ADAPTING BROYDEN METHOD TO HANDLE LINEAR CONSTRAINTS IMPOSED VIA LAGRANGE MULTIPLIERS

ANTONIO RODRÍGUEZ-FERRAN AND ANTONIO HUERTA *

Departament de Matemàtica Aplicada III, E.T.S. de Ingenieros de Caminos Edifici C2, Campus Nord,
Universitat Politècnica de Catalunya, E-08034 Barcelona, Spain

SUMMARY

Various non-linear equation solvers are adapted to handle linear constraints via the Lagrange-multiplier technique. This adaptation process turns out to be quite straightforward for Newton–Raphson methods and rank-two Quasi–Newton methods (BFGS and DFP), but rather more involved for Broyden method. In fact, two Broyden methods can be obtained: the standard one and a modified one, better adapted to the Lagrange-multiplier environment. Some numerical examples are used to assess the relative performance of the various adapted solvers. These tests illustrate the superiority of the modified Broyden method over the standard one.

KEY WORDS: Lagrange multipliers; Broyden method; linear constraints; non-linear equation solvers

1. INTRODUCTION

In finite element non-linear analysis, linear constraints are often employed to represent the boundary conditions. They range from the simple case of restrained or prescribed displacements, to more complex situations like skewed supports, symmetric or cyclic boundary conditions, rigid inclusions within a deformable body,... .

The non-linear system of equations is solved iteratively by using, for instance, the Newton–Raphson method. Prior to the imposition of boundary conditions, the tangent stiffness matrix is singular, because rigid motions are not precluded. To get a regular matrix, the linear constraints must be imposed at each iteration. Two strategies are commonly used: Transformation Methods and the Lagrange-multiplier technique.

The basic idea of Transformation Methods is to employ the m linear constraints to transform the singular matrix of order n into a regular matrix. The order of this regular matrix is n (i.e. same as the original matrix) for some Transformation Methods [1–3], and $n - m$ (i.e. reduced by the number of constraints) for other Transformation Methods [4, 5]. This strategy has three clear advantages: (1) the dimension of the original problem is either maintained or decreased, (2) the regular matrix is symmetric positive definite (SPD) for elliptic self-adjoint problems often

*Correspondence to: Antonio Huerta, Departament de Matemàtica Aplicada III, E.T.S. de Ingenieros de Caminos, Edifici C2, Campus Nord, Universitat Politècnica de Catalunya, E-08034 Barcelona, Spain. E-mail: antonio.huerta@upc.es

Contract/grant sponsor: CYCIT; contract/grant number: TAP98-0421

encountered in engineering analysis and (3) it is very simple for the usual situation of prescribed degrees of freedom (dof). In such a case, the Transformation Method reduces to the row-and-column adjustment typical of finite element analysis. However, it has an important drawback: it becomes very involved for the general case of multiple multipoint constraints, especially if some dof are affected by more than one constraint [2].

If the Lagrange-multiplier technique is chosen, on the other hand, the original non-linear system is enlarged by adding m equations (the linear constraints) and m unknowns (m Lagrange multipliers λ, one per constraint) [6, 7]. This method has two widely known drawbacks: (1) the dimension of the original problem is increased by the number of constraints and (2) the enlarged matrix is not SPD. However, it has two clear advantages: (1) general linear constraints can be handled in a very straightforward manner and (2) it is a technique naturally adapted to object-oriented programming [8–11], because the internal stiffness of the physical system and the boundary conditions are represented by two separate and distinct objects, the tangent stiffness matrix and the constraint matrix, respectively [12–14]. This independency between objects is a crucial concept in object-oriented programming.

Quasi–Newton non-linear solvers are employed to avoid the full recomputation of the tangent stiffness matrix associated with the Newton–Raphson method [15–17]. Easy-to-compute secant approximations to the tangent stiffness matrix are obtained according to some specifications.

The Lagrange-multiplier technique and Quasi–Newton solvers are nowadays classical numerical tools, and very clear presentations can be found at the textbook level (see for instance [6, 7, 15, 17]). However, to the authors' knowledge, the algorithmic issues related to the combination of Lagrange multipliers and Quasi–Newton methods have not been addressed in detail. The following questions arise: (1) how should Quasi–Newton methods be adapted to Lagrange multipliers? (2) is the structure of the enlarged matrix properly exploited by Quasi–Newton methods? The basic objective of this paper is to show how the most widely used Quasi–Newton methods (DFP, BFGS, Broyden) can be adapted to the Lagrange-multiplier technique. It is shown that this adaptation is straightforward for the rank-two methods (DFP and BFGS), but rather more involved for the Broyden method. The adaptation process leads to a modified Broyden method which preserves the structure of the problem more efficiently than the standard Broyden method. This modified Broyden method can be regarded as a least-change secant method in the sense of [18].

An outline of paper follows. Section 2 presents some standard non-linear solvers (Newton–Raphson, DFP, BFGS, Broyden) in the context of Lagrange multipliers. All of them are naturally adapted to the particular structure of the enlarged system except from the standard Broyden method. The modified Broyden method is then presented in Section 3, and its convergence properties are discussed in Section 4. The relative performance of the various solvers is assessed in Section 5 by means of some numerical examples. Finally, some concluding remarks are made in Section 6.

2. STANDARD NON-LINEAR SOLVERS FOR LAGRANGE MULTIPLIERS

The notation of non-linear structural analysis [15, 19] will be employed here to state the problem. The basic ideas and the algorithms, however, can be extended to other non-linear problems. The non-linear system and the linear constraints are written as

$$\mathbf{r}(\mathbf{u}) = \mathbf{f}_{\text{int}}(\mathbf{u}) - \mathbf{f}_{\text{ext}} + \mathbf{f}_{\text{reac}} = \mathbf{0} \tag{1a}$$

$$\mathbf{A}\mathbf{u} = \mathbf{b} \tag{1b}$$

where $\mathbf{u}$ is the vector of nodal displacements, $\mathbf{f}_{\text{int}}(\mathbf{u})$ is the vector of internal forces, which depends nonlinearly on displacements, $\mathbf{f}_{\text{ext}}$ is the vector of applied external forces, $\mathbf{f}_{\text{reac}}$ is the vector of reaction forces and $\mathbf{r}(\mathbf{u})$ is the vector of residual (off-balance) forces, which are null if the equilibrium represented by equation (1a) is verified. Equation (1b) represents the linear constraints, with $\mathbf{A}$ an $m \times n$ rectangular constraint matrix and $\mathbf{b}$ a vector with the prescribed values of the constraints. Following standard notation, $\mathbf{x}$ is a column vector and $\mathbf{x}^{\mathrm{T}}$ is a row vector, where T denotes transpose.

If the Lagrange-multiplier technique is chosen, the reaction forces are written as $\mathbf{f}_{\text{reac}} = \mathbf{A}^{\mathrm{T}}\boldsymbol{\lambda}$, where $\boldsymbol{\lambda}$ is a vector of m Lagrange multipliers, and the linear constraints are explicitly added to the non-linear equations, equation (1a). The enlarged system of order $n+m$ is

$$\begin{aligned} \mathbf{r}_1(\mathbf{u},\boldsymbol{\lambda}) &= \mathbf{f}_{\text{int}}(\mathbf{u}) - \mathbf{f}_{\text{ext}} + \mathbf{A}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0} \\ \mathbf{r}_2(\mathbf{u}) &= \mathbf{A}\mathbf{u} - \mathbf{b} = \mathbf{0} \end{aligned} \tag{2}$$

which can be written as $\mathbf{r}(\mathbf{x}) = 0$, with $\mathbf{x}^{\mathrm{T}} = (\mathbf{u}^{\mathrm{T}}\ \boldsymbol{\lambda}^{\mathrm{T}})$ and $\mathbf{r}^{\mathrm{T}} = (\mathbf{r}_1^{\mathrm{T}}\ \mathbf{r}_2^{\mathrm{T}})$. This non-linear system is partially linear, because $\mathbf{r}_1$ is linear on $\boldsymbol{\lambda}$ and $\mathbf{r}_2$ is linear on $\mathbf{u}$.

