A GENERALIZED GRAPH-THEORETIC MESH OPTIMIZATION MODEL

Andrey A. Mezentsev

Department of Earth Sciences and Engineering, Imperial College, London, UK.
A.Mezentsev@imperial.ac.uk

ABSTRACT

This paper presents a generic approach to mesh global optimization via node movement, based on a discrete graph-theoretic model. Mesh is considered as an electric system with lumped parameters, governed by the Kirchhoff's voltage and circuit laws. Each mesh element is treated as a multi-pole electric component, relating input electric potentials to the output via a transfer function. We automatically derive an element transfer function and finally a mesh optimization model using a formal analysis of the coefficients couplings in the finite element stiffness matrix, similar to the method, used in Algebraic Multigrid. Our mesh model is a transient dynamic system and proposed optimization can be also used for mesh deformation problems. We will show that new method works well for realistic 3D meshes and provide a number of mesh optimization examples and details of our implementation.

Keywords: mesh optimization, mesh smoothening, graph-theoretic mesh model, algebraic multigrid, hybrid mesh

1. INTRODUCTION

Most of partial differential equations (PDEs) in science and engineering cannot be solved analytically in closed form and require numerical methods for approximate solution. The Finite Element (FEM) and the Finite Volume (FVM) methods are among the most popular choices for such numerical approximations. Both the FEM and the FVM require domain discretization into a set of geometrical simplicies – a mesh [1]. With increasing complexity of discretized geometry the most efficient way of mesh generation is a fully automatic one. However, automatic unstructured meshing on free form geometry frequently produces meshes of unsatisfactory quality, especially for all – hexahedral and hybrid meshes. As stability and convergence of the FEM and the FVM depend on mesh quality, in recent years a lot of effort has been put into research on mesh quality improvement via node movement, also known as mesh smoothening or optimization [2].

A Posteriori methods of mesh optimization have been developed since early 1970 ([1][2] Chapter 33). Two general approaches based on the mesh proximity criterion are used for optimization – local and global. While local optimization solves optimization problem in the vicinity of a specific mesh node (like Laplacian smoothing [2][3][4]), global optimization is oriented on optimization of mesh quality metrics for an entire domain (like Jacquotte method [5], Dvinsky method [6]). As local smoothing cannot guarantee overall mesh improvement, global mesh optimization is a preferred choice though it is more computationally expensive. Therefore research on efficient methods of global mesh optimization has become an important issue recently.

One of the key aspects of the optimization technique is mesh distortion metrics. There are two main groups of metrics: firstly, related to interpolation error of numerical method and secondly, to condition number of discretized PDE matrices in the FEM or FVM ([1][7]). In the first group *geometric* metrics are popular for mesh optimization (e.g. [8][9][10]). Geometric metrics provide meaningful results, but suffer from a certain lack of generality. Freitag and Knupp [11][12]] proposed a more general approach, based on matrix norms and Jacobian condition numbers, associated with the transformation to a reference equilateral element. This metrics uses a distance of a given element to some ideal element, and it can also be attributed to the first group of metrics. The other popular choice in the first group is a range of *physical* metrics. As a rule it provides some energetic criterion, forcing a physical analog of a mesh to change its state to some state with a minimal energy, corresponding to a globally optimal mesh shape. Different physical models are used in this approach, ranging from a simple lumped spring-mass analogy (e.g. [13]) to complex hyper-elastic models (e.g. Jacquotte [5]). Different formalisms are used to derive a mathematical model of mesh in this group - spanning from an elasticity theory to neural networks [14] and graph-theoretic methods [15].

The second group, condition number related metrics, is very effective, but it is less popular because of its dependence on solved PDE and the general complexity of analytical expressions for bounds of quality measures [7]. Therefore, it would be highly desirable to have an automatic method for the derivation of that metrics type for mesh optimization.

Improving the efficiency of *A Posteriori* global mesh optimization in an attempt to automate the stiffness matrix condition number related metrics, we formulate a new graph-theoretic method. It is based on an electrical model mesh analogy and automatic definition of an electric system topology. The ideal system state is formally defined using a new analogy between the mesh stiffness matrix and cutset matrix (see Addendum for definitions) of the mesh model in a graph-theoretic form.

The paper is outlined as follows. In Paragraph 2 we discuss the graph-theoretic methods for derivation of the global optimization models. In paragraph 3 we present new automatic mesh optimization algorithm and in paragraph 4 discuss the implementation issues. In paragraph 5 we provide the first results of the method application to mesh optimization in 2D and 3D, while paragraph 6 and 7 give future work and conclusions.

2. GRAPH-THEORETIC MESH OPTIMIZATION

As a starting point for our optimization we have taken the procedure of the automatic graph-theoretic systems modeling (e.g. Karnopp et al. [16]). For the electrical schemes, graph theory has been used since Kirchhoff's contributions to network theory back in 1840s, but it was Trent [17] who recognized the generality of the graph theory approach in all physical systems. The key point of this modeling technique is that a linear graph becomes a system graph when it is used in a system model, in our case a mesh model.

Using the graph theory in the context of mesh optimization is not entirely new. Djidjev in [15] proposed a force-based mesh optimization model. The method uses the technique developed in graph drawing. A mesh is represented as a mechanical system with vertices replaced by steel rings and edges by springs that exert repulsive or attractive forces on their endpoints, depending on the edge lengths. The method showed general applicability and robustness of the graph based modeling for mesh global optimization, but as meshes correspond to very large graphs the efficiency of optimization is still a problem.

