



PERGAMON

International Journal of Solids and Structures 39 (2002) 689–698

INTERNATIONAL JOURNAL OF
SOLIDS and
STRUCTURES

www.elsevier.com/locate/ijsolstr

A comparison of three algorithms for tracing nonlinear equilibrium paths of structural systems

S.A. Ragon ^a, Z. Gürdal ^{a,*}, L.T. Watson ^b

^a Department of Engineering Science and Mechanics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060-0219, USA

^b Departments of Computer Science and Mathematics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

Received 1 June 2000; received in revised form 6 August 2001

Abstract

The relative efficiencies of the Riks/Wempner, Crisfield, and normal flow solution algorithms for tracking nonlinear equilibrium paths of structural systems are compared. It is argued that the normal flow algorithm may be both more computationally efficient and more robust compared to the other two algorithms when tracing the path through severe nonlinearities such as those associated with structural collapse. This is demonstrated qualitatively by comparing the relative behaviors of each algorithm in the vicinity of a severe nonlinearity. Quantitative results are presented for the collapse of a blade stiffened panel. © 2002 Published by Elsevier Science Ltd.

Keywords: Curve tracking; Nonlinear equilibrium path; Normal flow; Probability-one homotopy algorithm

1. Introduction

For many structural systems, it is necessary to compute the nonlinear relationship between the applied loads and the resulting structural deformations. By tracking this nonlinear equilibrium path, it is possible to identify critical buckling and collapse phenomena and, in general, gain insight into the load carrying capability of the structure. In structural mechanics, the nonlinear equilibrium equations are often solved using the Newton Raphson method in conjunction with an incremental/iterative solution method (Haisler and Stricklin, 1972). In order to trace the equilibrium paths through limit points in the response, the “arc length” methods of Riks/Wempner (Wempner, 1971; Riks, 1979) or Crisfield (Crisfield, 1981) are commonly employed. A closely related method, which is based on a “normal flow” algorithm (Watson et al., 1987; Watson et al., 1997), is less popular despite the fact that it may have advantages in terms of computational efficiency and robustness. It is the purpose of this paper to call greater attention to the normal flow algorithm and its potential advantages for the solution of structural mechanics problems. The normal flow algorithm is well known in numerical analysis, and was discussed in the literature as early as 1981 by

* Corresponding author. Fax: +1-540-231-4574.

E-mail address: zgurdal@vt.edu (Z. Gürdal).

Georg (1981) and others. Fried (1984) made similar observations in the mechanics literature. The first robust numerical implementation, coupled with a sophisticated prediction and step size control scheme, was in the HOMPACK software package (Watson et al., 1987), and most recently in HOMPACK90 (Watson et al., 1997).

In the following, a brief description of the Riks/Wempner, Crisfield, and normal flow solution methods is provided. The qualitative behavior of each algorithm is discussed in the context of both a “well behaved” equilibrium path and a more complicated equilibrium path exhibiting severe nonlinearities. The latter path was devised to demonstrate potential pitfalls of the Riks/Wempner and Crisfield algorithms and to highlight the qualitative advantages of the normal flow algorithm in negotiating severe nonlinearities. Finally, numerical results are presented for the collapse of a blade stiffened panel. The relative efficiencies of each method are compared and the advantages of the normal flow algorithm in this case are illustrated. The implementation of the normal flow algorithm used here is that of HOMPACK90.

2. Solution methods

The discrete set of nonlinear equilibrium equations representing the response of a structural system may be written in the following general form:

$$\mathbf{f}(\lambda, \mathbf{d}) = 0, \quad (1)$$

where $\mathbf{f} \in R^m$ is a nonlinear function of a scalar loading parameter, λ , and the displacement vector, $\mathbf{d} \in R^m$. In an incremental/iterative solution method, these equations are solved in a series of steps or increments, usually starting from the unloaded state ($\lambda = 0$). The n th step begins from a known solution on the equilibrium path, $\mathbf{z}_n^* = (\lambda_n^*, \mathbf{d}_n^*)^T$, and consists of a *prediction* phase and a *correction* phase. In the prediction phase, an estimate for the next point on the equilibrium path, $\mathbf{z}_{n+1}^0 = (\lambda_{n+1}^0, \mathbf{d}_{n+1}^0)^T$, is generated. Beginning at this point, Newton Raphson iterations are employed during the correction phase to find a new point on the equilibrium curve. The Riks/Wempner, Crisfield, and normal flow algorithms discussed here all utilize an approximation to the arc length of the equilibrium path, Δl , to fix the size of each step. The i th iteration executed by each algorithm at step ($n + 1$) can be described using the same general set of equations:

