УДК 624.074.5 + 624.94.014.2+ 69.033.2 + 69.04

РОЗРОБКА НОВОГО АЛГОРИТМУ ОБРОБКИ ДАНИХ ДЛЯ ТОПОЛОГІЧНОЇ РАЦІОНАЛІЗАЦІЇ

САВИЦЬКИЙ М. В.[1] , *д.т.н, проф.*,
ГРОСМАН С. О. [2*] , *аспірант*,
ТИТЮК А.О.[3], *к.т.н., доц.*

[1] Кафедра залізобетонних та кам'яних конструкцій, Державний вищий навчальний заклад "Придніпровська державна академія будівництва та архітектури", вул. Чернишевського, 24-а, 49600, Дніпропетровськ, Україна,
тел. +38 (0562) 47-02-98, e-mail: sav15@ukr.net, ORCID ID: 0000-0003-4515-2457
[2 *] Кафедра залізобетонних та кам'яних конструкцій, Державний вищий навчальний заклад "Придніпровська державна академія будівництва та архітектури", вул. Чернишевського, 24-а, 49600, Дніпропетровськ, Україна,
тел. +38 (0673) 96-73-02, e-mail: stas10.04.91@gmail.com, ORCID ID: 0000-0003-4474-7838
[3] Кафедра залізобетонних та кам'яних конструкцій, Державний вищий навчальний заклад "Придніпровська державна академія будівництва та архітектури", вул. Чернишевського, 24-а, 49600, Дніпропетровськ, Україна

Анотація: ***Мета.*** Існуючі алгоритми обробки результатів топологічної раціоналізації недосконалі: можуть ефективно обробити результати лише у випадку простої геометрії та при однаковому розмірі скінченних елементів, займають дуже багато часу при формуванні необхідних матриць та неефективно оперують над великими обсягами даних. Потрібен пошук нових, більш досконалих методів обробки результатів топологічної раціоналізації. ***Методика.*** Запропонований алгоритм обробки результатів топологічної раціоналізації базується на векторизації координат центрів елементів та на послідовному переносі базису системи координат до від одного центра елементу до іншого. Для реалізації запропонованої ідеї був розроблений програмний код у системі MATLAB. ***Результати.*** Зазначений алгоритм був випробуваний на тестових задачах різного розміру та складності геометрії. Отримані результати підтверджують ефективність розробленого алгоритму. Здобуто пришвидшення розрахунку у 2-6 разів в залежності від розміру задачі, що вирішується. Встановлено що зі збільшенням розміру задачі підвищується ефективність методу. Також підтверджене припущення про можливість значного прискорення розрахунку топологічної раціоналізації за рахунок зменшення затрат часу на розрахунок скінченно елементної моделі за допомогою ітеративних методів рішення систем лінійних рівнянь та за допомогою технологій збереження пам'яті. ***Наукова новизна.*** Удосконалення системи обробки результатів топологічної раціоналізації шляхом застосування векторизації даних та преалокації пам'яті . ***Практична значимість.*** Використання запропонованого методу може дозволити прискорити впровадження методів топологічної раціоналізації в повсякденне проектування.

Ключові слова: топологія, оптимізація, раціональне проектування, SIMP метод, технології фільтрування результатів

РАЗРАБОТКА НОВОГО АЛГОРИТМА ОБРАБОТКИ ДАННЫХ ДЛЯ ТОПОЛОГИЧЕСКОЙ РАЦИОНАЛИЗАЦИИ

САВИЦКИЙ Н. В.[1] , *д.т.н, проф.*,
ГРОСМАН С. А. [2*] , *аспирант*
ТЫТЮК А.А.[3], *к.т.н., доц.*

[1] Кафедра железобетонных и каменных конструкций Государственное высшее учебное заведение " Приднепровская государственная академия строительства и архитектуры", ул. Чернышевского, 24-а, 49600, Днепропетровск, Украина,
тел. +38 (0562) 47-02-98, e-mail: sav15@ukr.net, ORCID ID: 0000-0003-4515-2457
[2*] Кафедра железобетонных и каменных конструкций Государственное высшее учебное заведение " Приднепровская государственная академия строительства и архитектуры", ул. Чернышевского, 24-а, 49600, Днепропетровск, Украина,
тел. +38 (0673) 96-73-02, e-mail: stas10.04.91@gmail.com, ORCID ID: 0000-0003-4474-7838
[3] Кафедра железобетонных и каменных конструкций Государственное высшее учебное заведение " Приднепровская государственная академия строительства и архитектуры", ул. Чернышевского, 24-а, 49600, Днепропетровск, Украина

Аннотация: ***Цель.*** Существующие алгоритмы обработки результатов топологической рационализации несовершенны: могут эффективно обработать результаты только в случае простой геометрии и при одинаковом размере конечных