Independently Mezentsev et al. [18] proposed a generic mesh optimization model, based on functional of energy formulated in terms of electrical variables. The model used the graph-theoretic approach (e.g. presented in [17]) to derive a physical optimization model. As compared to the graph method used by Djidjiev [15], that method permits the use of an arbitrary physical model within unified graph-theoretical framework. It is not limited to ring-force analogy and permits to tune the model according to the requirements of mesh optimization. The other important advantage of that method is it's directly applicability for mesh metamorphosis (e.g. Baker [19]). However, this approach only provided efficient graph-theoretic formalism for automatic generation of different optimization models. It has not introduced new physical or geometric models for mesh optimization.

For the sake of completeness, we will go through derivation procedures of a directed graph-theoretic model of a mesh.

2.1 Nodal directed graph model for mesh optimization

We introduce an equivalent scheme (or network diagram) of a mesh *element* as an electric system (Figure 1). In the simplest case, each node of the equivalent scheme represents X, Y, Z coordinate of mesh node (here – node 1). We use the reference node (Node V, Figure 1), corresponding to the potential reference point of our electrical network.

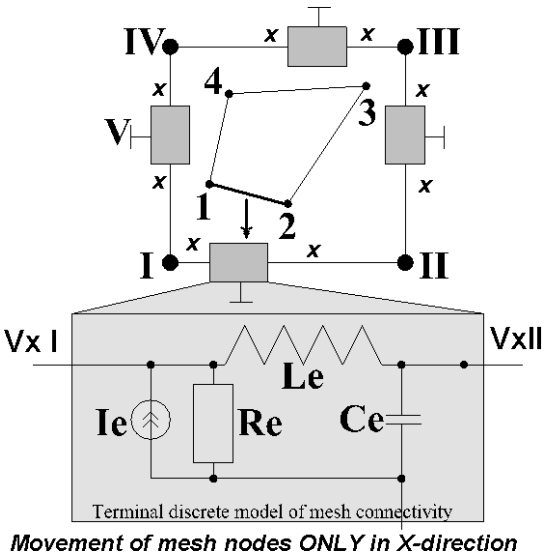


Figure 1. A 2D quadrilateral mesh element and its equivalent scheme. A call out shows equivalent scheme of multi-pole component with single DOF.

Terminal multi-pole *component* model (in Figure 1 – 3-pole component, hereinafter referred as component) links nodes I, II of the equivalent scheme, corresponding to mesh connectivity. Here we show only one electric pole, corresponding to the X-coordinate. Poles, corresponding to the degrees of freedom in Y and Z directions will be the same. A multi-pole component can have an arbitrary internal structure, representing any given physical behavior, i.e. spring-mass, beam or a hydraulic physical model. The user can combine different electrical primitives, realizing a different equivalent scheme of component and creating different physical models of the mesh (see [20]). Here for example we use 4 electric primitives: current source I_e, inductance L_e, resistance R_e and capacitance C_e (shown inside the callout on Figure 1). Note, that in this example we deliberately reproduce the topology of the known spring-mass model [13]. The current in the source can be driven by any of the known optimization metrics, for example it might be proportional to the distance from an ideal element in the chosen metric space.

We use a systematic graph-theoretic approach to automatically derive the model of our electric system, corresponding to the mesh. In terms of graph-theoretic terminology (e.g. [20]), the across variables (or potential type variables in electric systems) will form the vector of unknowns for system's mathematical model. Across variables define through type variables (currents). Any physical system: electrical, mechanical, hydraulic, etc. can be represented in this framework using the analogies of across and through variables. For example, across variables in electrical subsystem – voltages – will correspond to velocities in mechanical subsystem (see Table 1).

Properties of any component or primitive in a physical system are defined by component equations, providing functional dependence on through type variables from across type variables. In the simplest case of 2-pole resistance element the Ohm's law defines the component equation of through variables I from across variables U: $I = \frac{U}{R}$.

Component equations could be linear and non-linear, algebraic, differential or integral. Each multi-pole component of the modeled system has it's own set of component equations, describing specific physical behavior. Component equations are coded by developer/user and are included in libraries in graph theoretic software.

Subsystem	Through variables	Across variables
Mechanic translational	Force	Velocity
Mechanic rotational	Momentum	Angular velocity
Electric	Current	Voltage
Hydraulic/pneumatic	Flow rate	Pressure
Thermal	Heat flux	Temperature

Table 1. Through and across variables

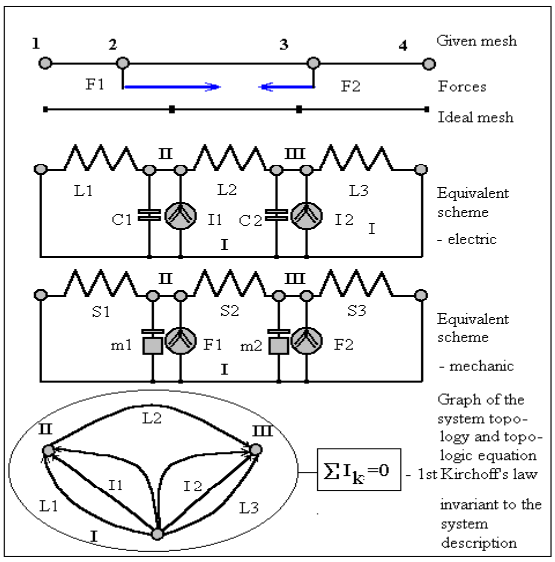


Figure 2. A 1D graph-theoretic mesh model

The lowest level of a multi-pole component hierarchy contains primitives or 2-poles. There are three main types of electrical primitives:

- Dissipation element – R-type element with corresponding component equation $I=U/R$;
- Capacitance element – C-type element with corresponding component equation $I=C(dU/dt)$;
- Inductance element - L-type element with corresponding component equation $U=L(dI/dt)$;

The R-type primitive transforms energy from one type to another, while the C-type and the L-type primitive accumulate potential and kinetic energy of the system.