$$[D\mathbf{f}(\mathbf{z}_{n+1}^{i-1})] \Delta \mathbf{z} = -\mathbf{f}(\mathbf{z}_{n+1}^{i-1}), \quad (2)$$

$$\mathbf{c}(\mathbf{z}_{n+1}^i) = 0, \quad (3)$$

where

$$\Delta \mathbf{z} = \mathbf{z}_{n+1}^i - \mathbf{z}_{n+1}^{i-1}, \quad i = 1, 2, \dots \quad (4)$$

and

$$[D\mathbf{f}(\mathbf{z}_{n+1}^{i-1})] = \left[\frac{\partial \mathbf{f}(\mathbf{z}_{n+1}^{i-1})}{\partial \lambda} \frac{\partial \mathbf{f}(\mathbf{z}_{n+1}^{i-1})}{\partial \mathbf{d}} \right] \quad (5)$$

is the $m \times (m + 1)$ Jacobian matrix of the system (1). Eq. (3) is an auxiliary “constraint” equation that is different for each algorithm. Upon convergence of the Newton Raphson iterations, a new solution, \mathbf{z}_{n+1}^* , is obtained and the prediction/correction process is continued for subsequent steps.

It should be noted that the system of equations (2) and (3) are not usually solved simultaneously, as Eq. (3) destroys the symmetric and banded qualities that Eq. (2) usually possess. Instead, a procedure similar to that first proposed by Batoz and Dhatt (1979) is often employed (Ramm, 1981). If the modified Newton Raphson algorithm is used, $[D\mathbf{f}(\mathbf{z}_{n+1}^{i-1})]$ and $\mathbf{f}(\mathbf{z}_{n+1}^{i-1})$ are updated only at the beginning of each step.

2.1. Riks/Wempner algorithm

In the Riks/Wempner algorithm, the auxiliary equation (3) takes the following form:

$$\mathbf{c}(\mathbf{z}_{n+1}^i) = (\bar{\mathbf{z}}_n^*)^T (\mathbf{z}_{n+1}^i - \mathbf{z}_n^*) - \Delta l = 0, \quad (6)$$

where $\bar{\mathbf{z}}_n^*$ is the tangent to the converged solution \mathbf{z}_n^* at step n . This equation defines a hyper-plane that is normal to $\bar{\mathbf{z}}_n^*$ and which is at a “distance” Δl from the previously obtained solution at step n . During the Newton Raphson iterations, the successive iterates are forced to return to the equilibrium path along this hyper-plane. This situation is illustrated in Fig. 1 for a one-dimensional problem. Starting at a converged point on the equilibrium path, point **A**, an initial step is taken in the direction of the tangent vector to point **B**. Successive iterates (illustrated by circular markers in the figure) are then confined to the plane through point **B** and normal to the tangent vector. The Riks/Wempner algorithm will converge to a new point on the equilibrium path at point **C**.

2.2. Crisfield algorithm

In the Crisfield algorithm, the auxiliary equation defines a hypersphere of radius Δl centered on the converged solution at step n :

$$\mathbf{c}(\mathbf{z}_{n+1}^i) = (\mathbf{z}_{n+1}^i - \mathbf{z}_n^*)^T (\mathbf{z}_{n+1}^i - \mathbf{z}_n^*) - \Delta l^2 = 0. \quad (7)$$

Successive iterates are confined to the surface of this hypersphere as they converge to the equilibrium solution. This situation is again illustrated in Fig. 1 for the solution of a one-dimensional problem. From

