Computers & Structures. Vol 13, pp 115–119. 1981
Printed in Great Britain

0045–7949/81/010115–05$02.00/0

METHODS FOR OPTIMAL ENGINEERING DESIGN PROBLEMS BASED ON GLOBALLY CONVERGENT METHODS

LAYNE WATSON†

Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, U.S.A.

and

W. H. YANG

Department of Applied Mechanics and Engineering Science, The University of Michigan, Ann Arbor, MI 48109, U.S.A.

(*Received* 25 *April* 1980)

Abstract— An optimal design problem is formulated as a system of nonlinear equations rather than the extremum of a functional. Based on the Chow–Yorke algorithm, another globally convergent homotopy method, and quasi-Newton methods, two algorithms are developed for solving the nonlinear system. Although the base algorithms are globally convergent (under certain fairly general assumptions), there is no theoretical proof of global convergence for the new methods. Some low dimensional numerical results are given.

1. INTRODUCTION

Most engineering designs are based on linear theories of physical phenomena. But since the parameters in the design problems are treated as variables, the mathematical formulations of the design problems are nonlinear. The usual formulation of an optimal design problem is to seek the extremum of a functional. If the optimality conditions are stated as differential or algebraic equations, the optimal design problem can be formulated directly as a system of nonlinear equations.

The finite element method is used to approximate the differential equations. The resulting nonlinear algebraic system is the projection of the original optimal design problem into a finite dimensional space. The solution of the algebraic system approximates that of the original problem. The algorithms developed here are based on globally convergent algorithms that have been used successfully in situations where Newton's method for nonlinear systems fails to converge. Examples of this approach are some nonlinear two-point boundary value problems [1], some fluid mechanics problems [2], the nonlinear complementarity problem [3], and the generalized plane stress problem of elasticity [4].

Two algorithms are developed here. One is a homotopy method and the other is a least change secant update (quasi-Newton) method. To illustrate the techniques, in this paper they are applied to a nonlinear algebraic system originating from a generalized plane stress problem of elasticity. This same model problem was solved in [4] by a globally convergent homotopy method. The homotopy map used in [4] was rather complicated, more so in order to be able to prove global convergence than from practical necessity. The homotopy map used here retains some of the essential features of the map in [4], but is much simpler, hence easier to implement. Unfortunately preliminary numerical results indicate that the homotopy method is not globally convergent. Creation of a new homotopy method is justified because both Newton's method and standard continuation diverge (unless the starting point is close to the solution) for this model problem [4]. Quasi-Newton methods are not theoretically globally convergent and in fact are known to fail for the model problem here [4], but a quasi-Newton method with a twist was very successful on the model problem.

The generalized plane stress problem of elasticity is chosen as a model problem of optimal design. The thickness of the sheet is assumed variable. The goal is to find the optimal thickness distribution of a given loading such that the strain energy density is uniform in the sheet.

The problem reduces to a nonlinear algebraic system by the use of the finite element approximation given in the next section.

2. FORMULATION

A generalized plane stress problem of elasticity describes the behavior of an elastic sheet under edge loading conditions. The sheet can be manufactured with an arbitrary thickness distribution. The optimal design problem is to seek a thickness distribution for a given loading such that the strain energy density is constant. This design uses material optimally in the elastic range. If the given load increases proportionally, the elastic limit of the material will be reached simultaneously throughout the sheet.

The problem must satisfy the equations of equilibrium,

$$\frac{\partial}{\partial x}(h\sigma_{xx})+\frac{\partial}{\partial y}(h\sigma_{xy})=0 \qquad (2.1)$$

$$\frac{\partial}{\partial x}(h\sigma_{xy})+\frac{\partial}{\partial y}(h\sigma_{yy})=0$$

†The work of this author was partially supported by NSF Grant MCS 7821337.

where h is the thickness and σ_{xx}, σ_{xy} and σ_{yy} are the components of the stress tensor.

