Optimal Pick-up Locations for Transport and Handling of Limp Materials

Part II: Two-Dimensional Parts

SHRINIVAS LANKALAPALLI[1]

Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, New York 12180, U.S.A.

JEFFREY W. EISCHEN[2]

Department of Mechanical & Aerospace Engineering, North Carolina State University, Raleigh, North Carolina 27695, U.S.A.

ABSTRACT

Pick-up locations are obtained on two-dimensional limp parts that minimize a measure of deformation (strain energy). The parts are modeled as shells undergoing arbitrarily large deformations and rotations by a geometrically exact nonlinear shell finite element formulation. The strain energy is computed from the finite element solution for deformation, and the optimal locations are obtained by solving a bound constrained optimization problem. Results are given for various two-dimensional shapes.

In the first part [1] of this work, we obtained optimal pick-up locations for strips of limp materials. We used small and large deflection beam theory to model the strips and solved the optimization problem as a continuous one. In this part of the series, we solve for pick-up locations that minimize the strain energy of two-dimensional limp parts. The limp parts are modeled as shells undergoing arbitrarily large deformations and rotations using the geometrically exact nonlinear shell finite element formulation described in Part I [1]. Strain energy is computed from a finite element solution for the deformation and requires the discretization of the limp part by a mesh. The pick-up locations correspond to enforcing boundary conditions in the mesh, and in the finite element method, boundary conditions can be enforced only at nodes in the mesh. Since optimization algorithms require evaluation of the objective function for different pick-up locations, an automatic mesh generation procedure that locates nodes at specified pick-up locations in the domain is required to solve the problem as a continuous optimization problem. We implemented such a procedure when we solved for optimal locations on one-dimensional domains. We implement a similar meshing procedure for the specific problem of four pick-up locations on square/rectangular domains and solve the corresponding optimization problem as a continuous one. Since automatic meshing for two-dimensional domains of arbitrary shape is difficult, we develop a new gradient-based algorithm to solve the problem as a discrete problem on a fixed mesh, and we use iterative mesh refinement to improve the accuracy of the solution.

The following is the minimization problem for two-dimensional domains:

$$\min f(\mathbf{x}) : \mathbb{R}^{2n} \rightarrow \mathbb{R}$$
$$\mathbf{x} \in D \quad . \tag{1}$$

where $f(\mathbf{x})$ is the strain energy and objective function, D is the domain bounded by the boundary of the limp material part, and $\mathbf{x}$ is a $(2n \times 1)$ vector of coordinates of pick-up locations where n is the number of pick-up locations.

Gradient Based Algorithm for a Fixed Mesh

With a fixed mesh, the pick-up locations are restricted to be at nodes in the mesh and the solution space is discrete. The algorithm is a discrete variant of the steepest descent method, which has good global convergence properties. It consists of a systematic search procedure using the steepest descent direction to determine the next set of nodes among the closest nodes to the current nodes that result in a decrease in the objective function. For a more accurate solution, the mesh is refined in selective regions and the problem is resolved. The inputs are the finite element mesh of the limp part, the material properties, and the initial guesses (xy coordinates) of the pick-up locations.

To describe the algorithm, we use the following notation: n = number of pick-up locations, $\mathbf{N}$ = vector of node numbers of dimension n, $f(\mathbf{N})$ = strain energy

[1] Corresponding author, post doctoral research associate, tel: 518-276-6195, fax: 518-276-4886, email: slankal@scorec.rpi.edu
[2] Associate professor, email: eischen@eos.ncsu.edu

computed from the finite element solution with boundary conditions applied at nodes in **N**, $\nabla \mathbf{f}(\mathbf{N})$ = the gradient of strain energy with boundary conditions applied at nodes in **N**. This is a $(2n \times 1)$ vector and is computed by finite differences. For this purpose, the x and y coordinates of each node of **N** are perturbed and a new finite element solution is computed with each perturbation.

The algorithm is as follows:

1. Read mesh data, physical properties, and initial guess.
2. Find **N** = nodes in the mesh closest to the initial guess (xy coordinates) of pick-up locations.
3. Compute $f_{\text{trial}} = f(\mathbf{N})$ and $\nabla f(\mathbf{N})$.
4. While $k <$ maximum allowable number of iterations:
   (a) find $\mathbf{N}_s$ = nodes closest to **N** in the steepest descent (negative gradient) direction
   (b) $\mathbf{N}_t = \mathbf{N}_s$, compute $f_{\text{new}} = f(\mathbf{N}_t)$
   (c) If ($f_{\text{new}} < f_{\text{trial}}$) then
   (i) $\mathbf{N} = \mathbf{N}_t$, $f_{\text{trial}} = f_{\text{new}}$
   (ii) compute $\nabla f(\mathbf{N})$
   (iii) $k = k + 1$, go to 4 a, or else
   (i) find a new trial set of nodes $\mathbf{N}_t$ (explained below), if number of tries is $>$ *maxtry* then EXIT,
   (ii) compute $f_{\text{new}} = f(\mathbf{N}_t)$
   (iii) go to 4 c, endif.
5. Refine mesh in selective areas and rerun to obtain a more accurate solution.