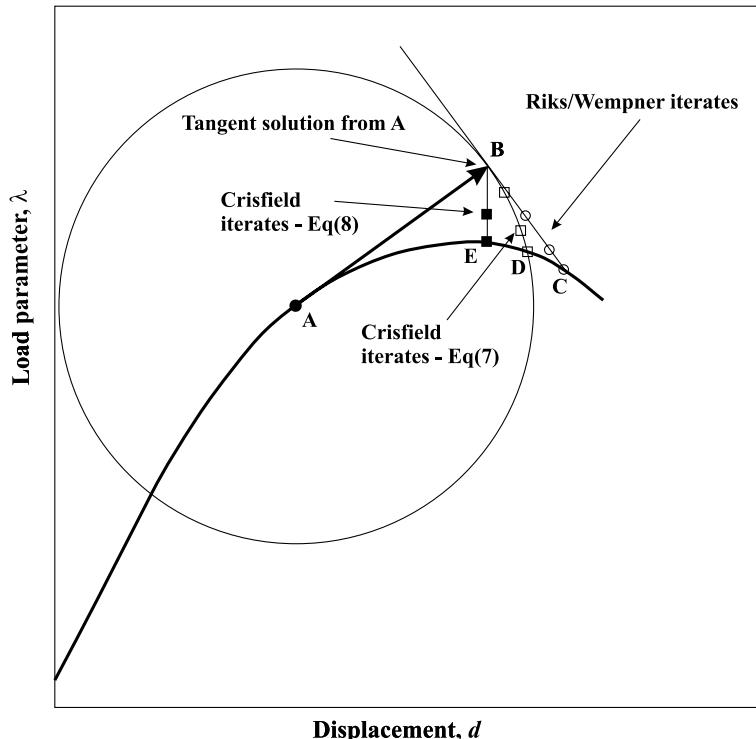


Fig. 1. Riks/Wempner and Crisfield algorithms.

point **B**, the Crisfield iterates (illustrated with open square markers in the figure) are confined to the circular path centered on point **A**. The Crisfield algorithm will converge to a new point on the equilibrium path at point **D**.

In practice, λ is sometimes omitted from Eq. (7), and the constraint is instead imposed in m -dimensional space (Crisfield, 1981):

$$\mathbf{c}(\mathbf{z}_{n+1}^i) = (\mathbf{d}_{n+1}^i - \mathbf{d}_n^*)^T (\mathbf{d}_{n+1}^i - \mathbf{d}_n^*) - \Delta l^2 = 0. \quad (8)$$

This confines the iterates to lie on a cylinder in (λ, d) space parallel to the λ axis. In the figure, these iterates (illustrated with filled square markers) will follow a line parallel to the λ axis and will converge at point **E**.

2.3. Normal flow algorithm

In the normal flow algorithm, successive Newton Raphson iterates converge to the equilibrium solution along a path which is normal (in an asymptotic sense) to the so-called Davidenko flow. The Davidenko flow can be described by considering a small perturbation, δ , to the nonlinear system equations:

$$\mathbf{f}(\lambda, \mathbf{d}) = \delta. \quad (9)$$

As the perturbation parameter varies, small changes will occur in the solution curve for Eq. (9). The family of curves generated by varying δ is known as the Davidenko flow (Allgower and Georg, 1990). The dashed lines in Fig. 2 are a representation of the Davidenko flow for a one-dimensional problem.

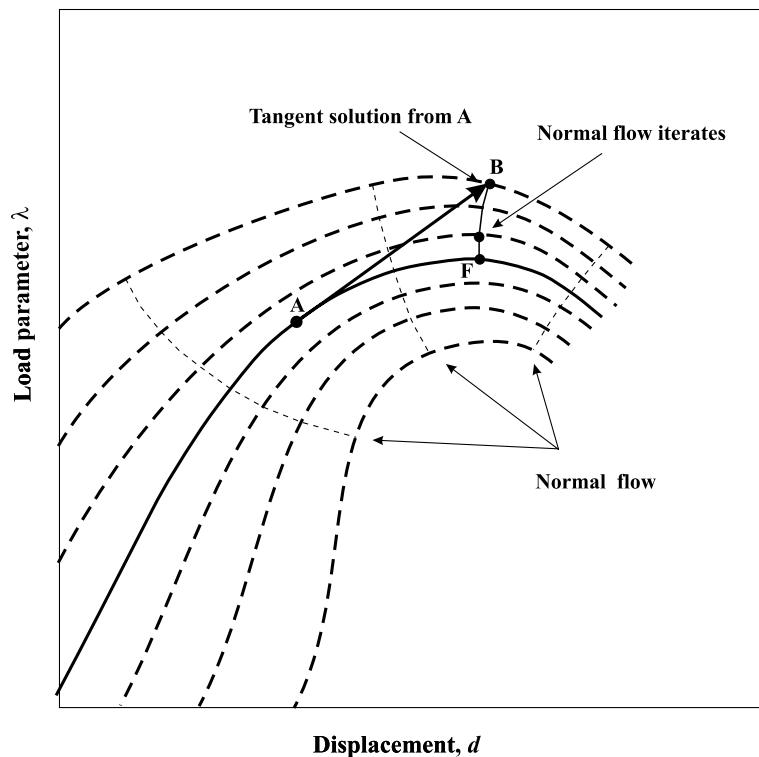


Fig. 2. Normal flow algorithm.