элементов, занимают очень много времени при формировании необходимых матриц и неэффективно оперируют над большими объемами данных. Необходим поиск новых, более совершенных методов обработки результатов топологической рационализации. ***Методика.*** Предложенный алгоритм обработки результатов топологической рационализации базируется на векторизации координат центров элементов и на последовательном переносе базиса системы координат от одного центра элемента к следующему. Для реализации предложенной идеи был разработан программный код в системе MATLAB. ***Результаты.*** Указанный алгоритм был испытан на тестовых задачах разного размера и сложности геометрии. Полученные результаты подтверждают эффективность разработанного метода. Получено ускорение расчета в 2-6 раз в зависимости от размера решаемой задачи. Установлено, что с увеличением размера задачи повышается эффективность метода. Также подтверждено предположение о возможности значительного ускорения расчета топологической рационализации за счет уменьшения затрат времени на расчет конечно элементной модели с помощью итеративных методов решения систем линейных уравнений и с помощью технологий сбережения памяти. ***Научная новизна***. Совершенствование системы обработки результатов топологической рационализации путем применения векторизации данных и преалокации памяти. ***Практическая значимость***. Использование предложенного метода может ускорить внедрение методов топологической рационализации в повседневное проектирование.

Ключевые слова: топология, оптимизация, рациональное проектирование, SIMP метод, технология фильтрации результатов

DEVELOPMENT OF NEW ALGORITHMS FOR DATA PROCESSING FOR TOPOLOGICAL RATIONALIZATION

SAVYTSKYI M. V. [1] , *Dr. Sc. (Tech.), Prof.*
GROSMAN S. O. [2*] , *MSc, Postgraduate student*
TYTJUK A.O.[3], *PhD*

[1] Department of Reinforce-Concrete and Stoune Constructions, State Higher Education Establishment “Pridneprovsk State Academy of Civil Engineering and Architecture”, 24-A, Chernishevskogo str., Dnipropetrovsk 49600, Ukraine, tel. +38 (0562) 47-02-98, e-mail: sav15@ukr.net, ORCID ID: 0000-0003-4515-2457
[2*] Department of Reinforce-Concrete and Stoune Constructions, State Higher Education Establishment “Pridneprovsk State Academy of Civil Engineering and Architecture”, 24-A, Chernishevskogo str., Dnipropetrovsk 49600, Ukraine, tel. +38 (0673) 96-73-02, e-mail: stas10.04.91@gmail.com, ORCID ID: 0000-0003-4474-7838
[3] Department of Reinforce-Concrete and Stoune Constructions, State Higher Education Establishment “Pridneprovsk State Academy of Civil Engineering and Architecture”, 24-A, Chernishevskogo str., Dnipropetrovsk 49600, Ukraine

Abstract: Existing algorithms for processing the results of topological rationalization are imperfect. They can effectively treat the results only in the case of simple geometry, and for the evenly sized finite elements. They take a lot of time during the formation of the necessary matrices and inefficiently operate on large volumes of data. That is why the need to search for new and improved methods of processing the results of topological rationalization arises. ***Methodology***. The proposed algorithm for processing the results of topological rationalization is based on the vectoring of coordinates of the centers elements and then on subsequent transfer of the basis of the coordinate system from one center of the element to another. To implement this idea, the program code in the MATLAB was developed. ***Results***. This algorithm has been tested on the test problems of different size and of different geometrical complexity. The results confirm the effectiveness of the discussed method. A time saving on matrix formulation is observed to vary between factors of 2 and 6 depending on the size of the problem being solved. It is also found that along the increase of size of the problem the efficiency of the method also increases. Also the assumption about the possibility of a significant acceleration of the calculation of the topological rationalization has been confirmed. This was achieved by reducing the time spent on solution of finite element model through the implementation of iterative methods for solving systems of linear equations and usage of memory savings technology. ***Scientific novelty***. Improvement in processing of the results of topological rationalization by applying data vectoring and memory preallocation. ***The practical significance***. Usage of the proposed method can help to accelerate the introduction of topological methods for rationalization in everyday civil engineering design practice.

Key words: topology optimization, sustainable design, SIMP method, technology of results filtering

Introduction

When designing bearing structures of a building choosing the right structural concept plays an important role. Generally to determine the best concept engineers have to rely on their experience and intuition. However, for complex and non-standard designs, the shape of which is determined on the basis of a large number of independent factors, or in the case of an implicit connection between the possible design and the loads acting on the structure, the decisions made on the basis of abstract concepts may not be optimal, and in some cases even catastrophic. The solution to this problem is the application of mathematical methods of optimization to find the most effective design solutions.

Mathematical methods for finding the optimal solution can be applied at all stages of the project. During the development of preliminary solutions (on this step a wide variety of choices is usually developed, only one from them will be subsequently selected) optimization techniques can be used to select the optimal topology of a structure (topology optimization) and to find the most effective form of the developed structure (shape optimization).