We can find the nodes of $\mathbf{N}_s$ by computing the angles between the unit vectors in the steepest descent direction at the corresponding nodes of **N** and the unit vectors in the direction of the immediate surrounding nodes in the mesh. The closest node is the one that gives the smallest angle. This procedure is illustrated in Figure 1.

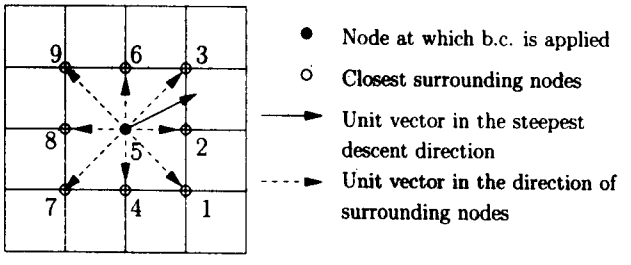


FIGURE 1. Node 3 is the closest node in the steepest descent direction.

The first trial set of nodes ($\mathbf{N}_t$) includes those in the steepest descent direction ($\mathbf{N}_s$). If this does not decrease the function value, then we choose $\mathbf{N}_t$ to be a combination of nodes from **N** and $\mathbf{N}_s$. For example, one choice for $n = 4$ would be $\mathbf{N}_t = [N_s(1), N(2), N(3), N(4)]$, *i.e.*, only the first node of **N** is replaced by the node in the steepest descent direction. The other tries involve replacing two nodes and three nodes of **N** by corresponding nodes in the steepest descent direction. The total number of such combinations (*maxtry*), including the first trial set, is given by $\sum_{k=1}^{n} C_k^n$. For $n = 4$ and $n = 3$, *maxtry* values are 15 and 7, respectively. When $n = 1$, *maxtry* $= 1$ and the algorithm simplifies to trying only the closest node in the steepest descent direction.

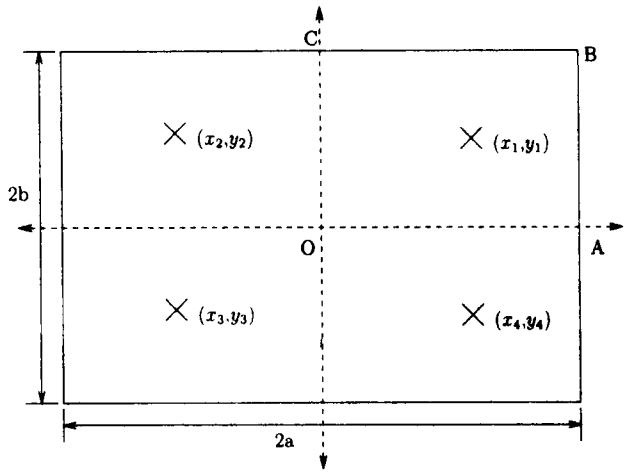
Initially, the limp part is discretized with a relatively coarse mesh and the algorithm is run for different initial guesses. Based on the results, the mesh is refined in selective regions and the algorithm is again run for different initial guesses. This process of iterative refinement and solution continues until an accurate enough solution is reached. Symmetry in the problem is exploited whenever possible because it reduces the size of the problem and the solution time.

Numerical Results

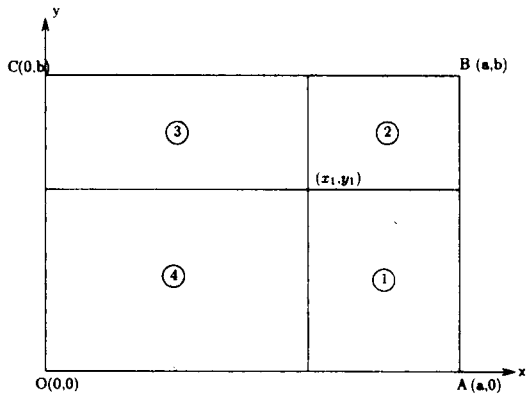
We obtained optimal pick-up locations for several different shapes and material properties. We first checked our algorithms by solving pick-up locations for relatively stiff parts. Because the deformations were small, we computed the objective function by solving a linear problem. This allowed us to quickly test the optimization algorithms. Then we systematically lowered the flexural rigidity and solved for optimal locations while computing the objective function from the solution to a nonlinear problem. We lowered the flexural rigidity until typical nonlinear phenomena (limit points and bifurcation points) associated with shells occurred frequently, resulting in divergence in the finite element computations and hence stalling of the optimization procedure. In order to demonstrate a practical application of these methods, we obtained optimal pick-up locations of parts used to make apparel and datawear products. In all the results, we assumed linear isotropic material behavior and set Poisson's ratio to $\nu = 0.3$.