The Newton Raphson iterate $\Delta\mathbf{z}$ for the normal flow algorithm is the unique minimum norm solution of the $m \times (m + 1)$ equation (2). (Note that Eq. (2) has infinitely many solutions, but a *minimum norm* solution is unique.) This solution may be obtained in two steps. First, a *particular* solution \mathbf{v} to the equations can be found by selecting an auxiliary equation (3) and solving the resulting system of equations (2) and (3). As long as this system of equations has rank $m + 1$, the auxiliary equation may be chosen arbitrarily. Once a particular solution is obtained, the minimum norm solution $\Delta\mathbf{z}$ is

$$\Delta\mathbf{z} = \mathbf{v} - \frac{\mathbf{v}^T \mathbf{u}}{\mathbf{u}^T \mathbf{u}} \mathbf{u}, \quad (10)$$

where \mathbf{u} is any vector in the kernel of $[D\mathbf{f}]$. A convenient choice for \mathbf{u} in this case is the tangent vector $\bar{\mathbf{z}}_n^*$, which has usually already been calculated. An illustration of the normal flow algorithm for the solution of a one-dimensional problem is presented in Fig. 2. Starting at a converged point on the equilibrium path, point **A**, an initial step is taken in the direction of the tangent vector to point **B**. Geometrically, the successive iterates (indicated using filled circular markers in the figure) return to the equilibrium solution along a path normal to the Davidenko flow. The normal flow algorithm will converge to a new point on the equilibrium path at point **F**.

3. Qualitative comparison of solution methods

All three of the solution algorithms described in this note are capable of traversing limit points in the response and are generally reliable and efficient as long as the nonlinearities in the load–displacement path are mild. For more severe nonlinearities, however, the Riks/Wempner and Crisfield solution algorithms often fail to converge unless the step size Δl is reduced by a significant amount. From the standpoint of computational efficiency it is desirable to trace the equilibrium curve using as few steps as possible in order to minimize the number of times that the system equations are formed and factored. It is here that the advantages of the normal flow algorithm become apparent. Because the iterates return to the equilibrium curve using a shorter and more direct path, the normal flow algorithm is able to maintain larger step sizes in the vicinity of severe nonlinearities. The advantages of the normal flow algorithm can be understood in a qualitative manner by referring to Fig. 3.

In Fig. 3, the behavior of the Riks/Wempner, Crisfield, and normal flow algorithms in the vicinity of a severe nonlinearity is illustrated for a one-dimensional problem. Starting from a converged solution at point **A**, a large initial step in the tangential direction results in an estimate for the new solution at point **B**. If the Riks/Wempner algorithm is used, successive iterates will proceed along the plane normal to the tangent vector (illustrated with open circular markers in the figure). In this example, this plane will not intersect the solution curve anywhere near point **A** unless the step size is considerably reduced. If the step size is not reduced, the algorithm will either not converge or will converge at a point far from point **A** (perhaps on another equilibrium path). If the Crisfield algorithm (Eq. (7)) is used, iterates will proceed along the circular path as indicated in the figure (open square markers). Unless the step size is reduced, the iterates will likely return to an earlier point on the equilibrium curve, point **D**. Such a failure can be difficult to detect. If the Crisfield algorithm is used with Eq. (8), the iterates will proceed along the line indicated using the filled square markers and may converge near point **E**.

If the normal flow algorithm is used, iterates will return to the equilibrium curve via a path normal to the Davidenko flow and will arrive on the solution curve near point **F**. The normal path will efficiently return the iterates to the solution curve for step sizes much larger than can be tolerated by the Riks/Wempner or Crisfield algorithms. In addition, the normal path is much more direct compared the paths defined by the hyperplane of the Riks/Wempner method or the hypersphere defined by the Crisfield method. This more direct path will usually result in fewer required iterations for convergence at each step.