At the stage of structural design various linear and nonlinear optimization methods may be employed to determine values for design parameters that governs characteristics of a structure such as the cross-sectional dimensions, the types of connections (rigid or hinged), and various physical, technical and aesthetic constraints defined by regulations.

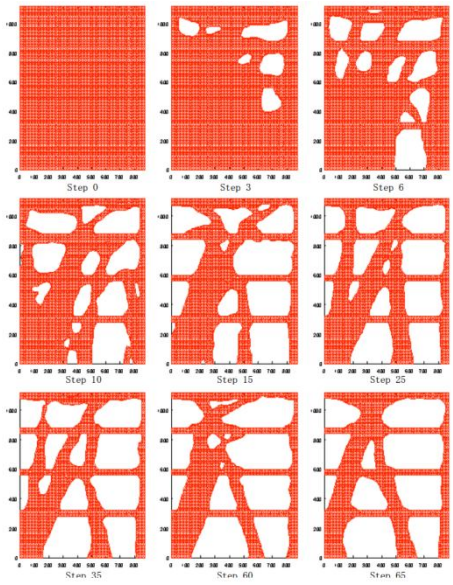
For the stage of final drawing development it is also possible to use the aforementioned methods for optimizing the joints of structural elements.

This paper is devoted to development of robust approaches to implementation of topology optimization in civil engineering.AT the moment topology techniques had very limited implementation in the building industry. Following examples are among most noticeable instances of this technique application to design process in civil engineering. Topology optimization was used to derive the optimal number, location and shape of holes in the exterior reinforced concrete walls of an office building near to the Takatsuki JR Station in Japan (see Figure 1). The walls were modelled as simple rectangular plates and optimized for vertical and horizontal loading combinations. The result was found to be both aesthetically pleasing and structurally sound. It should be noted that the architecture of the entire building was totally governed by structural considerations arising from the results of the topology optimization study.

Topology optimization has also been used for purely architectural purposes. The architectural aspiration of the Doha Education Centre's roof canopy support was to mimic the form of a Sidra tree[1]. Topology optimization studies were performed in order to define the geometry of the canopy support structure. It was found that the resulting form has strong resemblances to a tree trunk indeed (see Figure 2).

a)

b)



c)

Figure 1. Building designed on the basis of topologically optimized concept.

a) Concept of a structure
b) Practical realization
c) Stages of optimization

Figure 2. Educational center, Doha, Quatar

Apparatus of topology optimization has also been used to develop a project proposal of the Bionic Tower in 2007 in Dubai, United Arab Emirates. Due to the crisis the project has not been realized (see Figure 3)[2, 3].

Figure 3. Bionic tower concept, Dubai, UAE

This topic lies in sphere of interests of many of the world universities. The staff of Georgia Institute of Technology (USA) is currently developing an approach to structuring and regularization of design solutions obtained on the basis of topological optimization [4]. The preliminary result of their work is presented on the Figure 4.

Although the implementation of topology optimization can significantly benefit the building industry there are still several obstacles that prevent the universal usage of this approach by practicing engineers. Among these obstacles one of the most important is the time required to obtain the final solution. Since the topology optimization is an iterative procedure the time required to solve the problem is equivalent to the time necessary to solve the problem with predefined topology times the amount of iterations necessary to obtain converged solution (this number heavily depends on the complexity of the problem) times the amount of load cases considered on this stage.

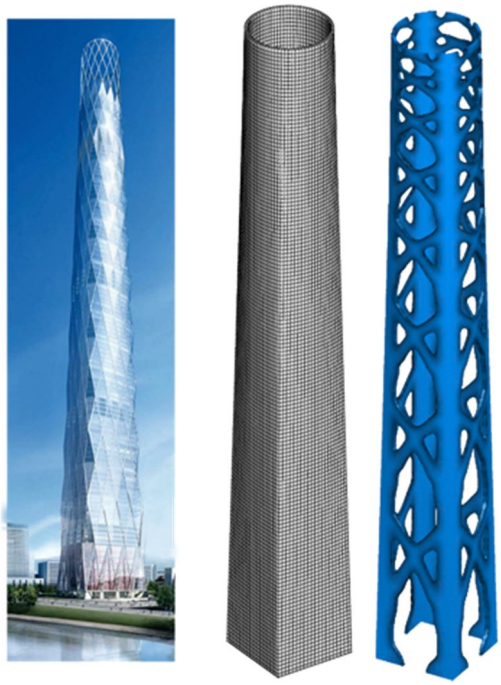


Figure 4. Design concept of tower with regular grating

The time that solution takes can be roughly divided into four categories: time for model generation, time for generation of constant topology entities, time for FE solution and time for the solution of topology optimization problem.

