Shape optimization using a matrix-free Newton method

by

Cheewee P. Chew

B.S.E. Electrical Engineering, University of Pennsylvania, 1995

B.A. Physics, University of Pennsylvania, 1995

B.S. Economics, University of Pennsylvania, 1995

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1997

Author ....

Department of Electrical Engineering and Computer Science

January 15, 1997

Certified by ....

Jacob K. White

Associate Professor, Department of Electrical Engineering and Computer Science

Thesis Supervisor

Accepted by ....

Arthur C. Smith

Chairman, Departmental Committee on Graduate Students

Acknowledgements and a dedication

I would like to thank my thesis advisor, Professor White, for his guidance, support and illuminating teaching in and out of the classroom. I am also grateful to Professor White for making it possible for me to do a significant part of this work at Princeton University. In addition, I would also like to thank Professor Alan Edelman for his numerical methods classes which helped me acclimatize to the field in a short period of time.

But, most of all, I would like thank my significant other, Lynn Loo, to whom this thesis is dedicated, for bringing me so much joy and happiness. Happy birthday!

Contents

1 Background **5**

2 The Newton Algorithm **6**

2.1 The physical model 6

2.2 The algorithm 7

2.3 Accelerating gradient approximations: The Method of Adjoints 9

2.4 Using an iterative solver: A matrix-free approach 10

2.5 Escaping from Newton stalls: A stochastic perturbative approach 11

3 Results: The importance of Δ **11**

4 Conclusions **14**

List of Figures

1 Transitivity of nodal perturbations 10

2 Initial 16 node structure with displacement vectors 16

3 Converged 16 node structure with displacement vectors (418 iterations, $\Delta = .01$) 16

4 Reduction of D with each iteration of 16 node structure (418 iterations, $\Delta = .01$) 17

5 Reduction of $||J||$ with each iteration of 16 node structure (418 iterations, $\Delta = .01$) 18

6 Converged 16 node structure with displacement vectors(5277 iterations, $\Delta = 5\text{x}10^{-4}$) 19

7 Reduction of D with each iteration of 16 node structure (5277 iterations, $\Delta = 5\text{x}10^{-4}$) 20

8 Reduction of $||J||$ with each iteration of 16 node structure (5277 iterations, $\Delta = 5\text{x}10^{-4}$) 21

Shape optimization using a matrix-free Newton method

by

Cheewee P. Chew

Submitted to the Department of Electrical Engineering and Computer Science on January 17, 1997 in partial fulfillment of the requirements for the degree of Master of Science

Abstract

In tackling the problem of minimizing the deformation of a loaded structure, by varying the shape of the original structure, past second order optimization efforts have focused on general Newton techniques like the Davidon-Fletcher-Powell (DFP) update formula and the Broyden- Fletcher-Goldfarb-Shanno (BFGS) formula which iteratively build estimates of the structure's Hessian. This thesis bypasses the above mentioned need for explicit estimation of the Hessian by exploiting the inherent symmetry of the problem and, thus, is able to use a matrix-free Krylov subspace method like GMRES as the Newton solver. In addition to this computationally efficient solver, the thesis algorithm also uses stochastic perturbations to escape from its Newton stalls. Results will be presented to illustrate the algorithm's convergence.

Thesis supervisor: Jacob K. White

Title: Assoc. Prof., Dept. of E.E.C.S.

1 Background

The task of minimizing the deformation of a particular structure, given a particular load vector, can be viewed as a non-linear optimization problem. Moreover, with the appropriate initial guess of the optimal shape, which can ensure that the iterates do not impinge upon any physical constraint and that the converged structure will have the general shape desired, the optimization problem becomes effectively unconstrained and local. Given these conditions, the shape optimization problem thus becomes amenable to local first and second order methods such as the Conjugate Gradient algorithm and the Newton method respectively.

Due to the Newton method's quadratic convergence property, there have been efforts to use the Newton method to perform unconstrained optimization. The main problems faced by researchers in applying this particular second order method have been the computationally expensive task of estimating the gradients and the Hessian and, especially for higher dimensional problems, the efficient storage and manipulation of the data structures required.

