Mecánica Computacional Vol 20. pp. 179—186
Compilado por Fernando G. Flores
Córdoba, 30 de octubre - 2 de noviembre de 2001

NUMERICAL MODELING OF METAL FORMING: PREFORM OPTIMIZATION

P. A. Muñoz-Rojas[1] & J. S. O. Fonseca
PROMEC – UFRGS
Rua Sarmento Leite, 425, CEP 90050-170
Porto Alegre – RS – Brasil
and
G. J. Creus
CEMACOM/PPGEC/PROMEC – UFRGS
Av. Osvaldo Aranha, 99, 3° andar, CEP 90035-190
Porto Alegre, RS - Brasil

ABSTRACT

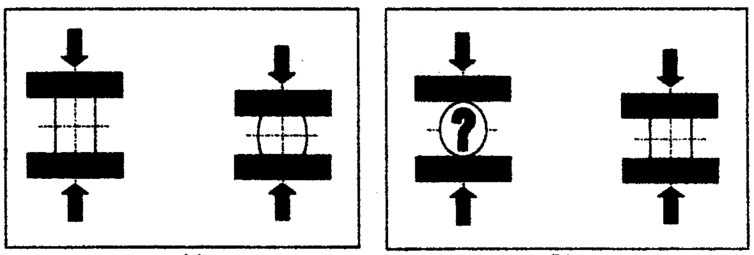
A numerical procedure for the design of optimized preforms for use in metal forming is described. The choice of design variables and objective functions and the procedures used for the determination of sensitivities are discussed, introducing new proposals. An example shows the performance of the code.

RESUMEN

Se describe un procedimiento numérico para la optimización de preformas en conformación mecánica. Se discute la selección de variables de diseño y funciones objetivo para esta aplicación bién como procedimientos para la determinación de la sensibilidad, introduciendo nuevas propuestas. Se presenta un ejemplo que demuestra el desempeño del código desarrollado.

INTRODUCTION

Fig. 1(a) shows a simple direct problem in finite plasticity: the determination of the form of an elastoplastic cylinder under axial compression with friction. A classical benchmark in preform optimization which is related to this problem is illustrated in Fig. 1(b): the axissymmetrical form in the left has to be determined, such that, under axial compression it deforms into the form of the perfect cylinder in the right.



(a) (b)

Fig 1. Preform optimization of a cylinder.

[1] on leave from Departamento de Engenharia Mecânica, Universidade do Estado de Santa Catarina, Campus Universitário Avelino Marcante, Bom Retiro, CEP 89223-100, Joinville, SC, Brazil.
e-mail: dem2pmr@joinville.udesc.br

This is a kind of inverse problem in mechanics, and can be solved with optimization techniques. As in all optimization problems, there are design variables (that determine the shape) and an objective function to be minimized. In preform optimization, the objective function is usually the difference between the obtained and the desired shapes, measured in an adequate norm. Some care must be taken with these concepts.

DESIGN VARIABLES

In principle, the design variables could be the coordinates of FE nodes, as initially proposed by Zienkiewicz & Campbell [1]. However, this approach is not efficient because of two main reasons: firstly, the number of design variables is mesh dependent and unnecessarily high and second, because as the nodes can change their positions independently, C^0 continuity of the shape cannot be assured and a jagged boundary is often obtained (see Haftka & Grandhi [2]), thus demanding a post processing stage for boundary smoothing at each iteration.

The modern approach is to parameterize the boundary of the preforms using NURBS. In this case, one possibility is to define the spline by its control points which are employed as design variables. Another one is to define the spline through a number of keypoints that it must match. At each keypoint a normal vector to the spline is calculated and the design variables are the variations of the original positions of the keypoints following the normal directions.

The last approach was adopted successfully by Chenot et al. [3] to solve the basic benchmark displayed in Fig. 1. Following a similar procedure we noticed that the results are very dependent on the initial positions of the keypoints. Thus, we introduced as new design variables the keypoint displacements tangent to the spline at each keypoint. Although the analytical sensitivity with respect to tangent displacements is zero, due to numerical approximation we expected to obtain a lower dependence on the choice of the initial position of the keypoints. On the contrary, difficulties in convergence and geometrical degeneracy were found in some cases [4]. We believe that the dependency on the initial position of the keypoints is related to local minima and a solution based on the enrichment of the design space is currently being developed.

