A spring system method for a mesh generation problem

A Romanov
Ishlinsky Institute for Problems in Mechanics of the Russian Academy of Sciences, Moscow, Russia
E-mail: malimo93@gmail.com

Abstract. A new direct method for the 2d-mesh generation for a simply-connected domain using a spring system is observed. The method can be used with other methods to modify a mesh for growing solid problems. Advantages and disadvantages of the method are shown. Different types of boundary conditions are explored. The results of modelling for different target domains are given. Some applications for composite materials are studied.

1. Introduction
Requirements to endurance and reliability of machines, constructions, and mechanisms are constantly increasing as the functionality of devices and their complexity are growing as a result of technical and technological progress. The methods for precise calculations of stresses, deformations, and distributions of different fields in machine parts are necessary for accepting new challenges. Various properties of devices, for instance, endurance and reliability, depend on the manufacturing process. One of the new and prospective processes is the additional manufacturing, so it is important to study its procedures [1–3]. Thus, we need to have methods for numerical solution of such problems. This is usually obtained by the finite element method (FEM) which consist in approximation of unknown displacements by a known function with parameters which are going to be determined. This approximation is based on a mesh in the solid under study. It is usually difficult to create a convenient mesh in 3d- and 2d-domains, especially for ones with complex geometry. There are a lot of methods for mesh generation methods and most of them provide a fine mesh either triangular or quadrangular. Composite materials [4] are also used in the growing process and, as a result, we can get composite solids (solids where the size of inclusions is comparable with the size of the body). Thus, we have to study such solids as well.

2. Description and comparison of the methods
All the methods can be divided into two large groups: methods for a triangular mesh and methods for a quadrangular mesh. The second ones are, in turn, divided into direct and indirect methods.

Triangular meshes. This class contains a lot of methods: Delauney triangulation [5], quad-tree-based method [6], LBWARP [7], and a bubble method [8]. They are not studied here because the quad-meshes are better than triangular ones in many cases for FEM.
**Quadrangular meshes. Direct methods.** In direct methods, the quadrilaterals are constructed at once. They use some kind of advancing front technique [9], regular grid-based methods, or quad-tree ones [10]. The advancing front methods for quads are considered to be nonrobust, and the quad-tree methods usually produce low quality elements close to the boundaries of the domain and are unable to fulfil general size constraints (anisotropy, strong variations).

**Indirect methods.** In indirect methods, a triangular mesh is built first. Then the triangle-merge methods use the triangles of the initial mesh and recombine them to form quadrangles [11, 12]. Other more sophisticated indirect methods use a mix of advancing front and triangle merge [13].

The methods described above do not suit us, because they generate a mesh from scratch every time, but in problems of growing bodies, this is improper because these problems are often solved in the quasi-state formulation [14, 15] and changing domains. So we have to find a method which only modified the mesh but does not regenerate. Our method is supposed to be as follows.

The method described in the paper is a direct method and can create meshes consisting only on quads for simply-connected domain. It was inspired by the theory of elasticity, the theory of conformal mapping [16], and the theory of neural net [17, 18].

The main point of the method is that we have a mesh in an initial domain. The initial domain can have a simple geometry, and it is easy to generate a mesh on it, or a mesh can be created for the initial state of the studied solid by using the techniques described above. Than if we have a mapping between the boundary of the initial domain and the boundary of the domain, where we would like to make a mesh, we have a mapping between the whole domains. And finally we can project the initial mesh on the manifold which we are interested in.

We do not have an exact mapping in the method, but only an approximate one. But if \( \| \partial \Omega(t) - l(t) \| < \varepsilon \), where \( \partial \Omega(t) \) is the boundary of the target domain, \( l(t) \) is a curve which approximates the boundary, \( t \) is a parameter, and \( \varepsilon \) is the required precision, then one can say that we have a mapping between the initial and target boundaries.

