Education of Engineers in Design Improvement Methods

Josef A. Tomas[1], Ryan Adams[2]

[1]Professor (Adjunct), [2]Postgraduate Student,

School of Aerospace, Mechanical & Manufacturing Engineering, RMIT University, Melbourne, Australia

ABSTRACT

Behind the history of objects invented and produced by man, is a history of incremental design improvements, aiming (for example) at lighter, more durable components, or at stiffer or softer structures, or at resonating objects at pleasant tones, and so on.

Today, where powerful computers and complex computer software of finite-element methods (FEM) are available for structural analysis, the usual process of design improvement is still based upon the results from the analyses, even if FEM can be extended to the synthesis of components and structures.

The aim of the course in *Design Improvement* offered at RMIT Melbourne is to educate and train students in computational methods of design synthesis. The course is offered to students in the sixth semester, following the introduction of linear FEM in semester five.

The prerequisite of the course is a sound understanding of FEM concepts taught in the introductory FEM course. The theoretical background of mathematical methods of optimization is reviewed first, and then exercised on selected applications. This is followed by learning the concept of design sensitivity. Finally, the problem of “reconciling” the discreteness of finite-element mesh with the smoothness of a CAD design is presented. A series of homework and exercises serves for gaining experience with selecting the most appropriate method for the solution of a given problem.

Each lesson is accompanied with solved examples and tutorial problems. Students are required to validate the solution by any alternative method available to them. At the end of the course, students solve one larger practical problem of their own choice, to demonstrate the level of their competence.

The software ReSHAPE, which has implemented most of the available methods for design improvement is used, along with the pre- and post-processors ANSA, HyperMesh and Patran.

keywords: education in FEM, shape optimization, sensitivity, design improvement

1. INTRODUCTION

At present, the most common computational analysis technique in engineering and sciences is the finite-element method. Even in its simplest linear implicit form, it has become a practical and reliable tool for calculation of deformations, stresses, natural frequencies, buckling critical forces, and so on. Based on the results from these calculations, designers decide on design modifications for improved performance. The decision is based on the designers’ knowledge, experience and more often than not on their “engineering feel”.

On the other hand, mathematical methods of optimisation, allowing the calculation of the best configuration of variables for prescribed objectives with constraints, have been known for a long time and have been applied successfully not only sciences and engineering, but also in economy, transportation, and other “less” scientific fields of general knowledge. The difficulties with their application to the solution of finite-element models are caused by the discreetness of the models, which consist of finite number of “communication points”, called nodes, and of continua between them, called elements, with internal displacement being approximated by displacement fields, defined by nodal displacements as parameters.

Surely, a given finite-element model can be improved by repeated analyses (direct search, random walk, steep descent, etc.), when design parameters are iteratively changed by use of a selected strategy until a satisfactorily improved model is obtained (see e.g. [1]). Each iterative step requires n+1 analysis, if the number of parameters is n, therefore these methods are applicable to the solution of described by a very limited number of parameters.

In the *Design Improvement* course, the students start with learning and understanding the theoretical background of selected mathematical methods of optimisation. They then acquire basic skills in using the software ReSHAPE [2], which has implemented most of available methods. Each lesson is accompanied by solved examples and by tutorial problems [3]. The solutions of the tutorial problems can be obtained after demonstrating that the solution has been attempted by the student. Finally, each student solves more complex problems of their own choice, and presents the results in a prescribed format of a publication paper.

Students are required to validate numerical solutions of each problem by any alternative method available to them, be it hand calculations of a simplified model, solution of a similar problem, or the best of all, an experiment, before attempting any improvement. It is really embarrassing trying to “improve” something which is not correct on the on-set.

Also, students are reminded to keep in mind that this is an engineering course in design improvement, not a mathematical course in optimisation. Surely, the numerical methods used in ReSHAPE are conforming to mathematical theories. In practice, however, the model, numerical methods and subsequently, the solution can only be approximations. In many cases there is no optimum design at all, only an improvement of the existing design.

2. ANALYTICAL METHODS OF OPTIMIZATION (e.g. [4])

In the first lecture, the analytical method of determining the minimum or maximum of a function $z = f(x,y)$ of two variables x and y, with a constraint $g(x,y) = 0$ is reviewed. The method of Lagrange multipliers is revised and a new compound function is created as

$$\Phi = f(x,y) + \lambda \cdot g(x,y),$$

and then differentiated with respect to x, y, and λ. The solution of the equations

$$\partial\Phi(x,y)/\partial x = 0, \quad \partial\Phi(x,y)/\partial y = 0, \quad \partial\Phi(x,y)/\partial\lambda = 0$$

delivers the unknowns x, y and λ for the constrained minimum or maximum.