The solution $\mathbf{x}$ is obtained in an incremental/iterative manner, according to the scheme

$${}^{n}\mathbf{x}^{k+1} = {}^{n}\mathbf{x}^{k} + \boldsymbol{\delta}\mathbf{x}^{k+1}$$

where the left superscript denotes time (or pseudotime load parameter for quasistatic problems) and the right superscript is the iteration counter.

2.1. *Newton–Raphson method*

If the Newton–Raphson method is employed, the iterative correction $\boldsymbol{\delta}\mathbf{x}^{k+1}$ is obtained by solving the linear system

$$\mathbf{J}({}^{n}\mathbf{x}^{k})\boldsymbol{\delta}\mathbf{x}^{k+1} = -\mathbf{r}({}^{n}\mathbf{x}^{k})$$

where the Jacobian matrix $\mathbf{J}(\mathbf{x}) = \partial\mathbf{r}/\partial\mathbf{x}$ can be expressed, accounting for equation (2), as

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} \partial\mathbf{r}_1/\partial\mathbf{u} & \partial\mathbf{r}_1/\partial\boldsymbol{\lambda} \\ \partial\mathbf{r}_2/\partial\mathbf{u} & \partial\mathbf{r}_2/\partial\boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{K}(\mathbf{u}) & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \tag{3}$$

with $\mathbf{K}(\mathbf{u}) = \partial\mathbf{r}_1/\partial\mathbf{u}$ the (singular) tangent stiffness matrix. It can be seen in equation (3) that this singular matrix $\mathbf{K}(\mathbf{u})$ is not altered (as it is done in a Transformation Method), but enlarged into a (regular) Jacobian matrix $\mathbf{J}(\mathbf{x})$ with the aid of the constraint matrix $\mathbf{A}$. This means that the internal stiffness of the physical system and the boundary conditions are represented by two independent objects, $\mathbf{K}(\mathbf{u})$ and $\mathbf{A}$, which can be merged to create a new object, $\mathbf{J}(\mathbf{x})$, without altering the original objects $\mathbf{K}(\mathbf{u})$ and $\mathbf{A}$. This process fits very naturally into object-oriented programming [8–11], where the independency between objects is a key feature. For instance, in [12], the linear constraints are handled in an object-oriented framework by means of a specific class, `LinearConstraint`, fully independent of the stiffness matrix. It must be pointed out that $\mathbf{K}(\mathbf{u})$ is the only variable block-matrix of $\mathbf{J}(\mathbf{x})$ in equation (3); the other three are constant, thus reflecting the partial linearity of the problem. It can be concluded that the Newton–Raphson method is (trivially) well adapted to the particular structure of equation (2). By replacing $\mathbf{K}(\mathbf{u})$ in equation (3) by the appropiate stiffness matrix, the modified Newton–Raphson and initial stress methods are obtained.

2.2. *Quasi–Newton methods*

The basic idea of (direct) Quasi–Newton methods is to employ secant approximations to the Jacobian matrix which are easy to compute [16, 17]. By doing so, the Jacobian matrix is neither recomputed at every iteration (full Newton–Raphson method) nor kept constant for several iterations (modified Newton–Raphson methods), but updated in a non-expensive, simple manner. This results in a good balance between computational cost per iteration and convergence properties: a superlinear rate of convergence is obtained [16]. For implementation purposes, inverse Quasi–Newton methods are often preferred [20, 21]. In this case, secant approximations to the inverse of the Jacobian matrix are employed.

The condition of secant approximation is expressed in the Quasi–Newton equation [16, 17], which reads

$$\mathbf{B}^k \boldsymbol{\delta}\mathbf{x}^k = \mathbf{r}^k - \mathbf{r}^{k-1} \tag{4}$$

and states that a secant approximation $\mathbf{B}^k$ to $\mathbf{J}^k$ must 'pass through' the points $(\mathbf{x}^{k-1},\mathbf{r}^{k-1})$ and $(\mathbf{x}^k,\mathbf{r}^k)$. Left superscripts denoting time are dropped to ease the notation. This Quasi–Newton equation does not uniquely determine the secant approximation, and additional condition on $\mathbf{B}^k$ are required. These additional conditions are specific of each particular Quasi–Newton method.

In the Lagrange-multiplier context of this work, a key issue is whether the secant approximation $\mathbf{B}^k$ has the same structure as the Jacobian matrix (three constant block-matrices and only one variable block-matrix, see equation (3)) or not. In other words, whether the Quasi–Newton methods are naturally adapted to the partial linearity of the problem or not. This issue is addressed for some of the most widely used Quasi–Newton methods in the rest of this Section.

2.2.1. Rank-two methods. The BFGS method is an inverse Quasi–Newton method [17, 20], so secant approximations $\mathbf{H}^k$ to the inverse of the Jacobian matrix, $(\mathbf{J}^k)^{-1}$, are employed. The additional conditions for the BFGS method [17], are hereditary symmetry and positive definiteness of the secant approximation (i.e. if $\mathbf{H}^{k-1}$ is SPD then $\mathbf{H}^k$ is also SPD). Together with an inverse version of the Quasi–Newton equation, equation (4), this requirement leads to the update formula

$$\begin{aligned}\mathbf{H}^k_{\mathrm{BFGS}} = \mathbf{H}^{k-1}_{\mathrm{BFGS}} + \mathbf{N}^k &= \mathbf{H}^{k-1}_{\mathrm{BFGS}} \\ &+ \frac{(-\mathbf{H}^{k-1}_{\mathrm{BFGS}}\mathbf{r}^k)(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}} + \boldsymbol{\delta}\mathbf{x}^k(-\mathbf{H}^{k-1}_{\mathrm{BFGS}}\mathbf{r}^k)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}}\mathbf{y}^k} - \frac{(-\mathbf{H}^{k-1}_{\mathrm{BFGS}}\mathbf{r}^k)^{\mathrm{T}}\mathbf{y}^k}{[(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}}\mathbf{y}^k]^2}\boldsymbol{\delta}\mathbf{x}^k(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}}\end{aligned} \tag{5}$$

where $\mathbf{y}^k = \mathbf{r}^k - \mathbf{r}^{k-1}$, and $\mathbf{N}^k$ is a rank-two modification matrix that allows to compute $\mathbf{H}^k_{\mathrm{BFGS}}$ as a simple update of $\mathbf{H}^{k-1}_{\mathrm{BFGS}}$.

