International Journal of Engineering and Applied Sciences (IJEAS)
Vol.1, Issue 1(2009)43-56

NUMERICAL SOLUTION OF SEEPAGE PROBLEM USING QUAD-TREE BASED TRIANGULAR FINITE ELEMENTS

B. Alyavuz, Ö. Koçyiğit, and T. Gültop
Department of Civil Engineering, Gazi University, 06570 Ankara, Turkey
Corresponding author. Tel.:+90 312 582 3231; Fax:+90 312-231-9223
E-mail address: tgultop@gazi.edu.tr (T. Gültop)

Accepted Date: 2 May 2009

Abstract

A triangular mesh based on the quad-tree grid is applied in the finite element solution of seepage flow under a sheet pile. After obtaining the quad-tree grid, cells are directly transformed into triangles by dividing a cell into four to eight triangles. Cells at the boundaries are turned into triangles using the Delaunay criterion for cell corner nodes and intersection nodes. Different mesh arrangements are considered in order to compare the flow characteristics with changing mesh size. Mesh patterns and results from finite element method are presented graphically for two test cases.

Keywords: Seepage, quad-tree grid, finite element method, triangulation

1. Introduction

As the numerical methods in the field of engineering applications have been developed to a considerable level and the computational capacity of modern computers greatly improved, the number of numerical modeling studies has also significantly increased. Depending on the features of the case studied, one, two or three-dimensional numerical models have been developed. Primarily, these models should be computationally efficient, and easily implemented with a satisfactory level of accuracy for most practical applications. In the implementation, many factors may affect the results of numerical model due to engineering problem diversity. One major factor is the mesh type which may be categorized mainly into two groups such as structured or unstructured meshes.

Structured meshes offer a simple and efficient approach for the solution of engineering problems using finite element and finite difference schemes. It is easier to access neighboring cells when computing a finite difference stencil. However, numerical models based on structured meshes have certain shortcomings. They are often unable to resolve features of a complicated geometric domain resulting in poor accuracy in the model predictions. Similarly, they become inefficient in regions where high velocity or concentration gradients are present due to the lack of local adaptation as a smaller grid size has to be used throughout the whole flow domain. Moreover, numerical problems may arise at the boundaries because of the poor resolution, producing excessive diffusion. Thus grid generation algorithms are required with which a mesh modified to local features can be generated easily and local refinement can be controlled. To achieve a fine resolution at boundaries and in regions having complex geometric features, unstructured grid techniques have been developed [1-3] where the spatial geometry of the problem needs to be approximated with a greater accuracy than a regular rectangular structured grid. One of the unstructured mesh generation techniques is based on

the quad-tree grids which offer a simple and efficient approach including adaptive type of procedure for the solution of the water engineering problems using finite difference schemes [4-10].
The quad-tree algorithm has been increasingly applied in engineering problems after first developed by Finkel and Bentley [11] in 1974, including image analysis [12], set up quad-tree grid generation methods for general computational fluid dynamic applications [13, 14], adaptive quad-tree models to solve shallow water equations [6, 7], and finite element mesh generation [15]. Afterwards quad-tree algorithm has been used as a method for creation of triangular unstructured meshes in various researches [16, 17].

Because of the importance of seepage and its influence in designing and building engineering works [18], triangular mesh based on quad-tree grid was considered in this paper to create flow region or flow net in which a graphical representation of the family of streamlines and their corresponding equipotential. Related details of the subject are given by Harr [19], and Wang and Anderson [20]. The seepage problem needs to use unstructured mesh since high potential gradients exist in the problem domain.

Quad-tree generation method, the triangulation procedure, formulation of seepage problem with finite element method incorporated with the quad-tree and solution process described in detail in this paper. The results from regular finite difference solution are also illustrated by graphical representations. Two test cases have been chosen to validate present solution method. The first test case is the seepage flow under a sheet pile and the second one is seepage flow into a cofferdam. Boundaries have been employed as quad-tree seeds in both examples. Equipotential and flow lines obtained from the solution of governing differential equation under specified boundary conditions are presented herein.