To bypass the difficulty of explicitly estimating the Hessian, Quasi-Newton or Variable-Metric algorithms, which seek to estimate the Hessian by iteratively building better approximations using gradient information, were pursued and had resulted in the Davidon-Fletcher-Powell (DFP) update formula [3] and, its successor, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula [1]. Subsequently, modifications by Toint [10], Shanno [9], Curtis, Powell and Reid [2] and Powell and Toint [8] have enabled this class of algorithms to preserve and exploit any inherent sparsity of the optimization problem.

However, as with any class of numerical algorithms — which had been designed to be robust for general problems — much of the implicit properties of a particular problem, such as symmetry, would be unexploited. Therefore, to achieve better computational efficiency for this particular problem of shape optimization, a customized matrix-free, i.e., the Hessian is never explicitly constructed, Newton method will be implemented instead. The gradients will be estimated using the method of adjoints while the Hessian operator will be constructed, by exploiting the fact that any given two successive nodal perturbations are transitive, from the estimated gradients.

2 The Newton Algorithm

The Newton method was chosen primarily because of its quadratic convergence property, its potential to use computationally efficient matrix-free techniques and the shape optimization the problem's need for only a local minimizer — a global minimizer, if unconstrained, would only collapse the initial shape of the structure to some globally minimal, but ultimately not too useful, shape.[1] As such, the Newton method will begin with a suitable initial configuration from which it will locally optimize the node positions.

2.1 The physical model

Since the main issue of this research is not the fidelity of the physical model used but the numerical issues of implementing a fast Newton solver, it was decided that a linearized struts and joints model would be used in favor of a model based on continuum mechanics. After satisfactory progress has been made in implementing the Newton method, it will only be a matter of switching the structural model to achieve better physical correspondence.

Definitions:

A	Connectivity matrix
f	Strut force vector
m	Nodal load vector
p	Nodal position vector
$\tilde{p}$	Nodal displacement vector
u	Strut displacement vector
$\tilde{u}$	Incremental strut displacement vector

[1] Please see the Results section for an example.

F	Linearized strut force matrix
D	Deformation
J	Deformation Jacobian
H	Deformation Hessian
n	Number of nodes

$$Af = m \tag{1}$$

$$A^T\tilde{p} = \tilde{u} \tag{2}$$

$$f = f(u) \approx F(p)\tilde{u} \tag{3}$$

By putting the linearized (3) into (1), we get

$$S\tilde{p} = m \tag{4}$$

where $S = S(p) = AFA^T$.

Also, the structural deformation, D, and its first and second derivatives, J and H respectively, are defined as follows:

$$D = \tilde{p}^T\tilde{p} \tag{5}$$

$$J = [\frac{\delta D}{\delta p_1}, \frac{\delta D}{\delta p_1}, \ldots, \frac{\delta D}{\delta p_{2n}}]^T \tag{6}$$

$$H = [\frac{\delta J}{\delta p_1}, \frac{\delta J}{\delta p_1}, \ldots, \frac{\delta J}{\delta p_{2n}}] \tag{7}$$

Hence, to minimize D, the algorithm will iteratively search, via an damped Newton method, for some p such that $\|J\| = 0$.

2.2 The algorithm

With the previous section's definitions, the iterative Newton algorithm is specified by the following pseudocode where ϵ_{rel}, ϵ_{abs}, $p_{initconfig}$ and $ztol$ are the relative and absolute tolerances, the initial nodal positions and the zero tolerance respectively.

```
$k = 1$
$p_k = p_{initconfig}$
**Do**
    $H(p_k)\Delta p = -J(p_k)$ [GMRES]
    **Do**
        **If** $(\|J(p_k + \Delta p)\| > \|J(p_k)\|$ **and** $\|\Delta p\| > ztol)$ **then** $\Delta p = \Delta p/2$
        **Else**
            **If** $\|\Delta p\| < ztol$ **then** randomly find a $\Delta p$ where
            $\|J(p_k + \Delta p)\| < \|J(p_k\|)$
        $p_{k+1} = p_k + \Delta p$
    **Until** $(\|J(p_{k+1})\| < \|J(p_k)\|)$
    k++
**Until** $(\frac{\Delta p}{p_k} < \epsilon_{rel}$ and $\|J\| < \epsilon_{abs})$
```

Other than the four tolerances mentioned above, the magnitude of nodal perturbations, Δ, is crucial to the solution of the Newton step. The accuracy of J and H estimates and the magnitude of the Newton step are all determined by Δ. Moreover, Δ also controls the magnitude of the stochastic perturbations. As such, Δ is closely linked to the following sections which will cover the algorithmic issues of estimating J, formulating the matrix-free representation of H and the stochastic perturbative method used. Following that, section 3 will illustrate how Δ determines, whether it is Newton or stochastic, the search technique used by the algorithm, i.e., this hybrid stochastic-Newton algorithm will switch between the two methods depending on the magnitude of the nodal perturbations, Δ.