Since the BFGS is an inverse Quasi–Newton method, it is difficult to check if the pattern of the Jacobian matrix is respected directly from equation (5). To do that, equation (5) must be inverted to produce the so-called direct version of the BFGS method [16]

$$\mathbf{B}^k_{\mathrm{BFGS}} = \mathbf{B}^{k-1}_{\mathrm{BFGS}} + \mathbf{M}^k = \mathbf{B}^{k-1}_{\mathrm{BFGS}} + \frac{\mathbf{y}^k(\mathbf{y}^k)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}}\mathbf{y}^k} + \frac{\mathbf{r}^k(\mathbf{r}^k)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}}\mathbf{r}^k} \tag{6}$$

The modification matrix $\mathbf{M}^k$, see the previous equation, can be written, in block-matrix format, as

$$\mathbf{M}^k = \begin{pmatrix} \mathbf{M}^k_{11} & \mathbf{M}^k_{12} \\ \mathbf{M}^k_{21} & \mathbf{M}^k_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{M}^k_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{7}$$

The key point about equation (7) is that only the block-matrix $\mathbf{M}_{11}^k$ is different from zero. This can be checked by expressing $(\mathbf{r}^k)^{\mathrm{T}}$ as $((\mathbf{r}_1^k)^{\mathrm{T}}\ (\mathbf{r}_2^k)^{\mathrm{T}})$ and $(\mathbf{y}^k)^{\mathrm{T}}$ as $((\mathbf{y}_1^k)^{\mathrm{T}}\ (\mathbf{y}_2^k)^{\mathrm{T}})$. If the iterative update scheme represented by equation (6) is initialized with a true Jacobian matrix (computed at the beginning of the time-step, for instance), then the secant approximation $\mathbf{B}_{\mathrm{BFGS}}^k$ is simply

$$\mathbf{B}_{\mathrm{BFGS}}^k = \begin{pmatrix} \left\{\mathbf{K} + \sum_i \mathbf{M}_{11}^i\right\} & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{0} \end{pmatrix}$$

where $\mathbf{K}$ is the tangent stiffness matrix of the Jacobian matrix used as $\mathbf{B}_{\mathrm{BFGS}}^0$. Since $\mathbf{M}^k$ has the structure shown in equation (7), the secant approximation $\mathbf{B}_{\mathrm{BFGS}}^k$ always has the structure of a Jacobian matrix, see equation (3), with three constant block-matrices. As a consequence, it can be concluded that the standard BFGS method is automatically well adapted to the partial linearity of equation (2).

Remark 1. Since $\mathbf{r}_2$ is a linear function of $\mathbf{u}$, it is null after the prediction, $\mathbf{r}_2^k = \mathbf{0}$ for $k \geqslant 1$. The fact that $\mathbf{M}^k$ has the structure shown in equation (7) relies on the assumption that the initial approximation $\mathbf{u}^0$ complies with the linear constraints ($\mathbf{r}_2^0 = \mathbf{0}$). This implies that $\mathbf{r}_2^k$ is null for $k \geqslant 0$, and equation (7) follows directly. If the initial approximation does not verify the linear constraints, then a minor modification is needed in the initialization of the BFGS algorithm [13].

A similar result is obtained with the DFP method, which is a direct rank-two Quasi–Newton method [17, 22], where hereditary symmetry and positive definiteness is imposed to secant approximations $\mathbf{B}^k$.

2.2.2. Broyden method. The Broyden method is a rank-one Quasi–Newton method, where the additional condition is that two consecutive matrices $\mathbf{B}^{k-1}$ and $\mathbf{B}^k$ have the same behaviour in all directions except $\boldsymbol{\delta}\mathbf{x}^k$, which can be written as

$$\mathbf{B}^k\mathbf{z} = \mathbf{B}^{k-1}\mathbf{z} \quad \text{for all } \mathbf{z} \text{ such that } \mathbf{z}^{\mathrm{T}}\boldsymbol{\delta}\mathbf{x}^k = 0 \tag{8}$$

Together with the Quasi–Newton equation, equation (4), this requirement renders the update scheme

$$\mathbf{B}_{\mathrm{Broyden}}^k = \mathbf{B}_{\mathrm{Broyden}}^{k-1} + \mathbf{M}^k = \mathbf{B}_{\mathrm{Broyden}}^{k-1} + \frac{\mathbf{r}^k(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}}\boldsymbol{\delta}\mathbf{x}^k} \tag{9}$$

Recalling that $(\mathbf{r}^k)^{\mathrm{T}} = ((\mathbf{r}_1^k)^{\mathrm{T}}\mathbf{0}^{\mathrm{T}})$ for $k \geqslant 1$ and $(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}} = ((\boldsymbol{\delta}\mathbf{u}^k)^{\mathrm{T}}\ (\boldsymbol{\delta}\boldsymbol{\lambda}^k)^{\mathrm{T}})$, the rank-one modification matrix $\mathbf{M}^k$ of equation (9) can be expressed as

$$\mathbf{M}^k = \begin{pmatrix} \dfrac{\mathbf{r}_1^k(\boldsymbol{\delta}\mathbf{u}^k)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}}\boldsymbol{\delta}\mathbf{x}^k} & \dfrac{\mathbf{r}_1^k(\boldsymbol{\delta}\boldsymbol{\lambda}^k)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{x}^k)^{\mathrm{T}}\boldsymbol{\delta}\mathbf{x}^k} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{10}$$

Equation (10) has an important difference with the rank-two methods just discussed: there are two non-zero block-matrices. This means that the secant matrix in a generic iteration is

$$\mathbf{B}_{\mathrm{Broyden}}^k = \begin{pmatrix} \left\{\mathbf{K} + \sum_i \dfrac{\mathbf{r}_1^i(\boldsymbol{\delta}\mathbf{u}^i)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{x}^i)^{\mathrm{T}}\boldsymbol{\delta}\mathbf{x}^i}\right\} & \left\{\mathbf{A}^{\mathrm{T}} + \sum_i \dfrac{\mathbf{r}_1^i(\boldsymbol{\delta}\boldsymbol{\lambda}^i)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{x}^i)^{\mathrm{T}}\boldsymbol{\delta}\mathbf{x}^i}\right\} \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \tag{11}$$

It can be seen in equation (11) that $\mathbf{B}^k_{\mathrm{Broyden}}$ does *not* have the structure of the Jacobian matrix, because the block-matrix $\mathbf{A}^{\mathrm{T}}$ is modified.

It can be concluded from equation (11) that the standard Broyden method does not profit efficiently from the special structure of the problem. A modified Broyden method which is adapted to the partial linearity of the problem is presented in the next section. Moreover, the inverse version of the method is discussed in detail.

3. A MODIFIED BROYDEN METHOD FOR LAGRANGE MULTIPLIERS

The basic idea of the modified Broyden method is to impose a priori that the secant approximation $\mathbf{B}^k$ has the three constant block-matrices of the Jacobian matrix. It consists of two steps:

(1) Obtain a secant approximation $\mathbf{B}^k_{11}$ to the tangent stiffness matrix $\mathbf{K}^k$:

$$\mathbf{B}^k_{11} \approx \mathbf{K}^k$$

(2) Enlarge $\mathbf{B}^k_{11}$ with the *constant* block-matrices $\mathbf{A}^{\mathrm{T}}$, $\mathbf{A}$ and $\mathbf{0}$ to get a secant approximation to the Jacobian matrix

$$\mathbf{B}^k_{\mathrm{Mod}} = \begin{pmatrix} \mathbf{B}^k_{11} & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \approx \mathbf{J}^k$$

To perform step 1, both the Quasi–Newton equation, equation (4), and the additional condition of the Broyden method, equation (8), must be adapted to this alternative approach. Imposing that the full $\mathbf{B}^k_{\mathrm{Mod}}$ satisfies the original Quasi–Newton equation, equation (4), renders

$$\begin{pmatrix} \mathbf{B}^k_{11} & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\delta}\mathbf{u}^k \\ \boldsymbol{\delta}\boldsymbol{\lambda}^k \end{pmatrix} = \begin{pmatrix} \mathbf{r}^k_1 - \mathbf{r}^{k-1}_1 \\ \mathbf{0} \end{pmatrix} \tag{12}$$

The first-block equation in equation (12) reads

$$\mathbf{B}^k_{11}\boldsymbol{\delta}\mathbf{u}^k = \mathbf{r}^k_1 - \mathbf{r}^{k-1}_1 - \mathbf{A}^{\mathrm{T}}\boldsymbol{\delta}\boldsymbol{\lambda}^k \tag{13}$$

which is the Quasi–Newton equation for matrix $\mathbf{B}^k_{11}$. Regarding the additional condition (8), it can be recast as

$$\mathbf{B}^k_{11}\mathbf{z}_1 = \mathbf{B}^{k-1}_{11}\mathbf{z}_1 \quad \text{for all } \mathbf{z}_1 \text{ such that } (\mathbf{z}_1)^{\mathrm{T}}\boldsymbol{\delta}\mathbf{u}^k = 0 \tag{14}$$