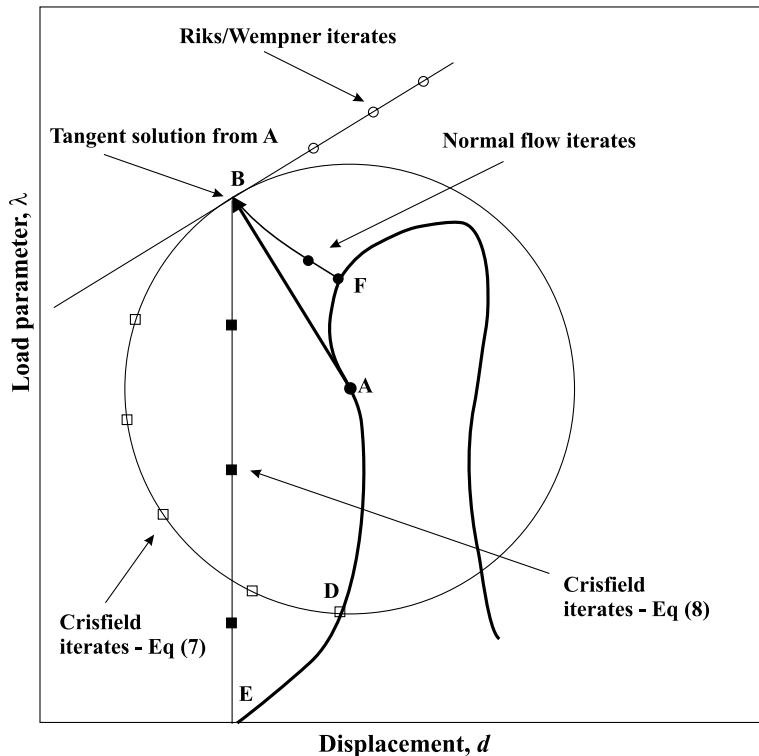


Fig. 3. Normal flow algorithm.

4. Numerical results

In order to quantitatively compare the relative performances of the solution algorithms, a sample problem was solved using each method. This sample problem consisted of computing the response of a compressively loaded graphite-epoxy blade stiffened panel loaded into the postbuckling regime. A “stiffener-unit” representation of the panel was utilized wherein the complete panel was modeled using a single repeating element of the structure comprised of a single stiffener and the adjacent skin (Fig. 4). The stiffener unit was 48.1 in. long, 6.28 in. wide, and had a stiffener height of 1.10 in. The skin layup was $(\pm 45/0_3)_s$, the stiffener layup was $(90/0)_s$, and the ply thickness was 0.006 in. The skin laminate was assumed to have an imperfection in the shape of seven half waves in the longitudinal direction with a maximum imperfection amplitude of 0.0001 in.

Uniform end shortening was imposed at the ends of the structure and symmetry boundary conditions were imposed at the longitudinal edges. At the ends, the skin was simply supported out of plane but the stiffener was not allowed to rotate in its own plane. The panel was analyzed using a finite strip analysis of the semi-analytical, multi-term type (Dawe, 1995; Ragon, 1998). The finite strips were modeled as balanced and symmetric laminated composite materials which were assumed to behave orthotropically in bending. Material properties were as follows: $E_1 = 20.0 \times 10^6$ lb/in.², $E_2 = 1.30 \times 10^6$ lb/in.², $G_{12} = 1.03 \times 10^6$ lb/in.², and $\nu = 0.30$.

All three of the solution algorithms were implemented using the full Newton Raphson method. The prediction phase of the Riks/Wempner and Crisfield algorithms (including the determination of the arc

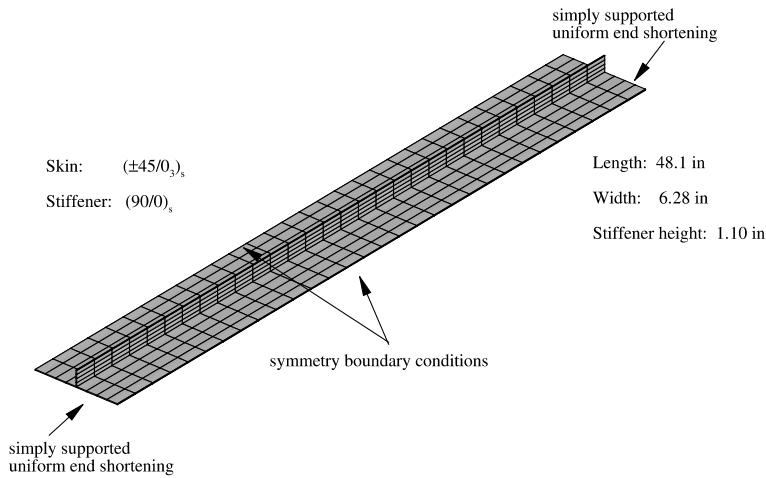


Fig. 4. Skin-stiffener unit.

length step size at each increment) were implemented as described in Crisfield (1981) under the heading “A Modified Riks Method”. This prediction scheme is described briefly described here.

