FROM OPTIMALITY CRITERIA TO STRUCTURAL APPROXIMATIONS

Pierre DUYSINX
LTAS – Automotive Engineering
Academic year 2019-2020

INTRODUCTION & MOTIVATION

Introduction

- Structural optimization applied to sizing (weight minimization) problem

$$\min_{x} \quad W(x) = \sum_{i=1}^{n} w_i \, x_i$$

$$\text{s.t.:} \quad g_j(x) \; \leq \; 0 \qquad (j = 1 \ldots m)$$

$$\underline{x}_i \; \leq \; x_i \; \leq \; \overline{x}_i \qquad (i = 1 \ldots n)$$

  - Finite element model
  - Design variables are the transverse sizes of the structural members (Fixed geometry and material properties)
  - Design restrictions

$$g_j(x) \;\equiv\; \begin{cases} u_j(x) - \overline{u}_j \;\leq\; 0 \\ \sigma_j(x) - \overline{\sigma}_j \;\leq\; 0 \\ \underline{\omega}_j^2 - \omega_j^2(x) \;\leq\; 0 \\ \underline{\lambda}_j^2 - \lambda_j^2(x) \;\leq\; 0 \end{cases}$$

Introduction

- Design constraints $g_j(x)<0$
  - Implicit functions
  - Non linear functions
  - One constraint evaluation requires a complete FE analysis

- Side constraints: simple and explicit
  - Fabrication / technological / physical constraints
  - Treated separately in most methods

- Iterative process → HIGH COST

INTRODUCTION

- Optimality criteria techniques (OC)
  - Highly specific
  - Intuitive techniques, simple
  - Convergence to a design that is not necessarily optimal (KKT conditions)
  - Difficulties in identifying the set of active constraints
  - Convergence instabilities
  - Small number of reanalyses, independent of the number of design variables

Résumé
  - Low cost
  - But uncertainty convergence

INTRODUCTION

- Pure Mathematical Programming methods
  - Very general
  - Rigorous methods, quite elaborated
  - Convergence to a local minimum
  - Stable and monotonic convergence
  - Large number of reanalyses, growing with the number of design variables

Résumé
  - Rigorous framework & guaranteed convergence
  - High cost (Growing with the size of the problem)

BERKE’S APPROXIMATION

FINITE ELEMENT DISCRETIZATION

- Let’s denote by
  - q the vector of nodal displacements
  - g the vector of nodal forces
  - K the matrix of finite element stiffnesses

- The total potential energy of the structure is:

$$\Pi = U(\mathbf{q}) - P(\mathbf{q}) = \frac{1}{2}\mathbf{q}^T\,\mathbf{K}\,\mathbf{q} \;-\; \mathbf{g}^T\,\mathbf{q}$$

- The principle of the minimum total potential energy yields the equilibrium equation

$$\boxed{\mathbf{K}\,\mathbf{q} \;=\; \mathbf{g}}$$

BERKE'S APPROXIMATION

- The virtual work theorem states that the equality holds for any kinematically admissible virtual fields

$$\delta \mathbf{q}^T \{\mathbf{K}\,\mathbf{q}\} = \delta \mathbf{q}^T \mathbf{g}$$

- If we consider the compatible displacement in equilibrium with any virtual load vector:

$$\mathbf{K}\,\tilde{\mathbf{q}} = \tilde{\mathbf{g}}$$

- The theorem of virtual work leads to:

$$T.V. = \tilde{\mathbf{q}}^T\,\mathbf{g} = \mathbf{q}^T\,\tilde{\mathbf{g}} = \mathbf{q}^T\,\mathbf{K}\,\tilde{\mathbf{q}}$$

BERKE'S APPROXIMATION

- Use the virtual work,

$$T.V. = \tilde{\mathbf{q}}^T \, \mathbf{g} = \mathbf{q}^T \, \tilde{\mathbf{g}} = \tilde{\mathbf{q}}^T \, \mathbf{K} \, \tilde{\mathbf{q}}$$