The discrete system of arbitrary complexity can be represented by a simple combination of R-L-C primitives and sources of current and potential. However, objects with complex component equations have large system graphs and should be modeled using multi-pole components [20]. Using multi-pole approach it is not difficult to derive a more complex mesh optimization models, for example, mimicking properties of different variational models (e.g. [5][10]). Note, that multi-pole component can be not just a superposition of *R-L-C* primitives, but be a "black box" with non-linear transfer function of input across variables to output.

Let us go through the process of the 1D mesh model derivation using 2-pole primitives, as shown in Figure 2. We assume that we do know *a priory* the reference ideal mesh and it will be equally spaced, as shown in Figure 2. The difference in positions of nodes 2 and 3 with respect to ideal state shown below generates the currents *I1* and *I2*. Nodes 1 and 4 of the mesh correspond to the reference node of our discrete model (node I) with constant potential. Optimization of potentials 2 and 3 is based on the minimization of the electric energy functional over capacitances C1 and C2 and inductances L1, L2, L3. The Kirchoff's voltage law (see equation (1)) presents equilibrium of the system. Our mesh optimization model can now be automatically derived through the analyses of the system's graph, also shown in Figure 2. Note, that we have chosen mesh element model (L-C-I) topology and we only use a graph theoretic method to generate the system's model in an automatic way. Currents *I1* and *I2* are taken proportional to displacements of node 2 and 3 of the mesh and will drive the system back to an ideal state.

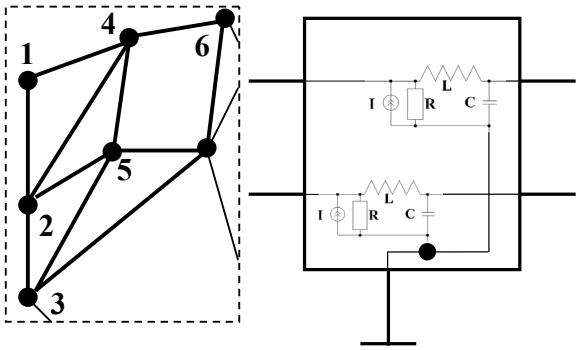


Figure 3. Mesh fragment and multi-pole model of component

The equivalent scheme of mechanical interpretation for the same discrete model is also given in Figure 2 and we can see correspondence of electric and mechanic formulations. Using mechanical analogy, our sources correspond to forces F1 and F2, inductances L1, L2, L3 to springs S1, S2, S3, etc. It is important, that our formulation treats mesh as a generic discrete dynamic system with lumped parameters, not necessarily mechanic. Let us discuss now how we represent mesh using multi-pole components. Figure 3 shows a fragment of a hybrid mesh in 2D with nodes 1–6 and internal structure of multi-pole component as well. Figure 4 presents the equivalent scheme of the given mesh as electric model with multi-pole components.

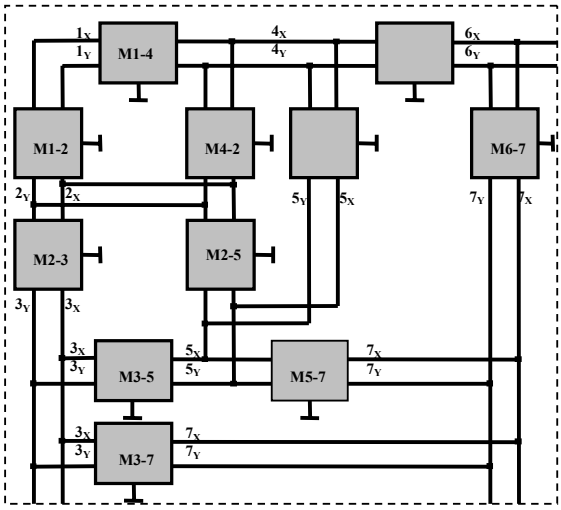


Figure 4. Equivalent scheme of the mesh fragment

Boxes in Figure 4, designated *M n-n* represent 5-pole components, corresponding to mesh edges. Solid links on the equivalent scheme are the electrical connections and poles of the scheme correspond to the x,y-components of the across variables (potentials) at nodes 1-7 of the mesh in Figure 3.

2.2. Automatic graph-theoretic method for modeling of system topology

The overall system's computational model is formulated using the nodal graph-theoretic method. A model is a unification of the component (see also paragraph 2.1) and topological equations [20]. Remember, that component equations provide functional dependence of through type variables from across type variables. Topological equations describe connection of system's components. They are generated using the fundamental energy conservation equation – the first Kirchoff's current law:

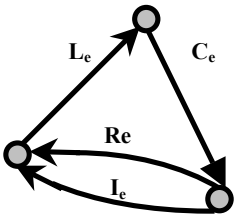
$$I(\varphi) = 0 \qquad (1)$$

where I – is the vector of the through type variables (currents, forces, thermal fluxes), φ - is the vector of across type variables, defining the state of the given system's node (potentials, velocities, temperatures).

We can derive topological equations from (1) using the incidence matrix (see Appendix for definitions) of a system graph:

$$AI = 0 \qquad (2)$$

where I – is the vector of the through type variables on real branches of the system's graph; A - is the system matrix of the system graph, also known as the matrix of contours and cuts. For mesh optimization we need to derive topological equations of our mesh model. This is easily formalized, using the information stored in the cutset matrix of the systems' graph. The cutset matrix M is derived from the graph of our physical system's equivalent scheme and specifically from the graph tree. The procedure of the cutset matrix formation is as following: each chord of the system's graph is included in turn into the tree forming a closed contour of the graph.



	I_e	L_e
R_e	-1	0
C_e	+1	+1

$$M = \begin{bmatrix} -1 & 0 \\ +1 & +1 \end{bmatrix}$$

Figure 5. Graph of the terminal model in Figure 1 and its M-matrix

Then we loop over each contour in the direction of the chord, entering +1 in the row of cutset matrix, corresponding to the chord, when the branch direction coincides with the loop direction and –1 if it does not. For example, Figure 5 shows the system graph of the 3-node terminal model with an equivalent scheme presented in the callout of Figure 1. The table in Figure 5 illustrates the formation of the *M*-matrix for the given system graph.