FOUR PICK-UP LOCATIONS FOR SQUARE/RECTANGULAR DOMAINS

The problem of four pick-up locations on a rectangular domain of dimensions $2a \times 2b$ can be simplified to one pick-up location on a quarter domain by exploiting symmetry, as shown in Figure 2. It is sufficient to solve for optimal location (x_1, y_1) on OABC by enforcing symmetrical boundary conditions along edges OA and OC. The meshing problem simplifies to that of generating a mesh of quadrilateral elements for a rectangle of dimensions $a \times b$, given the desired level of discretization and coordinates of pick-up locations (x_1, y_1). We have developed an automatic meshing procedure that distributes a specified number of



(a) Original Problem



(b) Symmetry used to solve for 1 pick-up location

FIGURE 2. Four pick-up locations on a rectangular domain.

elements in the x and y directions to the four different regions of the rectangle shown in Figure 2b in proportion to the size of each region. This procedure allows for systematic generation of nodal coordinates, element connectivity, element numbers, node numbers where symmetric boundary conditions are to be enforced, and the node number corresponding to (x_1, y_1). A limitation of the procedure is that bad meshes with quadrilateral elements of poor aspect ratio are generated when (x_1, y_1) is located very close to the boundary. During optimization, we prevent this from happening by perturbing (x_1, y_1) away from the boundary in the search direction. Also, the initial guesses are chosen on or away from the boundary.

We have included this meshing procedure in the nonlinear shell finite element code used to compute the objective function and solved the optimization problem using the L-BFGS-B code for bound constrained optimization developed by Zhu *et al.* [2]. We computed the objective function by summing the areas under the load deflection curve of every node using Equation 11 in Part I [1]. We used parameters *factr* = 1.0 and *pgtol* = $1e^{-8}$ to terminate the optimization and computed the gradient by finite differences. The variables were bounded by the dimensions of the rectangle (*i.e.*, $0 \le x_1 \le a$ and $0 \le y_1 \le b$), and we verified the solution by solving the problem for different levels of discretization and initial guesses. We characterized the material properties in terms of the following nondimensional number:

$$\gamma = \frac{wl^3}{D}, \qquad (2)$$

where w = weight per unit area, l = length of the square ($2a = 2b = l$), $D = \dfrac{Et^3}{12(1-\nu^2)}$ is the flexural rigidity, t = thickness, E = Young's modulus, and ν = Poisson's ratio. Increasing values of γ result in lower flexural rigidity and greater deformation.

Small Deformations

We first solved the problem for a square domain of length $l = 2.0$ m, thickness $t = 0.1$ m, $\gamma = 0.12$, and w and E corresponding to high flexural rigidity. Since the deformations are small, the objective function can be computed from a single load step. This allows us to quickly check our optimization algorithms. We first solved the problem as a continuous optimization problem using the L-BFGS-B code with automatic meshing. We specified a desired level of discretization of ten elements in the x and y directions in the automatic meshing procedure and tried different initial guesses. Figure 3 shows the iteration histories plotted on the square domain starting from different initial guesses on the boundary. Solutions from different initial guesses converged to any accuracy within four decimal places. The optimal location was at

$$x_1 = 0.54505 \text{ m}; \quad y_1 = 0.54505 \text{ m} \quad .$$

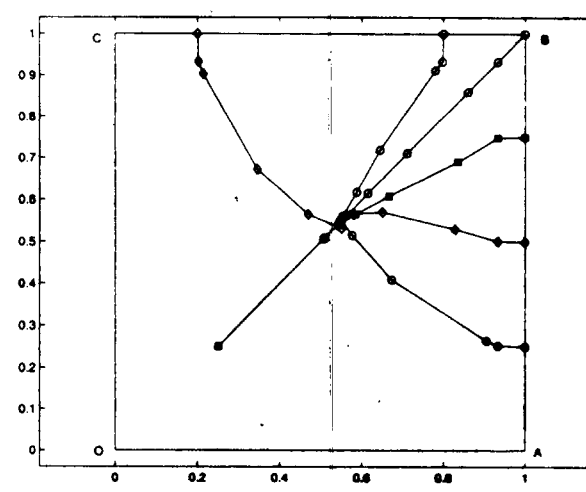


FIGURE 3. Iteration histories from different initial guesses.

Next, we solved the problem as a discrete optimization problem on a fixed mesh with iterative mesh refinement using the algorithm described earlier. The entire square domain was discretized and all four pick-up locations were solved for. Figure 4 shows the iteration histories of the optimization on a fine mesh obtained after three iterations of mesh refinement. Sets of four lines of the same type trace the movement of the pick-up locations from different initial guesses as optimization progresses. Note that the lines pass through nodes in the mesh because boundary conditions (pick-up locations) can be applied only at nodes. Solutions from different initial guesses converge to within an element. The final locations from different initial guesses along with the value of the objective function are shown in Table I. Note that there is less than a 1% difference in objective function between the solutions obtained from different initial guesses.