Combining equations (13) and (14) yields the update scheme

$$\mathbf{B}^k_{11} = \mathbf{B}^{k-1}_{11} + \frac{\mathbf{r}^k_1(\boldsymbol{\delta}\mathbf{u}^k)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{u}^k)^{\mathrm{T}}\boldsymbol{\delta}\mathbf{u}^k}$$

so the secant approximation to the Jacobian matrix in a generic iteration k is

$$\mathbf{B}^k_{\mathrm{Mod}} = \begin{pmatrix} \left\{\mathbf{K} + \sum_{i=1}^{k} \frac{\mathbf{r}^i_1(\boldsymbol{\delta}\mathbf{u}^i)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{u}^i)^{\mathrm{T}}\boldsymbol{\delta}\mathbf{u}^i}\right\} & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{0} \end{pmatrix}$$

where $\mathbf{B}^k_{\mathrm{Mod}}$ has, by construction, the structure of a Jacobian matrix.

The modified Broyden method just presented is a particular case of a ‘least-change secant method’, as defined in [17]. Least-change secant methods are designed for non-linear systems with Jacobian matrices of the form

$$\mathbf{J}(\mathbf{x}) = \mathbf{J}_1(\mathbf{x}) + \mathbf{J}_2(\mathbf{x}) \tag{15}$$

where $\mathbf{J}_1(\mathbf{x})$ must be secantly approximated but $\mathbf{J}_2(\mathbf{x})$ can be easily computed. To build secant approximations $\mathbf{B}_1(\mathbf{x})$ to $\mathbf{J}_1(\mathbf{x})$, least-change secant methods proceed in three steps [17]:

(1) Decide the secant condition (i.e. the modified Quasi–Newton equation) that matrices $\mathbf{B}_1(\mathbf{x})$ must verify.
(2) Choose an affine subspace $\mathscr{A}$ defined by properties such as symmetry or sparsity that all the approximants $\mathbf{B}_1(\mathbf{x})$ should have.
(3) Select $\mathbf{B}_1^k$ to be the matrix closest to $\mathbf{B}_1^{k-1}$ that simultaneously (i) verifies the secant condition of step 1 and (ii) belongs to the affine subspace $\mathscr{A}$ selected in step 2. In this manner, $\mathbf{B}_1^k$ is the least-change update to $\mathbf{B}_1^{k-1}$ with the desired properties.

It can be checked that the modified Broyden method fits into this framework. Recalling the expression of the Jacobian matrix, equation (3), it can be seen that the matrices $\mathbf{J}_1(\mathbf{x})$ and $\mathbf{J}_2(\mathbf{x})$ for the non-linear system under consideration, equation (2), are

$$\mathbf{J}_1(\mathbf{x}) = \begin{pmatrix} \mathbf{K}(\mathbf{u}) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{J}_2(\mathbf{x}) = \mathbf{J}_2 = \begin{pmatrix} \mathbf{0} & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \tag{16}$$

where $\mathbf{J}_2(\mathbf{x})$ is, in fact, a constant matrix $\mathbf{J}_2$.

By combining equations (15) and (16), it can be verified that the modified Quasi–Newton equation for the modified Broyden method, equation (13), can be put as

$$\mathbf{B}_1^k \boldsymbol{\delta}\mathbf{x}^k = \mathbf{r}^k - \mathbf{r}^{k-1} - \mathbf{J}_2 \boldsymbol{\delta}\mathbf{x}^k \tag{17}$$

which is the secant condition for matrix $\mathbf{B}_1^k$.

Regarding the choice of the affine subspace $\mathscr{A}$, all the approximations $\mathbf{B}_1^k$ are required to have the sparsity pattern of $\mathbf{J}_1(\mathbf{x})$ (i.e. the three null block-matrices, see equation (16)). This requirement defines $\mathscr{A}$.

Finally, the additional condition of the Broyden method, equation (8), can be interpreted as requiring $\mathbf{B}_1^k$ to be the matrix closest to $\mathbf{B}_1^{k-1}$ (measured with the Frobenius norm) which verifies equation (17) and has the sparsity pattern of $\mathbf{J}_1(\mathbf{x})$.

From an algorithmic viewpoint, the inverse Broyden method is preferred to the direct Broyden method just discussed [16, 21]. The basic idea is to employ the Sherman and Morrison lemma [16, 17] to invert the direct update formula. Assuming that $\mathbf{B}$ is a regular matrix, this lemma gives an explicit expression for the inverse of $\mathbf{B} + \boldsymbol{\alpha\beta}^{\mathrm{T}}$.

Some care is required in the inversion procedure. The update scheme for matrix $\mathbf{B}_{11}$, equation (13), *cannot* be inverted because matrix $\mathbf{B}_{11}$, which is an approximation of the tangent stiffness matrix $\mathbf{K}$, is singular (rigid modes not removed). The Sherman and Morrison lemma must be applied to the full matrix $\mathbf{B}_{\mathrm{Mod}}^k$, which is regular because $\mathbf{A}$ accounts for boundary conditions. To that aim, it is necessary that the update formula for the block-matrix $\mathbf{B}_{11}$ can be transformed into an update formula for $\mathbf{B}_{\mathrm{Mod}}$ of the form

$$\mathbf{B}_{\mathrm{Mod}}^k = \mathbf{B}_{\mathrm{Mod}}^{k-1} + \boldsymbol{\alpha}^k (\boldsymbol{\beta}^k)^{\mathrm{T}} \tag{18}$$

It can be checked that equation (18) holds for $\boldsymbol{\alpha}^k = \mathbf{r}^k$ and $(\boldsymbol{\beta}^k)^{\mathrm{T}} = \left(\frac{(\boldsymbol{\delta}\mathbf{u}^k)^{\mathrm{T}}}{(\boldsymbol{\delta}\mathbf{u}^k)^{\mathrm{T}}\boldsymbol{\delta}\mathbf{u}^k} \;\; \mathbf{0}^{\mathrm{T}} \right)$, and then it can be inverted with the Sherman and Morrison lemma into

$$\mathbf{H}^k_{\mathrm{Mod}} = \mathbf{H}^{k-1}_{\mathrm{Mod}} - \frac{\mathbf{H}^{k-1}_{\mathrm{Mod}}\mathbf{r}^k((\boldsymbol{\delta}\mathbf{u}^k)^{\mathrm{T}}\;\mathbf{0}^{\mathrm{T}})\mathbf{H}^{k-1}_{\mathrm{Mod}}}{((\boldsymbol{\delta}\mathbf{u}^k)^{\mathrm{T}}\;\mathbf{0}^{\mathrm{T}})\mathbf{H}^{k-1}_{\mathrm{Mod}}\mathbf{y}^k}$$

The inverse Broyden method is typically implemented by means of a recurrent product formula [21, 23] which relates $\mathbf{H}^k$ to the initialization matrix $\mathbf{H}^0$. This procedure has been adapted to both the standard and modified Broyden methods that arise in a Lagrange-multiplier context, see Box 1.