2. Quad-tree grids and triangulation

Mesh generation dates back to the beginning of finite element method in which the problem domain is divided into a finite number of geometric shapes such as triangles and quadrilaterals in two dimensions, tetrahedra and hexahedra in three dimensions. Among the examples of mesh generation, triangulation forms structured or unstructured interconnected

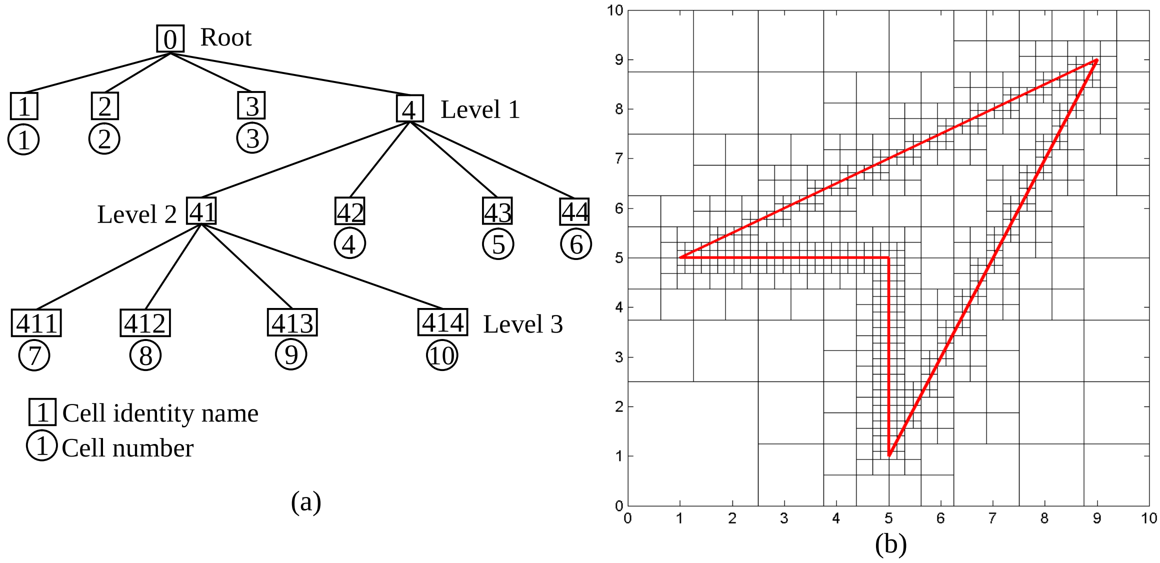


Figure 1. a) Representative quad-tree structure and b) quad-tree grid of test example

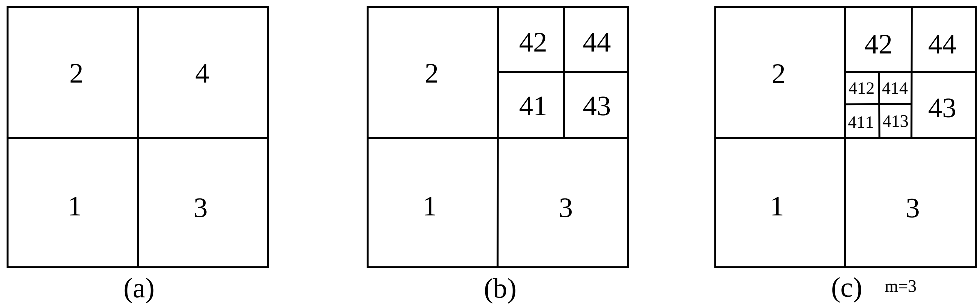


Figure 2. a) Level-one, b) level-two and c) level-three quadrants and their cell identity names

triangles over the problem domain. Another example is the quad-tree grid which differs from triangular or quadrilateral meshes in that a quad-tree grid contains certain hanging nodes. These nodes are corners of smaller elements which are placed along the edge of an adjacent element instead of its corner.

