a banded form of $\mathbf{K}$ by the nodal displacements. It was used primarily on IBM 7040 series computers, which cost millions of dollars and had a speed of far less than 1 megaflop and 32,000 words of RAM. The stiffness matrix was stored on a tape and the progress of a calculation could be gauged by watching the tape drive; after every step, the tape drive would reverse to permit a read of the stiffness matrix.

In 1969, in order to sell a proposal to the Air Force, we conceived what has come to be known as the element-by-element technique: the computation of the nodal forces without use of a stiffness matrix. The resulting program, SAMSON, was a two-dimensional finite element program which was used for a decade by weapons laboratories in the U.S. In 1972, the program was extended to fully nonlinear three-dimensional transient analysis of structures and was called WRECKER. This funding program was provided by the U.S. Department of Transportation by a visionary program manager, Lee Ovenshire, who dreamt in the early 1970's that crash testing of automobiles could be replaced by simulation. However, it was not to be, for at that time a simulation of a 300-element nodal
over ?? msec of simulation time took 30 hours of computer time, which cost the equivalent of three years of salary of an Assistant Professor $(\$ 35,000)$. The program funded several other pioneering efforts, Hughes' work on contact-impact (??) and Ted Shugar and Carly Ward's work on the modeling of the head at Port Hueneme. But the DOT decided around 1975 that simulation was too expensive (such is the vision of some bureaucrats) and shifted all funds to testing, bringing this far flung research effort to a screeching halt. WRECKER remained barely alive for the next decade at Ford.

Parallel work proceeded at the DOE national laboratories. In 1975, Sam Key completed PRONTO, which also featured the element-by-element explicit method. However, his program suffered from the restrictive dissemination policies of Sandia.

The key work in the promulgation of explicit finite element codes was John Hallquist's work at Lawrence Livermore Laboratories. John drew on the work which preceded his with discernment, he interacted closely with many Berkeley researchers such as Bob Taylor, Tom Hughes, and Juan Simo. Some of the key elements of his success were the development of contact-impact interfaces with Dave Benson, his awesome programming productivity and the wide dissemination of the resulting codes, DYNA-2D and DYNA-3D. In contrast to Sandia, LLN seemed to place no impediments on the distribution of the program and it was soon to be found in many government and academic laboratories and in industry throughout the world.

Key factors in the success of the DYNA codes was the use of one-point quadrature elements and the degree of vectorization which was achieved by john Hallquist. The latter issue has become somewhat irrelevant with the new generation of computers, but this combination enabled the simulation with models of suffiecient sizeto make full-scale simulation of problems such as car crash meaningful. The one-point quadrature elements with consistent hourglass control discussed in Chapter 8 increased the speed of three-dimensional analysis by almost an order of magnitude over fully integrated three-dimensional elements.

### 1.3 NOTATION

Nonlinear finite element analysis represents a nexus of three fields: (1) linear finite element methods, which evolved out of matrix methods of structural analysis; (2) nonlinear continuum mechanics; and (3) mathematics, including numerical analysis, linear algebra and functional analysis, Hughes(1996). In each of these fields a standard notation has evolved. Unfortunately, the notations are quite different, and at times contradictory or overlapping. We have tried to keep the variety of notation to a minimum and both consistent within the book and with the relevant literature. To make a reasonable presentation possible, both the notation of the finite element literature and continuum mechanics are used.

Three types of notation are used: 1. indicial notation, 2. tensor notation and 3. matrix notation. Equations in continuum mechanics are written in tensor and indicial notation. The equations pertaining to the finite element implementation are given in indicial or matrix notation.

Indicial Notation. In indicial notation, the components of tensors or matrices are explicitly specified. Thus a vector, which is a first order tensor, is denoted in
indicial notation by $x_{i}$, where the range of the index is the number of dimensions $n_{S}$. Indices repeated twice in a term are summed, in conformance with the rules of Einstein notation. For example in three dimensions, if $x_{i}$ is the position vector with magnitude $r$

$$
\begin{equation*}
r^{2}=x_{i} x_{i}=x_{1} x_{1}+x_{2} x_{2}+x_{3} x_{3}=x^{2}+y^{2}+z^{2} \tag{1.3.1}
\end{equation*}
$$

where the second equation indicates that $x_{1}=x, x_{2}=y, x_{3}=z$; we will always write out the coordinates as $x, y$ and $z$ rather than using subscripts to avoid confusion with nodal values. For a vector such as the velocity $v_{i}$ in three dimensions, $v_{1}=v_{x}, v_{2}=v_{y}, v_{3}=v_{z}$; numerical subscripts are avoided in writing out expressions to avoid confusing components with node numbers. Indices which refer to components of tensors are always lower case.

Nodal indices are always indicated by upper case Latin letters, e.g. $v_{i I}$ is the velocity at node I. Upper case indices repeated twice are summed over their range, which depends on the context. When dealing with an element, the range is over the nodes of the element, whereas when dealing with a mesh, the range is over the nodes of the mesh. Thus the velocity at a node $I$ is written as $v_{i I}$, where $v_{i I}$ is the $i$-component at node $I$.