Box 1. Standard and modified Broyden methods

```
    FOR EVERY LOAD-STEP:
 1. Choose an initial approximation $\mathbf{x}^0$ and compute the residual vector $\mathbf{r}^0$.
 2. Compute the initialization matrix $\mathbf{B}^0$.
 3. Solve the linear system $\mathbf{B}^0\boldsymbol{\delta}\mathbf{x}^1 = -\mathbf{r}^0$ and store $\boldsymbol{\delta}\mathbf{x}^1$.
 4. Update $\mathbf{x}^1 = \mathbf{x}^0 + \boldsymbol{\delta}\mathbf{x}^1$ and compute $\mathbf{r}^1$.
 5. Convergence control: if $\mathbf{x}^1$ is good enough, exit.
    $k \leftarrow 1$
 6. Solve the linear system $\mathbf{B}^0\mathbf{v} = -\mathbf{r}^k$.
 7. For $k = 1$ set $\mathbf{t} = \mathbf{v}$.
    For $k > 1$ recover $\boldsymbol{\omega}^i$, $\boldsymbol{\delta}\mathbf{x}^i$ for $i = 1, \ldots k-1$ from storage and compute
        $\mathbf{t} = \prod_{i=k-1}^{1} (\mathbf{I} + \boldsymbol{\omega}^i(\boldsymbol{\delta}\mathbf{x}^i)^{\mathrm{T}})\mathbf{v}.$
 8. Recover $\boldsymbol{\delta}\mathbf{x}^k$ from storage and define $\boldsymbol{\beta}^k$:
        STANDARD :   $\boldsymbol{\beta}^k = \begin{pmatrix} \boldsymbol{\delta}\mathbf{u}^k \\ \boldsymbol{\delta}\boldsymbol{\lambda}^k \end{pmatrix} = \boldsymbol{\delta}\mathbf{x}^k$
        MODIFIED :   $\boldsymbol{\beta}^k = \begin{pmatrix} \boldsymbol{\delta}\mathbf{u}^k \\ \mathbf{0} \end{pmatrix}$
 9. Compute and store $\boldsymbol{\omega}^k = \frac{\mathbf{t}}{(\boldsymbol{\beta}^k)^{\mathrm{T}}(\boldsymbol{\delta}\mathbf{x}^k - \mathbf{t})}$.
10. Compute and store $\boldsymbol{\delta}\mathbf{x}^{k+1} = \mathbf{t} + [(\boldsymbol{\beta}^k)^{\mathrm{T}}\mathbf{t}]\boldsymbol{\omega}^k$.
11. Update $\mathbf{x}^{k+1} = \mathbf{x}^k + \boldsymbol{\delta}\mathbf{x}^{k+1}$ and compute $\mathbf{r}^{k+1}$.
12. Convergence control: if $\mathbf{x}^{k+1}$ is good enough, exit.
13. Set $k \leftarrow k+1$ and go back to 6.
```

It can be seen in Box 1 that the two algorithms are identical, except for the definition of vector $\boldsymbol{\beta}^k$ in step 8. It is also worth mentioning that the scalar products in steps 9 and 10 involve only the displacement dof for the modified Broyden method, but both displacements and Lagrange multipliers for the standard Broyden method. This is due to the fact that $(\boldsymbol{\beta}^k)^{\mathrm{T}} = ((\boldsymbol{\delta}\mathbf{u}^k)^{\mathrm{T}}\;\mathbf{0}^{\mathrm{T}})$ in the modified Broyden method.

The major drawback of the inverse Quasi–Newton methods is that the computational cost *per iteration* increases at each iteration. For the inverse Broyden method (standard and modified), this can be checked from Box 1, where the number of stored vectors and scalar products increase with the iteration counter k. A similar situation is found with the BFGS method [15, 21]. The

Secant–Newton methods [15] were developed to remedy this problem. The basic idea is to replace matrix $\mathbf{H}^{k-1}$ with the initialization matrix $\mathbf{H}^0$ in the update schemes. From an algorithmic stand, this results in a constant number of scalar products and stored vectors, that is, a constant cost per iteration [15]. The simplifying strategy is general, so a Secant–Newton method can be obtained from every inverse Quasi–Newton method. Thus, in a Lagrange-multiplier setting, it is possible to devise two Secant–Broyden methods (standard and modified) and one Secant BFGS method [14].

4. CONVERGENCE OF THE MODIFIED BROYDEN METHOD

For the standard Quasi–Newton methods of Section 2 (BFGS, DFP and Broyden), the usual convergence theorems directly apply [17, 24]: under certain regularity conditions, these methods converge locally and superlinearly to a solution $\mathbf{x}^*$ of the non-linear system (that is, $\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leqslant c^k \|\mathbf{x}^k - \mathbf{x}^*\|$ for some sequence $\{c^k\}$ that converges to 0, provided that $\|\mathbf{x}^0 - \mathbf{x}^*\| \leqslant \varepsilon$ and $\|\mathbf{B}^0 - \mathbf{J}^*\| \leqslant \delta$).

The general theorem of Dennis and Walker [18] states the *superlinear* convergence of least-change secant methods under certain conditions. In particular, under the usual regularity assumptions, $\mathbf{J}_1(\mathbf{x}^*)$ belonging to the affine subspace $\mathscr{A}$ is a *sufficient* condition for local superlinear convergence to a solution $\mathbf{x}^*$ of the non-linear system.

For the modified Broyden method, this sufficient condition is trivially verified, because $\mathscr{A}$ is selected precisely by imposing the sparsity pattern of $\mathbf{J}_1(\mathbf{x})$. In conclusion, the modified Broyden method is locally superlinearly convergent.

5. NUMERICAL EXAMPLES

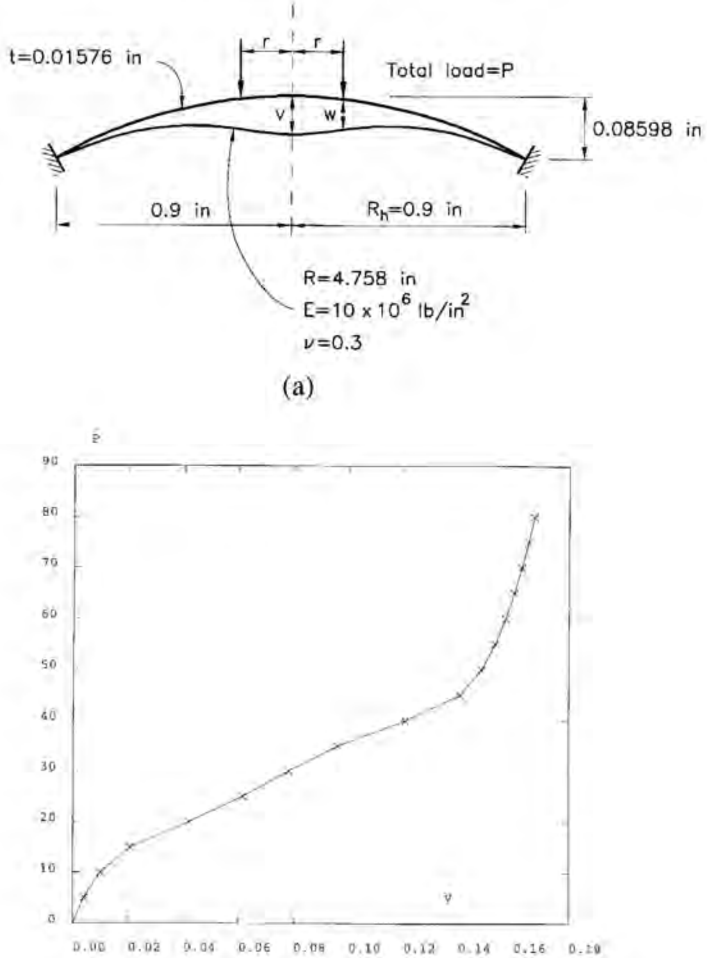
The algorithms just discussed have been implemented in CASTEM 2000, an object-oriented code that employs the Lagrange-multiplier technique to handle linear constraints [8, 25, 26]. Several numerical tests in the fields of non-linear mechanical and thermal analysis have been performed. Three of these numerical tests are shown in this section, with the purpose of (1) showing that the various methods can be effectively implemented in an object-oriented, Lagrange-multiplier environment, (2) assessing the relative performance of the different non-linear solvers and (3) comparing the standard and modified Broyden methods, both for Quasi–Newton and Secant–Newton versions.