Model generation is a stage that is different for each problem and thus is very complex to enhance, moreover since model generation occurs only once the time spent on this stage however significant it may be (it is far inferior to time required to generate a model for ordinary purposes) is irrelevant to the method performance.

Time required for the FE solution to take place is the most significant time consumers of the topology optimization cycle. This is so due to the fact that solution is performed on each iteration and load step. This results in the demand for productive methods to compute the necessary elemental results. One of the possible approaches to increasing the productivity of the solution stage will be briefly discussed further.

The time spent on the solution of topology problem is not as significant as the time spent in FE solver, especially for large problems but it is very important to use the appropriate techniques to achieve the best performance. MATLAB software has been used in this paper as a solver for topology problem. To achieve the best performance such techniques as loop vectorization and memory preallocation has been used [5]. Loop vectorization is the use of vector and matrix operations in order to avoid for and while loops. Memory preallocation means that the maximum amount of memory required for an array is reserved a priori, hence avoiding the costly operation of reallocating memory and moving data as elements are added to the array.

This paper will be devoted to the minimizing the time required for creation of topology constant entities. This process is extremely computationally intense and can

result in memory overfloat if handled carelessly. This can occur due to the creation of a matrix that relates each element position to another. This process results in creation of square matrix the size of which is equal to the amount of elements. It is obvious the amount of memory allocated for such matrix scales exponentially with the problem size. This paper presents a possible solution for creation of such a matrix with minimal amount of operations possible.

Methodology

General information about topology optimization

Topology optimization is a kind of shape optimization of structures, sometimes referred to as the optimization of the layout. Purpose of topology optimization is to determine the best use of the material for the scrutinized object or structure, so that the objective function (for example, the overall stiffness or natural frequency) reaches a maximal or minimal value in the presence of existing restrictions (such as volume reduction). Unlike traditional optimization, topological optimization does not require to set parameters of optimization explicitly (i.e. independent variables are optimized).

In topology optimization the role of variable in the objective function is played by the function of distribution of the material in design space. In contrast to the optimization of shape and size of bearing elements topology optimization allows to find an optimal distribution of the material in a given design space under certain load and boundary conditions.

The most common formulation of the topology optimization problem is a Solid Isotropic Material with Penalisation (SIMP) described in detail in the works of Sigmund and Bendose and Andrea [6–10]. This formulation suggests that the material of a scrutinized design space acts as a porous structure. The optimization is aimed at location the best position for such pores with respect to the predefined degree of porosity.

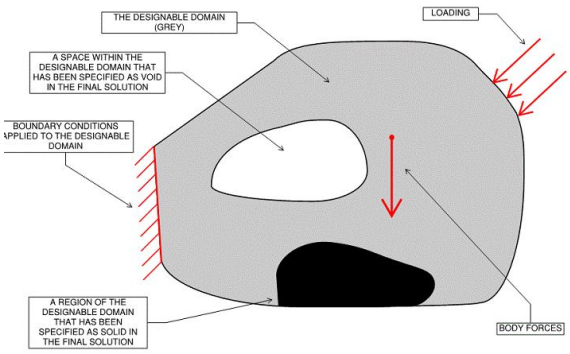


Figure 5. General outline of the optimization problem

Design space is defined as the space within which optimized design will be found. The space is divided into elemental regions each with its own porosity value. For such formulation objective function can be a total strain energy of the structure under the load and boundary conditions. The problem also has a boundary for condition that limits the maximum volume or weight of the structure.

For the purpose of optimization the design domain is discretized finite elements mesh. Then for each element a value of relative density x_e is assigned. This relative density governs the element basic characteristics such as mass, Young's modulus etc. For such approach Young's modulus of each element E_e is determined by the following formula:

$$E_e(x_e) = E_{min} + x_e{}^p \cdot (E_0 - E_{min}), x_e \in [0,1] \quad (1)$$

Where:

$\boldsymbol{E_0}$ – Young's modulus of material.

$\boldsymbol{E_{min}}$ – a very small stiffness assigned to void regions in order to prevent the stiffness matrix from becoming singular.

$\boldsymbol{p}$ – is a power law penalization factor, that is introduced into the system to transform a system from a continuum state with density values varying between 1 and 0 into a discrete system where only two of these values will be present. Practice had shown that the best results can be achieved when the value of this parameter is $\boldsymbol{p}=3$.