**3. Description of the method**

We use a system of springs to obtain mapping inside the domains. Let us assume that we have such a system in the initial domain free of stresses and deformations. If we change the position of each point belonging to the initial boundary, i.e., they have some displacements, then the inner points change their positions correspondingly. We write the elastic energy of the system as

\[
E = \sum_{i=1}^{K} \frac{k_i [(u_{i1}^x - u_{i2}^x)^2 + (u_{i1}^y - u_{i2}^y)^2]}{2} + \sum_{j=1}^{N} \lambda_j (u_{ij} - \tilde{b}_j)
\]

where \( K \) is the number of springs, \( k_i \) is the stiffness of the \( i \)th spring, \( u_{i1}^k \) is the displacement at the first side of the spring along the axis \( k \), \( u_{i2}^k \) is the displacement at the second side of the spring along the axis \( k \), \( N \) is the number of the boundary points, \( \lambda_j \) are Lagrangian coefficients, and \( \tilde{b}_j \) is the value of the displacement of a point on the boundary. Only the boundary conditions for displacements are used, because the explicit position of the target boundary is known.

The displacement of each point is found from the statement

\[ E[\bar{u}] \rightarrow \text{min}, \]
which is equal to
\[
\begin{aligned}
\frac{\partial E}{\partial u_i^k} &= 0, \\
\frac{\partial E}{\partial \lambda_j} &= 0.
\end{aligned}
\]
Thus
\[
\begin{align*}
\frac{\partial E}{\partial u_k^i} &= \sum_{i=1}^{K} k_i (u_{x_1}^i - u_{x_2}^i) (\delta_{i_1}^k - \delta_{i_2}^k) + \sum_{j=1}^{N} \lambda_j^1 \delta_j^k, \\
\frac{\partial E}{\partial u_k^i} &= \sum_{i=1}^{K} k_i (u_{y_1}^i - u_{y_2}^i) (\delta_{i_1}^k - \delta_{i_2}^k) + \sum_{j=1}^{N} \lambda_j^2 \delta_j^k, \\
\frac{\partial E}{\partial \lambda_k} &= \tilde{u}_k - \tilde{b}_k.
\end{align*}
\]
where \(\delta_k^i\) is the Kronecker delta. The equations of system (1) can be rewritten in matrix form
\[
A = \begin{pmatrix} u \\ \lambda \end{pmatrix} = f.
\]
The matrix \(A\) is the following one:
\[
A = \begin{pmatrix} B & \Lambda \\ \Lambda^T & 0 \end{pmatrix},
\]
where \(B\) is the stiffness matrix, \(\Lambda\) corresponds to the Lagrangian coefficients and contains only ones and zeros. Ones are in the places corresponding to \(u_k^i\) belonging to the initial boundary. The right-hand side is
\[
f = \begin{pmatrix} 0 \\ \tilde{b}_i \end{pmatrix}.
\]
In case where the mesh is regular, in a rectangular domain and at all inner points which have only four neighbors, we have the system of linear algebraic equations
\[
\begin{align*}
n_i k_i \tilde{u}_i - \sum_{j=1}^{n_i} k_{I(j)} \tilde{u}_{I(j)} + \lambda_i = 0, & \quad i = 1, K, \\
\tilde{u}_k = \tilde{b}_k, & \quad k = 1, N.
\end{align*}
\]
where \(n_i = 4\) if the point is inner one, \(n_i = 3\) if the point belongs to the boundary but is not a corner point, \(n_i = 2\) if the point is a corner point, \(I\) is the set of indexes of the neighbor points, and the cardinal number of \(I\) is \(n_i\). Different types of points and their neighbors are shown on figure 1, red color – an inner point, blue – a boundary one, green – a corner one.

The mapping between the initial and target boundaries is obtained as the matching of two sets of points \(J_1 \subset \partial \Omega_1\) and \(J_2 \subset \Omega_2\), \(|J_1| = |J_2|\), \(\partial \Omega_1\) is the initial boundary, and \(\partial \Omega_2\) is the target one. The set \(J_1\) also represents the boundary points of the initial mesh, and the set \(J_2\) gives their final positions. If \(X_1 = \{ (x, y) \mid x, y \in \mathbb{R} \}\) is a set of the positions of initial boundary points and \(X_2 = \{ (x, y) \mid x, y \in \mathbb{R} \}\) is a set of the positions of points in the set \(J_2\), then the boundary conditions are \(\tilde{b}_i = X_2^i - X_1^i\).