The method is demonstrated on the problem of a rectangle of height x and width y, where the objective is to find a rectangle of maximum area $A = xy$, with unchanged perimeter $p = 2(x+y)$. The compound function is $\Phi = xy + \lambda(p - 2(x+y))$. Then the solution results from
$\partial\Phi/\partial x \equiv y - 2\lambda = 0$, $\partial\Phi/\partial y \equiv x - 2\lambda = 0$, and $\partial\Phi/\partial\lambda \equiv p - 2(x+y) = 0$.
Solving for x, y and λ gives the solution $x = y = p/4$, and $\lambda = p/8$.

Subsequently, the students solve the problem graphically, for p=40 mm. The solution is shown in Figure 1, with curves A=const. and the straight line of the constraint $p = 2(x+y)$ are depicted. The solution is clearly at the point (10,10) with maximum area of 100 mm^2.

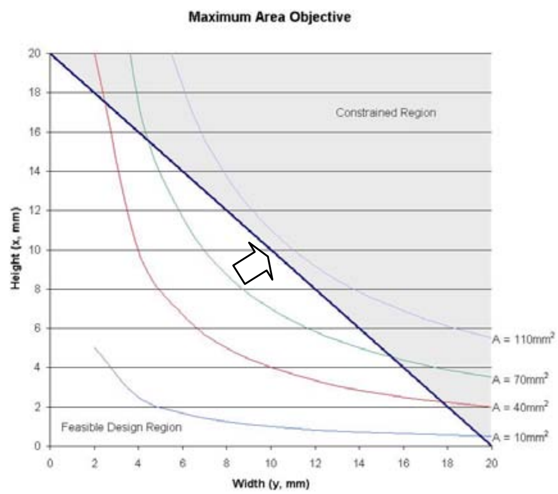

Figure 1: Solving the problem of max(A=xy) with a constraint p=40mm, graphically.

As homework, students are asked to maximize the area of an isosceles triangle with its base of *2x* and its height of *y*, keeping the circumference unchanged. They are made aware of an important fact: the constraint is now a curve, not a straight line.

3. NUMERICAL METHODS OF OPTIMISATION

3.1. METHODS OF STEEPEST ASCENT WITHOUT CONSTRAINTS

It becomes clear that more complex problems of optimization (even in two variables) cannot be solved analytically. The obvious choice is one of the numerical iterative techniques. The method of steepest ascent is one of such methods: it approximates the continuous gradient curve by a succession of straight lines.

Students solve the previous two problems with this method, again using Excel (with function formulae and iterative procedure), or in MATLAB. The result is shown in Figure 2.

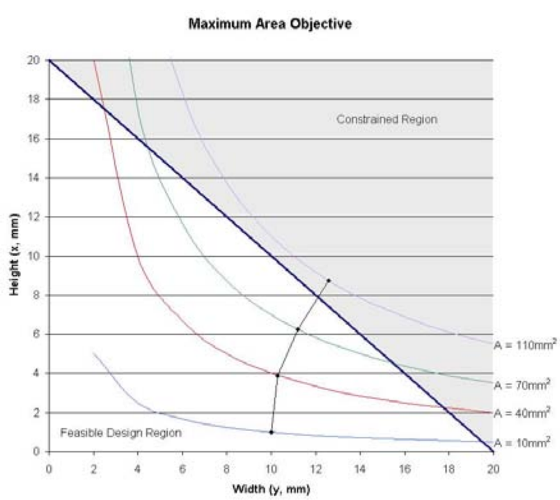

Figure 2: Contours with lines of steepest descent beginning from the point (10,1)

3.2. METHODS OF STEEPEST ASCENT WITH CONSTRAINTS

After analyzing previous results (shown in Figure 2), students are made aware of problems related to the continuation of the iterative process: that some action has to be taken after the constraint has been crossed, with necessary sub-iterations to "land" on the constraint; that next steps have to follow the constraint curve; and all that with some prescribed precision.

As the solution, the method of projected gradient is then presented, using the Gram-Schmidt ortho-normalization method. The objective function in n dimensions is written as $y=f(\mathbf{x})$ and the constraint as $c=g(\mathbf{x})$, where $\mathbf{x}$ *is an n*-dimensional vector of unknown variables.