Many research works adopt the NURBS control points as design variables. Among them we can cite the ones developed by the groups leaded by Zabaras [5] and Grandhi [6]. One advantage of this approach is related to the calculation of design sensitivity. If control points are used as design variables, the derivative of any quantity Ψ that depends on the boundary shape (and implicitly on the design variable) can be expressed by

$$\frac{\partial \Psi}{\partial p} = \frac{\partial \Psi}{\partial X}\frac{\partial X(t)}{\partial p} + \frac{\partial \Psi}{\partial Y}\frac{\partial Y(t)}{\partial p} \tag{1}$$

where $X(t)$ and $Y(t)$ are the coordinate equations of the parameterized boundaries. Changing notation we can represent $X(t)$ or $Y(t)$ by a cubic NURBS as

$$\begin{gathered} Q_i(t) = p_{i-3} b_{i-3,4}(t) + p_{i-2} b_{i-2,4}(t) + p_{i-1} b_{i-1,4}(t) + p_i b_{i,4}(t) \\ 3 \le i \le m,\ t_i \le t \le t_{i-1} \end{gathered} \tag{2}$$

where p_i is the i-th control point and t_i is the i-th knot [7]. If $Q_i(t)$ is the X coordinate of the point, p_i is the X coordinate of the control point and accordingly to coordinate Y. A simple inspection shows that this equation is explicitly dependent on the control points coordinates while this does not happen with respect to the keypoints locations. Thus, the spline derivative at point t is explicitly given by

$$\frac{\partial Q_i(t)}{\partial p_k} = b_{k,4}(t). \tag{3}$$

The terms $\partial\Psi/\partial X$ and $\partial\Psi/\partial Y$ are usually explicitly available. Another advantage of this approach is that it is possible to impose tangent continuity between two adjacent splines through linear equality constraints on their control point coordinates [8]. If keypoints were being used as design variables, highly nonlinear equality constraints could be necessary. These are always difficult to satisfy and it is a good practice to avoid them. The main disadvantage of this approach is the computational cost, since at each control point one has one design variable for each degree of freedom, while in the former approach there is only one design variable per keypoint.

OBJECTIVE FUNCTION

The objective function is determined, in the work of most researchers, as indicated in Fig. 2. and Eq. (4). In the Figure and Equation, (x_i, y_i) and (x_{i+1}, y_{i+1}) are the nodal coordinates of the i-th and (i+1)-th boundary nodes.

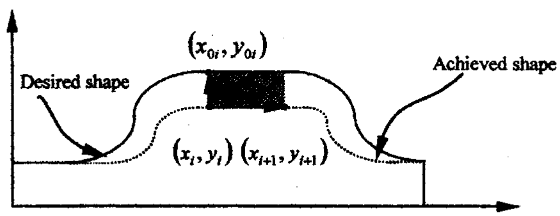
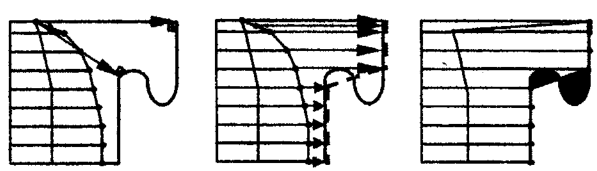


Fig. 2. Objective function as presented by Zhao et al. [6].

$$\Psi = \sum_{i=1}^{N} A_i^2 = \sum_{i=1}^{N} \left[(x_i - x_{0i})^2 + (y_i - y_{0i})^2\right]\left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2\right] \tag{4}$$

In order to find the coordinates (x_{0i}, y_{0i}), the procedure needs to find pointwise projections of the desired boundary on the obtained one. These projections are not always unique and this fact can lead to inadequate results. Furthermore, the minimization of Eq. (4) may not represent the proper matching of the obtained and desired geometries. These issues are depicted in Fig. 3.



(a) (b) (c)

Fig.3. Drawbacks of the usual objective function.