5.1. Test SHELL

The first example is a well-known benchmark test in non-linear mechanics [7, 27]. An axisymmetrical shell, see Figure 1(a), is clamped at its border and loaded with a vertical force in its apex ($r = 0$). The shell is made of an elastic material but undergoes large deformations, so a geometrically non-linear problem results.

A load-controlled incremental-iterative analysis is performed. A total load of $P = 80$ lb is applied in 16 load steps. A relative criterion in displacements measured with the l_∞ (or maximum) norm, with a tolerance of $\varepsilon = 10^{-4}$ is used as the basic convergence criterion, and convergence in forces is also checked. Figure 1(b) shows the load P versus the deflection of the apex, v.

The test has been performed with various Quasi–Newton and Secant–Newton methods. Table I shows their computational cost, both in terms of total number of iterations and CPU time. The



(a)

(b)

Figure 1. Test SHELL: (a) problem statement; and (b) load versus deflection

best Quasi–Newton method is the BFGS. This is an expected result, because the Jacobian (i.e. tangent stiffness) matrices are symmetric for this mechanical problem. Second comes the modified Broyden method. It has a higher computational cost than BFGS, both in terms of iterations and CPU time. It is interesting to note that a 45 per cent increase in the number of iterations (from 130 to 189) is associated with only a 36 per cent increase in CPU time, thus reflecting that the (rank-one) Broyden method has a lower cost per iteration than the (rank-two) BFGS method.

The standard Broyden method does not converge (a maximum number of 50 iterations per step is set), because it is not well adapted to the linear constraints (i.e. clamped border) imposed via Lagrange multipliers. Since the block-matrix $\mathbf{A}^{\mathrm{T}}$ is modified at each iteration, see equation (11), the matrix $\mathbf{B}^k_{\text{Broyden}}$ does not tend to the true Jacobian $\mathbf{J}^k$ when $k \to \infty$. As a consequence, convergence

Table I. Test SHELL. Computational cost of various methods

Method	Iterations	CPU time per cent
Modified Broyden (QN)	189	136
Standard Broyden (QN)	—	—
BFGS (QN)	130	100
Modified Secant–Broyden	225	137
Standard Secant–Broyden	—	—
Secant–BFGS	—	—

is more difficult than for the modified Broyden method. Among the Secant–Newton methods, the only one that achieves convergence is the modified Secant–Broyden method.

5.2. *Test THERMAL*

The second test studies the non-linear diffusion of heat [21]. Nonlinearity is associated to a non-constant, temperature-dependent thermal conductivity $K=1+2T^2$, where T is the temperature. For this problem, the conductivity matrices (and hence the Jacobian matrices) are non-symmetric [21].

The problem domain is a hollow disc with six rectangular perforations, see Figure 2(a). The boundary conditions, also shown in Figure 2(a), are: prescribed temperature in the inner circle ($T=0$) and in the rectangular inclusions ($T=10$), and a convection condition in the outer circle.

Since the problem has a cyclic periodicity of angle $\pi/3$, the solution must have the same periodicity. This allows to solve the problem with only one sixth of the total domain, see Figure 2(b). The temperature is set to be equal (pointwise) in the two straight lines, thus imposing the periodicity in the solution. This linear constraint can be implemented in a simple and efficient manner thanks to the combination of Lagrange multipliers and object-oriented environment: the two straight lines are two independent objects, and the constraint matrix $\mathbf{A}$ is produced with an appropriate operator. The resulting temperature field is shown in Figure 2(c).

This test allows the comparison of the Quasi–Newton methods for a problem with non-symmetric matrices. A single increment is performed. The results can be seen in Figure 2(d), which shows the convergence history (logarithm of the relative error versus iterations), and in Table II, which displays the computational cost. The modified Broyden method is clearly superior to the BFGS method (as expected, since Jacobian matrices are not symmetric) and to the standard Broyden method. The slow convergence rate of this latter method in the first 10 iterations is caused by the difficulties in dealing with the linear constraints imposed via Lagrange multipliers. It is also worth mentioning that the effect of the increasing cost per iteration associated to Quasi–Newton methods is manifest in Table II: the standard Broyden method takes 1·36 as many iterations as the modified Broyden method, but 1·68 as much CPU time.

5.3. *Test TWO HOLES*

A strip with two perforations, see Figure 3(a), is subjected to uniaxial compression. This test is inspired in an example found in Reference 28. In this reference, the two holes are along the longitudinal axis of symmetry, so only a quarter of the specimen is modelled. Here the position of

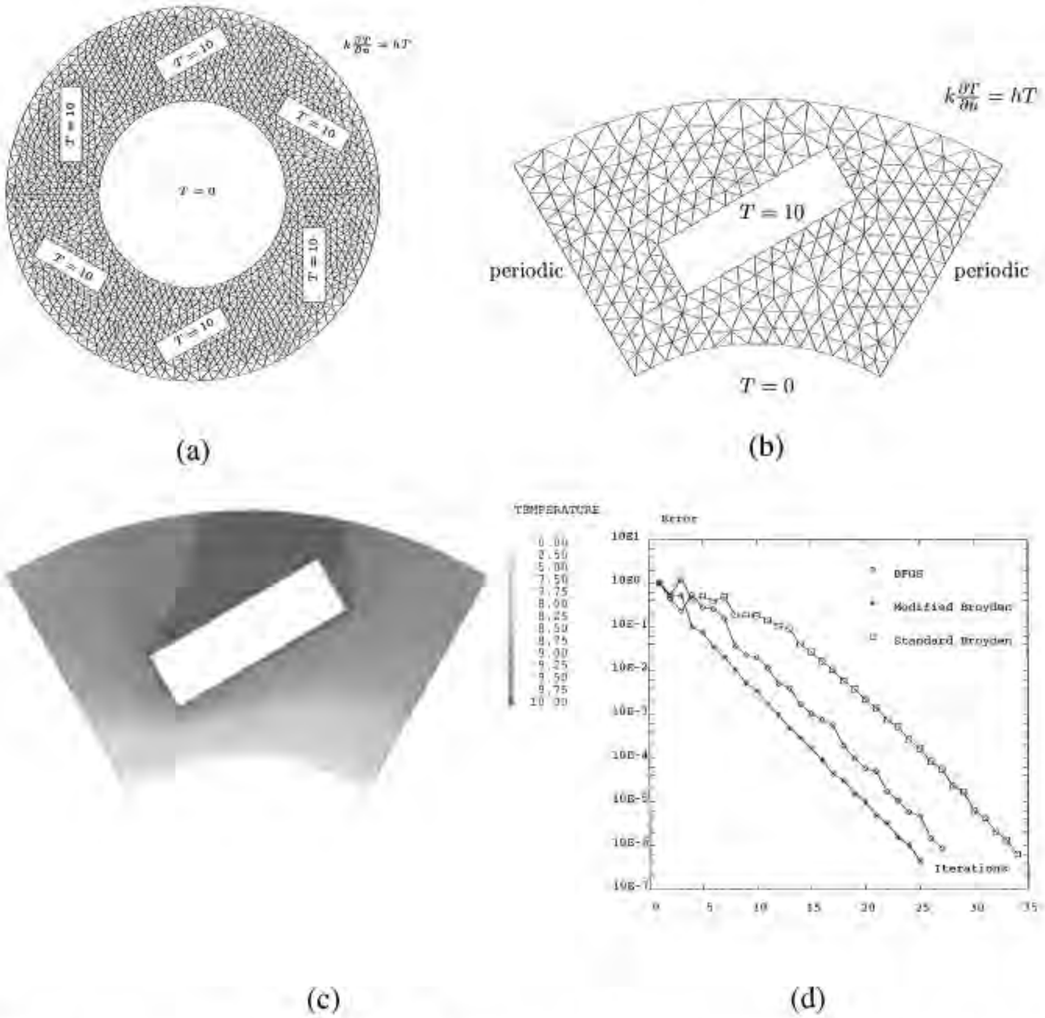


Figure 2. Test THERMAL: (a) problem statement; (b) computational domain; (c) temperature field; and (d) convergence history for Quasi-Newton methods