A second order tensor is indicated by two subscripts. For example, for the second order tensor $E_{i j}$, the components are $E_{11}=E_{x x}, E_{21}=E_{y x}$, etc.. We will usually use indicial notation for Cartesian components but in a few of the more advanced sections we also use curvilinear components.

Indicial notation at times leads to spaghetti-like equations, and the resulting equations are often only applicable to Cartesian coordinates. For those who dislike indicial notation, it should be pointed out that it is almost unavoidable in the implementation of finite element methods, for in programming the finite element equations the indices must be specified.

Tensor Notation. Tensor notation is frequently used in continuum mechanics because tensor expression are independent of the coordinate systems. Thus while Cartesian indicial equations only apply to Cartesian coordinates, expressions in tensor notation can be converted to other coordinates such as cylindrical coordinates, curvilinear coordinates, etc. Furthermore, equations in tensor notation are much easier to memorize. A large part of the continuum mechanics and finite element literature employs tensor notation, so a serious student should become familiar with it.

In tensor notation, we indicate tensors of order one or greater in boldface. Lower case bold-face letters are almost always used for first order tensors, while upper case, bold-face letters are used for higher order tensors. For example, a velocity vector is indicated by $\mathbf{v}$ in tensor notation, while the second order tensor, such as $\mathbf{E}$, is written in upper case. The major exception to this are the physical stress tensor $\mathbf{s}$, which is a second order tensor, but is denoted by a lower case symbol. Equation(1.3.1) is written in tensor notation as

$$
\begin{equation*}
r^{2}=\mathbf{x} \cdot \mathbf{x} \tag{1.3.2}
\end{equation*}
$$

where a dot denotes a contraction of the inner indices; in this case, the tensors on the RHS have only one index so the contraction applies to those indices.

Tensor expressions are distinguished from matrix expressions by using dots and colons between terms, as in $\mathbf{a} \cdot \mathbf{b}$, and $\mathbf{A} \cdot \mathbf{B}$. The symbol ":" denotes the contraction of a pair of repeated indices which appear in the same order, so

$$
\begin{equation*}
\mathbf{A}: \mathbf{B} \equiv A_{i j} B_{i j} \tag{1.3.3}
\end{equation*}
$$

The symbol " .." denotes the contraction of the outer repeated indices and the inner repeated indices, as in

$$
\begin{equation*}
\mathbf{A} \cdot \cdot \mathbf{B}=A_{i j} B_{j i}=\mathbf{A}^{T}: \mathbf{B} \tag{1.3.4}
\end{equation*}
$$

If one of the tensors is symmetric, the expressions in Eqs. (1.3.3) and (1.3.4) are equivalent. This notation can also be used for contraction of higher order matrices. For example, the usual expression for a constitutive equation given below on the left is written in tensor notation as shown on the right

$$
\begin{equation*}
\sigma_{i j}=C_{i j k l} \varepsilon_{k l} \quad \boldsymbol{\sigma}=\mathbf{C}: \varepsilon \tag{1.3.5}
\end{equation*}
$$

The functional dependence of a variable will be indicated at the beginning of a development in the standard manner by listing the independent variables. For example, $\mathbf{v}(\mathbf{x}, t)$ indicates that the velocity $\mathbf{v}$ is a function of the space coordinates $\mathbf{x}$ and the time $t$. In subsequent appearances of $\mathbf{v}$, the identity of the independent variables in implied. We will not hang symbols all around the variable. This notation, which has evolved in a some of the finite element literature, violates esthetics, and is reminiscent of laundry hanging from the balconies of tenements. We will attach short words to some of the symbols. This is intended to help a reader who delves into the middle of the book. It is not intended that such complex symbols be used working through equations. Mathematical symbols and equations should be kept as simple as possible.

Matrix Notation. In implementation of finite element methods, we will often use matrix notation. We will use the same notations for matrices as for tensors and but will not use connective symbols. Thus Eq. (2) in matrix notation is written as

$$
\begin{equation*}
r^{2}=\mathbf{x}^{T} \mathbf{x} \tag{1.3.6}
\end{equation*}
$$

All first order matrices will be denoted by lower case boldface letters, such as $\mathbf{v}$, and will be considered column matrices. Examples of column matrices are

$$
\mathbf{x}=\left\{\begin{array}{l}
x  \tag{1.3.7}\\
y \\
z
\end{array}\right\} \quad \mathbf{v}=\left\{\begin{array}{l}
v_{1} \\
v_{2} \\
v_{3}
\end{array}\right\}
$$

Usually rectangular matrices, of which second tensors are a special case, will be denoted by upper case boldface, such as $\mathbf{A}$. The transpose of a matrix is denoted by a superscript " $T$ ", and the first index always refers to a row number, the second to a column number. Thus a $2 \times 2$ matrix $\mathbf{A}$ and a $2 \times 3$ matrix $\mathbf{B}$ are written out as (the order of a matrix is also written with number of rows by number of columns, with rows always first):

$$
\mathbf{A}=\left[\begin{array}{ll}
A_{11} & A_{12}  \tag{1.3.8}\\
A_{21} & A_{22}
\end{array}\right] \quad \mathbf{B}=\left[\begin{array}{lll}
B_{11} & B_{12} & B_{13} \\
B_{21} & B_{22} & B_{23}
\end{array}\right]
$$

In summary, we show the quadratic form associated with $\mathbf{A}$ in three notations

$$
\begin{equation*}
\mathbf{x}^{\mathbf{T}} \mathbf{A} \mathbf{x}=\mathbf{x} \cdot \mathbf{A} \cdot \mathbf{x}=x_{i} A_{i j} x_{j} \tag{1.3.9}
\end{equation*}
$$

The above are all equivalent: the first is matrix notation, the second in tensor notation, the third in indicial notation.