Thus, a new procedure which uses Boolean XOR operations to measure the "difference" between the desired and achieved final shapes was implemented [4]. This method uses no normals and avoids the usual objective function's pitfall. On the other hand, we found situations in which the code would converge towards a shape not correct, as in Fig. 4.

Fig. 4. Objective function pitfall. Target shape: regular cylinder. Upper left quadrant modeled.

This was solved introducing the boundary dimension (perimeter in a 2D case) as an additional term in the objective function proposed, which became

$$f = W_1\left(\frac{A}{A_0}\right)^2 + W_2\left(\frac{P-P_d}{P_0-P_d}\right)^2 \tag{5}$$

where W_1 and W_2 are weights (W_1=W_2=0.5 in this work), A and A_0 are the XOR area and its initial value, P and P_0 are the obtained perimeter and its initial value, and Pd is the desired perimeter. The strategy showed up to behave very robustly.

SENSITIVITY ANALYSIS

The determination of sensitivities (i.e. objective function and constraints derivatives with respect to the design variables) is a crucial point of any optimization problem. If the sensitivities are inaccurate the convergence rate may be severely affected and the procedure may even diverge. On the other hand, sensitivity evaluation is usually one of the most expensive parts of optimization processes. Therefore it is highly desirable to use analytical derivatives when available. If this is not the case, a semi-analytical approach is a good option [9]. If one aims to test a formulation regardless of efficiency, finite differences is acceptable.

In non-linear analyses, the term *semi-analytical sensitivity* is usually employed for the application of finite differences to differentiate the non-linear iteration residual with respect to the design variables. In this work we use the term *semi-analytical* meaning that a chain rule is applied and some terms are evaluated analytically, while others are obtained using regular finite differences. Thus, differentiating Eq. (5) with respect to the j-th design variable, we get

$$\frac{df}{dp_j} = \frac{\partial f}{\partial x_i}\frac{dx_i}{dp_j}; \qquad x_i = [x_i \quad y_i]^T \tag{6-7}$$

$$\frac{df}{dp_j} = 2\left[W_1\frac{A}{(A_0)^2} \quad W_2\frac{P-P_d}{(P_0-P_d)^2}\right]\begin{bmatrix}\dfrac{\partial A}{\partial x_i} & \dfrac{\partial A}{\partial y_i}\\ \dfrac{\partial P}{\partial x_i} & \dfrac{\partial P}{\partial y_i}\end{bmatrix}\begin{Bmatrix}\dfrac{dx_i}{dp_j}\\ \dfrac{dy_i}{dp_j}\end{Bmatrix} \tag{8}$$

$$\frac{\partial x_i}{\partial p_j}=\frac{\partial X_i}{\partial p_j}+\frac{\partial u_i}{\partial p_j}=\begin{Bmatrix}\frac{\partial X_i}{\partial p_j}+\frac{\partial u_i}{\partial p_j}\\ \frac{\partial Y_i}{\partial p_j}+\frac{\partial v_i}{\partial p_j}\end{Bmatrix} \qquad (9)$$

where f is the objective function, p_j is the j-th design variable (in this case, a shape parameter) and u_i is the i-th nodal displacement vector. The term $\partial X_i/\partial p_j$ is called "velocity field" in the literature and its value on the boundary is explicitly known through the spline equation (2). In this work, the computation of du_i/dp_j was obtained using regular forward finite differences. This demands special routines to assure that the mesh topology is not changed during the design variable perturbations. The inverse power Laplacian relocation scheme was used for this purpose, following [10]. The power value adopted was 1.5.

OPTIMIZATION ALGORITHM

Up to the present stage we have tested basically two mathematical programming algorithms. The first one is sequential linear programming, which was successful but very inefficient because heuristic move limits had to be imposed throughout the procedure in order to assure the validity of the linearizations [4]. The second optimization algorithm employed was Svanberg's Globally Convergent Method of Moving Asymptotes – GCMMA [11]. This was applied together with a structured mesh generation approach and overall finite difference sensitivity. Excellent results were obtained leading to a much more efficient procedure [12]. Nevertheless the efficiency is still not as desired because of the finite differences. In the present work we will present the same optimization algorithm but this time together with a non-structured mesh generation scheme and semi-analytical sensitivity. As already pointed out, these issues demand proper attention and the adopted procedure will be detailed in a forthcoming paper [13]. The non-structured mesh generation allows to model complex domains concentrating fine elements near the boundary while employing a fairly coarse discretization elsewhere. The non-structured mesh generation was accomplished using GiD 6.1.2. – a general purpose pre and post-processor [14] developed by the Universidad Politécnica de Catalunya and the forging simulation was performed using METAFOR – a metal forming elastic-plastic finite element program [15] developed at the University of Liège. A managing code was developed which iteratively calls GiD, METAFOR, calculates the sensitivities and calls GCMMA, leading to the solution of the problem.