2.3 Accelerating gradient approximations: The Method of Adjoints

The deformation Jacobian, J, can be estimated by successively perturbing each node and noting the change in D due to each nodal perturbation. However, this naive approach would entail $2n$ matrix inversions. To handle this computational bottleneck, the algorithm uses the method of adjoints which only requires a single matrix inversion and $2n$ dot products for each structure.

By noting that the nodal perturbations are small, i.e. $\|\delta S\| \ll \|S\|$, the resultant small changes in the nodal positions, $\delta\tilde{p}$, can be approximated in the following manner :-

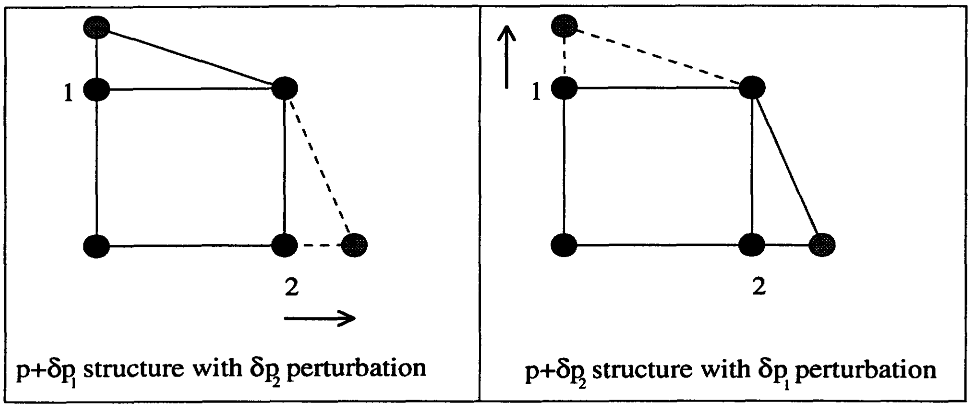
From (4),

$$\begin{aligned}(S+\delta S)(\tilde{p}+\delta\tilde{p}) &= m \\ Sp + S\delta\tilde{p} + \delta S\tilde{p} &\approx m \\ \delta\tilde{p} &\approx -S^{-1}\delta S\tilde{p}\end{aligned}$$

With this approximation for $\delta\tilde{p}$, the variation in D due to the nodal perturbations can be found rapidly for different perturbations — requiring only a single matrix solve for y and the use of the δS operator for each perturbation.

$$\begin{aligned}\delta D &= \delta(\tilde{p}^T\tilde{p}) \\ &= \delta\tilde{p}^T\tilde{p} + \tilde{p}^T\delta\tilde{p} \\ &\approx 2(-S^{-1}\delta S\tilde{p})^T\tilde{p} \\ &\approx -2(\delta S\tilde{p})^T y\end{aligned}$$

where $y = (S^T)^{-1}\tilde{p}$



p+δp$_1$ structure with δp$_2$ perturbation

p+δp$_2$ structure with δp$_1$ perturbation

Figure 1: Transitivity of nodal perturbations

2.4 Using an iterative solver: A matrix-free approach

In line 4 of the pseudocode, Δp can be obtained without explicitly forming H by using a Krylov subspace iterative solver such as GMRES — which only needs an operator that performs the H matrix-vector product. This implicit formation of H from the J vector is possible because of the transitivity of any two sequential nodal perturbations with respect to the final deformation, i.e. perturbing node 1 followed by node 2 gives the same resultant deformation as perturbing node 2 first before node 1. See Figure 1.

$$\frac{\delta D(p+\delta p_j)}{\delta p_i} \approx \frac{\delta D(p+\delta p_i)}{\delta p_j} \tag{8}$$

where $i \neq j$.