The steepest descents of the objective function and the constraint are defined as row vectors

$$\partial y/\partial \mathbf{x} \quad \text{and} \quad \partial g/\partial \mathbf{x}.$$

The new direction of the objective function is calculated as follows:

$$(\partial y/\partial \mathbf{x})_{new} = (\partial y/\partial \mathbf{x})_{old} + \lambda .\ \partial g/\partial \mathbf{x}\ .$$

The unknown multiplier λ is calculated from the requirement that the new direction must be orthogonal to the direction $\partial g/\partial \mathbf{x}$ i.e. $(\partial g/\partial \mathbf{x})\cdot(\partial y/\partial \mathbf{x})^T_{new} = 0$. The parameter λ is then calculated from

$$\lambda = -\ (\partial y/\partial \mathbf{x})_{old} \cdot (\partial g/\partial \mathbf{x})^T / (\partial g/\partial \mathbf{x}) \cdot (\partial g/\partial \mathbf{x})^T.$$

Then the new direction $(\partial y/\partial \mathbf{x})_{new}$ can be calculated.

In the above example, when the constraint is reached, $(\partial y/\partial \mathbf{x})_{old}$ = [0.35, 0.94], and $(\partial g/\partial \mathbf{x})^T$ = [1, 1]. Then $\lambda = -\ 0.645$, and $(\partial y/\partial \mathbf{x})_{new}$ = [-0.295, 0.295], i.e. along the straight line of the constraint towards the maximum.

Satisfying multiple constraints during the iterative process is a complicated problem. Firstly, some detection procedure has to be designed, which could decide whether a constraint should be considered as active or not. This would require certain ε-band to be defined around the constraint. The constraints outside the ε-band are ignored, but those inside are activated and the objective is projected on the hyper-tangent to the hyper-line of the intersection of constraint hyper-surfaces.

As homework, students are asked to solve the previously solved problem of the isosceles triangle, simulating the whole process by using Excel or MATLAB.

3.2. DIRECT SEARCH METHODS

A brief review is presented about the direct search methods, such as the piece-wise simplex method, the complex method, the random search and similar methods. It is emphasized that these methods can be used for any problem, linear or non-linear, but at the price of extreme computing costs because for an n-dimensional problem, n+1 analyses are required.

4. FINITE-ELEMENT METHODS IN OPTIMISATION

4.1. NODAL (RAW) SENSITIVITY

In the previous chapter, the gradient vector has been calculated either analytically or numerically by repeated re-analyses of the perturbed model. This is hardly efficient for large finite-element models. If possible, the standard implicit linear finite element method should be employed for design improvement. It is then natural to ask the question: how much will the value of a quantity (objective) change if one co-ordinate of a node is changed.

Let us define such a change of the response *r* as the nodal sensitivity as a vector

$$\boldsymbol{s} = dr/d\boldsymbol{x},$$

where the column vector $\boldsymbol{x}$ contains co-ordinates of all nodes participating in the design change, thus creating a design domain. If there is a design constraint to be satisfied, the sensitivity of the constraint response is calculated in a similar way. In ReSHAPE, the response sensitivities are calculated analytically.

Students get acquainted with ReSHAPE by solving the previous "rectangle" problem modeling it in finite elements. Figure 3 shows the rectangle of dimensions 100x10 mm meshed with one plate element 1 and four bar element 2 to 5. It is required to maximize the area while keeping the perimeter constant.

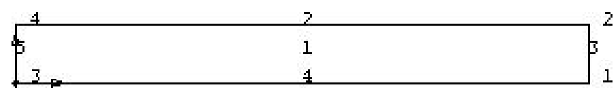

Figure 3: A rectangle meshed with one plate and four bar elements

The model has been meshed in a pre-processing program and exported as a NASTRAN input file.

The objective of maximum area and the constraint of constant perimeter are defined by the following command file:

```
reshape(improvement)
control
   steps(n=10,s=1)
responses()
   area(target=max,tol=0,stype=2d)=elem(all)
   length(bound=hold,tol=0,stype=1d)=elem(all)
process(raw)
    domain(xy)=elem(1thru5)
    locked(x)=node(3,4)
    locked(y)=node(1,3)
end
```

Most of the command is self-explanatory. The parameter *stype* allows using elements of different types in the domain, but selecting the appropriate types for objective and constraint without adding (in common nodes) sensitivity contributions from different elements.

The optimum configuration achieved by a number of iterations is shown in Figure 4.

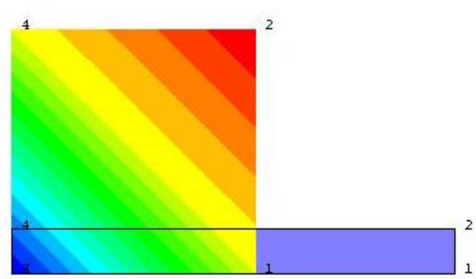

Figure 4: A rectangle with maximum area and unchanged perimeter

As an important exercise, students are asked to calculate and record individual sensitivities, i.e. first for the change of the area of the plate element, and then for the change of the perimeter using the beam elements, and use them for hand calculation of the constrained sensitivity by projected gradient method. The result is compared with the constrained sensitivity as obtained from the program. In this way, the students see that the projected gradient method used in ReSHAPE for models of any size is the same as the one used in hand calculation of the problem with two variables.