- By choosing a smart virtual displacement / vector field for instance, if the virtual load vector is chosen as a unit load vector under the displacement u that one wants to determine,

$$\tilde{\mathbf{g}} = (0\,0\,0 \ldots 1 \ldots 0\,0)^T$$

- One gets

$$T.V. = \mathbf{q}^T \, \tilde{\mathbf{g}} \, = \, u \, \times \, 1 \, = \, \mathbf{q}^T \, \mathbf{K} \, \tilde{\mathbf{q}}$$

- With

$$\mathbf{K} \, = \, \sum_i \mathbf{L}_i^T \mathbf{K}_i \, \mathbf{L}_i$$

BERKE'S APPROXIMATION

- For truss and plate design variables, the stiffness matrix takes the interesting form:

$$\boldsymbol{K}_e = x_e\,\bar{\boldsymbol{K}}_e$$

  - Truss structures $x_e = A_e$
  - Plate structures $x_e = t_e$

- One can decompose the contribution of each element:

$$\begin{aligned} u &= \boldsymbol{q}^T\,\boldsymbol{K}\,\tilde{\boldsymbol{q}} = \sum_e \boldsymbol{q}_e^T \boldsymbol{K}_e\,\tilde{\boldsymbol{q}}_e \\ &= \sum_e (x_e\,\boldsymbol{q}_e^T \bar{\boldsymbol{K}}\,\tilde{\boldsymbol{q}}_e) \end{aligned}$$

BERKE'S APPROXIMATION

- For statically determinate structures, internal loads remain constant. Therefore, increasing the cross section of the bars, will scale down the displacement accordingly.

$$\boldsymbol{K}_e = x_e \, \bar{\boldsymbol{K}}_e \qquad \boldsymbol{q}_e \div \frac{1}{x_e}$$

$$\tilde{\boldsymbol{q}}_e \div \frac{1}{x_e}$$

- So the following expression is constant for statically determinate structures

$$\boxed{\boldsymbol{q}_e^T \boldsymbol{K}_e \, \tilde{\boldsymbol{q}}_e \, x_e = c_e}$$

- For other structures, one can also assume a moderate redistribution of the internal loads around the current design point and has also constant value.

BERKE'S APPROXIMATION

- Considering that the coefficients c_e are constant, Berke's criterion provides an explicit expression of the displacement u in terms of the design variables

$$\tilde{u} = \sum_e \frac{c_e^0}{x_e} \quad \text{with} \quad c_e^0 = (\boldsymbol{q}_e^{0T}\, \boldsymbol{K}_e^0\, \tilde{\boldsymbol{q}}_e^0)\, x_e^0$$

BERKE’S APPROXIMATION ARE FIRST ORDER APPROXIMATIONS

A first order explicit approximation of displacement

- The Berke's expansion provide a first order explicit approximation of the displacement around x^0

$$\tilde{u} = \sum_i \frac{c_i^0}{x_i} \quad \text{with} \quad c_i^0 = (\boldsymbol{q}_i^{0T} \boldsymbol{K}_i^0 \tilde{\boldsymbol{q}}_i^0)\, x_i^0$$

  - In general (indeterminate structures) the c_i^0 are not constant
  - The expression is exact for statically determinate structures, but for statically indeterminate structures, it is only exact in the current point x^0

- The Berke's expression is an approximation of the displacement u around the current design point x^0 in terms of the design variables.

A first order explicit approximation of displacement

- The value of the approximation is exact in x_i^0

$$\tilde{u}(\boldsymbol{x}_i^0) = \sum_i \frac{c_i^0}{x_i^0} = \sum_i \frac{(\boldsymbol{q}_i^{0T}\,\boldsymbol{K}_i^0\,\tilde{\boldsymbol{q}}_i^0)\,x_i^0}{x_i^0} = \sum_i (\boldsymbol{q}_i^{0T}\,\boldsymbol{K}_i^0\,\tilde{\boldsymbol{q}}_i^0) = u^0$$

  - As c_i^0 remains constant only along $D(x^0)$. It is also true for all points along the scaling line