By using (8), the approximation of H becomes anti-symmetric:

$$\begin{aligned}\frac{\delta^2 D}{\delta p_j \delta p_i} &\approx \frac{\frac{\delta D(p)}{\delta p_j} - \frac{\delta D(p)}{\delta p_i}}{\Delta} \\ &\approx -\frac{\delta^2 D}{\delta p_i \delta p_j}\end{aligned}$$

where $i \neq j$ and Δ is the magnitude of the p perturbations.

However, since the above approximation will result in zero entries for the diagonals of H, these entries are approximated by the following perturbation:

$$\frac{\delta^2 D}{\delta p_i^2} \approx \frac{\frac{\delta D(p+\delta p)}{\delta p_i} - \frac{\delta D(p)}{\delta p_i}}{\Delta}$$

From the above approximations of the H entries, it is therefore possible to construct an approximate H operator, except for the diagonal H entries, from pair-wise differences of the J entries. The H operator has only to perform $2n$ individual gradient estimations — instead of $n(2n-1)$ gradient estimations — in addition to the simple differences required for the off-diagonal entries.

2.5 Escaping from Newton stalls: A stochastic perturbative approach

While testing the algorithm, without the random perturbation code (lines 8 - 10), it was found that the Newton method stalled frequently in the later iterations, i.e., no $||J||$ reduction was possible in the current Newton direction. To overcome these stalls, [11] suggested many approaches but the random perturbative approach was chosen because of its simplicity and ease of implementation: the algorithm was augmented to randomly select search vectors, $u\Delta$, that do result in $||J||$ reductions where $\{u \in R^{2n} : u_i$ are uniformly distributed between $-\frac{1}{2}$ and $\frac{1}{2}$ for $i = 1, 2, \ldots, 2n\}$. The effect of these random perturbations can be seen in the discontinuous reductions of $||J||$ and D during later iterations (See Figures 4,5,7 and 8).

3 Results: The importance of Δ

After the above mentioned algorithm was implemented successfully, a simple 16 node test structure (See Figure 2) was optimized with different magnitudes for the nodal perturbation parameter, Δ, to investigate the behavior of the hybrid stochastic-Newton algorithm. As will be demonstrated below, higher Δ resulted in poor Jacobian estimates which had caused the Newton component of the algorithm to stall early. After this initial stall, the algorithm effectively switched to a simple stochastic search method. Fortunately, the test Δ used had been sufficiently large thereby allowing the algorithm to find a structure which was close to the global minimum. However, there is neither an apriori way of choosing a suitable Δ nor is the global minimum necessarily the desired result since the globally minimum structure may be physically unsuitable. Therefore, the algorithm has to be a predominantly local optimizer that starts from an appropriate structure which takes into account the desired physical characteristics of the structure, e.g., general shape and span of the structure. During implementation, this means that the Newton method has to remain dominant for as many iterations as possible by using some relatively small Δ, e.g., in the second optimization described below, the Δ was set to $\frac{1}{200}$th of the initial nodal spacing.

Starting with a relatively large $\Delta = .01 = 10\%$ of the initial nodal spacing, the algorithm managed to converge, in 300 iterations, to a structure with 95% less deformation (See Figures 4 and 5) relative to the initial structure's deformation. This dramatic reduction in deformation relied mainly on random perturbations (See Figures 4 and 5) as the deformation had fallen smoothly only until the 68-th iteration, i.e., Newton's method was active only during the early stages of the optimization. After the Newton phase, the perturbations had resulted in discontinuous variations, mostly reductions, of the deformation and its Jacobian. In fact, the predominantly stochastic optimization was able to converge near the expected global minimum for the unconstrained problem — a structure with the loaded node located internally (See Figure 3: Note that the load is exerted in the negative vertical direction at the node in the center of the dashed diamond symbol). Despite the algorithm's convergence near the global minimum, this particular performance was due to the serendipitously large Δ used where the relatively large random perturbations (see previous section) and the initial

Newton steps had enabled the algorithm to sample this particular search space sufficiently. Since the goal of this particular optimization is not to find the global minimum but to locate the closest local minimum, the accuracy of the local Jacobian estimates had to be improved so as to maintain the Newton convergence for as many iterations as possible.