The elastic material properties are described by the generalized Hooke's law

$$\frac{\partial u}{\partial x}-\frac{1}{E}(\sigma_{xx}-\nu\sigma_{yy})=0$$
$$\frac{\partial v}{\partial y}-\frac{1}{E}(\sigma_{yy}-\nu\sigma_{xx})=0 \qquad (2.2)$$
$$\frac{\partial u}{\partial y}+\frac{\partial v}{dx}-\frac{2(1+\nu)}{E}\sigma_{xy}=0$$

where u, v are the displacement components in the x, y directions respectively; E is the Young's modulus; and ν is the Poisson's ratio.

There are six unknowns in the system of eqns (2.1)–(2.2) in terms of the thickness, stress and displacement components.

If the thickness is regarded as a known parameter, the problem can be interpreted as an operator equation,

$$\begin{matrix} 0 & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial x} & 0 & \frac{-1}{Eh} & 0 & \frac{\nu}{Eh} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 & \frac{-2(1+\nu)}{Eh} & 0 \\ 0 & \frac{\partial}{\partial y} & \frac{\nu}{Eh} & 0 & \frac{-1}{Eh} \end{matrix} \quad \begin{matrix} u \\ v \\ q_1 \\ q_2 \\ q_3 \end{matrix} = 0 \qquad (2.3)$$

where

$$q_1=h\sigma_{xx},\ q_2=h\sigma_{xy},\ q_3=h\sigma_{yy} \qquad (2.4)$$

are the edge stress resultants. The differential matrix operator is a function of the thickness. A finite element method given by Jespersen [8] may reduce (2.3) to an algebraic system,

$$K(\mathbf{h})\mathbf{u}=\mathbf{f} \qquad (2.5)$$

where K is an n by n positive definite matrix called the stiffness matrix, $\mathbf{h}$ is the vector of thicknesses of the elements, $\mathbf{u}$ is the nodel displacement vector and $\mathbf{f}$ is the load vector. If $\mathbf{h}$ is known, (2.5) may be solved uniquely.

We shall assume the strain energy density

$$(\sigma_{xx}-\sigma_{yy})^2+\sigma_{xy}^2=U_0 \qquad (2.6)$$

to be constant, U_0, everywhere. In terms of displacements, (2.6) is a differential equation. By the same finite element scheme, the condition (2.6) for each element has the form

$$\mathbf{u}^t B_i \mathbf{u}=1 \quad i=1,2,\ldots,m \qquad (2.7)$$

where the B_i are n by n positive semidefinite matrices, and the constant U_0 is used for normalization. The total number of elements m is usually smaller than the number of nodes n.

Since the thickness $\mathbf{h}$ is non-negative, let $h_i=t_i^2$. The nonlinear algebraic system is

$$K(\mathbf{t})\mathbf{u}=\mathbf{f} \qquad (2.8)$$
$$\mathbf{u}^t B_i \mathbf{u}=1 \quad i=1,\ldots,m$$

where $\mathbf{t}$ is an m-vector; $\mathbf{u}$ is an n-vector and $\mathbf{f}$ is a given vector ($m<n$). The stiffness matrix has the structure

$$K(\mathbf{t})=\sum_{i=1}^{m} t_i^2 K_i \qquad (2.9)$$

where the K_i are the element stiffness matrices which are positive semidefinite.

3. HOMOTOPY ALGORITHM

The algorithm developed here has the same theoretical basis as the fixed point algorithm in [5] and [6]. The theory is summarized in the following lemmas. See [5] for the proofs and [6] for an elementary exposition. Let E^n denote n-dimensional real Euclidean space.

Lemma 1. Let $\rho: E^m\times[0,1)\times E^n\to E^n$ be a C^2 map such that the Jacobian matrix $D\rho(a,\lambda,x)$ has full rank on $\rho^{-1}(0)=\{(a,\lambda,x)|\rho(a,\lambda,x)=0\}$. Then for almost all $a\in E^m$, the Jacobian matrix of $\rho_a(\lambda,x)=\rho(a,\lambda,x)$ also has full rank on $\rho_a^{-1}(0)=\{(\lambda,x)|\rho_a(\lambda,x)=0\}$.