The mathematical formulation of the optimization problem reads as follows[7]:

$$\left.\begin{array}{c} \min:_x C(x) = \mathbf{U}^{\mathrm{T}}\mathbf{KU} = \sum_{e=1}^{N} E_e(x_e) \cdot \mathbf{u}_e{}^T \mathbf{k}_0 \mathbf{u}_e \\ \frac{V(x)}{V_0} = f \\ \mathbf{KU} = \mathbf{F} \\ 0 \le x \le 1 \end{array}\right\} \quad (2)$$

Where:

C – compliance of the system;

$\mathbf{U}$ and $\mathbf{F}$ – the global displacement and force vectors, respectively;

$\mathbf{K}$ – the global stiffness matrix;

$\mathbf{u}_e$ – the element displacement vector;

$\mathbf{k}_e$ – the element stiffness matrix for an element with unit Young's modulus.

x – the vector of design variables (i.e. the element densities) ;

N – the number of elements used to discretize the design domain;

$V(x)$ and V_0 – the material volume and design domain volume, respectively;

f – the prescribed volume fraction.

The optimization problem (2) is solved by means of a standard optimality criteria method:

$$x_e^{new} = \begin{cases} \max(0, x_e - m) \text{ if } x_e \cdot B_e{}^{\eta} \le \max(0, x_e - m) \\ \min(1, x_e + m) \text{ if } x_e \cdot B_e{}^{\eta} \ge \min(1, x_e + m) \\ x_e \cdot B_e{}^{\eta} \text{ otherwise} \end{cases} \quad (3)$$

Where:

m – positive maximal move limit of variable on each iteration;

η – numerical damping coefficient, equals to: $\eta = 1/2$;

$$B_e = \frac{-\frac{\partial C}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}} \quad (4)$$

Where:

λ – Lagrangian multiplier that is chosen in such a way that the volume restriction is observed;

The sensitivities of the objective function c and the material volume V with respect to the element densities x_e are given by:

$$c = \frac{\partial C}{\partial x_e} = -p \cdot {x_e}^{p-1} \cdot (E_0 - E_{min}) \cdot \mathbf{u}_e{}^T \mathbf{k}_0 \mathbf{u}_e \quad (5)$$

$$\frac{\partial V}{\partial x_e} = V_e \quad (6)$$

Where:

V_e – is a volume of a finite element.

In order to ensure existence of solutions to the topology optimization problem and to avoid the formation of checkerboard patterns and dependency on finite element discretization, some restriction on the design parameters must be imposed[11], [12].

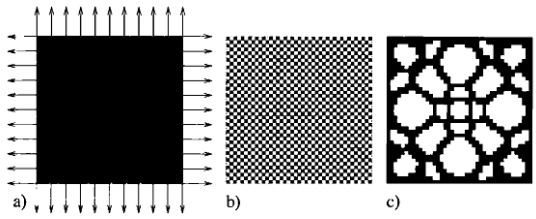


Figure 6. The checkerboard problem demonstrated on a square structure subject to biaxial stress

a) Design problem;

b) Solution without checkerboard control;

c) Solution with sensitivity filtering.

A common approach to solve this problem is the application of a filter to either the sensitivities (5) or the relative densities. Filter sets a new value for design parameters in element on the basis of weighted average of the results stored in the elements located in the vicinity (this zone is governed by r_{min} parameter) of the considered element. Two of the possible filtering schemes are presented here.

The sensitivity filter modifies the sensitivities $\partial c/\partial x_e$ as follows:

$$\frac{\widehat{\partial C}}{\partial x_e} = \frac{1}{\max(\gamma, x_e) \cdot \sum_{i \in N_e} H_{ei}} \cdot \sum_{i \in N_e} H_{ei} x_i \frac{\partial C}{\partial x_i} \quad (7)$$

Where

γ – a small positive number introduced in order to avoid division by zero.

$$H_{ei} = \max\left(0, r_{min} - \Delta(e,i)\right) \quad (8)$$

$\Delta(e,i)$ – center-to center distance between elements

The density filter transforms the original densities x_e as follows:

$$\widetilde{x_e} = \frac{1}{\sum_{i \in N_e} H_{ei}} \cdot \sum_{i \in N_e} H_{ei} x_i \quad (9)$$

In the case where a density filter is applied, the sensitivities of the objective function c and the material volume V with respect to the physical densities $\widetilde{x_e}$ are still given by equations (5) and (6), provided that the variable x_e is replaced with $\widetilde{x_e}$. The sensitivities with respect to the design variables x_j are obtained by means of the chain rule:

$$\frac{\partial \psi}{\partial x_j} = \sum_{i \in N_j} \frac{\partial \psi}{\partial \widetilde{x_e}} \frac{\partial \widetilde{x_e}}{\partial x_j} = \sum_{i \in N_j} \frac{1}{\sum_{i \in N_e} H_{ei}} \cdot H_{je} \cdot \frac{\partial \psi}{\partial \widetilde{x_e}} \quad (10)$$

Where

$\partial\psi$ – either the objective function c or the material volume V

Results and discussion

Filtering pattern generation

The application of a sensitivity filter according to equation (7) involves a weighted average over different elements. This is a linear operation; it can therefore be implemented as a matrix product of a coefficient matrix and a vector containing the original sensitivities $\partial C/\partial x_i$ (multiplied with the design variables x_i). Dividing the result by a factor $\max(\gamma, x_e) \times \sum_{i \in Ne} H_{ei}$ yields the filtered sensitivities $\widehat{\partial C/\partial x_e}$. The matrix H and the vector H_s contain the coefficients H_{ei} and the normalization constants $\sum_{i \in Ne} H_{ei}$, respectively.

