The LS-TaSC™ Software

TOPOLOGY AND SHAPE COMPUTATIONS USING THE LS-DYNA® SOFTWARE

THEORY MANUAL

January 2014
Version 3.0

Corporate Address
Livermore Software Technology Corporation
P. O. Box 712
Livermore, California 94551-0712

Support Addresses
Livermore Software Technology Corporation
7374 Las Positas Road
Livermore, California 94551
Tel: 925-449-2500 ♦ Fax: 925-449-2507
Email: sales@lstc.com
Website: www.lstc.com

Livermore Software Technology Corporation
1740 West Big Beaver Road
Suite 100
Troy, Michigan 48084
Tel: 248-649-4728 ♦ Fax: 248-649-6328

Disclaimer

22-Jan-14

I must say it looks a bit like science fiction to many people – Ofir Shor, June 2009, while evaluating the alpha version.

1. TOPOLOGY THEORY

1.1. Background

The traditional approach for solving topology optimization problems is based on sensitivity analysis that is inexpensive to obtain for linear-static problems. However, deriving analytical sensitivities for dynamic analysis is very difficult due to the complex interactions among material nonlinearities, geometry and mesh, and transient nature of load and boundary conditions. Numerical computation of sensitivities is also not practical due to the high computational expense. Hence the conventional sensitivity based approach of topology optimization is not practical for crashworthiness problems. To overcome the aforementioned difficulties in topology optimization, a different approach was proposed. This approach does not require gradients and hence there is no need to compute the sensitivities. In version 1, the approach was refer to as Hybrid Cellular Algorithm [1,2], but academics doing a literature review should also consult other standard views of topology optimization and our patent portfolio to understand what is currently actually implemented. With there being no cellular algorithm in the current version, the methodology is best referred to as LS-TaSC 3.0.

1.2. Implementation

The algorithm for structural optimization is shown pictorially in Figure 1-1. After defining the problem, the topology is evolved using the simple rules defined on the variables. The constraints are accommodated during the state update procedure.

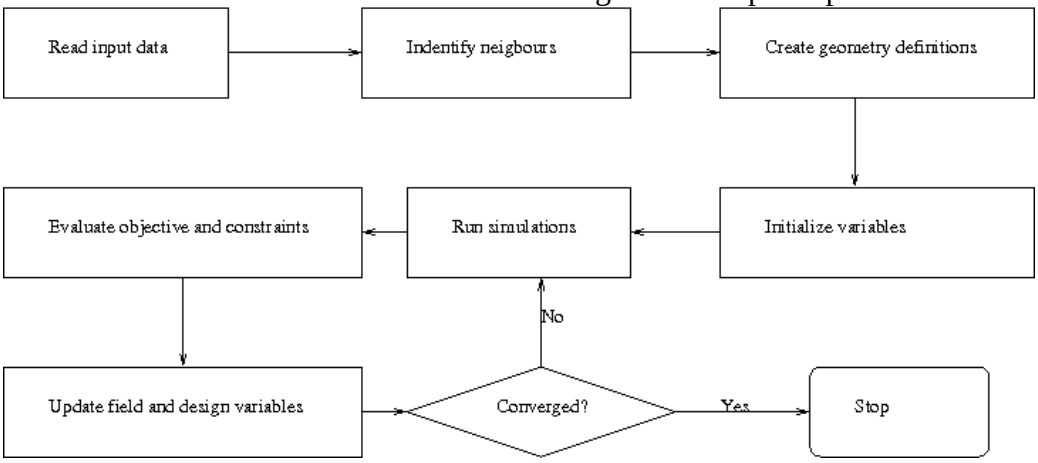

Figure 1-1: The topology optimization algorithm

1.2.1. Definition

The input data is used to identify the design domain and design material model. The input data comprises of method data e.g., number of iterations, convergence tolerance, and the problem data, e.g. load cases, design part, etc.

1.2.2. Creating the variables

The finite element model is mapped to design variables. Each design variables is assigned to a solid element in the design domain. For extrusion and symmetry constraints, the equality constraints are defined between the variables. For casting constraints, inequality constraints are established.