Second-order tensors are often converted to Voigt. Voigt notation is described in the Glossary.

### 1.4. MESH DESCRIPTIONS

One of the themes of this book is partially the different descriptions that are used in the formulation of the governing equations and the discretization of the continuum mechanics. We will classify the finite element model in three parts, Belytschko (1977):

1. the mesh description;
2. the kinetic description, which is determined by the choice of the stress tensor and the form of the momentum equation;
3. the kinematic description, which is determined by the choice of the strain measure.

In this Section, we will introduce the types of mesh descriptions. For this purpose, it is useful to introduce some definitions and concepts which will be used throughout this book. The first are the definitions of material coordinates and spatial coordinates. Spatial coordinates are denoted by $\mathbf{x}$; spatial coordinates are also called Eulerian coordinates. A spatial (or Eulerian coordinate) specifies the location of a point in space. The material coordinates, also called Lagrangian coordinates, are denoted by $\mathbf{X}$. The material coordinate labels a material point: each material point has a unique material coordinate, which is usually taken to be its spatial coordinate in the initial configuration of the body, so at $t=0, \mathbf{X}=\mathbf{x}$. Since in a deforming body, the positions of the material points change with time, the spatial coordinates of material points will be functions of time.

The motion or deformation of a body can be described by a function $\boldsymbol{\phi}(\mathbf{X}, t)$, with the material coordinates $X$ and the time $t$ as the independent
variables. This function gives the spatial positions of the material points as a function of time through:

$$
\begin{equation*}
\mathbf{x}=\boldsymbol{\phi}(\mathbf{X}, t) \tag{1.4.1}
\end{equation*}
$$

This is also called a map between the initial and current configurations. The displacement of a material point is the difference between its current position and its original position, so it is given by

$$
\begin{equation*}
\mathbf{u}(X, t)=\boldsymbol{\phi}(X, t)-\mathbf{X} \tag{1.4.2}
\end{equation*}
$$

To illustrate these definitions, consider the following motion in one dimension:

$$
\begin{equation*}
x=\boldsymbol{\phi}(X, t)=(1-X) t+\frac{1}{2} X t^{2}+X \tag{1.4.3}
\end{equation*}
$$

This motion is shown in Fig. 1.1; the motion of several points are plotted in space time to exhibit their trajectories; we obviously cannot plot the motion of all material points since there are an infinite number. The velocity of a material point is the time derivative of the motion with the material coordinate fixed, i.e. the velocity is given by

$$
\begin{equation*}
v(X, t)=\frac{\partial \phi(X, t)}{\partial t}=1+X(t-1) \tag{1.4.4}
\end{equation*}
$$

The mesh description is based on the choice of independent variables. For purposes of illustration, let us consider the velocity field. We can describe the velocity field as a function of the Lagrangian (material) coordinates, as in Eq. (1.4.4) or we can describe the velocity as a function of the Eulerian (spatial) coordinates

$$
\begin{equation*}
\overline{\mathbf{v}}(\mathbf{x}, t)=\mathbf{v}\left(\phi^{-1}(\mathbf{x}, t), t\right) \tag{1.4.5}
\end{equation*}
$$

In the above we have placed a bar over the velocity symbol to indicate that the velocity field when expressed in terms of the spatial coordinate $x$ and the time $t$ will not be the same function as that given in Eq. (1.4.4), although in the remainder of the book we will usually not distinguish different functions which are used to represent the same field. We have also used the concept of an inverse map which to express the material coordinates in terms of the spatial coordinates

$$
\begin{equation*}
X=\varphi^{-1}(x, t) \tag{1.4.6}
\end{equation*}
$$

Such inverse mappings can generally not be expressed in closed form for arbitrary motions, but for the simple motion given in Eq. (1.4.3) it is given by

$$
\begin{equation*}
X=\frac{x-t}{\frac{1}{2} t^{2}-t+1} \tag{1.4.7}
\end{equation*}
$$

Substituting the above into (3) gives

$$
\begin{equation*}
\bar{v}(x, t)=1+\frac{(x-t)(t-1)}{\frac{1}{2} t^{2}-t+1}=\frac{1-x+x t-\frac{1}{2} t^{2}}{\frac{1}{2} t^{2}-t+1} \tag{1.4.8}
\end{equation*}
$$

Equations (1.4.4) and (1.4.8) give the same physical velocity fields, but express them in terms of different independent variables, so that they are different functions. Equation (1.4.4) is called a Lagrangian (material) description for it expresses the dependent variable in terms of the Lagrangian (material) coordinate. Equation (1.4.8) is called an Eulerian (spatial) description, for it expresses the dependent variable as a function of the Eulerian (spatial) coordinates. Henceforth in this book, we will not use different symbols for these different functions, but keep in mind that if the same field variable is expressed in terms of different independent variables, then the functions must be different. In other words, a symbol for a dependent field variable is associated with the field, not the function.