Comparing the A -matrix in (2) with the cutset matrix M and also taking into account, that each fictious branch with given current has a node, from which it originates, we can observe that

$$A = -M^{t} \qquad (3)$$

Further we can modify the topological equations of the system with respect to potentials U and currents I:

$$MU_{dep} + U_{chords} = 0 \qquad (4)$$

$$I_{branches} - M^{t} I_{chords} = 0 \qquad (5)$$

where U_{dep} – is the potential on dependent branches of the system's graph.

As all branches of the graph's tree are fictious, then $I_{branches} = 0$ and from (2) we receive:

$$M^{t} I = AI = 0 \qquad (6)$$

where I is the through type variable on the real branches of the system graph.

From (3) we can obtain dependence of across variables from through variables on real branches of the system graph tree. As a first approximation, we can represent the voltage drop in dependent branches of the graph as the potential on the branch: $U_{dep} = \varphi$ and therefore

$$M\varphi + U = 0 \tag{7}$$

or

$$A^t\varphi - U = 0 \tag{8}$$

So we include φ to the vector of unknowns in our mesh model, after that component equations are discretized and we impose the restriction on the form of the component equation to be strictly the dependence of through variable from the across variable, i.e. $I = I(\varphi)$ or $I = I(t)$, where t is the time.

Further on the discretized and linearized system of equations for our mesh model will be as follows:

$$\begin{bmatrix} 1 & 0 & 0 & M \\ 0 & 1 & -M^t & 0 \\ Y_{31} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta U \\ 0 \\ \Delta I \\ \Delta\varphi \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ K \\ 0 \end{bmatrix} \tag{9}$$

where $Y_{31} = \dfrac{\partial I}{\partial U}$ the matrix of partial derivatives of the component equations over the across type variables, K – the nodal non-convergence vector of the system. After exclusion of the sub-vectors ΔU and ΔI from (9) we can write:

$$Y\Delta\varphi = -M^t K \tag{10}$$

where $Y = M^t Y_{31} M$ is the matrix of currents derivatives over voltages $\dfrac{\partial I_i}{\partial U_i}$ at node i of the model, $M^t K$ - is the vector of the through type variables in the nodes of the model. Finally we end up with the standard algebraic system of equations:

$$G\varphi = b \tag{11}$$

The main advantage of this nodal formulation is the efficiency of multi-pole models that permits us to represent mesh as a discrete system with smaller system graph, as compared to 2-pole primitives.

Our initial graph-theoretic approach in [18] uses a direct analogy of a mesh topology and topology of a corresponding discrete system with lumped parameters. We can generate a physical model automatically from the mesh description and the mathematical model is derived automatically using any appropriate graph-theoretic software (e.g. PA7, [24]). An accurate choice of component equations for discrete elements, representing the energy links between mesh nodes, resolves the problem of the singular energy functional behavior in the neighborhood of non-convex boundaries and ensures the convergence of the optimization algorithm. Euristic optimization models could be easily changed without any coding effort, using multi-pole elements with different component equations. But this method only provides a systematic and automated way of using different physical models for mesh global optimization. The main drawback is a large system graph for realistic unstructured meshes, making optimization process rather computer intensive. However, this is true for any other global mesh optimization techniques, and therefore it is common to combine global and local optimization methods. In the graph-based force approach Djidjev [15] circumvented the problem of large mesh graphs, using a combination of Laplacian-like smoothening and a global graph optimization method, but that potentially could degenerate to local optimization. As compared to the approach in [15] our multi-pole representation is 2-5 times more memory efficient and it can be used for realistic global optimization. We can handle meshes up to 10^6 elements on a 32-bit PC within 5 hours of computing time. However, we managed to improve our approach further, introducing system graph coarsening and automatic derivation of mesh optimization models for hyperbolic PDEs as discussed below.

3. AUTOMATIC COARSENING AND DERIVATION OF THE GRAPH-THEORETIC MODEL

Let us consider fully discretized FEM problem in a standard matrix form:

$$A \cdot x = b \ or \ \sum_{j=1}^{n} a_{ij} x_j = b_i \ (i = 1,2,...,\ n) \tag{12}$$

where A is the *n by n* coefficient matrix (in the FEM terminology – the stiffness matrix), x – *1 by n* vector of unknowns, b – is a *1 by n* right hand vector. We use formal analyses of the stiffness matrix structure to derive automatically a mesh optimization model in the graph-theoretic form. Unlike in the discussed method with primitives, where topology of a physical mesh optimization model needs to be defined by the user and then computational model is automatically derived using graph-theoretic framework, this approach provides a completely automatic workflow.

The idea is to provide the optimal conditioning of matrix A in the discretized problem (12). A number of publications (see, for example, Shewchuk [7]) show, that the difficulty of solving of linear system (12), typically grows with the condition number $k = \dfrac{\lambda^K_{\max}}{\lambda^K_{\min}}$ of the global stiffness matrix A, where $\lambda^K_{\max}$ and $\lambda^K_{\min}$ are the largest and smallest eigenvalues of A. Global matrix is assembled from an element stiffness matrices K_e and therefore it is roughly proportional to the largest eigenvalue of the stiffness matrices of the elements [8].