As home work, students solve the problem of the isosceles triangle, and also the problem of a quarter of a circular ring using local cylindrical co-ordinates.

4.2. APPLICATION OF NODAL (RAW) SENSITIVITY TO TRUSS STRUCTURES

In the previous lesson, the use of the raw improvement process was successful only because there were no internal nodes in the model. If the mesh of the rectangle contained internal elements and nodes, only the outside layer of plate elements would change, which would stop the iterative process long before the maximum were reached.

For truss and beam, elements are connected by isolated nodes. Therefore the use of raw improvement process can be successfully applied for shape improvement, as demonstrated in the following example.

A simple two-dimensional truss structure is shown in Figure 5. It is loaded by a concentrated load in node 3 and fixed in nodes 1 and 5. The NASTRAN input file for the example has been created by a preprocessor (Medina, Hypermesh, Ansa and Patran are generally used by students).

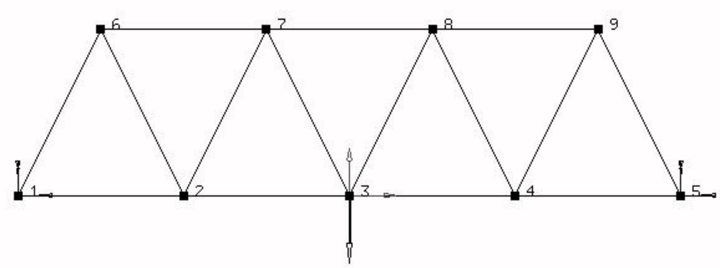

Figure 5: The truss structure with applied loading

The objective is to reduce the vertical displacement in node 3 by 25% (OBJECTIVE) but keep the mass of the structure unchanged (CONSTRAINT).

The RESHAPE command file for the iterative design improvement reads:

```
reshape(improvement)
control
   steps(n=50,s=2)
responses()
   displacement(var=ty,target=-25%)=node(3)
   volume(bound=hold)=elem(all)
process(raw)
   domain(xy)=elem(all)
   locked(y)=node(2THRU4)
   locked(xy)=node(1,5)
end
```

The improved structure is shown in Figure 6. The displacement was reduced from 2.2 mm to 1.65 mm i.e., reduced by exactly 25%. The mass remained unchanged.

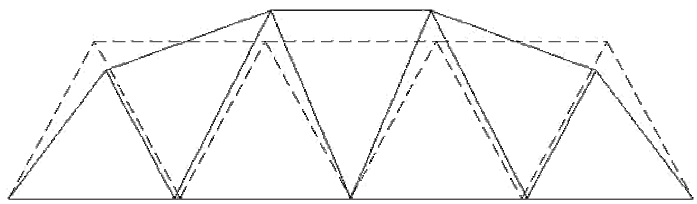

Figure 6: The improved truss structure

As homework, students solve a truss structure of their own design and exercise various combinations of objectives and constraints, including the use of more than one response for objectives or constraints.

Possible Objectives	**Possible Constraints**
Minimize volume	Displacement
Maximize 2nd modal frequency	1st modal frequency
Minimize reaction force in ‘x’ direction	
Minimize axial force in a member	
Two frequencies to prescribed values	Constant volume

5. HOW TO RESHAPE FINITE-ELEMENT MODELS WITH CONTINUOUS ELEMENTS

5.1 NODAL (RAW) SENSITIVITY

The unknown variables calculated in ReSHAPE are nodal sensitivities i.e., the sensitivities of selected responses to the change of nodal co-ordinates. If r is the scalar value of a response (e.g. a component of a nodal displacement, a component of the stress in an element, a natural frequency etc.), and $\boldsymbol{x}$ is the vector of nodal co-ordinates, the nodal sensitivity **s** is defined as a row vector:

$$\boldsymbol{s} = dr/d\boldsymbol{x} \; .$$

The nodal (‘raw’) sensitivity is calculated in ReSHAPE in all selected nodes analytically. This is, however, of little use for a detailed re-shaping of a component because the original smoothness of the model cannot be preserved (Figure 7).

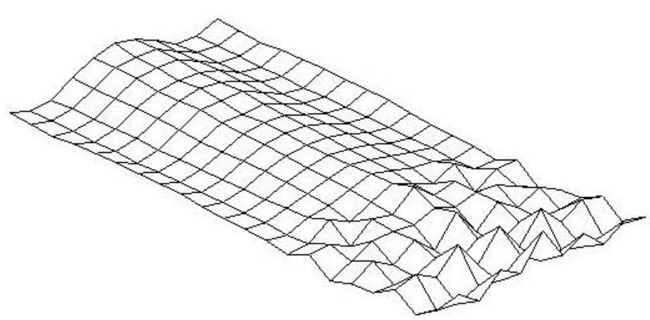

Figure 7: PROCESS(RAW) shape change for increased bending stiffness

After fifty iterations, the maximum displacement was reduced substantially, because a double-corrugated plate is indeed the stiffest structure possible. The “improved” plate, however, is not conserving any smoothness, because the raw process treats every node individually, not as belonging to a smooth surface.