## T. Belytschko, Introduction, December 16, 1998



Nodal
Trajectory

- Material Point - - - ${ }_{\text {Trajectory }}^{\text {Material Point }}$

Fig. 1.1 Space time depiction of a one dimensional Lagrangian, Eulerian, and ALE (arbitrary Lagrangian Eulerian) elements.

The differences between Lagrangian and Eulerian meshes are most clearly seen in terms of the behavior of the nodes. If the mesh is Eulerian, the Eulerian coordinates of nodes are fixed, i.e. the nodes are coincident with spatial points. If the mesh is Lagrangian, the Lagrangian (material) coordinates of nodes are time invariant, i.e. the nodes are coincident with material points. This is illustrated in Fig. 1.1 for the mapping given by Eq. (1.4.3). In the Eulerian mesh, the nodal trajectories are vertical lines and material points pass across element interfaces. In the Lagrangian mesh, nodal trajectories are coincident with material point trajectories, and no material passes between elements. Furthermore, element quadrature points remain coincident with material points in Lagrangian meshes, whereas in Eulerian meshes the material point at a given quadrature point changes with time. We will see later that this complicates the treatment of materials in which the stress is history-dependent.

The comparative advantages of Eulerian and Lagrangian meshes can be seen even in this simple one-dimensional example. Since the nodes are coincident with material points in the Lagrangian mesh, boundary nodes remain on the boundary throughout the evolution of the problem. This simplifies the imposition of boundary conditions in Lagrangian meshes. In Eulerian meshes, on the other hand, boundary nodes do not remain coincident with the boundary. Therefore, boundary conditions must be imposed at points which are not nodes, and as we shall see later, this engenders significant complications in multi-dimensional problems. Similarly, if a node is placed on an interface between two materials, it remains on the interface in a Lagrangian mesh, but not in an Eulerian mesh.

In Lagrangian meshes, since the material points remain coincident with mesh points, the elements deform with the material. Therefore, elements in a Lagrangian mesh can become severely distorted. This effect is apparent in a onedimensional problem only in the element lengths: in Eulerian meshes, the element length are constant in time, whereas in Lagrangian meshes, element lengths change with time. In multi-dimensional problems, these effects are far more severe, and elements can get very distorted. Since element accuracy degrades with distortion, the magnitude of deformation that can be simulated with a Lagrangian mesh is limited. Eulerian elements, on the other hand, are unchanged by the deformation of the material, so no degradation in accuracy occurs because of material deformation.

To illustrate the differences between Eulerian and Lagrangian mesh descriptions in two dimensions, a two dimensional example will be considered. In two dimensions, the spatial coordinates are denoted by $\mathbf{x}=[x, y]^{T}$ and the material coordinates by $\mathbf{X}=[X, Y]^{T}$. The deformation mapping is given by

$$
\begin{equation*}
\mathbf{x}=\boldsymbol{\phi}(\mathbf{X}, t) \tag{1.4.9}
\end{equation*}
$$

where $\boldsymbol{\phi}(\mathbf{X}, t)$ is a vector function, i.e. it gives a vector for every pair of the independent variables. For every pair of material coordinates and time, this function gives the pair of spatial coordinates corresponding to the current position of the material particles. Writing out the above expression gives

$$
\begin{align*}
& x=\phi_{1}(X, Y, t)  \tag{1.4.10}\\
& y=\phi_{2}(X, Y, t)
\end{align*}
$$

As an example of a motion, consider pure shear in which the map is given by

$$
\begin{align*}
& x=X+t Y \\
& y=Y \tag{1.4.11}
\end{align*}
$$


original configuration


L

deformed configuration


Fig. 1.2 Two dimensional shearing of a block showing Lagrangian (L) and Eulerian (E) elements.
In a Lagrangian mesh, the nodes are coincident with material (Lagrangian) points, so the nodes remain coincident with material points, so
for Lagrangian nodes, $\mathbf{X}_{I}=$ constant in time
For an Eulerian mesh, the nodes are coincident with spatial (Eulerian) points, so we can write
for Eulerian nodes, $\mathbf{x}_{I}=$ constant in time
Points on the edges of elements behave similarly to the nodes: in Lagrangian meshes, element edges remain coincident with material lines, whereas in Eulerian meshes, the element edges remain fixed in space.

To illustrate this statement we show Lagrangian and Eulerian meshes for the shear deformation given by Eq. (11) in Fig. 1.2. As can be seen from the figure, a Lagrangian mesh is like an etching on the material: as the material is deformed, the etching deforms with it. An Eulerian mesh is like an etching on a sheet of glass held in front of the material: as the material deforms, the etching is unchanged and the material passes across it.