Although the quad-tree grids and triangular meshes are of two different kinds of domain discretization methods, one can be used to construct the other. In the present work the triangular mesh employed in the solution of a seepage problem with the finite element method is generated from a quad-tree grid. We limited the number of hanging nodes to one in order to ensure a gradual change in cell size. It also helps to improve the quality of triangles. This restriction is called as 2:1 ratio rule or balancing condition.

2.1. Quad-tree grid generation

The basic idea behind the quad-tree grid generation is the recursive division of a square cell called root cell into four equal sized child cells until the maximum level of refinement is obtained. This repeating process can be shown by a tree of cell identity names shown in Fig. 4-a where the brunches represent the division. A child cell in this tree becomes a leaf cell, which is not subdivided further, or a parent cell which is going to be divided in the next layer.

In general, the problem domain is enclosed within a square cell. If a cell contains any boundary segment, that cell should be divided into four sub-cells. Cells other than boundary segment containing cells, if necessary, should be divided in order to satisfy the 2:1 rule. According to this rule, a cell can only be adjacent to two smaller cells. The cell “2” in Fig. 2-b is a neighbor of cells “41” and “42”, and in Fig. 2-c it is neighbor to cells “411”, “412”, and “42” where the 2:1 rule is violated. In such a situation the cell “2” has to be divided. A complete quad-tree grid after six-level of refinements is shown in Fig. 1-b.

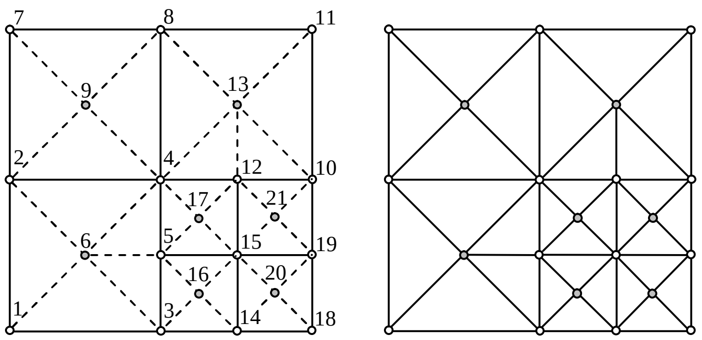


Figure 3. Node numbering and triangular mesh generated from quad-tree grid

An essential step in mesh generation process is the numbering of cells and nodes which makes available the mesh to be used in the numerical method. In the present work, a cell identity name is assigned as the cell is created. These names are formed from ***n*** digit numbers where ***n*** is the level number of the cell. Leaf cells at each level are numbered from left to right as shown in Fig. 1-a.

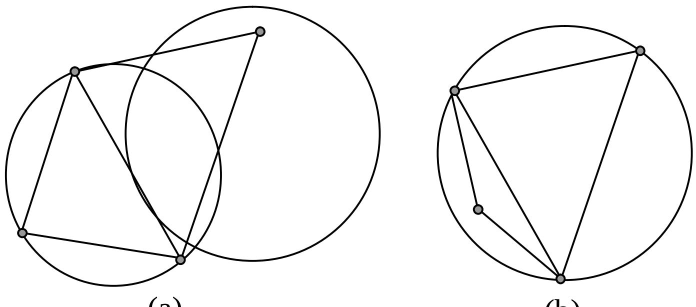
2.2. Triangulation based on quad-tree grid

Generation of triangles can be carried out by a number techniques including advancing front method [21, 22], Delaunay triangulation [23-25], and quad-tree grids [16]. All of these methods give unstructured meshes having generally irregular orientation of triangles. They are flexible in fitting complicated domains and they offer smooth transitions from large to small elements under certain conditions.

Dividing a quadrilateral into triangles is a common and well known approach. A similar way is the division of quad-tree cells into triangles. This can be summarized as follows; once the quad-tree grid is established over the problem domain, the triangulation is obtained from each cell by dividing the cell into four to eight triangles. If the cell is adjacent to a cell with the same size, then four triangles are generated in that cell. If it is neighbor to two smaller cells at one edge, a hanging node occurs at the mid edge of this cell. In such a case, one of the four triangles adjacent to smaller cells is also subdivided into two more triangles (Fig. 3). Generated triangles are stored with their node numbers ordered in counter clockwise direction. It should be noted that quad-tree grid obeys 2:1 dimension ratio between adjacent cells. This ratio limits the number of hanging nodes at the cell edge to one.