1.2.3. Filtering of results

Past work were based on the structured grid arrangement of cells. This assumption would breakdown for industrial applications where structured grids are not always possible. Hence, a radius based strategy is used to identify neighbors. In this strategy, a virtual sphere of user-defined radius is placed at the centroids of an element. All elements that are within this sphere are considered the neighbors of the corresponding element, and the results are averaged over the elements in the neighborhood

$$U_i = \sum_{j=1}^{n} w_j U_j \Big/ \sum_{j=1}^{n} w_j \,. \tag{2}$$

1.2.4. Material Parameterization

The material model is parameterized using a so-called *density approach*. In this approach, a design variable is directly linked to the individual material element such that each variable has its own material model. The material properties corresponding to the values of design variables are obtained using an appropriate interpolation model. The solid isotropic material with penalization (SIMP) model [6] is the most popular interpolation method. This model is power law approach that drives the intermediate material properties towards the boundaries to obtain a 0-1 topology. According to SIMP model, the material properties are defined as,

$$\rho(x) = x\rho_0, \tag{3}$$

$$E(x) = x^p E_0, \tag{4}$$

$$\sigma(x) = x^q \sigma_0, \tag{5}$$

$$E_h(x) = x^q E_{h0}, \tag{6}$$

where ρ denotes the density of the material, E represents the Young's modulus, σ is the yield stress, and E_h is the strain hardening modulus. The last two material properties represent material non-linearities and are required for dynamic problems like crash that involve material yielding. The subscript '0' refers to the base material properties. The design variable x, also known as relative density, varies from 0 to 1 where '0' indicates void and '1' represents full material. A more detailed description of the material model

parameterization, one should refer to Bendsøe and Sigmund [7], and Patel [8]. The elements with design variable value less than a user-defined minimum value are deleted to improve numerical stability. To enable the use of very large FE models, this approached was implemented using a discrete material model approach [9].

1.2.5. Design Objectives and Constraints

The typical goal of topology optimization is to obtain a structure with the best use of the material. Compliance and the strain energy density are the most commonly used objectives for linear-static problems. For dynamic problems like crashworthiness simulations, the structure needs to absorb maximum energy while maintaining the structural integrity and keeping the peak loads transmitted to the occupants low.

Following the formulation proposed by Patel [8], the goal of obtaining uniform internal energy density in the structure is defined as the objective for optimization. This concept is similar to the fully-stressed design and uniform strain energy density approaches (Haftka and Gurdal [10], Patnaik and Hopkins [11]) that are well established in literature for linear-static problems. The use of the internal energy density in optimization, its relationship with the design sensitivity information for crash problems, and its usefulness for ranking variables has been extensively studied by Öman [12,13].

The optimization problem is formulated as,

$$\min_{x} \sum_{i=}^{N} \sum_{j=1}^{L} \left(w_j U_j(x_i) - U_j^* \right), \tag{7}$$

$$\begin{aligned} subject\ to: & \sum_{i=1}^{N} \rho(x_i) V_i \le M^* \\ & C_j^l \le C_j \le C_j^u, \quad j = 1,2,...,J \\ & x_{\min} \le x_i \le 1.0. \end{aligned} \tag{8}$$

where U represents the internal energy density of the i^{th} element, V_i is the volume of i^{th} element, U^* represents internal energy density set point, and C_j is the j^{th} constraint. There are L load cases with a total of J constraints. The superscripts 'l' and 'u' represent lower and upper bounds on the constraints, respectively.

1.2.6. Design Variable Initialization

The design variables are initialized to satisfy the material constraint. All elements are assigned the same design variable values. All associated field variables are also initialized to zero.

1.2.7. Simulation to Obtain Field Variables

The elements in the finite element model are modified by changing the material models, adding or deleting elements, at each iteration. So the input deck is re-written at each iteration. This modified input deck is analyzed using LS-DYNA® [11]. One can take advantage of multiple processors using the MPP version of LS-DYNA. The relevant field

variables for all elements are obtained from the output to completely define the state of each variable. For multiple load case conditions, the state variable is based on the output from simulations of different load cases.

For dynamic problems, it was observed that accounting for the history of evolution induces stability by reducing the element deletion rate. Hence, the field variable (internal energy density) of i^{th} variable at iteration t is updated by defining a weighted sum on the field variable of three previous iterations as follows,

$$U_i^t = \sum_{j=0}^{3} (x_i)^{j+1} U_i^{t-j} \Big/ \sum_{j=0}^{3} (x_i)^{j+1}. \tag{9}$$

where x_i is the design variable associated with the i^{th} variable at iteration t. If the load cases are a mixture of dynamic and static problems, then this weighing is followed for all the load cases.