- The derivatives of the approximations are exact in x_i^0

$$\left.\frac{\partial \tilde{u}}{\partial x_i}\right|_{\boldsymbol{x}^0} = \left.\frac{\partial u}{\partial x_i}\right|_{\boldsymbol{x}^0}$$

A first order explicit approximation of displacement

- The derivatives of the approximation in x^0:

$$\left.\frac{\partial \tilde{u}}{\partial x_i}\right|_{\boldsymbol{x}^0} = -\frac{c_i^0}{(x_i^0)^2}$$

- If one remembers the definition of the mutual energy coefficients

$$c_i^0 = x_i^0 \, \boldsymbol{q}_i^{0T} \boldsymbol{K}_i \, \tilde{\boldsymbol{q}}_i^0$$

- It comes

$$\left.\frac{\partial \tilde{u}}{\partial x_i}\right|_{\boldsymbol{x}^0} = -\frac{\boldsymbol{q}_i^{0T} \boldsymbol{K}_i \, \tilde{\boldsymbol{q}}_i^0}{(x_i^0)}$$

A first order explicit approximation of displacement

- Let's calculate the true derivative of the displacement using sensitivity analysis.
- The displacement *u* can be expressed as follows :

$$u = \boldsymbol{b}^T \boldsymbol{q}$$

- Expression of sensitivity of u:

$$\frac{\partial u}{\partial x_e} = \boldsymbol{b}^T \frac{\partial \boldsymbol{q}}{\partial x_e}$$

- One knows the expression of the derivatives of the generalized displacements Kq=g

$$\frac{\partial \boldsymbol{q}}{\partial x_e} = \boldsymbol{K}^{-1} \{ \frac{\partial \boldsymbol{g}}{\partial x_e} - \frac{\partial \boldsymbol{K}}{\partial x_e} \boldsymbol{q} \}$$

A first order explicit approximation of displacement

- For dead load,

$$\frac{\partial u}{\partial x_e} = -\boldsymbol{b}^T \boldsymbol{K}^{-1} \frac{\partial \boldsymbol{K}}{\partial x_e} \boldsymbol{q} \qquad \frac{\partial \boldsymbol{g}}{\partial x_e} = 0$$

- Direct approach

$$\frac{\partial \boldsymbol{q}}{\partial x_e} = -\boldsymbol{K}^{-1} \frac{\partial \boldsymbol{K}}{\partial x_e} \boldsymbol{q}$$

- Adjoin approach

$$\tilde{q} = \boldsymbol{K}^{-1} b$$

  - Which is exactly the same as the unit load approach

$$\tilde{g} = \boldsymbol{b} \qquad \boldsymbol{K}\tilde{q} = \tilde{g} = \boldsymbol{b}$$

A first order explicit approximation of displacement

- The key point lies in the fact one can identify the adjoin load vector and the virtual load vector used in the Virtual Work Principle considered in Berke’s criterion

$$\boxed{\tilde{g} = b}$$

- It comes:

$$K\tilde{q} = \tilde{g} = b = K\Lambda$$

$$\tilde{q} = \Lambda$$

A first order explicit approximation of displacement

- Since $$\frac{\partial \boldsymbol{K}}{\partial x_e} = \bar{\boldsymbol{K}}_e = \frac{\boldsymbol{K}_e}{x_e}$$

- One obtains

$$\frac{\partial u}{\partial x_e} = -\boldsymbol{b}^T \boldsymbol{K}^{-1} \bar{\boldsymbol{K}}_e \boldsymbol{q} = -\tilde{\boldsymbol{g}}^T \boldsymbol{K}^{-1} \bar{\boldsymbol{K}}_e \boldsymbol{q} = -\tilde{\boldsymbol{q}}^T \bar{\boldsymbol{K}}_e \boldsymbol{q}$$

$$= -\frac{(x_e \tilde{\boldsymbol{q}}^T \boldsymbol{K}_e \boldsymbol{q})}{x_e^2} = -\frac{c_e}{x_e^2}$$

- Expressing **x**°

$$\left.\frac{\partial u}{\partial x_e}\right|_{x_e^0} = -\frac{c_e^0}{x_e^{0\,2}}$$

- Which proves again the fact the Berke's criterion is a first order approximation of the displacement *u*.