To overcome this problem, generalized co-ordinates $\boldsymbol{q}$ are introduced and the relationship

$$\mathbf{x} = \mathbf{x}(\mathbf{q})$$

called *mesh-geometry associativity,* is formed.

This relationship can take many forms, with generalized co-ordinates such as:

- Modal vectors ($\boldsymbol{q}$ being the amplitudes of individual modes),
- Displacement vectors ($\boldsymbol{q}$ is the scaling factor of the applied virtual load),
- Splines ($\boldsymbol{q}$ being the spline parameters), or
- Influence functions (with $\boldsymbol{q}$ as generalized co-ordinates, calculated in user defined control points).

Underlying the whole concept of smoothing is the idea of creating a best fit to nodal changes $\Delta\boldsymbol{x}$. Without smoothing, this change would be proportional to the sensitivities $dr/d\boldsymbol{x}$. To achieve a smooth change**,** the nodal sensitivities must be first recalculated as the sensitivities $dr/d\boldsymbol{q}$ of generalized co-ordinates $\boldsymbol{q}$:

$$dr/d\boldsymbol{q} = dr/d\boldsymbol{x}\cdot d\boldsymbol{x}/d\boldsymbol{q}\ ,$$

where the matrix $d\boldsymbol{x}/d\boldsymbol{q}$ is called *influence matrix*, expressing the effect of a change in $\boldsymbol{q}$ on the change in $\boldsymbol{x}$.

The change of design variables $\Delta\boldsymbol{q}$ by steepest descent is then calculated as:

$$\Delta\boldsymbol{q} = \alpha\cdot(dr/d\boldsymbol{q})^{T}\cdot\|(dr/d\boldsymbol{q})\|^{-1},$$

where $||dr/d\boldsymbol{q}||$ indicates the norm of the vector $dr/d\boldsymbol{q}$, and α *is* the iteration step defined by the user. The mesh change $\Delta\boldsymbol{x}$ is then calculated, using the influence matrix $d\boldsymbol{x}/d\boldsymbol{q}$, as

$$\Delta\boldsymbol{x} = d\boldsymbol{x}/d\boldsymbol{q}\cdot\Delta\boldsymbol{q}\ .$$

This process, first from nodal sensitivities to the sensitivities in generalized co-ordinates, and then, in reverse, from the change of generalized co-ordinates to change of nodal co-ordinates (re-meshing) is correct only theoretically i.e., for regular meshes and regular distributions of generalized co-ordinates.

In practice meshes can be very irregular, and the generalized co-ordinates cannot be distributed regularly either. The remedy is to enforce the same best fit both on the nodal changes and on the changes of generalized co-ordinates, i.e.

$$\Delta\mathbf{x}^{\mathbf{T}}.\ \Delta\mathbf{x} = \Delta\mathbf{q}^{\mathbf{T}}.\ \Delta\mathbf{q},$$

Resulting into the condition of ortho-normality

$$(d\mathbf{x}/d\mathbf{q})^{T}.\ (d\mathbf{x}/d\mathbf{q}) = \boldsymbol{I}.$$

The ortho-normalization can be achieved, for example, by Gram-Schmidt ortho-normalization algorithm, or by any other suitable method.

5.2. PROCESS FUNCTION

In ReSHAPE *process function*, virtual displacement fields of special shapes (so called *influence functions*) emanate from selected control points (see ReSHAPE Theory Manual). These fields have *unlimited support*, i.e. they permeate the whole space whilst not disappearing at the boundary of the structure. This has an advantage that all internal nodes can be easily affected; the disadvantage, however, is that also the nodes on the fixed boundaries (command *locked*) will move. The remedy is to generate functions also from all fixed nodes, and subsequently enforce constraints of no change at these nodes.

As an exercise, the process is again applied to the solution of the problem of a rectangle. Figure 8 shows a mesh. It consists of 8 by 8 rectangular plate elements and 4 by 8 bar elements around the perimeter.