The use of a density filter not only implies filtering of the densities according to equation (9) but also a chain rule modification of the sensitivities of the objective function and the volume constraint according to equation (10). Both operations involve a weighted average over different elements. Use is made of the same coefficients H and normalization constants Hs as described above.

Both the matrix H and the vector Hs remain invariant during the optimization and are computed a priori. The coefficient matrix H establishes a relationship between all elements. However, according to the filter kernel defined in equation (8), only neighboring elements affect one another. As a consequence, the majority of the coefficients are zero and the matrix H is sparse.

Sparse matrix is a matrix in which most of the elements are zero. By contrast, if most of the elements are nonzero, then the matrix is considered dense. When storing and manipulating sparse matrices on a computer, it is beneficial and often necessary to use specialized algorithms and data structures that take advantage of the sparse structure of the matrix. Operations using standard dense-matrix structures and algorithms are slow and inefficient when applied to large sparse matrices as processing and memory are wasted on the zeroes. Sparse data is by nature more easily compressed and thus require significantly less storage. Some very large sparse matrices are infeasible to manipulate using standard dense-matrix algorithms. A matrix is typically stored as a two-dimensional array. Each entry in the array represents an element $a_{i,j}$ of the matrix and is accessed by the two indices i and j. Conventionally, i is the row index, numbered from top to bottom, and j is the column index, numbered from left to right. For an $m \times n$ matrix, the

amount of memory required to store the matrix in this format is proportional to $m \times n$.

In the case of a sparse matrix, substantial memory requirement reductions can be realized by storing only the non-zero entries. Depending on the number and distribution of the non-zero entries, different data structures can be used and yield huge savings in memory when compared to the basic approach.

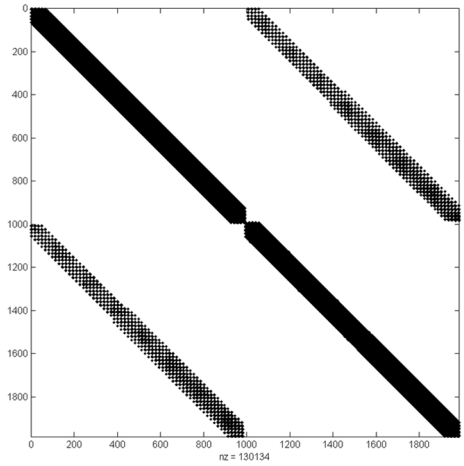


Figure 7. Sparse matrix representation of 2D design space

One way to decrease time for FE solution

Analyzing the time spent in the FE solver the most time consuming process is creating of global stiffness matrix with following factorization of it. Since for each iteration the stiffness matrix will be different this operation has to be repeated each time. This results in a significant increase of computational time. To solve this problem one can consider taking different approach to the solution FE problem. Among such approaches one of the most promising is usage of preconditioned conjugate gradients method to solve the linear equations. They do not require a costly matrix factorization of the assembled matrix, and they always run in memory and do only minimal I/O [13]. Another plus of usage of this method can arise from application of additional technique intended on memory saving. This technique uses an element-by-element approach (rather than globally assembling the stiffness matrix) and as such significantly up to 10 times decreases amount of time required for the solution of a well-conditioned model.

New filtering pattern creation algorithm

The existing approaches to the problem are meticulously described in the works of Bendose & Sigmund, Jensen and Andreassen [7], [8], [11]. The idea is to create a sparse matrix that will relate the results in neighboring elements and is such a way get rid of fictious void regions caused by mesh dependency or checker-board phenomena.

The sparse function used in this paper takes three vectors as input arguments: the first and second contain the row and column indices of the nonzero matrix entries, which are collected in the third vector. It is the built-in sparse MATLAB function. It is constructed by means of Row and column index vectors *iH* and *jH* as well as a vector *sH* with non-zero entries are assembled. In order to avoid continuous resizing of these vectors as entries are added, a sufficient (but slightly too high) amount of memory is preallocated. The entries that remain unused in the vectors *iH, jH*, and *sH* have no effect: they preserve their initial value (1, 1, and 0, respectively) and result in the addition of a zero term to the first element of the sparse matrix *H*.