The advantages and disadvantages of the two types of meshes are similar to those in one dimension. In a Lagrangian mesh, element edges and nodes which are initially on the boundary remain on the boundary, whereas in Eulerian meshes edges and nodes which are initially on the boundary do not remain on the boundary. Thus, in Lagrangian meshes, element edges (lines in two dimensions, surfaces in three dimensions) remain coincident with boundaries and material interfaces. In Eulerian meshes, element sides do not remain coincident with boundaries or material interfaces. Hence tracking methods or approximate
methods, such as volume of fluid approaches, have to be used for treating moving boundaries treated by Eulerian meshes; such as volume of fluid methods described in Section 5.?. Furthermore, an Eulerian mesh must be large enough to enclose the material in its deformed state. On the other hand, since Lagrangian meshes deform with the material, and they become distorted in the simulations of severe deformations. In Eulerian meshes, elements remain fixed in space, so their shapes never change.

A third type of mesh is an arbitrary Lagrangian Eulerian mesh, in which the nodes are programmed to move so that the advantages of both Lagrangian and Eulerian meshes can be exploited. In this type of mesh, the nodes can be programmed to move arbitrarily, as shown in Fig. 1.1. Usually the nodes on the boundaries are moved to remain on the boundaries, while the interior nodes are moved to minimize mesh distortion. This type of mesh is described and discussed further in Chapter 7.

## REFERENCES

T. Belytschko (1976), Methods and Programs for Analysis of Fluid-Structure Systems," Nuclear Engineering and Design, 42, 41-52.
T. Belytschko and T.J.R. Hughes (1983), Computational Methods for Transient Analysis, North-Holland, Amsterdam.
K.-J. Bathe (1996), Finite Element Procedures, Prentice Hall, Englewood Cliffs, New Jersey.
R.D. Cook, D.S. Malkus, and M.E. Plesha (1989), Concepts and Applications of Finite Element Analysis, 3rd ed., John Wiley.
M.A. Crisfield (1991), Non-linear Finite Element Analysis of Solids and Structure, Vol. 1, Wiley, New York.
T.J.R. Hughes (1987), The Finite Element Method, Linear Static and Dynamic Finite Element Analysis, Prentice-Hall, New York.
T.J.R. Hughes (1996), personal communication
M. Kleiber (1989), Incremental Finite element Modeling in Non-linear Solid Mechanics, Ellis Horwood Limited, John Wiley.
J.T. Oden (1972), Finite elements of Nonlinear Continua, McGraw-Hill, New York.
O.C. Zienkiewicz and R.L. Taylor (1991), The Finite Element Method, McGrawHill, New York.
Z.-H. Zhong (1993), Finite Element Procedures for Contact-Impact Problems, Oxford University Press, New York.

## GLOSSARY. NOTATION

Voigt Notation. In finite element implementations, Voigt notation is often useful; in fact almost all linear finite element texts use Voigt notation. In Voigt notation, second order tensors such as the stress, are written as column matrices, and fourth order tensors, such as the elastic coefficient matrix, are written as square matrices. Voigt notation is quite awkward for the formulation of the equations of continuum mechanics. Therefore only those equations which are related to finite element implementations will be given in Voigt notation.

Voigt notation usually refers to the procedure for writing a symmetric tensor in column matrix form. However, we will use the term for all conversions of higher order tensor expressions to lower order matrices.

The Voigt conversion for symmetric tensors depends on whether a tensor is a kinetic quantity, such as stress, or a kinematic quantity, such as strain. We first consider Voigt notation for stresses. In Voigt notation for kinetic tensors, the second order, symmetric tensor $\boldsymbol{\sigma}$ is written as a column matrix:

$$
\text { tensor } \quad \rightarrow \text { Voigt }
$$

in two dimensions:

$$
\boldsymbol{\sigma} \equiv\left[\begin{array}{ll}
\sigma_{11} & \sigma_{12}  \tag{A.1.1}\\
\sigma_{21} & \sigma_{22}
\end{array}\right] \rightarrow\left\{\begin{array}{l}
\sigma_{11} \\
\sigma_{22} \\
\sigma_{12}
\end{array}\right\}=\left\{\begin{array}{l}
\sigma_{1} \\
\sigma_{2} \\
\sigma_{3}
\end{array}\right\} \equiv\{\boldsymbol{\sigma}\}
$$

in three dimensions:

$$
\boldsymbol{\sigma} \equiv\left[\begin{array}{lll}
\sigma_{11} & \sigma_{12} & \sigma_{13}  \tag{A.1.2}\\
\sigma_{21} & \sigma_{22} & \sigma_{23} \\
\sigma_{31} & \sigma_{32} & \sigma_{33}
\end{array}\right] \rightarrow\left\{\begin{array}{c}
\sigma_{11} \\
\sigma_{22} \\
\sigma_{33} \\
\sigma_{23} \\
\sigma_{13} \\
\sigma_{12}
\end{array}\right\}=\left\{\begin{array}{c}
\sigma_{1} \\
\sigma_{2} \\
\sigma_{3} \\
\sigma_{4} \\
\sigma_{5} \\
\sigma_{6}
\end{array}\right\} \equiv\{\boldsymbol{\sigma}\}
$$

We will call the correspondence between the square matrix form of the tensor and the column matrix form the Voigt rule. For stresses the Voigt rule resides in the relationship between the indices of the second order tensor and the column matrix. The order of the terms in the column matrix in the Voigt rule is given by the line which first passes down the main diagonal of the tensor, then up the last column, and back across the row (if there are any elements left). As indicated in the bottom row, the square matrix form of the tensor is indicated by boldface, whereas brackets are used to distinguish the Voigt form. The correspondence is also given in Table 1.