1.2.8. Global Constraint Handling

In presence of constraints other than the mass constraints, the target mass constraint is adjusted to satisfy the structural constraints. The mass target (M^*) is increased in proportion to the constraint violation for all constraints except force constraints for which the mass target is reduced.

$$\begin{aligned} M^* &= M^* + \Delta M, \\ \Delta M &= \left(\sum_j K_j^c \varepsilon_j \right) / J, \end{aligned} \tag{10}$$

where J is the total number of constraints, K_j^c is the coefficient used to scale the constraint violation of the j^{th} constraint, and ε_j is the violation of the j^{th} constraint. The total change in mass target (ΔM) is bounded to allow gradual changes in the structure.

1.2.9. Dynamic Load Case Weighing

The desired behavior is $k_1 C_1 + offset_1 = k_2 C_2 + offset$ with C the constraint value, k a scale factor, and an offset added as shown. The weight w_i of load case i is adjusted to change constraint C_i. The target value is computed as

$C_{t\arg et} = \dfrac{\sum_{i}^{i=n} (k_i C_i + offset_i)}{n}$ from which we compute $\Delta w_i = (C_{t\arg et} - k_i C_i - offset_i) / \partial C_i / \partial w_i$ with the derivative approximated as ±1 and a maximum bound is place on Δw to ensure convergence in a reasonable number of iterations.

1.2.10. State Update Rules

This is the heart of topology optimization method. In this step, the state of a variable is updated based on the state of its neighbors. The state update is carried out in two steps:

1. Field variable update: The field variable (internal energy density) of a variable is updated as accounting for the field variable values of its *n* neighbors as,

$$U_i = \sum_{j=0}^{n} U_j \Big/ \sum_{j=0}^{n} 1. \tag{10}$$

2. Variable/Material Update: Once the field-variable state of each variable is defined, the design variable is updated to reflect the changes. While numerous rules are proposed in literature [6] to update design variables, a control based rule used by Patel [8] is implemented here (Figure 1-2).

The change in the design variable of i^{th} variable (Δx_i) is computed as,

$$\Delta x_i^t = K\left(U_i^t - U^*\right)/U^*. \tag{11}$$

where K is a scaling factor and U^* denotes the internal energy density set point. The design variable is updated as,

$$x_i^{t+1} = x_i^t + \Delta x_i^t. \tag{12}$$

The change in the variable is constrained by the bounds on the value of the design variable i.e.,

I. if $x_i^{t+1} < LB$, then $x_i^{t+1} = LB$,

II. if $x_i^{t+1} > UB$, then $x_i^{t+1} = UB$,

and only certain discrete values are allowed.

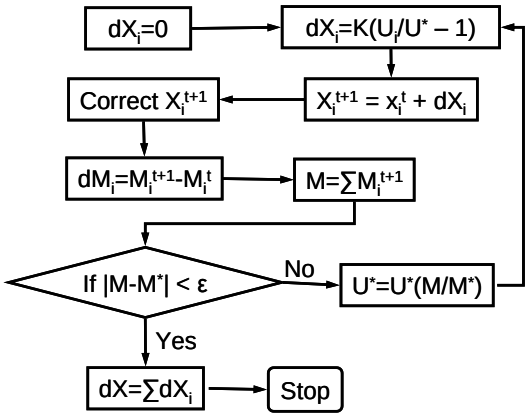

Figure 1-2: Design variable update.

The mass of each element is then calculated by using the appropriate material model associated with the design variables. If the total mass of the structure meets the constraint, the total change in design variables in this iteration is calculated, and the design variable update is considered completed. If the mass constraint is not satisfied, the IED set point is updated iteratively to accommodate the material constraint as,

$$U^* \equiv U^* = U^* M / M^*. \tag{13}$$

where M is the mass of the structure.

1.2.11. Stopping Criteria

Two termination conditions are used to stop the optimization process.

1. The number of iterations has exceeded the maximum number of iterations, or
2. The change in the topology is smaller than the tolerance, i.e.,

$$dX^t = \sum_{i=1}^{N} \Delta x_i^t \leq \varepsilon. \tag{14}$$

The numerical oscillations in convergence are limited by averaging the total change in topology over two iterations.