EXAMPLE

The example presented in Figs. 5-8 considers the upsetting of an axissymmetrical preform in order to achieve a perfect cylinder after 20% height reduction. The component's material is steel (E=210000 MPa, ν=0.3, Isotropic Hardening Modulus = 2100 MPa, σ_e=270 MPa). Sticking contact exists between the component and the die. A penalization factor $\alpha=10^7$ was applied to impose normal contact.

Due to symmetry conditions only the upper right quadrant of the longitudinal section was modeled and discretized. Fig. 5 presents the geometric model which uses a cubic B-spline with 7 control points. There are 11 design variables: the horizontal coordinates of points 4, 5, 6, 7, 8 and 9, and the vertical coordinates of points 5, 6, 7, 8 and 9. The unstructured mesh generated for the initial design is depicted in Fig. 6.

Two procedures were applied for optimization. In both, the sensitivity was calculated using the semi-analytical approach with a perturbation of 5×10^{-3} times the spline length. In the first procedure the mesh topology is kept unchanged during each optimization iteration so that the sensitivity information keeps consistent in each subproblem. In this stage the mesh is modified proportionally to the velocity field, calculated in the domain using an inverse power Laplacian smoothing. After convergence of each subproblem, mesh regeneration including topology modification is allowed so that a less distorted mesh is defined for the new geometry. In the second procedure, only Laplacian relocation is allowed in the whole process.

The final shapes achieved using the optimization procedures without and with remeshing are presented in Fig. 7 and Fig. 8, respectively.

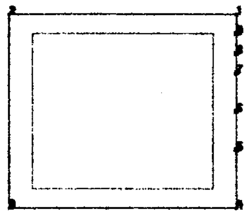


Fig. 5. Geometric model of the upper right quadrant.

Fig. 6. Initial preform and forged shape.

Fig. 7. Optimized preform and forged shape (GCMMA without remeshing).

Fig. 8. Optimized preform and forged shape (GCMMA with remeshing).

The graphs displayed in Fig. 9 show that while both strategies work well, the one that includes remeshing achieves a lower objective function in the same number of iterations. We believe that this behavior can be attributed to a less distorted initial mesh in each subproblem. Hence, the larger the deformations, the more pronounced this effect is expected to be.

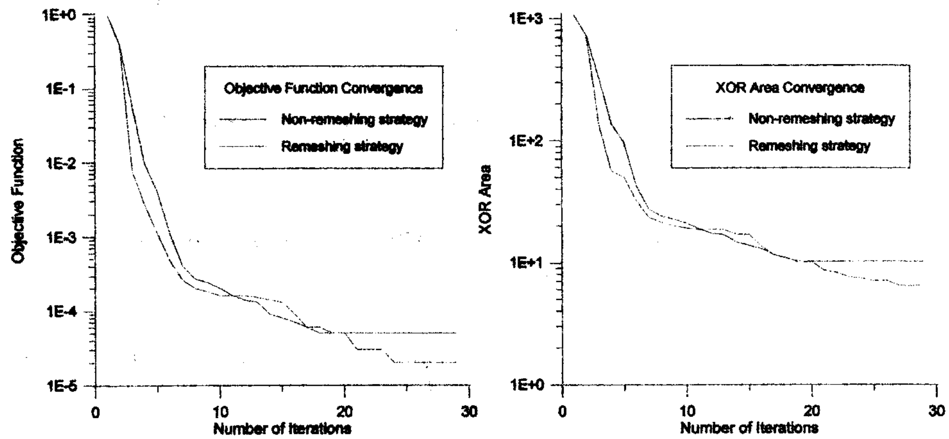


Fig. 9. Convergence behavior: (a) Objective function; (b) XOR area.