## TABLE 1

Two-Dimensional Voigt Rule

| $\sigma_{i j}$ |  | ${ }^{\prime}$ |
| :---: | :---: | ---: |
| $i$ | $j$ | $\sigma_{a}$ |
| 1 | 1 | $a$ |
| 2 | 2 | 1 |
| 3 | 3 | 2 |
| $\sigma_{i j}$ |  |  |
| Three-Dimensional Voigt Rule |  |  |
|  |  |  |
| 1 | $j$ | $\sigma_{a}$ |
| 1 | 1 | 1 |
| 2 | 2 | 2 |
| 3 | 3 | 3 |
| 2 | 3 | 4 |
| 1 | 3 | 5 |
| 1 | 2 | 6 |

When the tensors are written in indicial notation, the difference between the Voigt and tensor form of second order tensors is indicated by the number of subscripts and the letter used. We use subscripts beginning with letters $i$ to $q$ for tensors, and subscripts $a$ to $g$ for Voigt matrix indices. Thus

$$
\sigma_{i j} \text { is replaced by } \sigma_{a}
$$

in going from tensor to Voigt notation. The correspondence between the subscripts $(i, j)$ and the Voigt subscript $a$ is given in Table 1 for two and three dimensions.

For a second order, symmetric kinematic tensor such as the strain $\varepsilon_{i j}$, the rule is almost identical: the correspondence between the tensor indices and the row numbers are identical, but the shear strains, i.e. those with indices that are not equal, are multiplied by 2. Thus the Voigt rule for the strains is

$$
\text { tensor } \quad \rightarrow \text { Voigt }
$$

two dimensions

$$
\boldsymbol{\varepsilon} \equiv\left[\begin{array}{ll}
\varepsilon_{11} & \varepsilon_{12}  \tag{A.1.3}\\
\varepsilon_{21} & \varepsilon_{22}
\end{array}\right] \rightarrow\left\{\begin{array}{c}
\varepsilon_{11} \\
\varepsilon_{22} \\
2 \varepsilon_{12}
\end{array}\right\}=\left\{\begin{array}{c}
\varepsilon_{1} \\
\varepsilon_{2} \\
\varepsilon_{3}
\end{array}\right\} \equiv\{\boldsymbol{\varepsilon}\}
$$

in three dimensions

$$
\boldsymbol{\varepsilon} \equiv\left[\begin{array}{lll}
\varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13}  \tag{A.1.4}\\
& \varepsilon_{22} & \varepsilon_{23} \\
\text { sym } & & \varepsilon_{33}
\end{array}\right] \rightarrow\left\{\begin{array}{c}
\varepsilon_{11} \\
\varepsilon_{22} \\
\varepsilon_{33} \\
2 \varepsilon_{23} \\
2 \varepsilon_{13} \\
2 \varepsilon_{12}
\end{array}\right\} \equiv\{\boldsymbol{\varepsilon}\}
$$

The Voigt rule requires a factor of two in the shear strains, which can be remembered by observing that the strains in Voigt notation are the engineering shear strains.

The inclusion of the factor of 2 in the Voigt rule for strains and strain-like tensors is motivated by the requirement that the expressions for the energy be equivalent in matrix and indicial notation. It is easy to verify that an increment in energy is given by

$$
\begin{equation*}
\rho d w^{i n t}=d \varepsilon_{i j} \sigma_{i j}=d \boldsymbol{\varepsilon}: \boldsymbol{\sigma}=\{d \boldsymbol{\varepsilon}\}^{T}\{\boldsymbol{\sigma}\} \tag{A.1.5}
\end{equation*}
$$

For these expressions to be equivalent, a factor of 2 is needed on the shear terms in the Voigt form; the factor of 2 can be added to either the stresses or the strains (or a coefficient of $\sqrt{2}$ on both the stresses and strains), but the preferred convention is to use this factor on the strains because the shear strains are then equivalent to the engineering strains.