2. SURFACE DESIGN THEORY

2.1. Background

The traditional approach for solving shape design problems is based on sensitivity analysis that is inexpensive to obtain for linear-static problems. However, deriving analytical sensitivities for dynamic analysis is very difficult due to the complex interactions among material nonlinearities, geometry and mesh, and transient nature of load and boundary conditions. Numerical computation of sensitivities is also not practical due to the high computational expense. Hence this approach is not practical for crashworthiness problems. To overcome the aforementioned difficulties, a different approach was proposed. This approach does not require gradients and hence there is no need to compute the sensitivities. The methodology is best referred to as LS-TaSC 3.0.

2.2. Implementation

The algorithm is shown pictorially in Figure 1-1. After defining the problem, the surface shape is evolved using the simple rules defined on the variables.

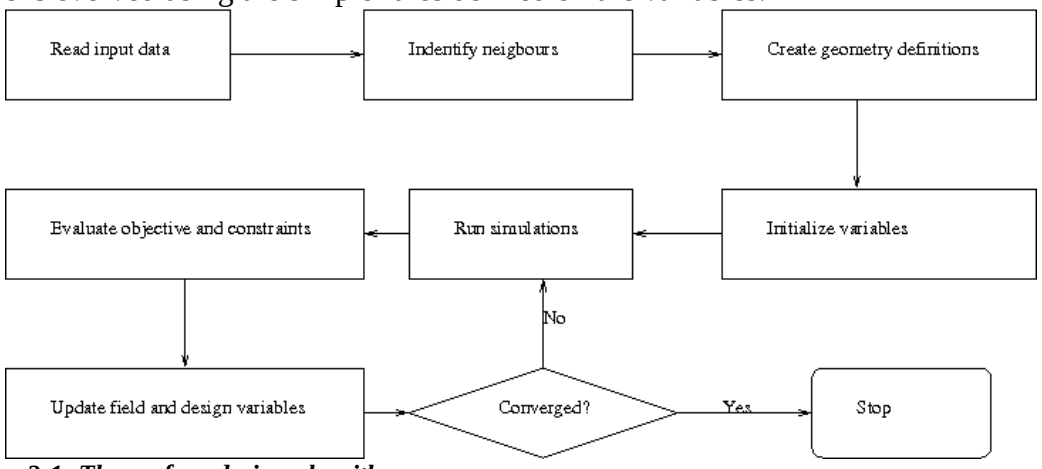

Figure 2-1: The surface design algorithm

2.2.1. Definition

The input data is used to identify the design problem. The input data comprises of method data e.g., number of iterations, convergence tolerance, and the problem data, e.g. load cases, design surface, etc.

2.2.2. Creating the variables

The discrete surface is mapped to design variables. The normal displacement to each node in the design surface assigned to a design variable. For extrusion and symmetry constraints, the equality constraints are defined between the variables.

2.2.3. Filtering of results

A radius based strategy is used to identify neighbors. In this strategy, a virtual sphere of user-defined radius is placed at the centroids of an element. All elements that are within this sphere are considered the neighbors of the corresponding element, and the results are averaged over the elements in the neighborhood

$$U_i = \sum_{j=1}^{n} w_j U_j \Big/ \sum_{j=1}^{n} w_j . \tag{2}$$

2.2.4. Design Objective

The goal of shape design is to obtain surface with a uniform stress.

The optimization problem is formulated as,

$$\min_x \sum_{i=1}^{i=n} [U(x_i) - U_{target}] \tag{7}$$

(8)

where U represents the design field (typically the von Mises stress) at the node associated with design variable x_i, and U_{target} represents the target value of the design field.

2.2.5. Target Stress

The goal of shape design is to obtain surface with a uniform stress. In order to complete this task we need to define a target stress. There are the following possibilities of selecting a target stress:

- Average over the surface
- The maximum value on the surface
- The minimum value on the surface
- A user-defined value

Using the above target stresses it should be noted that the goal becomes more subtle than obtaining the a surface with a uniform stress: if selecting the maximum is as the target stress, then the weight of the structure will be reduced; while if the minimum is selected, then the average stress is reduced.

2.2.6. Design Variable Initialization

All design variables are also initialized to zero.