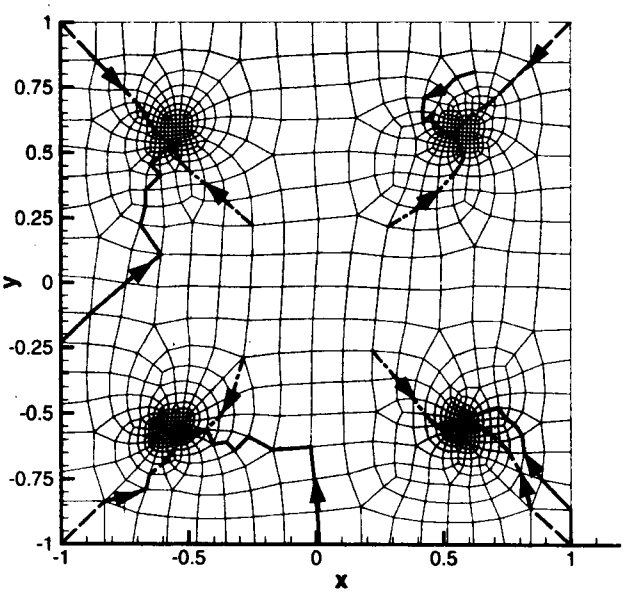


FIGURE 4. Iteration histories from different initial guesses.

Comparing the results for (x_1, y_1) from the two methods, we note that the discrete solution on a fixed mesh matches up to two decimal places with the continuous solution. This exercise, in addition to checking our implementation of various procedures, serves to validate the performance of the algorithm developed earlier.

Effect of flexural rigidity and size on optimal locations for square domains: We studied the effect of flexural rigidity and size by systematically increasing the nondimensional number γ and solving for optimal locations for different l/t ratios. We used different initial guesses and discretizations for each set of γ and l/t values. Figures 5 and 6 show the variations of x and y coordinates of optimal locations as γ and l/t vary. The locations lie along the diagonal, and as the flexural rigidity is lowered (higher value of γ), the locations tend to move

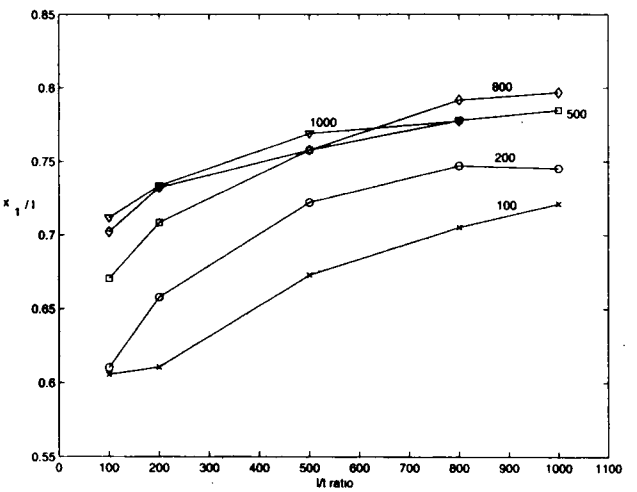


FIGURE 5. Effect of γ and l/t ratio on x_1/l.

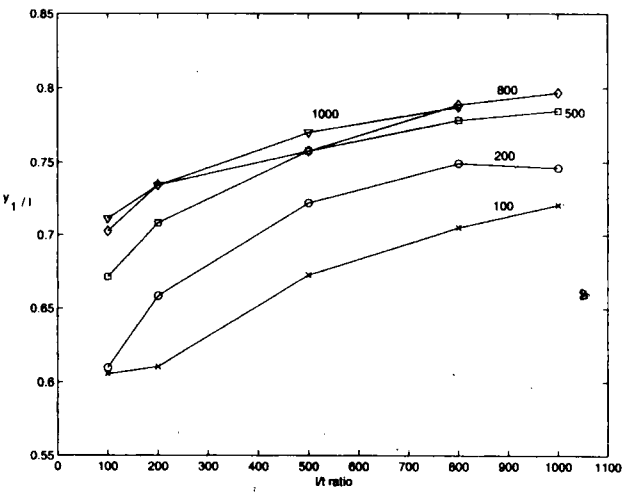


FIGURE 6. Effect of γ and l/t ratio on y_1/l.

TABLE I. Coordinates to which different initial guesses converge for the mesh in Figure 4.