The first step is the selection of an appropriate value for the arc length, Δl . The arc length is adjusted from one step to the next using the following simple formula:

$$\Delta l_{n+1} = \Delta l_n \frac{\bar{m}}{m}, \quad (11)$$

where m is the number of iterations that were required at the previous step and \bar{m} is the (user specified) desired number of iterations at each step. This procedure allows larger steps to be taken when the solution is converging easily, and forces the solver to take smaller steps when convergence is more difficult. For the present work, \bar{m} was selected between 3 and 5 so as to achieve the most favorable results. Once a value for the arc length has been computed at step $n + 1$, the following equation is used to predict $\Delta \lambda_{n+1}^0$:

$$\Delta \lambda_{n+1}^0 = \pm \Delta l [(\Delta d_{n+1}^0)^T \Delta d_{n+1}^0], \quad (12)$$

where Δd_{n+1}^0 is taken to be the tangential solution from the previously converged point.

The prediction strategy used for the normal flow algorithm was that implemented in subroutine STEPNX in HOMPACK90. This strategy utilizes a Hermite cubic interpolating polynomial through previously converged points and a sophisticated step size estimation algorithm. This step size control algorithm, with proper choice of the error control parameters, will automatically adapt to external scaling of the problem.

Performance differences between the three methods can be attributed to differences in (1) the prediction formula, (2) the step size control algorithm, and (3) the iteration trajectory in returning to the equilibrium path (hyperplane vs. hypersphere vs. normal curve). These three items will interact with each other and with the specific problem in complicated ways. Such interactions have been studied in detail (Watson et al., 1987; Watson et al., 1997), and it is the intent here to compare intelligently implemented versions of the three algorithms rather than to sort out the exact causes of the differences. (For instance, a version of the Riks/Wempner algorithm that uses the tangent direction to the Davidenko curve through point **B** in Fig. 3, rather than the tangent direction at point **A**, is much more robust.)

The load/end-shortening behavior of the structure as obtained using the normal flow algorithm is illustrated in Fig. 5, where N_x is plotted against a scalar loading parameter, λ . A value of $\lambda = 1.0$ corresponds

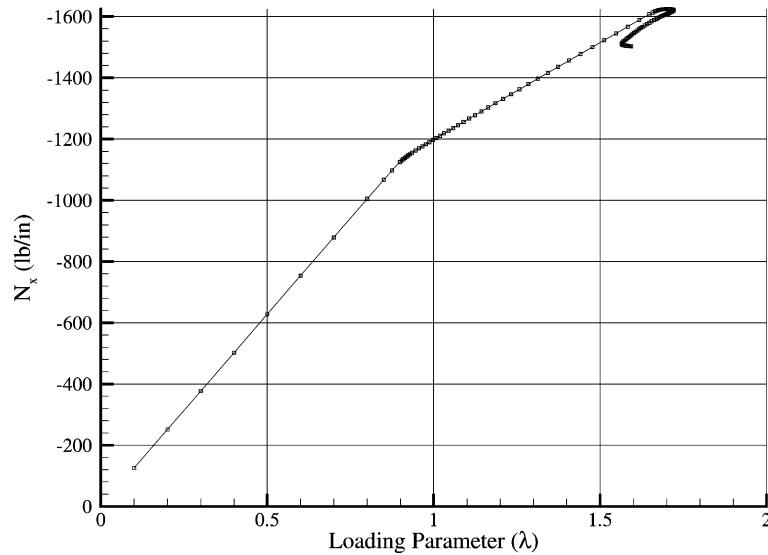


Fig. 5. N_x vs. loading parameter (normal flow algorithm).

to a uniform axial strain $\varepsilon_x = -2.34 \times 10^{-4}$ and a uniform transverse strain in the skin of $\varepsilon_y = +1.40 \times 10^{-4}$. The scaling for λ was chosen so that local buckling of the skin occurred at approximately $\lambda = 1.0$. The response is linear up until $N_x \approx 1130$ lb, where the skin buckles into seven half waves along the length of the stiffener unit. Beyond this point, the panel continues to carry load up to a maximum of $N_x \approx 1620$ lb, where the stiffener unit as a whole deforms into a single half wave and collapses (the seven half-wave buckled pattern of the skin is superimposed on the single half wave of the structure). The normal flow algorithm was able to continue tracking the equilibrium path through the collapse point with no difficulty; the algorithm was eventually terminated at the user's request at the point shown (the algorithm did not terminate because of convergence difficulties).