Corners, mid edge, and center nodes are numbered for each leaf cell. The neighboring cells should be determined in order to avoid duplicated node numbering. Here, we used an indexing system, which helps to identify the neighbors of a cell, proposed by Cruz [5] where every cell is denoted by three integer numbers associated with row, column and level information. Whole domain is divided into m numbers of rows and columns, where m=2^n and n is the level number.

Boundaries of problem domain generally pass through the points other than nodes which are used to construct the triangles. Therefore, another triangulation scheme is needed for the cells near the boundaries. To handle the boundaries, we utilize the Delaunay criterion. First, cells containing boundary line segments are identified. Then, each members of this group of cells is treated individually. After determination of intersections of boundary lines and cell edges, nodes at the intersection points and corner nodes of the cell are triangulated according to a



(a) (b)

Figure 4. a) Triangulation which obeys empty circle criterion, b) triangulation which does not obey the criterion

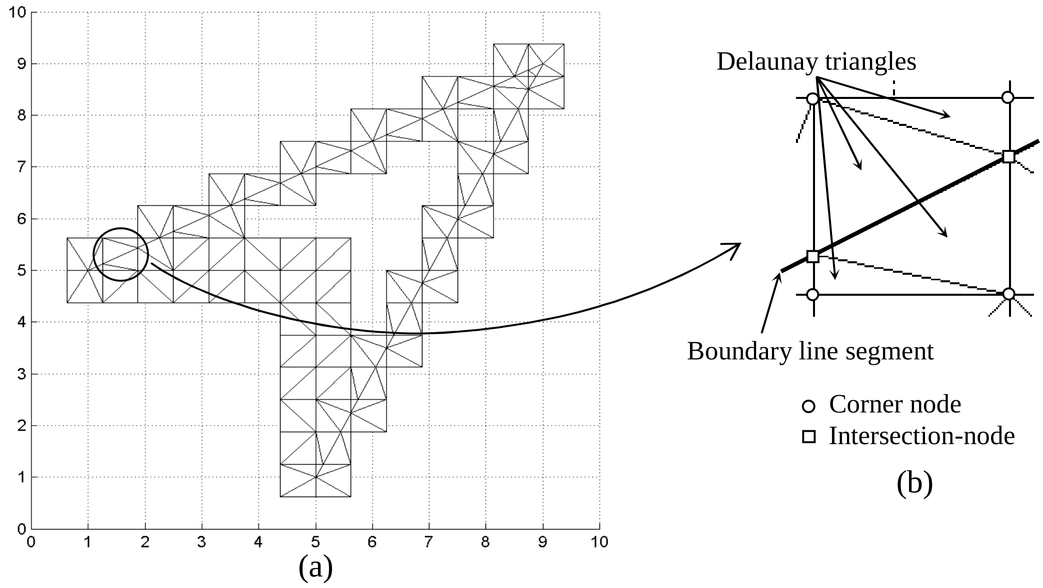


Figure 5. Test example; a) boundary line containing cells , b) nodes and Delaunay triangles for a chosen cell

rule called Delaunay rule or empty circle criterion. According to this rule, in a circumcircle of a triangle there must be no node other than corner nodes of the triangle as shown in Fig. 4. Nodes in the cell are scanned from first node to the last node selecting three candidate nodes which will obey empty circle criterion.