This is expressed in differential geometry jargon by saying if $\rho(a,\lambda,x)$ is transversal to zero, then for almost all a $\rho_a(\lambda,x)$ is also transversal to zero. "Almost all" means every point except those in a set of Lebesgue measure zero. Alternatively one could say $\rho_a(\lambda,x)$ is transversal to zero with probability one. Lemma 1 is known as a "parameterized Sard's Theorem". Now suppose ρ_a is chosen such that $\rho_a(0,x)=s(x)$ is a simple function with unique zero $x=\alpha$, and $\rho_a(1,x)=f(x)$ is the function for which a zero is desired. The next lemma merely spells out the implications of Lemma 1.

Lemma 2. Under the hypothesis of Lemma 1, for almost all a there exists a zero curve γ of ρ_a emanating from (o, α) along which the Jacobian matrix $D\rho_a(\lambda,x)$ has full rank. γ is a simple C^1 curve, is disjoint from any other zeros ρ_a might have, and either wanders off to infinity or reaches a zero of $f(x)$ (at $\lambda=1$).

Note that if the zero curve γ is bounded, it *must* reach a zero of $f(x)$. In general terms, the homotopy method is: construct the homotopy map $\rho_a(\lambda,x)$, then track the zero curve γ emanating from $(0,\alpha)$. If γ is bounded, then the algorithm is *globally divergent* with probability one. It turns out that γ is bounded for many important problems [1–6], hence there are globally convergent algorithms for these problems. The homotopy map ρ_a may be simple, as for the Brouwer fixed point problem [6], or quite complicated, as for the optimal design problem [4].

Another observation is that this homotopy algorithm is *not* just continuation or embedding. λ is not an embedding parameter that increases monotonically from 0 to 1, but is a *dependent* variable that can both increase and decrease along γ. Furthermore, the full rank of $D\rho_a$ along γ and the way in which the algorithm is implemented guarantee that there are never any "singular points" along γ. Singular points occur frequently in standard embedding techniques, resulting in their failure.

The nonlinear system under consideration here is (2.8). For comparison, the homotopy map used in [4] will be given. Define

$$\psi: E^n \times (0,1)^m \times [0,1) \times E^m \times E^n \to E^{m+n}$$

by

$\psi(a, b, \lambda, t, u)$

$$\begin{pmatrix} [\lambda K(t)+(1-\lambda)\,\mathrm{diag}(t_1^2,\ldots,t_m^2,1,\ldots,1)]u \\ \qquad -f-(1-\lambda)a \\ u^t[\lambda B_1+(1-\lambda)e_1e_1^t]u-1-(1-\lambda)b_1 \\ \cdot \\ \cdot \\ \cdot \\ u^t[\lambda B_m+(1-\lambda)e_me_m^t]u-1-(1-\lambda)b_m \end{pmatrix}$$

where e_j is an n-vector with 1 in the jth component and zeros elsewhere, diag $(t_1^2,\ldots)$ is an $n\times n$ diagonal matrix with diagonal elements $t_1^2,\ldots$. Now regard a, b, t, u as being complex vectors, so that ψ defines a complex map

$$\bar{\psi}: \mathbf{C}^{n+m} \times [0,1) \times \mathbf{C}^{m+n} \to \mathbf{C}^{m+n}.$$

Next convert every complex vector $v=(v_1, v_2, \ldots)$ to a real vector $(Re\ v_1,\ Im\ v_1,\ Re\ v_2, \ldots)$. This converts $\bar{\psi}$ back to a real map

$$\rho: E^{2n+2m} \times [0,1) \times E^{2m+2n} \to E^{2m+2n},$$

and

$$\rho_d(\lambda, x) = \rho(d, \lambda, x)$$

is the homotopy map actually used. The obtuse definition of ψ and the complexification of ψ make it possible to prove that the zero curve γ of ρ_d emanating from $(0, \alpha)$ reaches a zero of (2.8) at $\lambda=1$.