To prevent the algorithm from collapsing the structure, as had happened previously with high Δ, the algorithm was restricted to local searches by reducing the Δ. This reduction would result in better Jacobian estimates and hence smoother D and $\|J\|$ evolution. For the 16 node structure mentioned above, the algorithm's Δ was reduced by a factor of 20. As expected, after the algorithm had converged, the deformation was reduced by only 75% while the structure had retained much of the nodal ordering of the initial rectilinear configuration (See Figures 6, 7 and 8).

Unlike the globally optimal structure mentioned previously, the deformation of this structure smoothly dropped with iteration number until around the 2004-th iteration where the Newton method had finally stalled. After that first Newton stall, the deformation continued lower mostly due to random perturbations in the same manner as that of the higher Δ optimization. In effect, the local optimizer had been able to approximate the local J and H with sufficient accuracy such that the Newton method was able to find the local minimum closest to the initial structure.

Even though the locally and globally optimal structures differed in their convergence behaviors, both structures managed to align most of their struts to coincide with their respective predicted nodal displacement vectors (See Figures 3 and 6). From this observed alignment of the struts, the algorithm had been able to replicate the obvious optimization heuristic of positioning the struts such that they are aligned to the predicted nodal movements or forces. Moreover, in comparison to the topological optimization result of [6, page 542], the locally optimal structure is in agreement with the converged structure found by Bendsoe and Kikuchi. Relative to pre-existing mechanical designs, the familiar crane-like upper two nodal rows of the locally optimal structure can be interpreted as the optimal truss to support the downward acting load while the lower nodes form a structure similar to a cantilever truss.

4 Conclusions

From the results obtained, the algorithm was able to produce physically reasonable optimized structures based on the magnitude of Δ and the initial structure used. Although relatively[2] large Δ allowed for much faster convergence in the test structure's optimization, the physically unconstrained algorithm had wandered too far from the initial structure to arrive at a physically reasonable structure. This was corrected when the algorithm used a smaller Δ. With better estimates of J and H, the algorithm used the damped Newton search for a sufficient number of iterations to approach the local minimum of D before the simple stochastic converged upon it. Therefore, the algorithm would be greatly improved if an adaptive Δ scheme can be devised to retain both the rapid convergence and the locally optimal characteristics. A possible scheme is one where the algorithm starts with some small Δ which is then increased with each iteration while checking that $||\tilde{p}||$ and $||p - p_{init}||$ are both below some suitable thresholds.

In addition to benefiting the shape optimization problem tackled in this thesis, the dependence of the hybrid stochastic-Newton algorithm on Δ also suggests the possibility of a locally-accelerated, global optimizer where the thesis algorithm can be used to provide a seamless integration of the Newton method within a more sophisticated stochastic optimizer like simulated annealing. This approach promises a global optimizer which can smoothly accelerate its local or "low temperature" searches using the Newton method where Δ will be closely linked to the annealing algorithm's temperature parameter.

[2] Relative to the physical scale of the initial structure and the magnitude of the load

References

[1] C. G. Broyden. *The convergence of a class of double-rank minimization algorithms.* J. Inst. of Math. Appl. 6, 222-231, 1970.

[2] A. R. Curtis, M. J. D. Powell, J. K. Reid. *On the estimation of sparse Jacobian matrices.* J. Inst. Math. Appl. 13, 117-119, 1974.

[3] W. C. Davidon. *Variable metric method for minimization.* Atomic Energy Commission Research and Development Report. ANL-5990 (Rev.), Nov. 1959.

[4] G. H. Golub and C. Van Loan. *Matrix Computations.* Academic Press, NY, 1981.

[5] R. T. Haftka and Z. Gudal. *Elements of Structural Optimization.* 1992.

[6] M. P. Kamat. *Structural Optimization: Status and Promise.* American Institute of Aeronautics and Astronautics, 1992.

[7] U. Kirsch. *Structural Optimization: Fundamentals and applications.* Springer, NY, 1995.

[8] M. J. D. Powell and Ph. L. Toint. *On the estimation of sparse Hessian matrices.* SIAM J. Num. Anal. 11, 1060-1074, 1979.

[9] D. F. Shanno. *On variable metric methods for sparse Hessians.* Math. Comput. 34, 499-514, 1980.

[10] Ph. L. Toint. *On sparse and symmetric matrix updating subject to a linear equation.* Math. Comput. 31, 954-961, 1977.