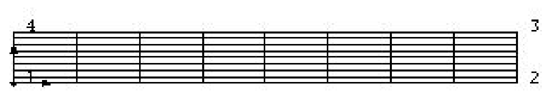

Figure 8: Mesh of a 100x10mm rectangle

The iterative improvement process is controlled by the following set of ReSHAPE commands:

```
reshape(improvement)
   eset(top)=2
   eset(left)=3
eset(bot)=4
   eset(rite)=5
control
   steps(n=100,s=0.5)
responses()
   area(target=max,stype=2d,tol=5%)=elem(all)
   length(bound=hold,stype=1d,tol=0)=elem(all)
process(function)
   domain(xy)=elem(all)
   passive(internal)=set(geom)
   locked(x)=set(left)
   locked(y)=set(bot)
   fcontrol(wrt=xy)=node(3)
   fparameter(boundary)=linear
end
```

Explanation of three special commands:

1) `passive` – do not calculate sensitivities either in *all* or in the *internal* nodes of the set, but remesh, i.e. calculate the influence function in the passive nodes too.
2) `fcontrol` – In the case of one control point only, the influence function emanating from the user defined control point (node 3 in the example) is constant in all directions.
3) `fparameter(boundary)=linear` – enforces a linear change of the influence function, i.e., the value of the function decreases linearly from the maximum in the locked point to the value 0 in the user defined control point. The internal functions are subsequently eliminated by enforcing zero changes in the locked points.

The resulting shape is shown in Figure 9.

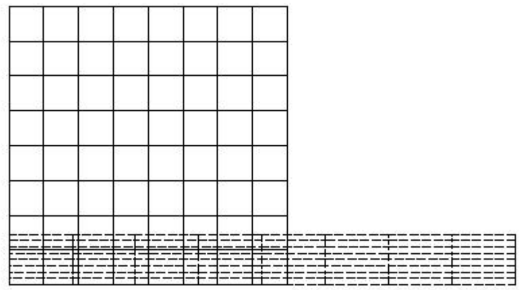

Figure 9: Rectangle of maximum area and a constant perimeter.

The method can be applied to any shape. As an exercise, students are asked to use a shape of their own choice. One such shape is shown in Figure 10. The command files for the sensitivity and improvement calculations are practically identical with those shown above. Figure 10 shows the original and the optimized shapes for a skewed and freely meshed quadrilateral.

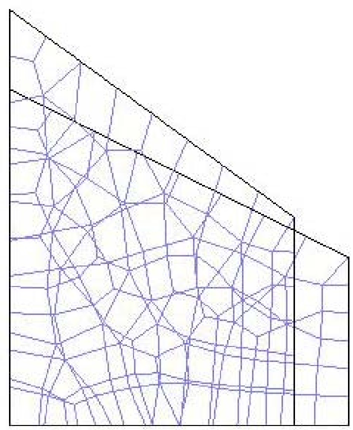

Figure 10: Freely meshed quadrilateral of maximum area and a constant perimeter

As home work, students are required to model a 3D solid box and maximize its volume while keeping the surface area unchanged.

5.3. PROCESS GEOMETRY

The geometry process (as used in ReSHAPE) does not calculate the influence matrix $d\mathbf{x}/d\mathbf{q}$, but smoothes the calculated sensitivities, scaled by user defined step size, to change the position of nodes.

The sensitivities of the internal nodes of plate elements are theoretically equal to zero. Numerically, however, they would not be zero and would represent numerical noise resulting from changing the shape of the elements. For this reason, sensitivity calculations should be avoided in the internal nodes. The ReSHAPE command `passive,` serves this reason.

In order to recalculate the mesh, the patch of elements must be defined by patch parameters in the corner nodes of the patch. Three and four sided patches are allowed in ReSHAPE.

The elements in the individual patches have to be in separate element sets. Then the ReSHAPE command `gset` in the `reshape` block of the command file will calculate the patch parameters and parametric co-ordinates of the nodes.

During the improvement process, the new nodes on the outline of the patch are best-fitted with cubic parabolas (quadratic parabolas in the case of three nodes along the edge, and straight lines in the case of two nodes). The approximated model using the patches is output by ReSHAPE, and can be viewed in a post processor.

As an exercise, students are asked to apply the geometry process to the previous example (Figure 8). The ReSHAPE command file reads:

```
reshape(improvement)
   gset(geom)=1
   gset(beam)=2,3,4,5
control
   steps(n=50,s=1)
responses()
   area(target=10000,stype=2d)=elem(all)
   length(bound=hold,tol=0,stype=1d)=elem(all)
process(geometry)
   domain(xy)=set(geom,beam)
   locked=elem(all)
   unlock(x)=node(2)
   unlock(xy)=node(3)
   unlock(y)=node(4)
end
```

Note: Unlocking only corner nodes results in boundary consisting of straight lines only.

The resulting shape is identical with the one shown in Figure 9.