Line type	x_1	y_1	x_2	y_2	x_3	y_3	x_4	y_4	Objective function
- - -	0.5445	0.5505	−0.5454	0.5382	−0.5413	−0.5584	0.5525	−0.5371	2.201E-04
-	0.5445	0.5505	−0.5454	0.5382	−0.5399	−0.5444	0.5513	−0.5524	2.206E-04
- · -	0.5445	0.5505	−0.5454	0.5382	−0.5413	−0.5584	0.5525	−0.5371	2.201E-04

toward the corner. When the size of the domain increases (higher l/t ratio), the locations tend to move toward the corner.

Effect of aspect ratio on optimal locations for a rectangular domain: We studied the effect of aspect ratio on optimal locations for blue denim fabric, using the material properties for denim from Part I [1] and obtaining the optimal locations for different aspect ratios. Figure 7 shows the variations of x_1/a and y_1/b as the aspect ratio varies. Note that the optimal locations do not lie along the diagonal for aspect ratios other than 1.0. Also, the change in the location of y_1/b is greater than that in x_1/a.

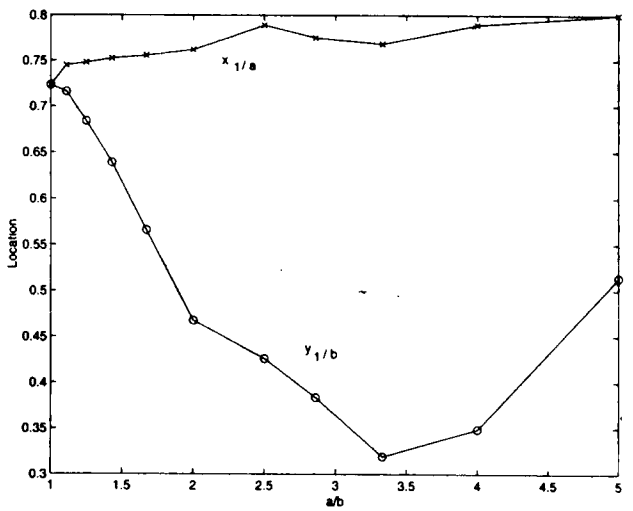


FIGURE 7. Effect of aspect ratio on x_1/a and y_1/b for blue denim fabric.

Optimal locations for square and rectangular fabric parts: We obtained optimal locations for several square and rectangular fabric parts. Figure 8 shows the deformed shape at optimal locations for a 30 × 30 cm square cotton twill fabric part (γ = 474.87 and l/t = 499.16). Figure 9 shows the deformed shape of a 40 × 24 cm rectangular blue denim fabric part.

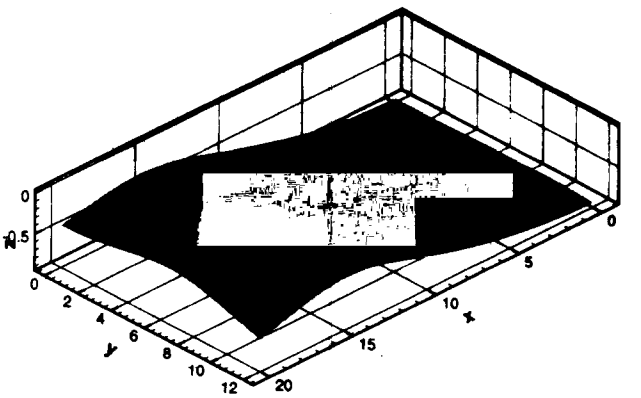


FIGURE 9. Deformed shape of rectangular blue denim fabric part at optimal locations: x_1 = 15.11 cm, x_2 = 6.79 cm.

PICK-UP LOCATIONS FOR ARBITRARY DOMAINS

We used the algorithm for a fixed mesh described earlier to solve for optimal locations for limp parts of various shapes and material properties. We present results for parts obtained from datawear and apparel products. Dimensions of the parts used to make up the cover of the Symbol 4340 hand-held computer came from Medcovers Inc. (Raleigh, North Carolina), a company that manufactures equipment covers. Two parts (symbol-1 and symbol-2) made of laminated 1000 denier Cordura are shown in Figure 10. Dimensions of typical shirt and shorts parts came from the Apparel Lab, College of Textiles, North Carolina State University. A short front and shirt front part are shown in Figure 11. The material properties used are listed in Table V in Part I [1].