The Voigt rule is particularly useful for converting fourth order tensors, which are awkward to implement in a computer program, to second order matrices. Thus the general linear elastic law in indicial notation involves the fourth order tensor $C_{i j k l}$ :

$$
\begin{equation*}
\sigma_{i j}=C_{i j k l} \varepsilon_{k l} \quad \text { or in tensor notation } \quad \boldsymbol{\sigma}=\mathbf{C} \boldsymbol{\varepsilon} \tag{A.1.6}
\end{equation*}
$$

The Voigt or matrix form of this law is

$$
\begin{equation*}
\{\boldsymbol{\sigma}\}=[\mathbf{C}]\{\boldsymbol{\varepsilon}\} \tag{A.1.7}
\end{equation*}
$$

or writing the matrix expression in indicial form: $\sigma_{a}=C_{a b} \varepsilon_{b}$
and as indicated on the right, when writing the Voigt expression in matrix indicial form, indices at the beginning of the alphabet are used. The Voigt matrix form of the elastic constitutive matrix is

$$
[\mathbf{C}]=\left[\begin{array}{lll}
C_{11} & C_{12} & C_{13}  \tag{A.1.9}\\
C_{21} & C_{22} & C_{23} \\
C_{31} & C_{32} & C_{33}
\end{array}\right]=\left[\begin{array}{lll}
C_{1111} & C_{1122} & C_{1112} \\
C_{2211} & C_{2222} & C_{2212} \\
C_{1211} & C_{1222} & C_{1212}
\end{array}\right]
$$

The first matrix refers to the elastic coefficients in in tensor notation, the second to Voigt notation; note that the number of subscripts specifies whether the matrix is expressed in Voigt or tensor notation. The above translation is completely consistent. For example, the expression for $\sigma_{12}$ from (A.1.6) is

$$
\begin{equation*}
\sigma_{12}=C_{1211} \varepsilon_{11}+C_{1212} \varepsilon_{12}+C_{1221} \varepsilon_{21}+C_{1222} \varepsilon_{22} \tag{A.1.10}
\end{equation*}
$$

The above translates to the following expression in terms of the Voigt notation

$$
\begin{equation*}
\sigma_{3}=C_{31} \varepsilon_{1}+C_{33} \varepsilon_{3}+C_{32} \varepsilon_{2} \tag{A.1.11}
\end{equation*}
$$

which can be shown to be equivalent to (A.1.10) if we use $\varepsilon_{3}=\varepsilon_{12}+\varepsilon_{21}=2 \varepsilon_{12}$ and the minor symmetry of $\mathbf{C}: C_{1212}=C_{1221}$.

It is convenient to reduce the order of the matrices in the indicial expressions when applying them in finite element methods. We will denote nodal vectors by double subscripts, such as $u_{i I}$, where $i$ is the component index and $I$ is the node number index. The component index is always lower case, the node number index is always upper case; sometimes their order is interchanged. The following rule is used for converting matrices involving node numbers and components to column matrices:

$$
\begin{equation*}
\text { matrix } u_{i I} \text { is transformed to a column matrix }\{d\} \text { by } \tag{A.1.12a}
\end{equation*}
$$

$$
\begin{equation*}
\text { elements of }\{d\} \text { are } u_{a} \text { where } a=(I-1) n_{S D}+i \tag{A.1.12b}
\end{equation*}
$$

(The symbol for the column matrix associated with displacements is changed because $\mathbf{u}$ is used for the components, i.e. $\mathbf{u}=u_{x}, u_{y}, u_{z}$.) This rule is combined with the Voigt rule whenever a pair of indices on a term pertain to a second order symmetric tensor. For example in the higher order matrix $B_{i j K k}$ is often used to related strains to nodal displacements by

$$
\begin{equation*}
\varepsilon_{i j}=B_{i j K k} u_{k K} \tag{A.1.13}
\end{equation*}
$$

where

$$
\begin{align*}
& u_{i}(\mathbf{x})=N_{I}(\mathbf{x}) u_{i I}  \tag{A.1.14}\\
& \varepsilon_{i j}=\frac{1}{2}\left(\frac{\partial u_{i}}{\partial x_{j}}+\frac{\partial u_{j}}{\partial x_{i}}\right)=\frac{1}{2}\left(\frac{\partial N_{I}}{\partial x_{j}} \delta_{i k}+\frac{\partial N_{I}}{\partial x_{i}} \delta_{j k}\right) u_{k I} \equiv B_{i j l k} u_{k I} \tag{A.1.15}
\end{align*}
$$

To translate this to a matrix expression in terms of column matrices for $\varepsilon_{i j}$ and a rectangular matrix for $B_{i j a}$, the kinematic Voigt rule is used for $\varepsilon_{i j}$ and the first two indices of $B_{i j K k}$ and the nodal component rule is used for the second pair of indices of $B_{i j K k}$ and the indices of $u_{k K}$. Thus

$$
\begin{equation*}
\text { elements of }[\mathbf{B}] \text { are } B_{a b} \text { where }(i, j) \rightarrow a \text { by the Voigt rule, } \tag{A.1.16a}
\end{equation*}
$$

$$
\begin{equation*}
b=(K-1) n_{S D}+k \tag{A.1.16b}
\end{equation*}
$$

The expression corresponding to (??) can then be written as

$$
\begin{equation*}
\varepsilon_{a}=B_{a b} u_{b} \quad \text { or } \quad\{\boldsymbol{\varepsilon}\}=[\mathbf{B}] \mathbf{d} \tag{A.1.17}
\end{equation*}
$$

The correspondence of the indices depends on the dimensionally of the problem. In two dimensional problems, the matrix counterpart of $B_{i j K k}$ is then written as

$$
\mathbf{B}_{K}=\left[\begin{array}{cc}
B_{11 x K} & B_{11 y K}  \tag{A.1.18}\\
B_{22 x K} & B_{22 y K} \\
2 B_{12 x K} & 2 B_{12 x K}
\end{array}\right]
$$