5.4. OTHER PROCESSES

There are other processes exercised by students:

process(vector) in three modes of virtual mesh change: using displacements enforced by external load, using modes of vibrations, and using externally generated loads.

process(traction) where pre-calculated sensitivities on the boundaries are used as virtual loads to generate virtual mesh change.

process(sizing), applicable to plate and shell elements to change their thickness.

process(topology), changing the pseudo-density of the elements to suggest an improved structure.

The reader can obtained all the tutorial exercises from [2].

6. STUDENT ASSIGNMENTS

During the second half of the semester, each student works on a more complex problem of their own choice, which is advised and supervised by tutors. Two problems have been selected for this presentation.

6.1 Example 1 – Stress reduction in a slotted pipe

A pipe of 200mm length, 100mm radius and 3 mm thickness is fixed at both ends to rigid plates. One of the plates can rotate about the axis of the pipe and is loaded by a torsional moment of 5,000,000 Nmm. The pipe has a slot of 24 mm width at the bottom.

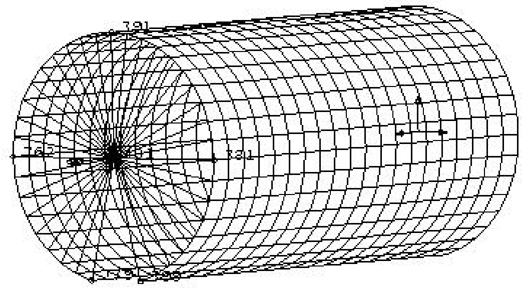

Figure 11: Finite-element model of a slotted pipe

The finite-element model of the pipe is shown in Figure 11. The pipe is modeled by thin shell elements and the rigid plates by rigid constraints (RBE2). The master of the constraint at one end is fully fixed and the master at the loaded end can rotate around the pipe axis which is loaded with the moment.

The maximum von Mises stress appears at the four corners with a value of 400 MPa.

The objective is to change the shape of the cross-section of the pipe in such a way that the von Mises maximum stress is reduced, whilst keeping the pipe prismatic and ensuring that the total volume does not increased by more than by 10%. As an added manufacturing constraint, the width of the slot must not change.

ReSHAPE `process(function)` is the most suitable process for this task. Five control points have been selected along the edge of the loaded end, as shown in Figure 11.

The command file for the iterative improvement process function is:

```
reshape(improvement)
   nset(srow)=1
control
   steps(n=30,s=2)
responses()
   stress(var=vmmax,target=-50%,avgw=90%)=elem(all)
   volume(bound=ceiling,tol=10%)=elem(all)
process(function)
   domain(xy)=element(all)
   locked(x)=node(399,779)
   locked=node(400,401)
   fcontrol=nodes(381,391,399,762,779)
end
```

The resulting shape is shown in Figure 12, together with the original circular shape.

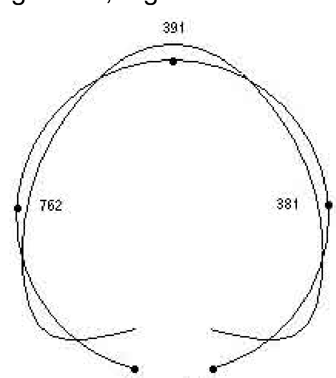

Figure 12: Original and new shape of the pipe

The maximum von Mises stress of 400 MPa was reduced to 360 MPa.

6.2 Example 2 – Stress reduction in a cylinder with a hole

Figure 16 shows a section of a cylinder with a hole in the middle, modeled using symmetry and loaded in the longitudinal (z) direction. The hole in the cylinder is experiencing high stress. The objective is to modify the shape of the hole so that the maximum stress is reduced. The hole must remain on the cylindrical surface. Also, the hole must not go inside its original radius, or outside the rectangle into which the hole is inscribed.

The constraint for the hole to remain on the surface of the cylinder can be satisfied, only if the shape of the hole is forced to change in y and z coordinates interpreted in a cylindrical co-ordinate system (shown in Figure 16). In this system, y represents the change in the circumferential direction (with no change in the radial and axial directions).

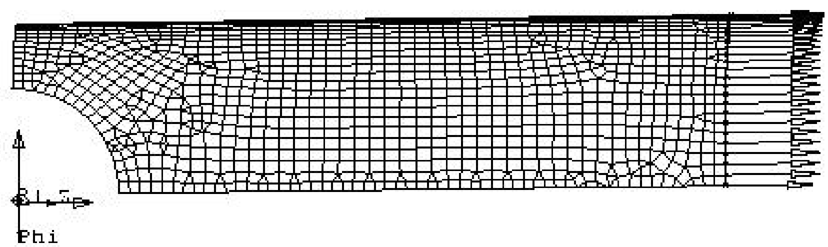

Figure 16: Cylinder with hole modeled using symmetry

Figure 17 shows the system again, together with a second cylindrical system, needed to keep the hole in the limits as prescribed above. This second system has the radial direction pointing along the radius of the hole.