The proposal here is to use a simpler ψ, but the same complexification process. Let $v=(t, u)$,

$$F(v)=\begin{pmatrix} K(t)u-f \\ u^tB_1u-1 \\ \cdot \\ \cdot \\ \cdot \\ u^tB_mu-1 \end{pmatrix}$$

and take

$$\psi(a, \lambda, v)=\lambda F(v)+(1-\lambda)(v-a).$$

Converting ψ to complex and back to real again as above leads to the homotopy map

$$\rho_d(\lambda, x)=\lambda G(x)+(1-\lambda)(x-d), \qquad (3.1)$$

where $x \in E^{2m+2n}$. Lemmas 1 and 2 apply to this ρ_d, hence the proposed homotopy method is to track the zero curve γ of ρ_d emanating from $(0, d)$. There is no proof that γ reaches $\lambda=1$ but it is possible to prove that γ cannot turn back toward $\lambda=0$.

The details of tracking γ are in [6], so that aspect will only be sketched here. Parameterize γ by arc length so $\lambda=\lambda(s)$, $x=x(s)$ along γ, and

$$\rho_d(\lambda(s), x(s))=\lambda(s)G(x(s))+(1-\lambda(s))(x(s)-d)=0.$$

Then

$$\frac{d}{ds}\rho_d(\lambda(s), x(s))=D\rho_d(\lambda, x)\begin{pmatrix} d\lambda/ds \\ dx/ds \end{pmatrix}=0, \qquad (3.2)$$

and since the parameter is arc length,

$$\left\|\left(\frac{d\lambda}{ds}, \frac{dx}{ds}\right)\right\|_2=1. \qquad (3.3)$$

Thus γ is the trajectory of the initial value problem (3.2)–(3.3) with initial conditions

$$\lambda(0)=0, \quad x(0)=d. \qquad (3.4)$$

Note that (3.2) does not explicitly specify the derivative $(d\lambda/ds,\ dx/ds)$, which is required by any ODE subroutine. However, the full rank of $D\rho_d(\lambda(s), x(s))$, the condition (3.3), and the continuity of $(d\lambda/ds,\ dx/ds)$ along γ permit the unique determination of the derivative. The details of the numerical calculation of $(d\lambda/ds,\ dx/ds)$ are in [6] for dense $D\rho_d$ and in [4] for sparse $D\rho_d$.

The initial value problem (3.2)–(3.4) is most efficiently solved by a variable step, variable order Adams algorithm as in [7], for example. Since the ultimate goal is to solve $G(x)=0$ and not to track γ, some special strategies are called for. These strategies, based on computational experience, are discussed in [6] and [1]. Since λ is a dependent variable and the ODE solver is taking discrete steps, it is unlikely that λ will hit 1 exactly. As soon as $\lambda(s)>1$, inverse interpolation with previous points saved by the ODE solver yields an $\bar{s}$ such that $\lambda(\bar{s})=1$. The corresponding $x(\bar{s})$ is a zero of $G(x)$. Note that no extra derivative evaluations or steps by the ODE solver are required for the inverse interpolation.

4. QUASI-NEWTON ALGORITHM

Define $v=(t, u)$ and

$$F(v)=\begin{pmatrix} K(t)u-f \\ u^tB_1u-1 \\ \cdot \\ \cdot \\ \cdot \\ u^tB_mu-1 \end{pmatrix}$$

as in Section 3. The most modern quasi-Newton methods, known as least change secant update methods, are based on solving

$$F(v)=0$$

by minimizing

$$\|F(v)\|_2.$$

When intelligently programmed, they are in practice usually globally convergent because they guarantee a decrease in $\|F(v)\|$ at each move [10]. This feature is necessary for robustness, but it results in their failure on (2.8), because $\|F(v)\|$ has local minima at which $F(v)\neq 0$.

A least change secant update method applied directly to $F(v)$ will generally fail (unless the starting point is sufficiently close to the solution). The proposal here is to apply the least change secant update method in [12] to the *complexification*

$$G(x)$$

of $F(v)$, where G is the same as in Section 3. At least for the model problems tried here, this trick worked very well. $G(x)$ does not have the local minima difficulties afflicting $F(v)$. The drawback is that the dimension of the problem doubles, but clever programming can partially overcome this.