CONCLUDING REMARKS

A code for the optimization of preforms is described. The use of adequate design variables and a new objective function has made it fairly robust. The use of an inverse Laplacian relocation scheme provides the sensitivities efficiently. The procedure is still under development and test.

ACKNOWLEDGEMENTS

We thank Prof. Krister Svanberg for permission to use of his GCMMA code and Prof. M. Hogge for the use of the METAFOR system. The financial support given by CNPq is greatly acknowledged. The first author would like to thank UDESC for the waving of his lecture duties in order to perform his Ph.D. research studies.

REFERENCES

[1] Zienkiewicz O.C. & Campbel, J.S., "Shape Optimization and Sequential Linear Programming", *Optimum Structural Design (Ed. by R.H. Gallagher and O.C. Zienkiewicz)*, John Wiley, New York, pp. 109-126, 1973.

[2] Haftka, R.T. & Grandhi, R.V., "Structural Shape Optimization – A Survey", *Comput. Methods Appl. Mech. Engrg.* 57, pp. 91-106, 1986.

[3] Chenot, J.L., Massoni, E. and Fourment, L., "Inverse Problems in Finite Element Simulation of Metal Forming Processes", *Engineering Computations*, vol. 13, nº 2/3/4, pp. 190-225, 1996.

[4] Muñoz-Rojas, P.A., Bittencourt, E., Fonseca, J.S.O. & Creus, G.J., "Preform Optimization of 2D Forged Components Using Sequential Linear Programming", *XXI Iberian Latin-American Congress on Computational Methods in Engineering, CILAMCE 2000*, Rio de Janeiro, Brazil, 2000.

[5] Srikanth, A., Zabaras, N. & Frazier, W.G. "An Updated Lagrangian Finite Element Sensitivity Analysis of Large Deformations using Quadrilateral Elements", ", *Int. J. Numer. Meth. Engng.*, to appear, 2000.

[6] Zhao, G., Wright, E. & Grandhi, R.V., "Preform Die Shape Optimization in Metal Forming Using an Optimization Method", *Int. J. Numer. Meth. Engng*, 40, pp. 1213-1230, 1997.

[7] Foley, J.D., van Dam, A., Feiner, S.K. & Hughes, J.F., "Computer Graphics: Principles and Practice", 2nd Edition, Addison-Wesley Publishing Company, 1990.

[8] Wang, X., Zhou, J. & Hu, Y., "A Physics-Based Parameterization Method for Shape Optimization", *Comput. Methods Appl. Mech. Engrg.*, 175, pp. 41-51, 1999.

[9] Kleiber, M., Antúnez, H., Hien, T.D. & Kowalczyk, P., "Parameter Sensitivity in Nonlinear Mechanics: Theory and Finite Element Computations", John Wiley & Sons, 1997.

[10] Duysinx, P., Zhang, W.H. & Fleury, C., "Sensitivity Analysis with Unstructured Free Mesh Generators in 2-D Shape Optimization", *The World Congress on Optimal Design of Structural Systems, "STRUCTURAL OPTIMIZATION 93"*, Rio de Janeiro, Brazil, August 2-6, 1993.

[11] Svanberg, K., "The MMA for Modeling and Solving Optimization Problems", *Third World Congress of Structural and Multidisciplinary Optimization – WCSMO3*, Niagara Falls/Amherst, New York, USA, 1999.

[12] Muñoz-Rojas, P.A., Fonseca, J.S.O. & Creus, G.J., "Preform Optimization of 2D Forged Components Using Sequential Convex Programming", submitted, 2001.

[13] Muñoz-Rojas, P.A, Fonseca, J.S.O. & Creus, G.J., "A New Approach for 2D Preform Optimization Using an Elastic-Plastic Material Model and Unstructured Mesh Generators", submitted, 2001.

[14] http://gid.cimne.upc.es

[15] Ponthot, J.P. and Hogge, M., "The Use of the Euler Lagrange Finite Element Method in Metal Forming Including Contact and Adaptive Mesh", in: Proceedings of the ASME Winter Annual Meeting, Atlanta, USA, 1991.