As was said before, the approximation of the boundaries should be enough precise. And as before, \(\| \partial \Omega_1(t) - l_1(t) \| < \varepsilon_1, \| \partial \Omega_2(t) - l_2(t) \| < \varepsilon_2\), \(l_1(t)\) and \(l_2(t)\) are linear approximations of the appropriate boundary based on the appropriate set \(J_i\). This leads to the following statement.
Statement. All corner points of the boundary should be in the set for the compliance $J$. If not, then one can choose such an $\varepsilon$ smaller than $\|\partial\Omega(t) - l(t)\|$, and thus the condition of precision is broken. This fact is demonstrated in figure 2.

Also an amount of points near parts of the boundary with high curvature should be considerable, otherwise the precision will be low, but this is a common rule for every mesh.

The authors of [19] stated that their algorithm is the fastest one among the methods for quadrangular meshing and gives an estimate of the time required for the mesh generation. The time $t_1$ is $O(n_p(n_e + \ln n_p))$, where $n_p$ is the number of points, $n_e$ is the number of edges. The estimate can be slightly rewritten to compare with the time needed for our method, $t_1 \approx O(n_p^2)$. The time $t_2$ for our method is $O(n_p^2)$. So our method is not slower than the others, but we can improve the time using iteration methods to solve a system of linear algebraic equations.

Here is the algorithm for using this method in problems of growing [20–23].

1. Create a mesh for either a simple domain or the initial domain. If chosen, the initial one go to 3.
2. Project the mesh onto the initial domain.
3. Solve one step of the problem of a growing solid.
4. If there are cells of area larger than the maximum area $S_{\text{max}}$, then create new elements on the growing surface and go to 5.
5. Stretch the mesh on the grown area. The displacement for each point is zero except for points belonging to the growing surface. Here the displacement is $v(\vec{x})\, dt$, where $\vec{x}$ belongs to the growing surface.
6. If this is the last time step, complete the program or else go to 3.
4. Numerical realization

The program was developed to demonstrate the method. It was written using the programming language C++ and QT libraries. One should enter an amount of points of the mesh, scales of the initial rectangle, and the boundary of the target manifold into the program. All springs are considered equal with the same stiffness \( k \). The initial domain with the mesh is shown in figure 3. The result of this work is shown in figures 4 and 5. All results were obtained with the mesh of 10 \( \times \) 10 cells. Both the initial and final boundaries were divided into the same amount of uniform parts, and a mapping was created between these divisions.

The resulting meshes consist of non-uniform cells, and the mesh in the second test near the peaks on the lateral surfaces is rough. Probably, it would be better to use different stiffness for each spring and mapping between the boundaries.

Conclusions

The method is valid only for simply-connected domains, but if other methods are used to create the initial mesh, then it is valid for any domain. There are two possible modifications of the method for multi-connected domains. The first one makes the initial manifold to be a multi-connected one. The second one first makes the mesh not to take the holes in the target domain into account, then deletes every point inside the holes except for the points which are the closest ones to the holes boundary. Then these points are projected onto the holes boundaries, and the mesh is recalculated. An additional requirement is that the mesh cell size should be smaller than the smallest one of the holes.

The same techniques can be used for composite solids, but it is not the optimal way. It is better to apply another method. Let us assume that the solid under study consists of several domains with very different physical properties and the precise boundary surfaces are given. Then one can generate meshes for each domain and modify the meshes near the boundaries to obtain one continuous mesh using a system of springs. Such a modification is required to
Figure 4. The first test.

Figure 5. The second test.
avoid additional restrictions on the boundary points during the modelling, which simplifies the computational process.

Acknowledgments
The author thanks professor A.V. Manzhirov for the statement of the problem and helpful comments. The work was supported by the Russian Science Foundation (project No. 17-19-01257).