Least change secant update methods have the form

$$x^{(k+1)}=x^{(k)}-\gamma H_k G(x^{(k)}),$$

where H_k is an approximation to the inverse of the Jacobian matrix $DG(x^{(k)})$, and γ is chosen to guarantee certain desirable properties of the sequence $\{x^{(k)}\}$. H_k is updated at each iteration in $O((n+m)^2)$ operations, compared to $O((n+m)^3)$ for Newton's method. The precise details are complicated, and can be found in [10]. The efficiency per iteration, ultimate superlinear convergence, and robustness make least change secant update methods very attractive.

5. NUMERICAL RESULTS

To illustrate the dependence on m and n, the problem (2.8) was solved for various m and n using the data described below. Let $n=km$, and K_i be an $n\times n$ block diagonal matrix with $k\times k$ diagonal blocks. All the diagonal blocks are zero except the ith, which is

$$B=\begin{pmatrix} 2 & -1 & 0 & & & \\ -1 & 2 & -1 & & \bigcirc & \\ 0 & -1 & 2 & & & \\ & & & \ddots & & \\ & \bigcirc & & & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}$$

Thus

$$K_i=\begin{pmatrix} 0 & & & & & \\ & \ddots & & & \bigcirc & \\ & & 0 & & & \\ & & & B & & \\ & & & & 0 & \\ & \bigcirc & & & & \ddots \\ & & & & & \dots 0 \end{pmatrix} \qquad (5.1)$$

B_i is an $n\times n$ block diagonal matrix with the same block structure as K_i, and has the form (4.1) with $B=I$. f is an n-vector with the jth component being $0.01j$.

Table 1 shows the results. The execution times are in seconds on an IBM 370/158. The stopping criterion was $\|G(x)\|_1\leq 10^{-4}$, and the starting points were chosen arbitrarily, but always at least a distance 1 from the solution. Failure of the homotopy method to converge is indicated in Table 1 by a dash. The number of Jacobian evaluations is reported in parentheses after the CPU times. The computer code used for the homotopy method was the fixed point code in [11], which also produced the numerical results in [1] and [6]. The quasi-Newton code used was the FORTRAN subroutine HYBRJ, part of the MINPACK package being developed at Argonne National Laboratory [12].

The CPU times in Table 1 are from codes which assume that the Jacobian matrix is dense, and use direct (elimination) methods to compute the kernel of the Jacobian matrix $D\rho_a(\lambda, x)$ (for the homotopy algorithm) or factor the Jacobian matrix $DG(x)$ (for the quasi-Newton method). Typically the Jacobian matrix in optimal design problems is very sparse, and a production code would use iterative sparse matrix techniques to find the kernel of $D\rho_a$ or factor DG. An important aspect of Table 1 is the number of Jacobian evaluations, which would be the same independent of how the matrix calculations are done. Note that the number of Jacobian evaluations is relatively insensitive to the dimensions m and n for the homotopy method, which is typical behavior [6]. Since the whole point of quasi-Newton methods is to avoid explicitly calculating the Jacobian, the quasi-Newton CPU times are the important figures.

Large scale computational results for a realistic generalized plane stress problem in elasticity will be

Table 1.

			CPU time	
m	n	total dimension	homotopy	quasi-Newton
2	4	12	8.8 (95)	.4 (1)
2	6	16	----	.6 (1)
3	6	18	----	.9 (1)
4	8	24	98.0 (109)	1.8 (1)
3	9	24	80.1 (86)	2.0 (1)
2	10	24	60.0 (70)	3.0 (2)
5	10	30	----	3.6 (1)
4	12	32	----	8.3 (2)
3	15	36	----	20.8 (4)
5	15	40	----	18.9 (2)
4	20	48	----	61.1 (4)
10	20	60	----	41.0 (1)
5	25	60	----	131.1 (4)
6	30	72	----	262.6 (4)
10	30	80	----	310.4 (3)
15	30	90	----	183.5 (1)

reported in a future paper.