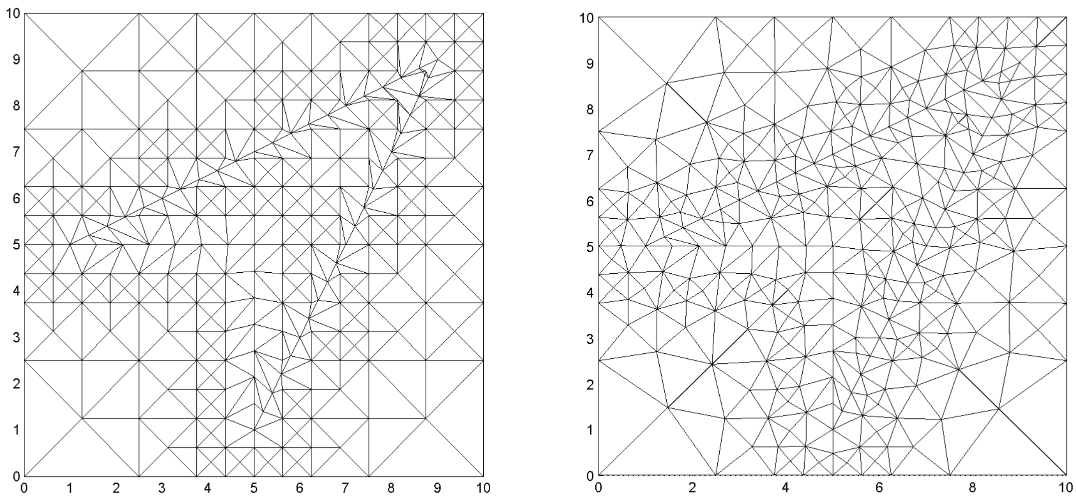
Fig. 5-a shows the cells which intersect the boundary lines. At certain locations, the distance between intersection-nodes is immoderate causing possibly bad shaped triangles adjacent to the boundaries of problem domain. In order to eliminate this issue, we define a line smoothing procedure which produces equally spaced boundary nodes. The nodes on a boundary line are repositioned after the triangulation of each boundary containing cell. A triangulation after the application line smoothing procedure can be seen in Fig. 6-a. Although nodes on the boundaries are repositioned, it can be seen that there still exists a requirement for the adjustment of free nodes, i.e. nodes other than boundary nodes, to get rid of needle shaped triangles.

The triangular mesh obtained from quad-tree grid can be improved using some smoothing techniques which make certain relocations of adjustable triangle vertices. For example, Laplacian smoothing method [26, 27] adjusts each free node by moving the node to a position which is the arithmetic average of node locations adjacent to it. This technique is inexpensive in terms of computational time and very powerful for two dimensional problems. The displacement corresponding to the relocation of the node is,

$$\Delta u = \frac{1}{n}\sum_{i=1}^{n} U_i \, , \tag{1}$$

where $U_i = (x_i - x_0, y_i - y_0)$ and n is the number of adjacent nodes. New location of the node is,

$$(x, y)^* = (x, y)_0 + \Delta u \, . \tag{2}$$



(a) (b)

Figure 6. Test triangulation a) before and b) after Laplacian smoothing

Fig. 6-b shows the final triangulation based on the quad-tree grid of test example after line and Laplacian smoothing procedures.

3. Formulation of Seepage Problem

A number of phenomena including steady state heat flow, electrostatics, torsion of elastic rods, and flow of viscous liquids are governed by the Laplace equation with a single variable. For two dimensional flow, the assumption of homogeneous soil and laminar flow conditions let the governing equation be the Laplace equation.

3.1. Governing differential equation

The governing differential equation of seepage flow is given by following second order elliptic differential equation,

$$k_x \frac{\partial^2 h}{\partial x^2} + k_y \frac{\partial^2 h}{\partial y^2} = 0 \tag{3}$$

where h is total head, k_x and k_y are the coefficients of permeability along the x and y directions, respectively. The governing differential equation is also written in terms of velocity potential function as,

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0, \tag{4}$$

where $f(x,y)$ is a potential function from which two velocity components are derived as

$$v_x = \frac{\partial f}{\partial x} = -k_x \frac{\partial h}{\partial x} \text{ and } v_y = \frac{\partial f}{\partial y} = -k_y \frac{\partial h}{\partial y}, \tag{5}$$

for homogeneous, anisotropic soil. A water particle flows through the curve $y(x,y)$=*Constant* which defines a flow line perpendicular to the curves, i.e. equipotential lines, along which $f(x,y)$=*Constant*. The stream function $y(x,y)$ also satisfies the Laplace equation, that is,

$$\frac{\partial^2 y}{\partial x^2} + \frac{\partial^2 y}{\partial y^2} = 0 \cdot \tag{6}$$