The full matrix for a 3-node triangle is

$$
[\mathbf{B}]=\left[\begin{array}{cccccc}
B_{x x 1 x} & B_{x x 1 y} & B_{x x 2 x} & B_{x x 2 y} & B_{x x 3 x} & B_{x x 3 y}  \tag{A.1.19}\\
B_{y y 1 x} & B_{y y 1 y} & B_{y y 2 x} & B_{y y 2 y} & B_{y y 3 x} & B_{y y 3 y} \\
2 B_{x y 1 x} & 2 B_{x y 1 y} & 2 B_{x y 2 x} & 2 B_{x y 2 y} & 2 B_{x y 3 x} & 2 B_{x y 3 y}
\end{array}\right]
$$

where the the first two indices have been replaced by the corresponding letters.
The Voigt rule is particularly useful in the implementation of stiffness matrices. In indicial notation, the stiffness matrix is written as

$$
\begin{equation*}
K_{I J r s}=\int_{\Omega} B_{i j l r} C_{i j k l} B_{k l J s} d \Omega \tag{A.1.20}
\end{equation*}
$$

The above stiffness is a fourth order matrix and maultiplying it with a matrix of nodal displacements is awkward. The indices in the above matrices can be converted by the Voigt rule, which gives

$$
\begin{equation*}
K_{a b}=\int_{\Omega} B_{a e} C_{e f} B_{f g} d \Omega \rightarrow[\mathbf{K}]=\int_{\Omega}[\mathbf{B}]^{T}[\mathbf{C}][\mathbf{B}] d \Omega \tag{A.1.21}
\end{equation*}
$$

where the indices " $I r$ " and " $J s$ " have been converted to " $a$ " and " $b$ ", respectively, by the column matrix rule and the indices " $i j$ " and " $k l$ " have been converted to " $e$ " and " $f$ " respectively by the Voigt rule. Another useful form of the stiffness matrix is obtained by transforming only the indices " $i j$ " and " $k l$ ", which gives

$$
\begin{equation*}
[\mathbf{K}]_{I J}=\int_{\Omega}[\mathbf{B}]_{I}^{T}[\mathbf{C}][\mathbf{B}]_{J} d \Omega \tag{A.1.22}
\end{equation*}
$$

where $[\mathbf{B}]_{I}$ is given in Eq. (A.1.18).
T. Belytschko, Introduction, December 16, 1998
4.5.3. Indicial to Voigt Notation. The above equations in Voigt notation can also be derived from the indicial equations if we start with the following expressions for the displacement and velocity fields

$$
\begin{equation*}
u_{i}=N_{i b} d_{b} \quad v_{i}=N_{i b} \dot{d}_{b} \tag{4.5.20}
\end{equation*}
$$

The key difference in (4.5.20) as compared to (4.4.5) is that the component index is ascribed to the shape function. In this notation, different displacement and velocity components can be approximated by different shape functions. This added generality is seldom used because it generally destroys the capability of the element to represent rigid body rotation. The gradient of the velocity field in this notation is obtained by differentiating (4.5.20):

$$
\begin{equation*}
\frac{\partial v_{i}}{\partial x_{j}}=\frac{\partial N_{i b}}{\partial x_{j}} \dot{d}_{b} \quad D_{i j}=\frac{1}{2}\left(\frac{\partial v_{i}}{\partial x_{j}}+\frac{\partial v_{j}}{\partial x_{i}}\right)=\frac{1}{2}\left(\frac{\partial N_{i b}}{\partial x_{j}}+\frac{\partial N_{j b}}{\partial x_{i}}\right) \dot{d}_{b}=B_{i j b} \dot{d}_{b}(4 . \tag{4.5.21}
\end{equation*}
$$

where

$$
\begin{equation*}
B_{i j b}=\frac{1}{2}\left(\frac{\partial N_{i b}}{\partial x_{j}}+\frac{\partial N_{j b}}{\partial x_{i}}\right)=\operatorname{sym} \frac{\partial N_{i b}}{\partial x_{j}} \tag{4.5.22}
\end{equation*}
$$

so the $B_{i j b}$ extracts the symmetric part of the gradient of the shape functions. The rectangular $\mathbf{B}$ matrix is defined by

$$
B_{a b}=B_{i j b} \text { where }(i, j) \rightarrow a \text { by the symmetric kinematic Voigt rule, see }
$$ Appendix B

and Eqs.(4.5.18) and (4.5.19) hold as before. The conversion of the internal nodal force expression to the Voigt form (4.5.19) is shown in the following

$$
\begin{equation*}
f_{b}^{i n t}=\int_{\Omega} \frac{\partial N_{i b}}{\partial x_{j}} \sigma_{j i} d \Omega=\int_{\Omega} B_{i j b} \sigma_{j i} d \Omega \rightarrow \mathbf{f}^{i n t}=\int_{\Omega} \mathbf{B}^{T}\{\boldsymbol{\sigma}\} d \Omega \tag{4.5.23}
\end{equation*}
$$

T. Belytschko, Introduction, December 16, 1998