APPROXIMATION IS A FIRST ORDER TAYLOR EXPANSION IN THE RECIPROCAL VARIABLE SPACE

Constraint linearization in reciprocal space

- We have shown that Berke's criterion is a first order approximation of the displacement *u*.

$$\tilde{u} = \sum_i \frac{c_i^0}{x_i} \quad \text{with} \quad c_i^0 = (\boldsymbol{q}_i^{0T} \, \boldsymbol{K}_i^0 \, \tilde{\boldsymbol{q}}_i^0) \, x_i^0$$

- Furthermore, the expression of Berke's criterion suggests that it is a first order Taylor expansion in terms of the reciprocal variables

$$z_i = \frac{1}{x_i}$$

- The Berker's approximation writes

$$\tilde{u}_j(\boldsymbol{z}) = \sum_i c_{ij} \, z_i$$

Constraint linearization in reciprocal space

- Let's show that the virtual energy densities c_i are the first derivatives (gradients) of the constraints with respect to the reciprocal variables $z_i = 1/x_i$ that is

$$c_{ij} = \left.\frac{\partial u_j}{\partial z_i}\right|_{z^0} = -\frac{c_{ij}}{x_i^2}$$

- At first, the derivatives of the approximation is obviously :

$$\left.\frac{\partial \tilde{u}}{\partial z_i}\right|_{z^0} = c_i = x_i^0 \, \boldsymbol{q}_i^{0T} \boldsymbol{K}_i \, \tilde{\boldsymbol{q}}_i^0$$

Constraint linearization in reciprocal space

- Let's now compute the derivative of the (real) displacement with respect to the reciprocal design variable $z_i = 1/x_i$. It is clear that

$$\frac{\partial u}{\partial z_i} = \frac{\partial u}{\partial x_i} \frac{\partial x_i}{\partial z_i}$$

- Since

$$\frac{\partial x_i}{\partial z_i} = \frac{\partial}{\partial z_i}\left(\frac{1}{z_i}\right) = -\frac{1}{z_i^2} = -x_i^2$$

- It comes

$$\begin{aligned} \left.\frac{\partial u}{\partial z_i}\right|_{\boldsymbol{z}^0} &= \left.\frac{\partial u}{\partial x_i}\right|_{\boldsymbol{x}^0} (-x_i^{0\,2}) \\ &= -\frac{\boldsymbol{q}_i^{0T} \boldsymbol{K}_i \tilde{\boldsymbol{q}}_i^0}{x_i^0} (-x_i^{0\,2}) = \boldsymbol{q}_i^{0T} \boldsymbol{K}_i \tilde{\boldsymbol{q}}_i^0 \, x_i^0 = c_i^0 \end{aligned}$$

Constraint linearization in reciprocal space

- We now show that the generalized optimality criterion using first order approximation of the displacements are achieved by linearizing the constraints in the reciprocal space.

$$c_j(\boldsymbol{z}) \;=\; c_j(\boldsymbol{z}^0) + \sum_i \left(\frac{\partial c_j}{\partial z_i}\right)_0 (z_i - z_i^0) \;\leq\; \bar{c}_j$$

- Applied to the displacement constraint

$$u(\boldsymbol{z}) \;=\; u(\boldsymbol{z}^0) + \sum_i c_i \,(z_i - z_i^0) \;\leq\; \bar{u}$$

$$u(\boldsymbol{z}) \;=\; [u_j(\boldsymbol{z}^0) - \sum_i c_i z_i^0] + \sum_i c_i \, z_i \;=\; \sum_i c_i \, z_i$$