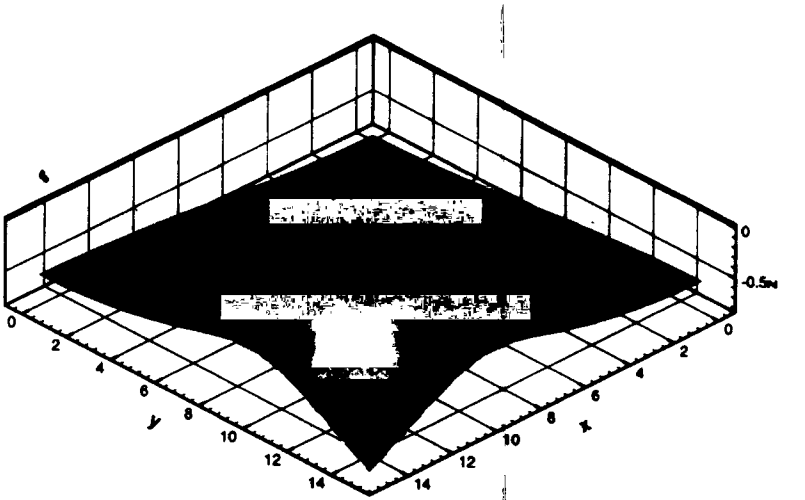


FIGURE 8. Deformed shape of square cotton twill fabric part at optimal locations: x_1 = 11.16 cm, x_2 = 11.16 cm.

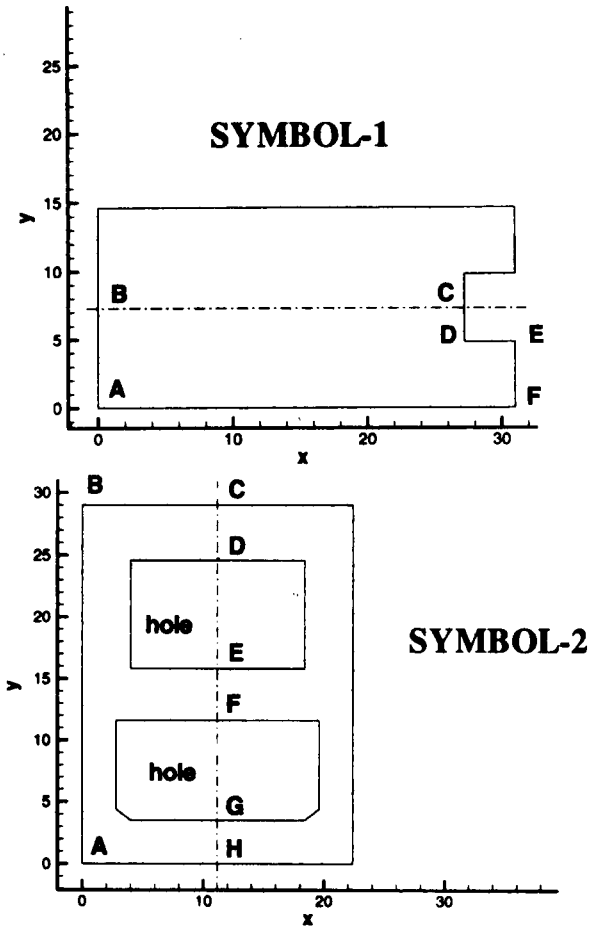


FIGURE 10. Shapes of two symbol parts (all dimensions in centimeters).

Symbol-1 part: The symbol-1 part is symmetrical about the x-axis. Hence, when solving for four pick-up locations, it is sufficient to solve for two pick-up locations on the domain bounded by ABCDEF by enforcing symmetrical boundary conditions along edge BC. Figure 12 presents the mesh on which solutions from different initial guesses are shown by lines that connect the initial and final nodes. Numbers are used to identify lines corresponding to different initial guesses. Table II shows the coordinates to which different initial guesses converge, along with the final objective function values. Note that different initial guesses converge to nodes that are close by, if not within, an element. In addition, there is a very small difference in the value of the final objective function. Although we could obtain a more accurate

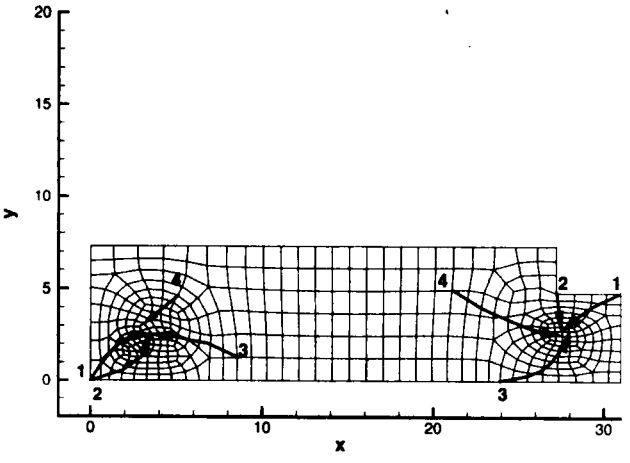


FIGURE 12. Mesh with iteration histories for symbol-1.

TABLE II. Coordinates to which different initial guesses converge for symbol-1.