Analyzing the typical structure of matrix *A* (12) for different PDEs and meshes and comparing it to the structure of matrix *G* (11) in graph-theoretic model with different realizations of mesh optimization model we observed a systematic similarity of *A* and *G*. It appears possible to directly derive optimization model using formal analyses of matrix *A*. The only concern is high dimensionality of obtained graph-theoretic model (11). In our method we came up with a new approach for reduction of that model. It is based on automatic algebraic coarsening of the FEM matrix *A* (see equation 12) and then the automatic generation of mesh optimization model using (9). For coarsening of matrix *A* we use the Algebraic Multigrid method (AMG) [21][22]. The AMG is successfully applied for solving discretized PDE on unstructured meshes using coarsening and also for certain classes of discrete problems, not arising from differential equations [23].

The coarse matrix A_H is formed from fine level matrix A_h taking a coarse subset C^h from a set of FEM coefficients Ω^h so that

$$\Omega^h = C^h \cup F^h \tag{13}$$

AMG is working with the directed weighted connectivity graph of the matrix *A*. The matrix coefficients (*1,...,n*) are the vertices of connectivity graph. If coefficient a_{ij} not equal to 0, there is an edge e_{ij} from *i* to *j* with weight w_{ij}. The larger w_{ij} the stronger the connection from *i* to *j*. Formally eliminating coefficients with weak connections we can construct a coarse subset C^h. So, defining $\Omega^H = C^h$, the coarse level AMG system will be as follows:

$$A_H x^H = b_H \;\; or \;\; \sum_{l \in \Omega^H} a_{kl}^H x_l^H = b_k^H \;\; (k \in \Omega^H) \tag{14}$$

It is constructed using Galerkin coarsening [22], i.e. coarse matrix A_H is defined as the Galerkin operator:

$$A_H := I_h^H A_h I_H^h \tag{15}$$

where I_h^H - is the restriction operator performing mapping from fine to coarse level, I_H^h - interpolation operator performing mapping from coarse to fine level. In contrast to geometric multigrid, AMG uses no information on the problem mesh. A Galerkin operator is constructed purely algebraically, satisfying variational principle for the coarse subset correction process.

The traditional application of AMG produces a set of coarsening matrix coefficients on different levels for (12). Our experience shows, that one level of algebraic coarsening is sufficient in terms of *A*-matrix coarsening to produce a subset C^h of matrix coefficients, small enough to reduce the optimization problem.

The next step is an automatic derivation of the graph-theoretic mesh optimization model. For our multi-pole terminal presentation we propose a formal correspondence of the A_H matrix and accumulated graph-theoretic model cutset matrix *M* (see paragraph 2.1). Each positive non-zero entry of the coarsened A_H matrix corresponds to +1 value of the cutest matrix *M*, each negative non-zero entry corresponds to –1, zero values are the same. The procedure of optimal automatic definition of sub-matrix Y_{31} is still open to further research in our first prototype (Paragraph 4 and 5) we used the euristic approach, based on the direct scaling of the AMG weights w_{ij} on the coarse level.

Ultimately, our algorithm of the graph-theoretic mesh global optimization using *A*-matrix condition number with AMG coarsening will be as follows:

1. Accumulate a stiffness *A* matrix for a given problem and material properties.
2. Define the condition number of matrix *A* as optimization criterion.
3. Using the AMG coarsening algorithm, reduce the FEM matrix *A* (12) to a coarse subset C^h (equations 13-15).
4. Analyze the structure of coarsened matrix A_H and automatically generate graph-theoretic cutset and nodal conductance matrices.
5. Formally derive a discrete graph-theoretic mesh model corresponding to defined cutest and nodal conductance matrices.
6. Perform a dynamic evolution of the graph-theoretic mesh model up to a steady state. That state will define optimal positions of coarse mesh nodes with respect to the optimal conditioning of matrix *A*.
7. Using I_H^h interpolate coarse level positions to fine level.
8. Check the validity of the optimized mesh. If invalid, correct weights in 4 and return to 5.

It is important, that based on this formulation we derive the mesh optimization model as a union of multi-pole terminal components in a graph-theoretic framework. Together with coarsening, this method permits us to reduce significantly the number of edges in a graph, representing a mesh. Moreover, in some cases it is possible to represent boundary elements with fixed nodal positions and “good” elements as multi-pole components with a reduced number of outer poles. While "bad" elements have as outer terminals *all the nodal potentials* plus an energy pole, the "good" elements will have only energy-related outer node, not contributing to node movement.

4. IMPLEMENTATION

Our implementation of the described mesh optimization model is based on the PA7 graph-theoretic code [24]. This code provides internal meta-language for a definition of modeled system topology and parameters. Exstensive libraries of different electric, mechanic, hydraulic and pneumatic multi-pole models are available for users. These libraries permit us to derive graph-theoretic mesh optimization models directly as discussed in paragraph 2.1. A

number of optimal matrix and numerical integration algorithms are implemented in the code, making graph-theoretic modeling very efficient (for details, see [25]).

Initial 2D meshes for further optimization were generated by the research indirect hybrid code of Dr. O. Hassan (University of Wales, Swansea), initial 3D all-tetrahedral and hybrid meshes were generated by the MezGen unstructured constrained Delaunay mesh generator [25]. MezGen is coded in an Object-Oriented style using polymorphism and meta-templates. It is based on the constrained Delaunay mesh generation and boundary recovery algorithms, developed at the University of Wales Swansea [26]. However, MezGen also provides a number of algorithmic extensions related to indirect hybrid mesh generation and reflecting application of the Standard Template Library - STL. It is also capable of generating prismatic meshes, using a normal offsetting method and has direct CAD interface to ACIS (Spatial Inc.) model.

The analysis of the finite element stiffness matrices is carried out within the framework of the CSP4.0 FEM code, developed by S. Matthäi et al. [27]. For a given PDE accumulation of the problem's stiffness matrix is carried out in a standard way, see [27] and paragraph 5.2 for details.