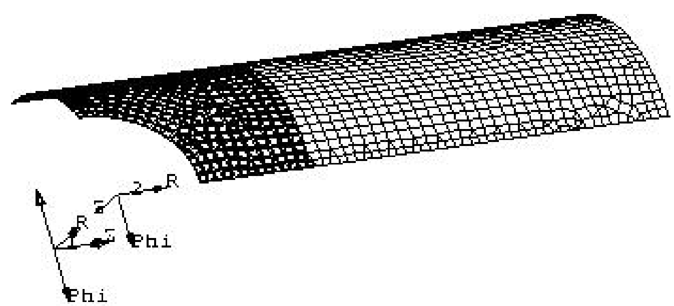

Figure 17: Second cylindrical coordinate system and selected domain

Figure 17 also shows the domain used for the shape change. It makes no sense to change the whole model, because the only region up to a certain distance from the hole will reduce the stress, hence it would be a waste of CPU time to change the whole model.

The maximum value of Mises stress is 661 MPa and appears at the expected location at the hole edge.

The following ReSHAPE command file is used for the stress reduction:

```
reshape(improvement)
   nset(fixed)=1thru4
   eset(hole)=5
   eset(active)=6
   nset(nhole)=7
control
  steps(n=200,s=2,m=0.95)
responses()
   stress(var=vmmax,target=-50%,avgw=95%)=set(hole)
```

```
process(function)
   domain(yz,cid=1)=set(active)
   passive(internal)=set(active)
   locked=set(fixed)
   fcontrol(wrt=yz,rot=y)=nodes(9,15)
   fparam(bound)=4
side conditions
   alimits(xl=29.9,cid=2)=set(nhole)
   alimits(xu=30.01,zu=30.01)=set(nhole)
end
```

Notice the increased value of the `fparameter(boundary)`from the default of 1.0 to 4.0. The use of the default value shows too much mesh change inside the model and not along the hole. This is caused by the internal influence function being “too week”, i.e. decreasing too fast from the maximum value at the boundary. The combined influence functions (adding user defined to the internal ones) drop suddenly too sharply to zero at the boundary. The increased value results into less sudden change, and, as the consequence, into a better mesh.

The resulting shape of the hole is shown in Figure 18, together with contours of Mises stresses, the maximum of which was reduced from 660 MPa to 335 MPa.

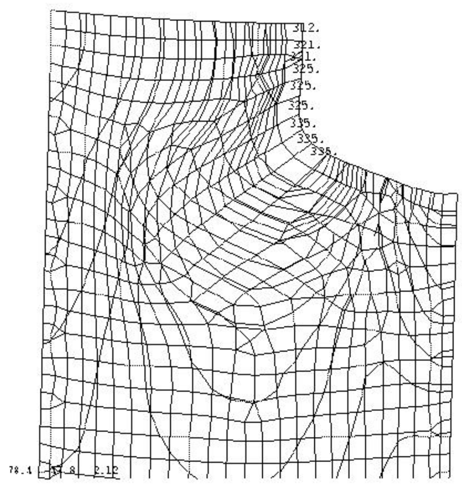

Figure 18: Contour of von Mises stress for the improved shape

Figure 19 shows the axial view of the cylinder after the shape change. It is evident that the cylindrical shape was not affected by the shape change of the hole.

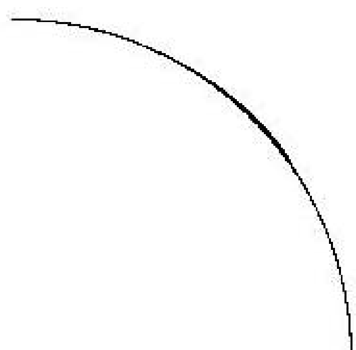

Figure 19: Axial view of the improved cylinder

7. CONCLUSION

With the introduction of the undergraduate *Design Improvement* course (started in year 2000), as a follow up of the introductory finite-element course, engineering students are educated in modern methods of design improvement. In such a way, the industry will be continuously provided with structural engineers who are at the fore-front of the application of finite-element methods and so will be thus be capable of contributing more effectively to designing better products.

REFERENCES

[1] Tomas, J. A. 1976, *Optimum Design of a Gas Turbine Wheel for some Ceramic Materials,* SAE Paper No. 760241, Detroit

[2] ReSHAPE Manuals and Case Studies, www.advea.com/reshape/

[3] ReSHAPE Manuals and Case Studies, www.advea.com/reshape/downloads_uni_course.htm

[4] Beveridge-Schechter: Optimization Theory and Practice, McGraw Hill, 1970.