Initial guess	x_1	y_1	x_2	y_2	Objective function
1	3.541	2.526	27.569	2.503	0.68268
2	3.541	2.526	27.569	2.503	0.68268
3	3.541	2.526	27.569	2.503	0.68268
4	**3.082**	**3.107**	**27.534**	**2.757**	**0.67715**

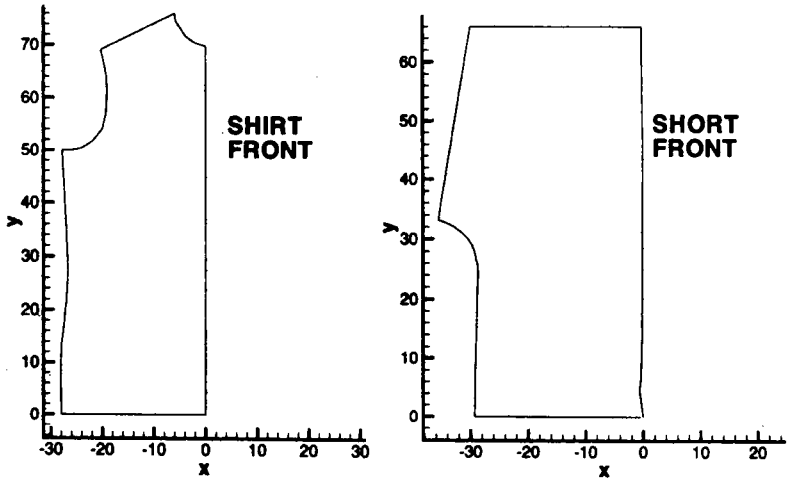


FIGURE 11. Shapes of shirt and short parts (all dimensions in centimeters).

solution by using a more refined mesh, for practical purposes we can consider the solution with the lowest objective function as optimal. The deformed shape at optimal locations is shown in Figure 13.

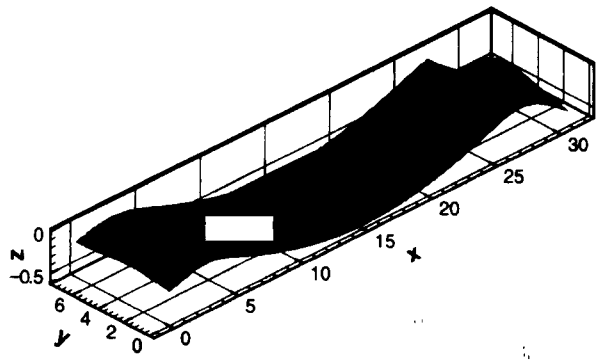


FIGURE 13. Deformed shape of symbol-1 at optimal locations.

Symbol-2 part: The symbol-2 part is symmetrical about the y-axis. We obtained four and six pick-up locations by solving for two and three locations, respectively, on half the domain bounded by ABCDEFGH. Symmetrical boundary conditions were enforced along edges CD, EF, and GH. Figure 14 shows the mesh for two locations on which solutions from different initial guesses are shown by lines that connect the initial and final nodes. Numbers identify the lines corresponding to different initial guesses. In this case, all initial guesses converge to the same nodes. The optimal locations are at

$$x_1 = 0.8793, \quad y_1 = 2.3885 \quad ,$$
$$x_2 = 1.5835, \quad y_2 = 26.567 \quad .$$

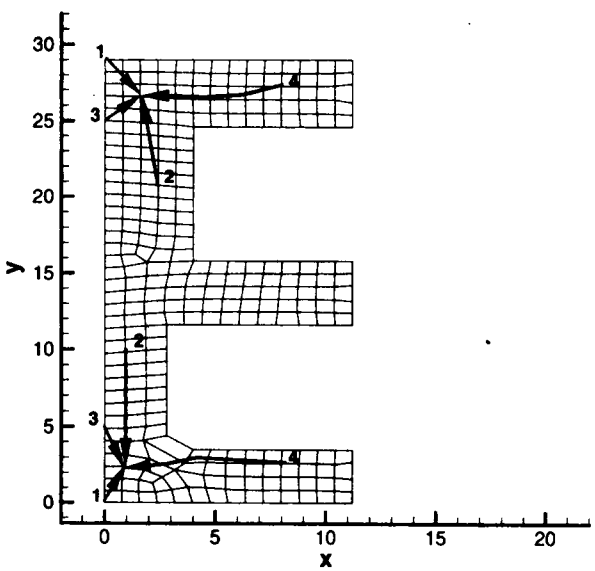


FIGURE 14. Mesh with iteration histories for symbol-2.

The deformed shapes at optimal locations are shown in Figures 15 and 16.

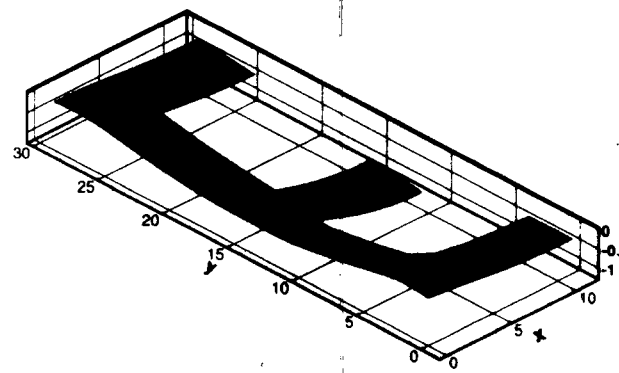


FIGURE 15. Deformed shape of symbol-2 at optimal locations.