The Riks/Wempner and Crisfield algorithms computed the same load/end-shortening curve as did the normal flow algorithm, but they both encountered difficulties in tracing the complete curve. In Fig. 6, a zoomed in portion of the load/end-shortening curve in the vicinity of the collapse point is illustrated and the solutions obtained using the Riks/Wempner algorithm and the normal flow algorithm ($\bar{m} = 4$) are compared. The normal flow algorithm was able to traverse this portion of the curve using a relatively small number of steps and continue on with the remainder of the curve. In contrast, the Riks/Wempner algorithm was able to make forward progress only by taking significantly smaller steps. Furthermore, the Riks/Wempner algorithm was unable to continue tracing the curve past a certain point; at this point the algorithm began chattering back and forth over already computed portions of the curve, perhaps due to the step size control rules. A Crisfield solution obtained using Eq. (8) ($\bar{m} = 4$) and the normal flow solution on the same portion of the curve are compared in Fig. 7. The Crisfield algorithm is able to compute more of the curve than the Riks/Wempner algorithm, but it also failed to make forward progress after reaching a certain point on the curve. As with the Riks/Wempner algorithm, it is forced to take significantly smaller (and more numerous) steps as compared to the normal flow algorithm in order to obtain converged solutions. Eq. (7) was also used to obtain solutions to the problem, but this algorithm performed no better than the Eq. (8) algorithm.

Using the normal flow algorithm, only one or two iterations were required to obtain a converged solution at each step, whereas the Riks/Wempner and Crisfield algorithms typically required from three to five

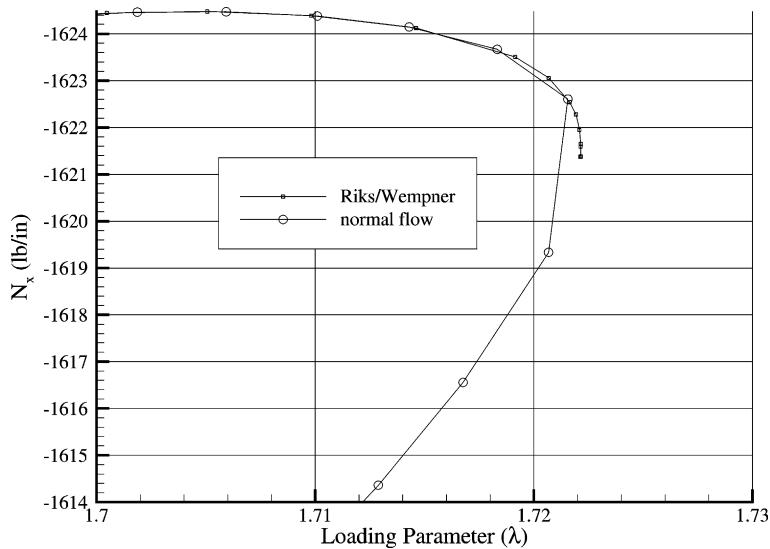


Fig. 6. Comparison of Riks/Wempner and normal flow algorithms.

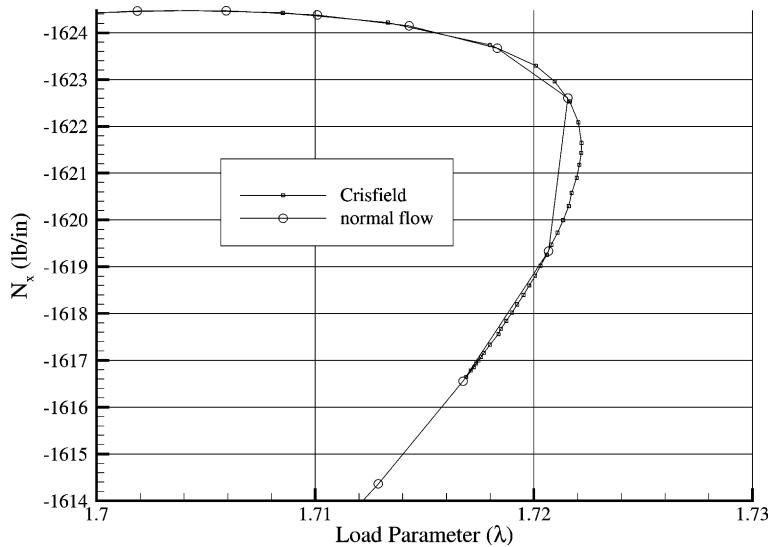


Fig. 7. Comparison of Crisfield and normal flow algorithms.

iterations at each step. Depending on the value of \bar{m} , the Riks/Wempner and Crisfield algorithms each required from 250 to 400 matrix formations/factorizations in tracing the load/end-shortening curve up to the points where the algorithms failed; the normal flow algorithm reached the same portion of the curve with only 78 formations/factorizations.