In conclusion, recall that Newton's method, quasi-Newton methods and standard continuation fail when applied directly to (2.8). A complicated nonlinear homotopy based on the Chow–Yorke algorithm was developed in [4], and proven globally convergent for (2.8). The existence of a globally convergent homotopy algorithm for (2.8) motivated the algorithm of Section 3. Unfortunately the simple homotopy algorithm of Section 3 is not always globally convergent, which suggests that the intricacies of the homotopy map in [4] may be necessary. Hence there is no completely satisfactory homotopy algorithm for optimal design problems of the form (2.8) yet. The Section 4 algorithm is perhaps obvious, but it is interesting that it works. At present the best least change secant update methods destroy sparsity (H_k is dense even though $DG(x)$ may be very sparse), and thus the quasi-Newton approach is (at present) infeasible for large $m+n$. There are sparse matrix techniques for the quasi-Newton updating and factoring of a new quasi-Newton method (which retains sparsity and superlinear convergence) [13–16], but the global behavior and ultimate convergence rate of this new method are untested on real problems. Sparsity is maintained by sacrificing other desirable features of the quasi-Newton update (such as symmetry or positive definiteness), and a satisfactory compromise remains to be found. Note that the kernel of a homotopy Jacobian can be computed by sparse matrix algorithms.

There is no simple, globally convergent, feasible algorithm for large dimensional problems like (2.8). The advantages of both homotopy and least change secant update methods are too great to rule either approach out, and both should be pursued with regard to optimal design problems.

REFERENCES

1. L. T. Watson, An algorithm that is globally convergent with probability one for a class of nonlinear two-point boundary value problems. *SIAM J. Numer. Anal.* **16**, 394–401 (1979).
2. L. T. Watson, T.-Y. Li and C.-Y. Wang, Fluid dynamics of the elliptic porous slider. *J. Appl. Mech.* **45**, 435–436 (1978).
3. L. T. Watson, Solving the nonlinear complementarity problem by a homotopy method. *SIAM J. Control Optimization* **17**, 36–46 (1979).
4. L. T. Watson and W. H. Yang, Optimal design by a homotopy method. *Applicable Anal.*, **10**, 275–284 (1980).
5. S. N. Chow, J. Mallet-Paret and J. A. Yorke, Finding zeros of maps: homotopy methods that are constructive with probability one. *Math. Comput.* **32**, 887–899 (1978).
6. L. T. Watson, A globally convergent algorithm for computing fixed points of C^2 maps. *Appl. Math. Comput.* **5**, 297–311 (1979).
7. L. F. Shampine and M. K. Gordon, *Computer Solution of Ordinary differential Equations: The Initial Value Problem*. W. H. Freeman, San Francisco (1975).
8. D. Jespersen, A least squares decomposition method for elliptic equations. Ph.D. thesis, Department of Mathematics, Univ. of Michigan, Ann Arbor, Michigan (1976).
9. J. E. Dennis, Jr. and J. J. More, Quasi-Newton methods, motivation and theory. *SIAM Rev.* **19**, 46–89 (1977).
10. J. E. Dennis, Jr. and A. Schnabel, Quasi-Newton methods for unconstrained nonlinear problems. *Book Manuscript* (1979).
11. L. T. Watson and D. Fenner, Chow-Yorke algorithm for fixed points or zeros of C^2 maps. *ACM Trans. Math. Software* **6**, 252–260 (1980).
12. J. J. More, MINPACK documentation, Applied Math. Div., Argonne National Lab Argonne, Illinois, 1979.
13. Ph. L. Toint, On sparse and symmetric matrix updating subject to a linear equation. *Math. Comput.* **31**, 954–961. (1977).
14. Ph. L. Toint, Some numerical results using a sparse matrix updating formula in unconstrained optimization. *Math. Comput.* **32** 839–851 (1978).
15. Ph. L. Toint, On the superlinear convergence of an algorithm for solving a sparse minimization problem. *SIAM J. Numer. Anal.* **16**, 1036–1045 (1979).
16. M. J. D. Powell and Ph. L. Toint, On the estimation of sparse Hessian matrices. *SIAM J. Numer. Anal.* **16**, 1060–1074 (1979).