We use a commercial version 21b1-12 of the AMG code [28] for coarsening. After the formal analyses of the stiffness matrix we generate corresponding graph-theoretic *M*-matrix and automatically derive a meta-language description of the corresponding discrete optimization model using the PA7 code. At this stage, the specific equivalent scheme and structures of multi-pole components of the mesh model are automatically written to a file.

5. TEST CASES

5.1 Graph-theoretic model with primitives

The proposed graph-theoretic optimization technique was tested in our first prototype of the automated mesh optimization module for unstructured meshes in 2D and 3D.

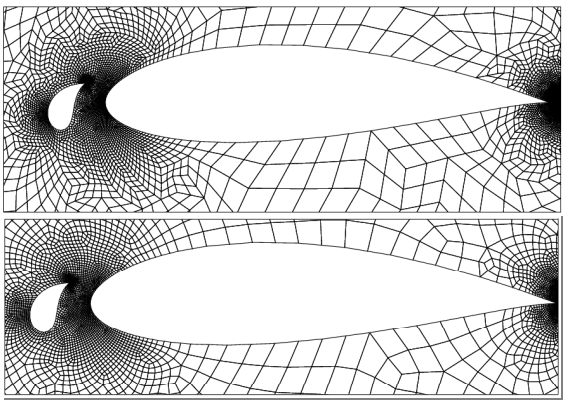


Figure 6. Optimized versus the initial quadrilateral mesh for multi-component airfoil. Graph-theoretic multi-pole optimization model, emulation of the Jacquotte method

Firstly, we checked the generality of our multi-pole graph-theoretic method presented paragraph 2.1 emulating a number of mesh optimization models, described in literature. For example, Figure 6 gives the comparison of mesh quality before and after optimization with the multi-pole graph-theoretic model similar to the Jacquotte method [5].

The multi-pole terminal model uses primitives in its structure to emulate the variational strain based mesh optimization algorithm. It calculates the nodal potentials at each optimization's iteration, minimizing the objective function. Summation of energy over all mesh cells gives total potential energy, stored in the mesh. Minimization of that energetic functional with respect to the across variables of the system is the objective of optimization. The optimal configuration of the system will correspond to the minimal potentials at the nodes and is achieved during Newton-Rahpson iterations.

Model	Nodes	Ele-ments	Angle before Min	Angle after Min	Time, seconds
Hybrid Airfoil	10786	10489	2.54	8.11	28.4
Tet Reservoir	131380	594934	2.0e-3	7.0e-1	1610.0
Hybrid Reservoir	125750	630531	4.5e-3	4.6e-1	1417.4

Table 2. Test cases characteristics

Compassion of the efficiency of the discrete model to the variational model is presented in Table 2. We would like to point out again that the graph mesh optimization model uses rather large system graph, requiring certain computer power for global optimization.

However, our tests show, that due to discussed dimensionality reduction assumptions in the model and high efficiency of graph-theoretic algorithms (paragraph 2.2) proposed optimization technique is nearly 70% faster than the Jacquotte optimization, which we implemented as a benchmark.

5.2 Graph-theoretic Multi-pole component model

The second test presents the efficiency of our fully automatic mesh optimization approach for the solution of the Darcy equation, used in subsurface flow simulation in porous media [29]. It is basically a Laplacian equation with the following definition of the advection velocity *v*:

$$v = \frac{k}{\mu} \nabla P \qquad (16)$$

where P - is pressure in the domain, Pa, k - is permeability, m^2, μ - is porosity of the rock mass.

We solve (16) using the standard Bubnov-Galerkin Finite Element method with linear elements, to define velocity for further transport simulation with a hyperbolic advection-dispersion equation. So, the equation (16) is used in spatio-temporal discretization scheme for multiphase flow simulation in realistic fractured reservoirs [30]. Due to large difference in the permeability tensor k in different regions of the mode (up to four orders of magnitude) spatial discretization results in an extremely stiff system of ordinary differential equations. It is therefore very important to provide quality spatial discretization, as a single poorly shaped element will further complicate temporal discretization of the problem. Here mesh optimization with respect to minimization of the FEM (stiffness) matrix condition number is required. Using the new method of mesh optimization as discussed in Paragraph 2.2, we managed to improve significantly both the geometric and computational quality of the mesh. Table 2 and Figure 7 provide the comparison of the geometric mesh quality of the mesh.

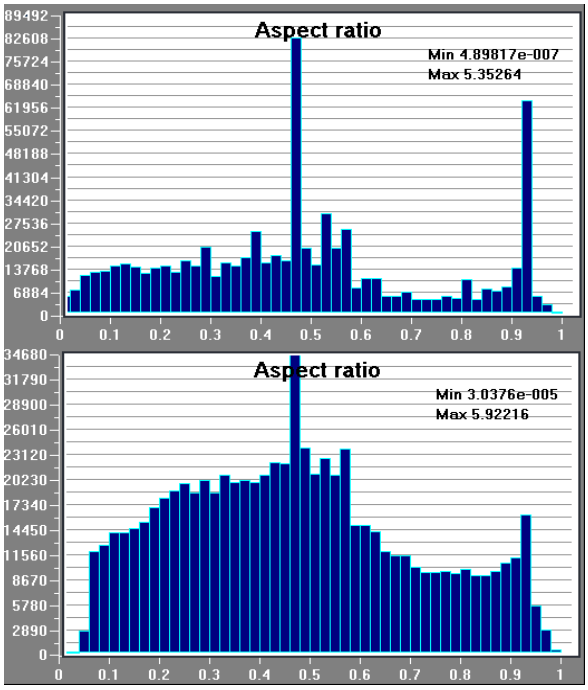


Figure 7. Mesh quality before and after optimization

Figure 8 and Figure 9 shows the elements of the tetrahedral unstructured mesh of the realistic fractured reservoir (mesh generated by the Mezgen code (A. Mezentsev [25]). In the Figures 8 and 9, solid blue outline depicts the elements in the mesh with the element stiffness matrix condition number [16] higher then $5*10^4$ before and after mesh optimization, based on the graph-theoretic model with multi-node terminal presentation of mesh elements.