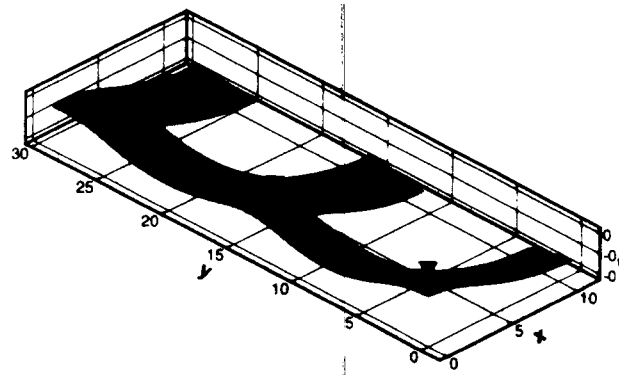


FIGURE 16. Deformed shape of symbol-2 at optimal locations.

Shirt and short front: We solved for four pick-up locations for a blue denim shirt front and a shorts front. Figures 17 and 18 show the deformed shape of the shirt and short parts, respectively, when picked up at optimal locations.

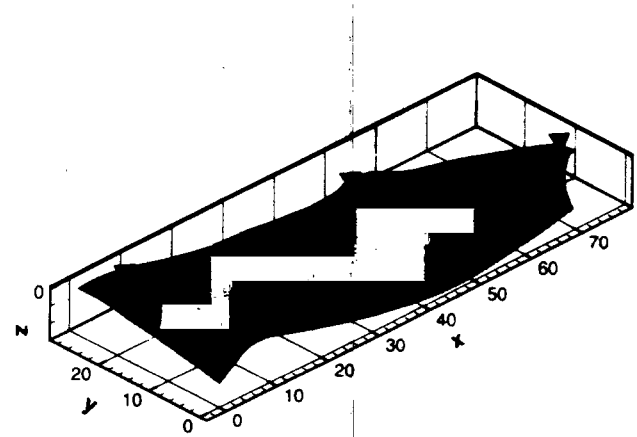


FIGURE 17. Deformed shape of shirt front at optimal locations.

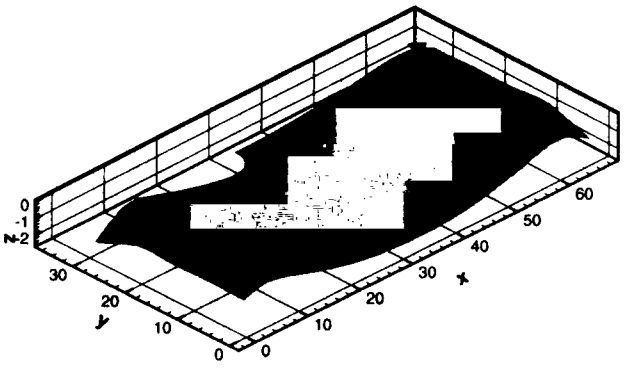


FIGURE 18. Deformed shape of shorts front at optimal locations.

Conclusions

In this paper, we have developed procedures to solve for optimal pick-up locations that minimize the strain energy of two-dimensional limp parts. The limp parts are modeled as shells undergoing large deformations and rotations by a geometrically exact nonlinear shell finite element formulation. The optimal locations are solved using gradient-based algorithms in which both objective function and gradient are computed from a nonlinear finite element solution. We have developed a novel gradient-based algorithm to solve for optimal locations on a fixed mesh to overcome the need for automatic meshing of arbitrarily shaped two-dimensional domains, and have successfully used these methods to obtain results for two-dimensional domains of different shapes.

ACKNOWLEDGMENTS

This research was funded by the National Textile Center, and we gratefully acknowledge that support. We would like to thank Professor T. G. Clapp, College of Textiles, North Carolina State University, for various useful discussions and suggestions. We would also like to acknowledge the excellent computing facilities of the North Carolina Supercomputing Center, which we used for the numerical results. We thank Medcovers Inc. for providing the dimensions of different products as well as the fabric samples.

Literature Cited

1. Lankalapalli, S., and Eischen, J. W., Optimal Pick-up Locations for Transport and Handling of Limp Materials, Part I: One-Dimensional Strips, *Textile Res. J.* **73,** 787–796 (2003).
2. Zhu, C., Byrd, R. H., Lu, P., and Nocedal, J., Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-Scale Bound Constrained Optimization, *ACM Trans. Math. Software* **23**(4), 550–560 (1997).

Manuscript received September 23, 2002; accepted January 27, 2003.