[11] J. K. White. *Lecture notes of 6.336 of Fall 1996.* MIT, 1996.

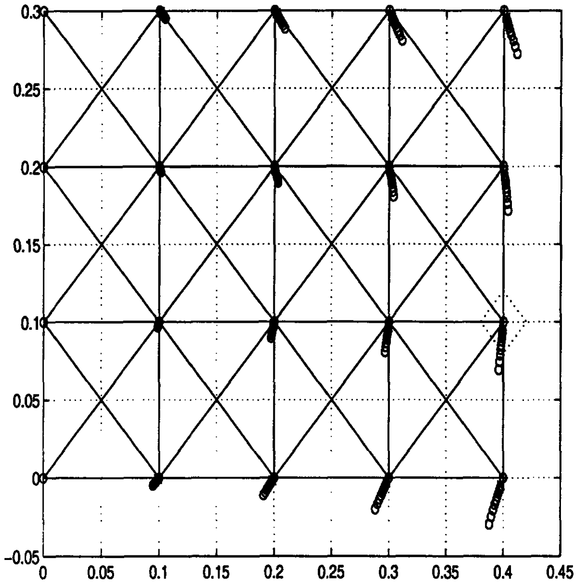


Figure 2: Initial 16 node structure with displacement vectors

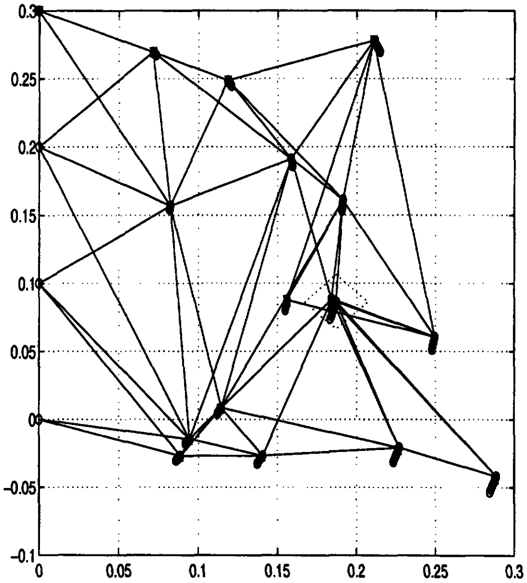


Figure 3: Converged 16 node structure with displacement vectors (418 iterations, $\Delta = .01$)

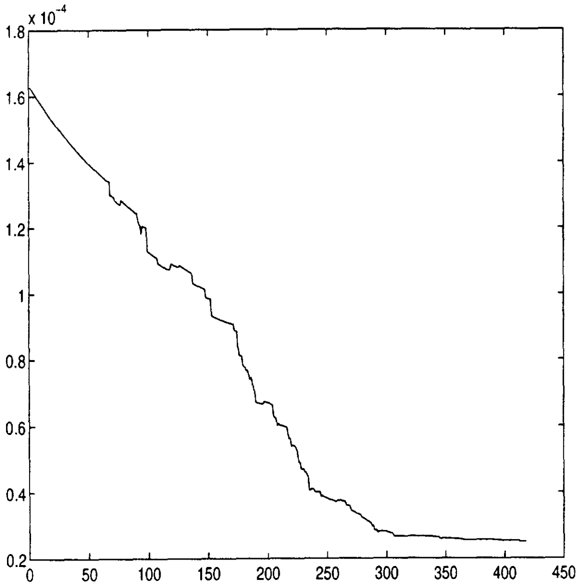


Figure 4: Reduction of D with each iteration of 16 node structure (418 iterations, $\Delta = .01$)

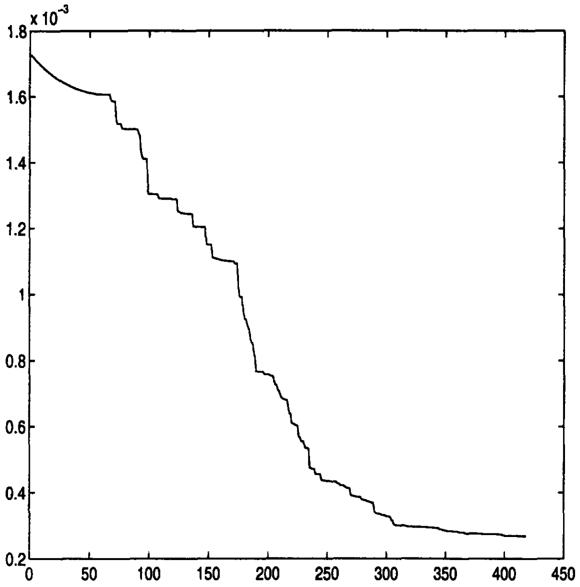


Figure 5: Reduction of $||J||$ with each iteration of 16 node structure (418 iterations, $\Delta = .01$)