Table II. Test THERMAL. Computational cost for Quasi–Newton methods

Method	Iterations	CPU time per cent
Modified Broyden	25	100
Standard Broyden	34	168
BFGS	27	150

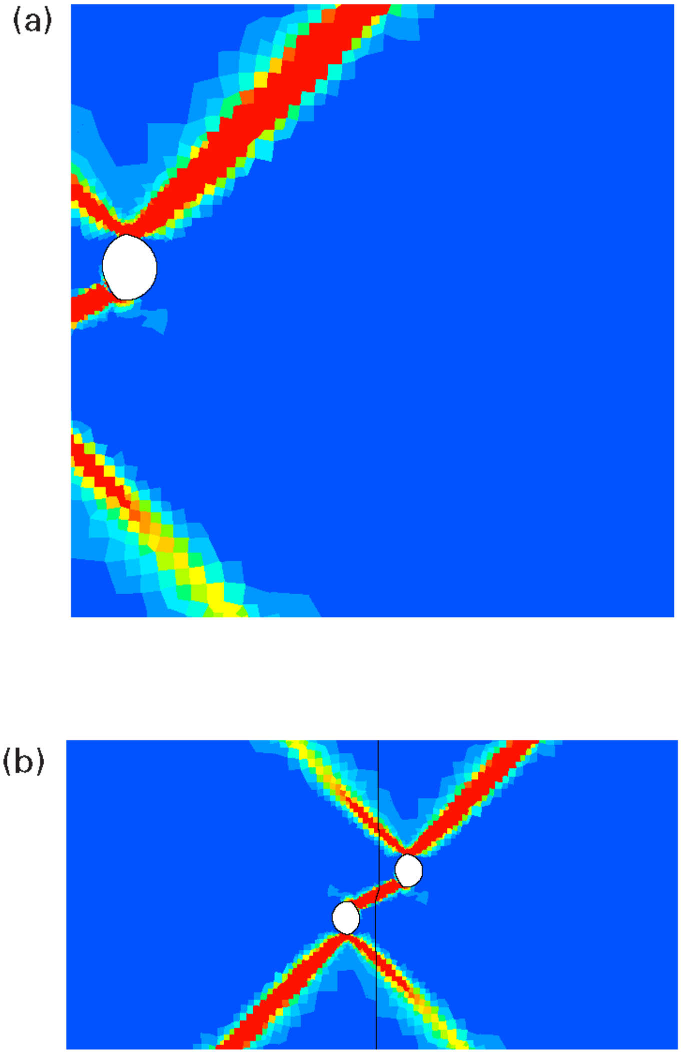


Plate 1. Test TWO HOLES. Deformed shape and equivalent plastic strain for: (a) the computational domain; and (b) the full domain

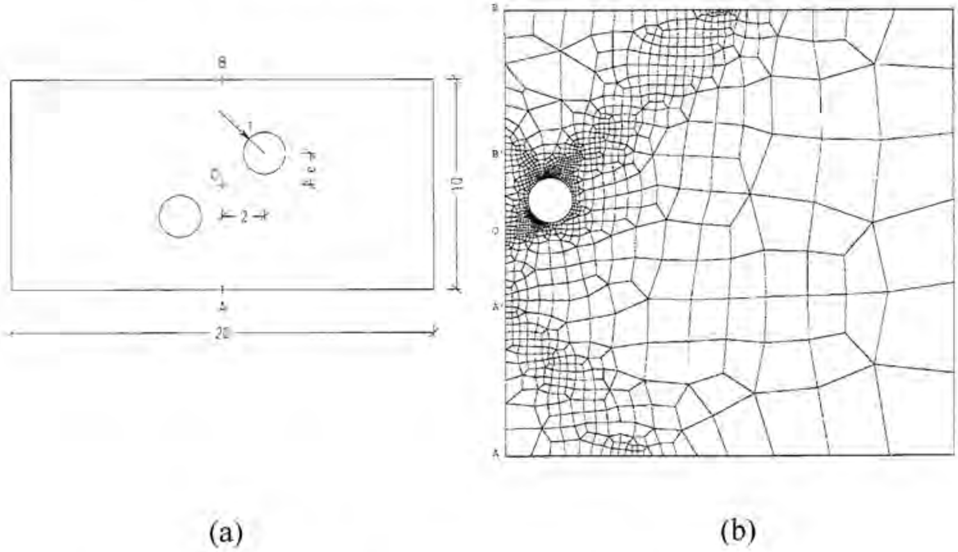


Figure 3. Test TWO HOLES: (a) problem statement; and (b) computational domain

the two holes is altered by a vertical eccentricity e. The domain has central symmetry, but there are no axes of symmetry.

A displacement-controlled, plane strain analysis of the piece, with large deformations and non-linear material behaviour, has been performed. The second-order algorithm discussed in [29] is used to treat large strains. Von Mises elastoplastic behaviour with linear hardening is assumed, with a ratio of plastic modulus to elastic (Young's) modulus of $E_p/E = 10^{-3}$. Consistent material stiffness matrices are employed to compute the tangent stiffness matrices. Different values of the eccentricity e have been tested, to study its effect on the response of the piece [13, 30]. The case $e = 2$ will be shown here.

Thanks to the central symmetry, the problem can be solved by modelling only half the specimen, see Figure 3(b). The finite element mesh has been obtained through a process of adaptive remeshing, which combines an unstructured quadrilateral mesh generator [31, 32], and an a-posteriori error estimator for non-linear finite element analysis [33, 34].

Plate 1(a) shows the equivalent plastic strain over the deformed shape after a 3 per cent reduction in length. The band formation due to strain concentration is clearly visible. The effect of the central-symmetry boundary conditions is also apparent, with a 'mysterious' band appearing from nowhere in the bottom left corner. Of course, this band originates due to the hole in the other half of the specimen. This can be seen in Plate 1(b), where the plastic strain is depicted over the whole piece. It must be remarked that the eccentricity e has a crucial influence on the band pattern, as discussed in Reference 30 in the context of viscoplastic softening materials.

All the non-linear solvers discussed in this paper have been employed for this test. The time step is automatically updated, as suggested in Reference 15, by comparing the number of iterations required in the previous step with a desired number of iterations per step prescribed a priori (5 for the full Newton–Raphson method and 10 for the other methods). The results are shown in

Table III. Test TWO HOLES. Behaviour of various non-linear solvers

Method	Steps	Iterations	Iterations/step	per cent of analysis
Full Newton–Raphson	35	193	5·5	100
Modified Newton–Raphson	Div.	Div.	Div.	25
Initial stress	Div.	Div.	Div.	25
Modified Broyden (QN)	35	389	11·1	100
Standard Broyden (QN)	Div.	Div.	Div.	63
BFGS (QN)	28	303	10·8	100
Modified Secant–Broyden	Div.	Div.	Div.	63
Standard Secant–Broyden	Div.	Div.	Div.	70
Secant–BFGS	Div.	Div.	Div.	64

Table III. Only three methods succeeded in carrying out the complete computation: full Newton–Raphson, BFGS and the modified Broyden. Note that the average number of iterations per step is close to the prescribed value, thus reflecting the good performance of the automatic time-stepping. The modified Newton–Raphson and the initial stress methods diverge as soon as the plastification begins, at 25 per cent of the analysis (a maximum of 100 iterations per step is set). The standard Broyden method, on the other hand, behaves like the Secant–Newton methods, which diverge at 60–70 per cent of the total planned displacement.