Constraint linearization in reciprocal space

- We have to prove

$$u(\boldsymbol{z}^0) = \sum_i c_i \, z_i^0$$

- This is performed by writing the expression of u($\mathbf{z}_0$)

$$u(\boldsymbol{z}^0) = u(\boldsymbol{x}^0) = \sum_i \boldsymbol{q}_i^{0T} \boldsymbol{K}_i \, \tilde{\boldsymbol{q}}_i^0 = \sum_i \boldsymbol{q}_i^{0T} \boldsymbol{K}_i \, \tilde{\boldsymbol{q}}_i^0 \frac{x_i^0}{x_i^0} = \sum_i \frac{c_i^0}{x_i^0} = \sum_i c_i^0 z_i^0$$

- It comes that the approximation writes

$$\tilde{u} = \sum_i c_i \, z_i = \sum_i \frac{c_i}{x_i}$$

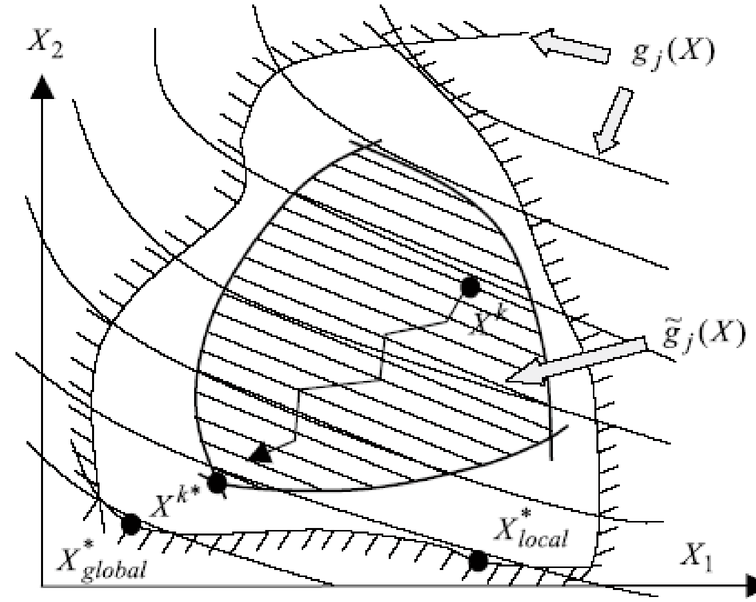
CONCLUSION

CONCLUSION

- Berke's approximation has been successful in providing high quality explicit approximations of displacement constraints
- Optimality Criteria can reduce substantially the number of function evaluation in solving costly problems in truss sizing
- They are used in building fast solution algorithms
- Berke's approximations are first order Taylor expansion of the displacement in terms of the reciprocal design variables
- How can we extend the principle to other engineering design problems?
  - Answer: Structural Approximations

SEQUENTIAL CONVEX PROGRAMMING APPROACH

Direct solution of the original optimisation problem which is generally non-linear, implicit in the design variables



$$\begin{aligned} \min_{\boldsymbol{x}} \quad & g_0(\boldsymbol{x}) \\ \text{s.t.} \quad & g_j(\boldsymbol{x}) \leq \bar{g}_j \qquad j = 1 \ldots m \end{aligned}$$

is replaced by a sequence of optimisation sub-problems

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \tilde{g}_0(\boldsymbol{x}) \\ \text{s.t.} \quad & \tilde{g}_j(\boldsymbol{x}) \leq \bar{g}_j \qquad j = 1 \ldots m \end{aligned}$$

by using approximations of the responses and using powerful mathematical programming algorithms

SEQUENTIAL CONVEX PROGRAMMING APPROACH

- Two basic concepts:
  - Structural approximations replace the implicit problem by an explicit optimisation sub-problem using convex, separable, conservative approximations; e.g. CONLIN, MMA
  - Solution of the convex sub-problems: efficient solution using dual methods algorithms or SQP method.
- Advantages of SCP:
  - Optimised design reached in a reduced number of iterations: 10 to 20 F.E. analyses
  - Efficiency, robustness, generality, and flexibility, small computation time
  - Large scale problems in terms of number of design constraints and variables