The increased computational efficiency that can be achieved using normal flow algorithm can be particularly advantageous in an optimization environment, where the computational efficiency of a single analysis is of utmost importance. If the equilibrium path must be traced with a higher degree of fidelity

(compared to that shown in Figs. 6 and 7), the normal flow algorithm can be forced to take smaller steps along the path. This will, of course, result in a degradation of computational efficiency.

5. Conclusions

The normal flow solution algorithm (as implemented in HOMPACK90 (Watson et al., 1987; Watson et al., 1997)) was found to be both more efficient and more robust than the Riks/Wempner and Crisfield solution algorithms (implemented according to (Crisfield, 1981)) for computing complicated nonlinear responses of structural systems. Advantages in efficiency derive from the fact that the normal flow algorithm is able to traverse sharp nonlinearities the load–displacement path using relatively larger step sizes and because successive iterates tend to converge more quickly to the equilibrium curve. The normal flow algorithm is evidently more robust than the competing algorithms because of the more direct path that successive iterates follow in returning to the equilibrium curve. Efficiency and robustness advantages also accrue from superior prediction and step size algorithms, but these do not discount the inherent value of the normal flow iteration.

Acknowledgements

This research was supported by the Air Force Office of Scientific Research (F49620-99-1-0128) and the NASA-Langley Research Center (NAG-1-2116). The support of the grant monitors Drs. Neal D. Glassman and James H. Starnes, Jr., respectively, is gratefully acknowledged.

References

- Allgower, E.L., Georg, K., 1990. Numerical Continuation Methods. Springer, Berlin.
- Batoz, J.L., Dhatt, G., 1979. Incremental displacement algorithms for nonlinear problems. *International Journal of Numerical Methods in Engineering* 14, 1262–1267.
- Crisfield, M.A., 1981. A fast incremental/iterative solution procedure that handles snap-through. *Computers and Structures* 13, 55–62.
- Dawe, D.J., 1995. Finite strip buckling and postbuckling analysis. In: Turvey, G.J., Marshall, I.H. (Eds.), *Buckling and Postbuckling of Composite Plates*. Chapman and Hall, London, pp. 108–153.
- Fried, I., 1984. Orthogonal trajectory accession to the nonlinear equilibrium curve. *Computer Methods in Applied Mechanics and Engineering* 47, 283–298.
- Georg, K., 1981. Numerical integration of the Davidenko equation. In: Allgower, E., Glashoff, K., Peitgen, H.-O. (Eds.), *Numerical Solution of Nonlinear Equations*. Lecture Notes in Math. Springer, Berlin, p. 878.
- Haisler, W.E., Stricklin, J.A., 1972. Development and evaluation of solution procedures for geometrically nonlinear analysis. *AIAA Journal* 10 (3) pp. 164–172.
- Ragon, S.A., 1998. Development of a Global/Local Approach and a Geometrically Nonlinear Local Panel Analysis for Structural Design. Center for Composite Materials and Structures Report No. CCMS-98-10, Blacksburg, Virginia.
- Ramm, E., 1981. Strategies for Tracing Nonlinear Response Near Limit Points. *Nonlinear Finite Element Analysis in Structural Mechanics: Proceedings of the Europe-US Workshop*, July 28–31, 1980. Springer, Berlin, pp. 63–89.
- Riks, E., 1979. An incremental approach to the solution of snapping and buckling problems. *International Journal of Solids and Structures* 15, 524–551.
- Watson, L.T., Billups, S.C., Morgan, A.P., 1987. Algorithm 652: HOMPACK: A Suite of Codes for Globally Convergent Homotopy Algorithms. *ACM Transactions on Mathematical Software*, vol. 13(3), September 1987, pp. 281–310.
- Watson, L.T., Sosonkina, M., Melville, R.C., Morgan, A.P., Walker, H.F., 1997. Algorithm 777: HOMPACK90: A Suite of Fortran 90 Codes for Globally Convergent Homotopy Algorithms. *ACM Transactions on Mathematical Software*, vol. 23(4), pp. 514–549.
- Wempner, G.A., 1971. Discrete approximation related to nonlinear theories of solids. *International Journal of Solids and Structures* 17, 1581–1599.