6. CONCLUDING REMARKS

The most common non-linear solvers have been adapted to handle linear constraints via the Lagrange-multiplier technique. The adaptation of the Newton–Raphson methods is straightforward: the Jacobian matrix is formed by enlarging the appropriate stiffness matrix with the constraint matrix, which accounts for linear constraints and reaction forces. The adaptation is also automatic for rank-two Quasi–Newton methods (BFGS and DFP), because these methods are naturally well adapted to the partial linearity of the problem associated to Lagrange multipliers.

The situation is quite different for the Broyden method. It has been shown that the standard Broyden method yields secant matrices that do not have the same block-structure of the Jacobian matrix. A modified Broyden method has been presented here, which is by construction well adapted to the Lagrange-multiplier context. This method, which can be regarded as a least-change secant method, has been discussed in detail: derivation, algorithmic implementation and convergence behaviour.

The numerical performance of the various solvers has been assessed by means of some mechanical and thermal non-linear problems. These examples illustrate the superiority of the modified Broyden method over the standard one, both for the Quasi–Newton versions and the Secant–Newton simplifications.

ACKNOWLEDGEMENTS

The authors are grateful to the Commisariat à l'Énergie Atomique (C.E.A.), France, for the availability of the object-oriented code CASTEM 2000. The partial financial support of CYCIT, Spain (grant contract: TAP98-0421) is also acknowledged.

REFERENCES

1. Abel JF, Shephard MS. An algorithm for multipoint constraints in finite element analysis. *International Journal for Numerical Methods in Engineering* 1979; **14**:464–467.
2. Schreyer HL, Parsons DA. Direct application of constraints to symmetric algebraic systems. *Communications in Numerical Methods in Engineering* 1995; **11**:563–573.
3. Webb JP. Imposing linear constraints in finite-element analysis. *Communications in Numerical Methods in Engineering* 1990; **6**:471–475.
4. Barlow J. Constraint relationships in linear and nonlinear finite element analyses. *International Journal for Numerical Methods in Engineering* 1982; **18**:521–533.
5. Shephard MS. Linear multipoint constraints applied via transformation as part of a direct stiffness assembly process. *International Journal for Numerical Methods in Engineering* 1984; **20**:2107–2112.
6. Hughes TJR. *The Finite Element Method.* Prentice-Hall International: Stanford, USA, 1987.
7. Zienkiewicz OC, Taylor RL. *The Finite Element Method,* vols. 1 and 2. McGraw-Hill: London, 1991.
8. Bretones MA, Rodríguez-Ferran A, Huerta A. La programación orientada al objeto aplicada al cálculo por el método de los elementos finitos. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería* 1995; **11**:423–449 (in Spanish).
9. Dubois-Pèlerin Y, Zimmermann Th, Bomme P. Object-oriented finite element programming concepts. In *New Advances in Computational Structural Mechanics,* Ladevèze P, Zienkiewicz OC (eds). Elsevier: Amsterdam, 1992; 457–466.
10. Mackie RI. Object-oriented programming of the FEM. *International Journal for Numerical Methods in Engineering* 1992; **35**:425–436.
11. Miller GR. An object-oriented approach to structural analysis and design. *Computers and Structures* 1991; **40**:75–82.
12. Dubois-Pèlerin Y, Pegon P. Linear constraints in object-oriented finite element programming. *Computer Methods in Applied Mechanics and Engineering* 1998; **154**:31–39.
13. Rodríguez-Ferran A. Arbitrary Lagrangian–Eulerian formulation of quasistatic nonlinear problems. *Doctoral Thesis,* E.T.S. de Ingenieros de Caminos, Universitat Politècnica de Catalunya, Barcelona, 1996.
14. Vila A, Rodríguez-Ferran A, Huerta A. Nonlinear finite element techniques using an object-oriented code. *Monograph No. 31,* International Centre for Numerical Methods in Engineering, Barcelona, 1995.
15. Crisfield MA. *Non-linear Finite Element Analysis of Solids and Structures.* Wiley: England, 1991.
16. Dennis JE, Moré JJ. Quasi–Newton methods, motivation and theory. *SIAM Review* 1977; **19**:46–89.
17. Dennis JE, Schnabel RB. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall Series in Computational Mathematics. Prentice-Hall: New Jersey, USA, 1983.
18. Dennis JE, Walker HF. Convergence theorems for least-change secant update methods. *SIAM Journal on Numerical Analysis* 1981; **18**:949–987.
19. Bathe KJ. *Finite Element Procedures in Engineering Analysis.* Prentice-Hall: NJ, USA, 1982.
20. Matthies H, Strang G. The solution of nonlinear finite element equations. *International Journal for Numerical Methods in Engineering* 1979; **14**:1613–1626.
21. Soria A, Pegon P. Quasi–Newton iterative strategies applied to the heat diffusion equation. *International Journal for Numerical Methods in Engineering,* 1990; **30**:661–677.
22. Fletcher R. *Practical Methods of Optimization.* Wiley: Chichester, England, 1987.
23. Engelman MS, Strang G, Bathe KJ. The application of Quasi–Newton methods in fluid mechanics. *International Journal for Numerical Methods in Engineering* 1981; **17**:707–718.
24. Broyden CG, Dennis JE, Moré JJ. On the local and superlinear convergence of quasi–Newton methods. *Journal of the Institute of Mathematics and its Applications* 1973; **12**:223–245.
25. LAMS. CASTEM2000 – Manuel d'utilisation. *Rapport 88/176,* Laboratoire d'Analyse Mécanique des Structures, Commissariat à l'Énergie Atomique, France, 1988 (in French).
26. Verpeaux P, Millard A. De l'existence à l'essence. De CASTEM à CASTEM2000. Quelques considérations sur le développement de grans codes du calcul. *Rapport 88/179,* Laboratoire d'Analyse Mécanique des Structures, Commissariat à l'Énergie Atomique, France, 1988 (in French).
27. Vila A, Rodríguez-Ferran A, Huerta A. A note on a numerical benchmark test: an axisymmetric shell under ring loads. *Communications in Numerical Methods in Engineering* 1997; **13**:181–192.
28. Zienkiewicz OC, Huang GC. A note on localization phenomena and adaptive finite-element analysis in forming processes. *Communications in Numerical Methods in Engineering* 1990; **6**:71–76.
29. Rodríguez-Ferran A, Pegon P, Huerta A. Two stress update algorithms for large strain solid mechanics: accuracy analysis and numerical implementation. *International Journal for Numerical Methods in Engineering* 1996; **40**:4363–4404.
30. Díez P, Arroyo M, Huerta A. Adaptivity based on error estimation for viscoplastic softening materials. *Mechanics of Cohesive-Frictional Materials,* in press.
31. Sarrate J, Gutiérrez MA, Huerta A. A new algorithm for unstructured quadrilateral mesh generation and rezoning. *Proceedings of the Eighth International Conference on Finite Elements in Fluids,* Barcelona, vol. 1, 1993; 745–754.
32. Sarrate J. Modelización numérica de la interacción fluido–sólido rígido: desarrollo de algoritmos, generación de mallas y adaptabilidad. *Doctoral Thesis,* E.T.S. de Ingenieros de Caminos, Universitat Politècnica de Catalunya, Barcelona, 1996 (in Spanish).

33. Díez P, Egozcue JJ, Huerta A. A posteriori error estimation for standard finite element analysis. *Computer Methods in Applied Mechanics and Engineering* 1998; **163**:141–157.

34. Huerta A, Díez P. Error estimation including pollution assessment for nonlinear finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, in press.