References
[1] Manzhirov A V 2016 A method for mechanical design of AM fabricated viscoelastic parts In Transactions on Engineering Technologies ed S Ao, G C Yang, and L Gelman (Singapore: Springer) 223–35
[2] Manzhirov A V 2015 Mechanical design of viscoelastic parts fabricated using additive manufacturing technologies Lect. Notes Engng Comp. Sci. 2 710–4
[3] Manzhirov A V 2014 Mechanics of growing solids: new track in mechanical engineering ASME 2014 Int. Mechanical Engineering Congr. and Exposition, Vol 9: Mechanics of Solids, Structures and Fluids Montreal, Quebec, Canada, November 14–20, 2014 (ASME) V009T12A039
[4] Harris P J 2004 Carbon nanotube composites Int. Mater. Rev. 49 (1) 31–43
[5] Chen L and Xu J C 2004 Optimal Delaunay triangulations J. Comp. Math. 22 (2) 299–308
[6] GVS P R, Montas H J, Samet H, and Shirmhahmadi A 2001 Quad-tree-based triangular mesh generation for finite element analysis of heterogeneous spatial data In The 2001 ASAE Annual Int. Meeting (St. Joseph, Mich.: ASAE) Paper No. 01-3072.
[7] Shontz S M and Vavasis S A 2003 A mesh warping algorithm based on weighted Laplacian smoothing In Proc. of the 12th Int. Meshing Roundtable 147–58
[8] Shimada K and Gossard D C 1995 Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing In Proc. of the Third ACM Symp. on Solid Modeling and Application ed C Hoffmann and J Rossignac (New York: ACM) pp 409–19
[9] Blacker T D and Stephenson M B 1991 Paving: A new approach to automated quadrilateral mesh generation Int. J. Num. Meth. Engng 32 (4) 811–47
[10] Liang X, Ebeida M S, and Zhang Y 2010 Guaranteed-quality all-quadrilateral mesh generation with feature preservation Comp. Meth. Appl. Mech. Engng 199 (29) 2072–83
[11] Lee C K and Lo S H 1994 A new scheme for the generation of a graded quadrilateral mesh Comput. Struct. 52 847–57 Borouchaki H and Frey P 1998 Adaptive triangular–quadrilateral mesh generation Int. J. Num. Meth. Engng 41 (5) 915–34
[12] Owen S J, Staten M L, Canann S A, and Saigal S 1999 Q-morph: An indirect approach to advancing front quad meshing Int. J. Num. Meth. Engng 44 (9) 1317–40
[13] Manzhirov A V 2017 Fundamentals of mechanical design and analysis for am fabricated parts Proc. Manufact. 7 59–65
[14] Manzhirov A V 2017 Advances in the theory of surface growth with applications to additive manufacturing technologies Proc. Engng 173 11-6
[15] Smith W and Volberg A 2011 A conformal mapping example Comptes Rendus Math. 349 (9-10) 511-4
[16] Hodge V J, O’Keefe S., and Austin J 2016 Hadoop neural network for parallel and distributed feature selection Neural Netw. 78 24–35
[17] Kim J, Kim J, Jang G J, and Lee M 2017 Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection Neural Netw. 2017 87 109–21
[18] Remacle J F, Lambrechts J, Seny B, et al. 2012 Blossom-Quad: A non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm Int. J. Num. Meth. Engng 89 (9) 1102–19
[19] Gabow H N 1990 Data structures for weighted matching and nearest common ancestors with linking In Proc. of the 1st Annual ACM-SIAM Symp. on Discrete Algorithms pp 434–43
[20] Manzhirov A V 1995 The general non-inertial initial-boundary value problem for a viscoelastic ageing solid with piecewise-continuous accretion J. Appl. Math. Mech. 59 (5) 805–16
[21] Lychev S A, Lycheva T N, and Manzhirov A V 2011 Unsteady vibration of a growing circular plate Mech. Solids 46 (2), 325–333
[22] Manzhirov A V and Lychev S A 2012 The mathematical theory of growing solids: finite deformations Dokl. Phys. 57 (4) 160–3
[23] Joubert S V, Shatalov M Y, and Manzhirov A V 2013 Bryan’s effect and isotropic nonlinear damping J. Sound. Vibr. 332 (23) 6169–76