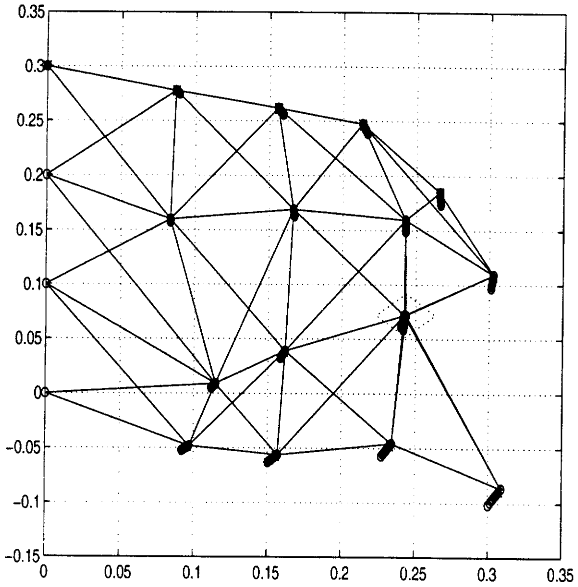


Figure 6: Converged 16 node structure with displacement vectors(5277 iterations, $\Delta = 5\text{x}10^{-4}$)

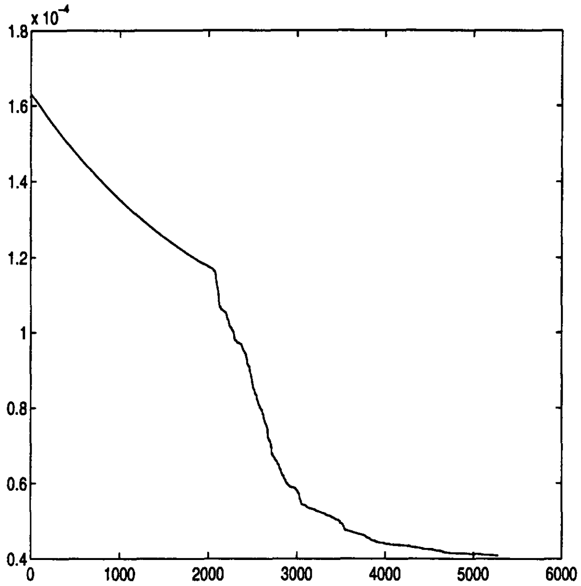


Figure 7: Reduction of D with each iteration of 16 node structure (5277 iterations, $\Delta = 5\text{x}10^{-4}$)

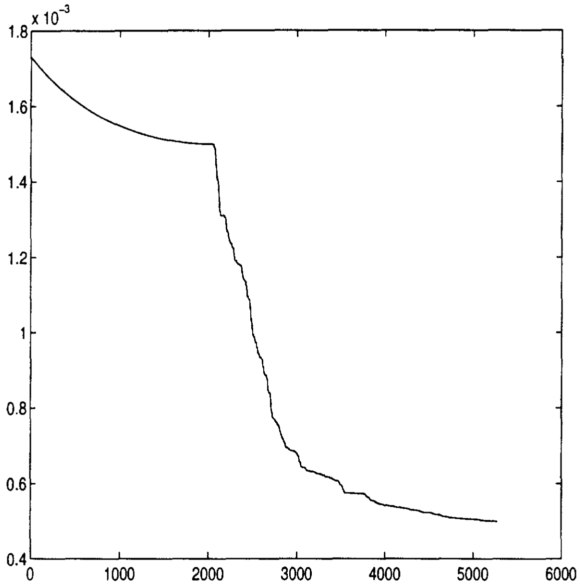


Figure 8: Reduction of $||J||$ with each iteration of 16 node structure (5277 iterations, $\Delta = 5\text{x}10^{-4}$)