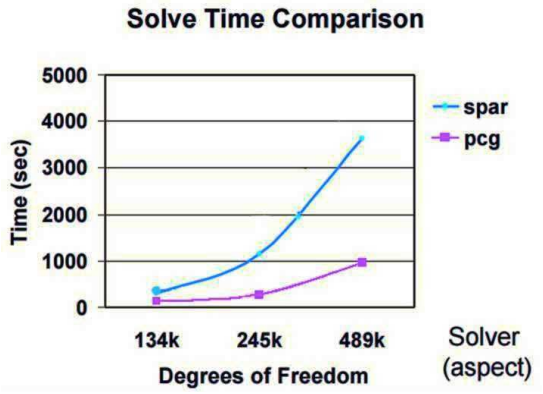


Figure 8. Performance diagram for Sparce and PCG solvers

The scope of this paper is development of an approach to speed up the process of sparse matrix *H* formulation. The general trend observed in the works of Bendose & Sigmund, Jensen and Andreassen [7], [8], [11] is to use the rectangular design space with the similar elements of similar size and basing on these prerequisites to operate on predefined elements when creating a filtering pattern. This approach helps to significantly decrease amount of time necessary to create the pattern, but it has very limited applicability for real life problems where design space can't be simplified to a rectangle or when the usage of square or cubic finite elements is impossible. For such cases general instructions are to conduct a search for each element to find the elements located in the vicinity of this element and fill the matrix with the appropriate weighting factors. This can be described in the following code lines:

```
for iii=1:elem_count
    for jjj=1:elem_count
        dist= sqrt((x(iii)-x(jjj))^2+...
                   (y(iii)-y(jjj))^2+...
                   (z(iii)-z(jjj))^2)
        pp=max(0,1-dist)/r_min);
        if pp>0
            k=k+1
            sH(1,k)=pp;
            iH(1,k)=iii;
            jH(1,k)=jjj;
        end
    end
end
```

It can be seen that to solve this problem it is necessary to perform `elem_count`^{2} operations to fill up the matrix. The use can be made of the matrix symmetry:

```
for iii=1:elem_count
    for jjj= iii:elem_count
        dist= sqrt((x(iii)-x(jjj))^2+...
                   (y(iii)-y(jjj))^2+...
                   (z(iii)-z(jjj))^2)
        pp=max(0,1-dist)/r_min);
        if pp>0
            k=k+1
            sH(1,k)=pp;
            iH(1,k)=iii;
            jH(1,k)=jjj;
            if iii ~=jjj
                k=k+1
                sH(1,k)=pp;
                iH(1,k)=jjj;
                jH(1,k)=iii;
            end
        end
    end
end
```

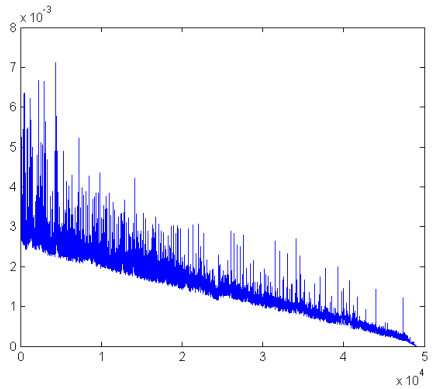


Figure 9. Time spend on each inner loop iterations for existing methods

This can reduce the computational cost from the `elem_count`^{2} operations to `elem_count`^{2}/2 operations. Both of these approaches are extremely inefficient when tackling the problem even of a small size. To overcome this problem an approach based on vectorization of the problem has been used. The coordinates of element centers have been formed as vectors. Then the coordinate transmutation has been performed. The purpose of this transformation is to move the coordinate centers to the center of the element under consideration. This simple transformation results in a very elegant way of obtaining a whole column and row (due to symmetry) of *H* matrix in just one operation instead of `elem_count` operations. This approach also indicates that each following iteration will operate on the smaller vector then the previous one. This can be described in the following code lines:

```
end_index=0;
for iii=1:elem_count
    dist= sqrt((x(iii)-x(jjj))^2+...
               (y(iii)-y(jjj))^2+...
               (z(iii)-z(jjj))^2)
    pp=max(0,1-dist)/r_min);
    elem_red_count=nnz(pp);
    [row_i,~,s_ij] = find(pp);
    start_index=1+end_index;
    end_index=start_index+...
    (elem_red_count*2-1)-1;
     sH(1,start_index:end_index)= ...
               [s_ij' s_ij(2: end)'];
     iH(1,start_index:end_index)= ...
                 [(row_i+iii-1)', ...
     ones(1,(elem_red_count-1))*iii];
     jH(1,start_index:end_index)= ...
    [ones(1, elem_red_count)*iii, ...
              (row_i(2:end)+iii-1)'];
end
```

This can reduce the computational cost from the `elem_count`^{2}/2 operations to `elem_count`/2 operations. This approach has been successfully implemented in MTLAB software and used for topology optimization of several test cases. The main advantage of this approach is ability to consider only relevant elements for each iteration. Another great advantage of this method is its independence from the shape and size of finite elements used for solution. This can greatly simplify solution process for complex geometries which are common in practice of engineer.