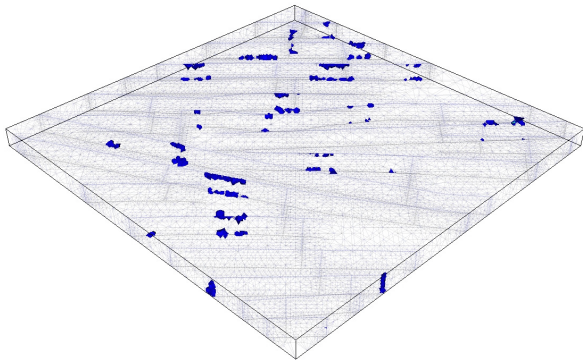


Figure 8. Elements of the fractured reservoir tetrahedral mesh with the stiffness matrix condition number larger than 50000 before optimization – 2874

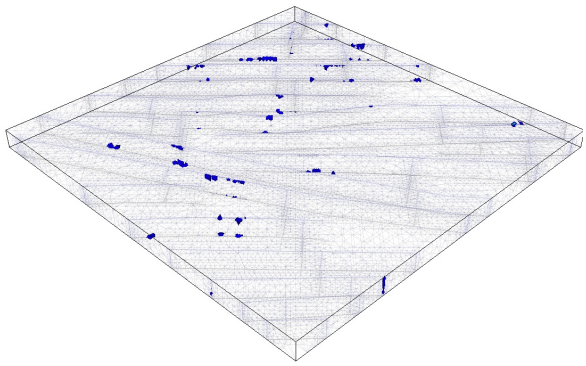


Figure 9. Elements of the fractured reservoir tetrahedral mesh with the stiffness matrix condition number larger than 50000 after optimization – 341

Table 2 provides the CPU times for all discussed optimizations for the SuSE Linux-64 AMD64 3200+ workstation with 3.0 GB. of RAM. As we can see, the efficiency of the mesh optimization permits us to state an applicability of the new method to the complex unstructured meshes with up to 1 million elements on complex free form geometry in Boundary Representation (BREP) format. Our tests show, that for pressure equation (16) proposed mesh optimization permits us to achieve convergence of the applied preconditioned Conjugate Gradient Solver. The solution does not converge without optimization. Using the AMG solver is more straightforward, however standard

settings also fail to provide convergence in numerical solution without mesh optimization.

6. DISCUSSION AND FUTURE WORK

Our first results show applicability of the general graph-theoretic approach to the problem of mesh optimization. It is combining physical interpretation of the optimization process with efficient automatic model derivation technique.

However, though the theory of graph-theoretic method is well formulated for discrete mesh models with primitives and multi-poles, heuristic assumptions of analogy of the FEM stiffness and cutest matrices in the graph-theoretic method require further validation and rigorous theoretical validation. It is also not clear what the limits are of AMG-type coarsening and how stable is the coarse/fine interpolation between meshes in Galerkin operator (equation 15). These questions should be addressed in future work.

Here we would also like to point out the applicability of the new method to the problem of mesh movement and metamorphosis. Due to the inherently dynamic nature of our approach, it appears possible to use a discrete graph-theoretic mesh model instead of Laplacian and elastic variational approaches, considered by Baker [19]. On the other hand, it is also possible to emulate practically an arbitrary model of the mesh element within our graph-theoretic framework. These two aspects of the graph-theoretic model stimulate further research in the dynamic mesh area.

7. CONCLUSIONS

We have developed and tested a new automated graph theoretic method for global mesh optimization via nodal movement. Two main techniques, using direct derivation of mesh optimization model using primitives and multi-pole components are proposed. Both methods are based on a new correspondence between the structure of Finite Element stiffness matrix and the structure of the graph-theoretic *M*-matrix We use also an original coarsening technique, realizing the algebraic multigrid principles. Our generic approach shows its applicability to mesh optimization problems on complex mono-element and hybrid meshes. Easy implementation of different optimization models permits us to tune optimization metrics and optimization function, forming the advantage of the new method. However, the method requires rigorous testing and theoretical proof for the isomorphism of the FE stiffness matrix and the *M*-matrix structures.

ACKNOWLEDGEMENTS

The author would like to thank colleagues from The Moscow State Technical University Prof. I. Norenkov, Dr. V. Manichev, Dr. A. Fedoruk and Dr. M. Uvarov for the provided graph-theoretic software, used in mesh optimization, Dr. O. Hassan and Prof. N.P. Weatherill, from the University of Wales Swansea, who participated in the initial discussions on the method applicability in mesh optimization and Dr. S. Matthäi from Imperial College London, who helped the author to incorporate the optimization module in the CSP4.0 FEM code.

ADDENDUM: DEFINITIONS FROM GRAPH THEORY

A *graph G(V,E)* is a pair of sets *V* and *E*, together with a relation which associates two elements of *V* with each element of *E*. *V* is called the *vertex* set and its elements are called *vertices*. *E* is called the *edge* set and its elements are called *edges*. The two vertices, associated with an edge are called *endpoints*. Two vertices of *G* are said to be *adjacent* if they the endpoints of the same edge *G*.

Sub-graph of a graph *G* is a graph obtained from *G* by deleting edges from the edge set. Sub-graph has an edge set which is subset of the edge set *G*.

Cycle of a graph *G* is a closed route inside the *G*.

Graph tree is connected sub-graph of graph *G* without cycles.

Tree *branches* are the edges of a graph *G*, included in a tree.

Tree *chords* are the graph edges, not included in a tree.

The *incidence matrix* of a graph gives the (0,1)-matrix which has a row for each vertex *v* and column for each edge *e*, and (*v*,*e*)=1 if vertex v is incident upon edge *e*.