2.2.7. Simulation to Obtain Field Variables

The elements in the finite element model are modified the nodal locations for all iterations. So the input deck is re-written for all iterations. The relevant field variables for all nodes are obtained from the output to completely define the state of each variable. For

multiple load case conditions, the state variable is based on the output from simulations of different load cases.

For dynamic problems, it was observed that accounting for the history of evolution induces stability by reducing the element deletion rate. Hence, the field variable (internal energy density) of i^{th} variable at iteration t is updated by defining a weighted sum on the field variable of three previous iterations as follows,

$$U_i^t = \sum_{j=0}^{3} (x_i)^{j+1} U_i^{t-j} \Big/ \sum_{j=0}^{3} (x_i)^{j+1}. \tag{9}$$

where x_i is the design variable associated with the i^{th} variable at iteration t. If the load cases are a mixture of dynamic and static problems, then this weighing is followed for all the load cases.

2.2.8. Variable Update

This is the heart of shape design method. In this step, the state of a variable is updated based on the state of its neighbors. The state update is carried out in two steps:

1. Field variable update: The field variable (internal energy density) of a variable is updated as accounting for the field variable values of its n neighbors as,

$$U_i = \sum_{j=0}^{n} U_j \Big/ \sum_{j=0}^{n} 1. \tag{10}$$

2. Variable update: Once the field-variable state of each variable is defined, the design variable is updated to reflect the changes..

The change in field value required is $\delta_i = U_i - U_{target}$. Now compute $\Delta x_i = \delta_i / \dfrac{\partial U}{\partial x}$ with x_i the required movement of node i normal to the surface.

2.2.9. Stopping Criteria

Two termination conditions are used to stop the optimization process.

1. The number of iterations has exceeded the maximum number of iterations, or
2. The change in the topology is smaller than the tolerance, i.e.,

$$dX^t = \sum_{i=1}^{N} \Delta x_i^t \le \varepsilon. \tag{14}$$

2.3. References

1. A Tovar, *Bone Remodeling as a Hybrid Cellular Automaton Optimization Process*, PhD thesis, University of Notre Dame, 2004.

2. NM Patel, B-S Kang, JE Renaud, Crashworthiness Design using a Hybrid Cellular Automata Algorithm, *In Proceedings of the 2006 International Design Engineering Technical Conference*, DETC-2006-99566, Philadelphia PA, Sep 10-13, 2006.
3. P Hajela, B Kim, On the Use of Energy Minimization of CA Based Analysis in Elasticity, *Structural and Multidisciplinary Optimization*, 23, 23-33, 2001.
4. J Forsberg, L Nilsson, Topology Optimization in Crashworthiness Design, *Structural and Multidisciplinary Optimization*, 33, 1-12, 2007.
5. http://mathworld.wolfram.com, Last accessed 23-March-2008.
6. MP Bendsøe, O Sigmund, Material Interpolation Schemes in Topology Optimization, *Archives of Applied Mechanics*, 69, 635-654, 1999.
7. MP BendsØe, O Sigmund. *Topology Optimization: Theory, Methods and Applications*, Springer-Verlag, Berlin, 1989.
8. NM Patel, *Crashworthiness Design Using Topology Optimization*, PhD thesis, University of Notre Dame, 2004.
9. Goel T, Roux WJ. Topology optimization for designing engineering product. US patent 8,126,684, filed April 10, 2009, and issued February 8, 2012.
10. RT Haftka, Z Gurdal, MP Kamat, *Elements of Structural Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2nd ed., 1990.
11. SN Patnaik, DA Hopkins, Optimality of Fully-Stressed Design, *Computer Methods in Applied Mechanics and Engineering*, 165, 215-221, 1998.
12. M Öman, *Optimization and Robustness of Structural Product Families*, PhD Thesis, Linköping University, 2011
13. M Öman and L Nilsson, Structural Optimization based on Internal Energy Distribution, *Engineering Optimization*, 45(4), 397-414, 2013.
14. JO Hallquist, *LS-DYNA Manual version 971*, Livermore Software Technology Corporation, October 2007.
15. RT Haftka, Z Gurdal, MP Kamat, *Elements of Structural Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2nd ed., 1990.
16. SN Patnaik, DA Hopkins, Optimality of Fully-Stressed Design, *Computer Methods in Applied Mechanics and Engineering*, 165, 215-221, 1998.