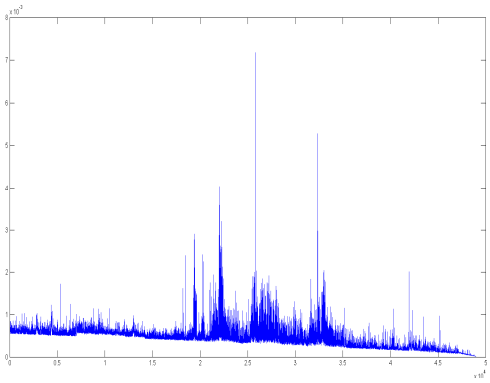


Figure 10. Time spent on each iteration for the proposed method

Conclusion

This paper presents a MATLAB code for creation of filtering pattern matrix for topology optimization. The major difference with respect to the code proposed by Andreassen [7] is the computational efficiency. An improvement in speed with a factor of 3.4 has been measured for an example problem with 48960 elements. It has also been found that the larger the problem the more efficient the proposed code is. This has mainly been accomplished by means of loop vectorization and memory preallocation.

The classical filter requires information about the neighbor elements, which for irregular meshes and complex geometries is obtained by a relatively expensive search. The computational complexity of previously applied codes, as well as the memory utilization, are proportional to r_{min}^2 in 2D and to r_{min}^3 in 3D respectively. The new approach decreases the computational cost so that it only depends linearly on the length parameter *r*. Therefore, for large filter radius, especially in 3D, the filtering scheme proposed in this paper should be the preferred choice. This approach is also beneficial for smaller problems with a complex geometries.

This paper also presents an approach to speed up the finite element solution stage of topology optimization. This is achieved by combining usage of iterative solver with the memory saving techniques that don't require to create a general stiffness matrix.

References

1. K. D. Tsavdaridis, J. J. Kingman, and V. V. Toropov, "Structural topology optimisation in steel structural applications," in *Proceedings of the Hellenic National Conference of Steel Structures*, 2014.
2. J. Kingman, K. Tsavdaridis, and V. Toropov, "Applications of topology optimisation in structural engineering: high-rise buildings & steel components," *Jordan Journal of Civil Engineering*. 20-Nov-2014.
3. J. J. Kingman, K. D. Tsavdaridis, and V. V. Toropov, "Applications of topology optimisation in structural engineering: high-rise buildings & steel components," *Jordan J. Civ. Eng.*, vol. 9, 2014.
4. L. L. Stromberg, A. Beghini, W. F. Baker, and G. H. Paulino, "Application of layout and topology optimization using pattern gradation for the conceptual design of buildings," *Struct. Multidiscip. Optim.*, vol. 43, no. 2, pp. 165–180, Feb. 2011.
5. The MathWorks Inc., *MATLAB and Statistics Toolbox Release 2012b*. Natick, Massachusetts, United States., 2012.
6. M. P. Bendsoe and O. Sigmund, *Topology Optimization: Theory, Methods and Applications*. Springer Science & Business Medi, 2003.
7. E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund, "Efficient topology optimization in MATLAB using 88 lines of code," *Struct. Multidiscip. Optim.*, vol. 43, no. 1, pp. 1–16, 2011.
8. O. Sigmund, "A 99 line topology optimization code written in matlab," *Struct. Multidiscip. Optim.*, vol. 21, no. 2, pp. 120–127, 2001.
9. J. S. Jensen, "TOPOLOGY OPTIMIZATION THEORY , METHODS AND APPLICATIONS INTRODUCTION TO COMPUTER EXERCISES," Delft, 2010.
10. G. H. Paulino, E. Silva, E. DE STURLER, S. WANG, C. C. SWAN, S. F. RAHMATALLA, C. TALISCHI, C. H. LE, and J. E. BAE, *New Advances in Topology Optimization*. Sao Paolo: Cornell University, 2008.
11. J. S. Jensen, "TOPOLOGY OPTIMIZATION THEORY, METHODS AND APPLICATIONS DESIGN IN VIBRATION AND WAVE PROPAGATION," Delft, 2010.
12. Z. Jianhua, J. Shoushan, C. Yongdang, and W. Jilong, "Application of Variable Density Topological Optimization Method to the Design of High-Speed Ground Vehicle's Chassis," *Int. J. Adv. Inf. Sci. Serv. Sci.*, vol. 4, no. 22, pp. 510–516, 2012.
13. ANSYS® Academic Research, *Release 14.0, Help System,Performance Guide*. ANSYS, Inc., 2014.

Статья поступила в редколлегию 10.08.2015