The *adjacency matrix* of a graph is a matrix with rows and columns labeled by graph vertices, with a 1 or 0 in position (v_i,v_j) according to whether v_i, and v_j are adjacent or not.

REFERENCES

[1] O.C. Zienkiewicz, R.L. Taylor, "The Finite Element Method", Volume 1, The Basis, Fifth Edition, Oxford, Butterworth-Heinemann (2000).

[2] J. F. Thomson, B.K. Soni, N.P. Weatherill, "Handbook of Grid Generation", CRC Press (1999).

[3] A.M. Winslow, "An irregular triangle mesh generator", *Technical report UCXRL-7880*, National Technical Information Service, Springfield, VA (1974).

[4] D.A. Field, "Laplacian smoothening and Delaunay triangulations", *Communications in Applied Numerical Analyses*, (4) pp.709-712 (1988).

[5] O.-P. Jacquotte, G. Coussement, "Structured mesh adaptation: Space accuracy and interpolation methods", *Comp. methods in Applied Mechanics and Engineering*, Vol 101 pp.397-432 (1992).

[6] A. Dvinsky, "Adaptive grid generation from harmonic maps on Reimannian manifolds", *Journal of Computational Physics*, Vol 63 pp.1-19, (1986).

[7] J.R. Shewchuk, "What is a good linear Element? Interpolation, conditioning, and quality measures",

Proceeding of the 11th International meshing Roundtable, South Lake Tahoe (1999).

[8] I. Babuska, A.K, Aziz. "On the angle condition in the finite element method", *SIAM Journal on Numerical Analysis*, Vol 13(2) pp.214-226 (1976).

[9] D.A. Field, "Qualitative measures for initial meshes", *International |Journal for Numerical Methods in Engineering*, Vol 47 pp.887-906 (2000).

[10] S. Steinberg, P. Roache, "Variational Grid Generation", *Numerical Methods for Partial Differential Equations*, Vol 2 pp.71 – 97 (1986).

[11] L.A. Freitag, P.M. Knupp, "Tetrahedral Element Shape Optimization via the Jacobian Determinant and Condition Number", *Proceeding of the 8th International meshing Roundtable*, South Lake Tahoe, (1999).

[12] P.M. Knupp, "Matrix norms & Condition number, A General Framework to Improve Mesh Quality via Node-movement", *Proceeding of the 8th International meshing Roundtable*, South Lake Tahoe, (1999).

[13] J.T. Batina, "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic meshes", *AIAA Journal*, Vol 28 (8) pp.1381–1388 (1990).

[14] C-A. Bohn, "An adaptive Neural Finite-Element Grid for the Simulation of Light Propagation", *Proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field Simulations*, Ed. B.K. Soni, J. Hauser, J.F. Thompson, P.R. Eiseman, pp.521–529 (2000).

[15] H.N. Djidjev, "Force-directed Methods for Smoothening Unstructured Triangular and Tetrahedral Meshes", *Proceedings of 9th International Meshing Roundtable*, pp.395-408 (2000).

[16] D.C. Karnopp, D.L. Margolis, R.C. Rosenberg. "System Dynamics: A Unified Approach", 2nd Edition, John Wiley & Sons (1990).

[17] H.M. Trent, "Isomorphisms Between Oriented Linear Graphs and Lumped Physical Systems", *Journal of Acoustical Society of America*, Vol 27(3) pp.500-527, (1955).

[18] A.A. Mezentsev, O. Hassan, N.P. Weatherill, K.Morgan, "Mesh Optimization Based on Discrete Graph-theoretical Model", *Proceedings of the 9th ACME Conference*, University of Birmingham, pp.36-40 (2001).

[19] T.J. Baker, "Mesh movement and metamorphosis", *Proceedings of the 10th Meshing Roundtable*, Newport Beach, (2001).

[20] V.A. Trudonochin, N.V. Pivovarova, Mathematical models of technical objects, Moscow, High School Publishers (1986).

[21] J.W. Runge, K. Stüben, "Algebraic multigrid", in *Multigrid Methods*, S.F. McCormick, ed., SIAM, Philadelphia, pp.73-130, (1987).

[22] K. Stüben, "Algebraic multigrid (AMG): An introduction with applications", *GMD Report 53* (1999).

[23] A. Brandt, S. McCormick, J. Ruge, "Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations", Institute for Computational studies, *POB 1852*, Fort Collins, Colorado (1982).

[24] A.A. Mezentsev, M.Y. Uvarov, "Pa7 Dynamic Simulation Environment: Methodology Of Organization and Application", *11th European Simulation Multiconference*, pp.16-20 (1997).

[25] A.Mezentsev. "Mezgen – an unstructured tetrahedral mesh generator with hybrid meshing capabilities", http://www.geocities.com/aamezentsev/Mezgen.htm, http://myweb.tiscali.co.uk/aamezentsev/Mezgen.htm, (2003).

[26] N. P. Weatherill, O. Hassan, "Efficient Three-dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Constraints", *International Journal for Numerical Methods in Engineering*, Vol 37 pp.2005-2039 (1994).

[27] S. Matthäi, S. Geiger, A. Mezentsev. "CSP4.0 user guide", Imperial College London, (2003).

[28] K. Stüben, "SAMG User manual", Release 21b1, Fraunhofer Institute SCAI, St. Augustin, Germany, (2002).

[29] R. Helmig. "Multiphase flow and transport processes in the subsurface: a contribution to the modeling of hydrosystems", Berlin, Springer, (1997).

[30] A. Mezentsev, S. Matthäi, C. Pain, M. Eaton, "A New High Order TVD Transport Algorithm for Hybrid Meshes on Complex Geological Geometry", *Proceedings of the ICFD Conference*, Oxford, UK, (2004).