The quantity of discharge, q, per unit width at the cross section between the flow lines y_i and y_j is given by,

$$q = y_i - y_j \, . \tag{7}$$

3.2. Finite Element Formulation

In the solution of seepage problems using finite element method, we approximate the head function in a linear triangular element by,

$$\bar{h}(x, y) = \mathbf{h}^e \bullet \mathbf{N} \, , \tag{8}$$

where $\bar{h}$ is the approximate function of h, $\mathbf{h}^e$ is the vector containing nodal values of head and **N** is the vector of interpolation functions of a linear triangular element, that is,

$$\boldsymbol{\varphi}^e = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix}, \mathbf{N} = \begin{Bmatrix} N_1 \\ N_2 \\ N_3 \end{Bmatrix}, \tag{9}$$

and components of interpolation function vector are defined by

$$N_i(x, y) = \frac{1}{2A_e}\left[a_i - b_i x + g_i y\right], \tag{10}$$

where $a_i = x_m y_n - x_n y_m$, $b_i = y_m - y_n$, $g_i = -(x_m - x_n)$, $i \neq m \neq n$, permute in natural order. By use of a weak formulation of Eq. 3 Three equations relation between the nodal values h_j and can be calculated from,

$$K_{ij}^e h_j = Q_i^e \, , \tag{11}$$

where the symmetric element coefficient matrix K_{ij}^e is,

$$K_{ij} = \iint_{\Omega_e} \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) dxdy \, , \tag{12}$$

or

$$K_{ij} = \frac{1}{4A_e}\left(b_i b_j + g_i g_j\right), \tag{13}$$

and A_e is the area of the element at hand. **Q** is the boundary vector for the specified inflow or outflow at relevant nodes.

$$Q_i^e = \oint N_i q_n ds\,, \tag{14}$$

3.3. Implementation of boundary nodes

If the boundary is an impervious layer, the specified flow perpendicular to this layer should be zero. Hence, Q_L is zero for the node L on this boundary. If the nodal head is specified at node L, the L^{th} equation is redundant and we need to remove the L^{th} equation from the global system of equations. For all interior nodes the components of boundary vector should be zero.

3.4. Calculation of total flow

The quantity of discharge under the sheet pile is determined as the summation of discrete discharge values between minimum ($y_{\min}$) and maximum ($y_{\max}$) stream-function values.

Velocity components v_x and v_y at each triangular element are calculated from Eq. 5 and the derivatives of Eq. 8 with respect to x and y as,

$$v_x = -\frac{k_x}{2A_e}\sum_{i=1}^{3} b_i h_i\,, \tag{15}$$

$$v_y = -\frac{k_y}{2A_e}\sum_{i=1}^{3} g_i h_i\,. \tag{16}$$

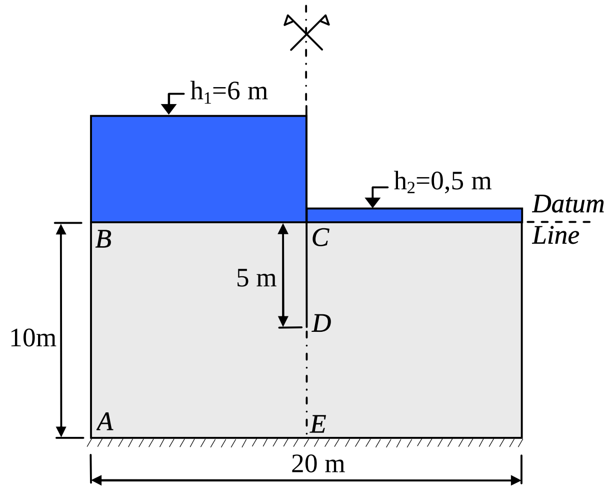
The velocities in Eq. 15 and Eq. 16 are calculated at the center of each linear triangular element. The velocity at a given node is estimated from the velocities calculated at the surrounding triangles.

4. Numerical Implementation

In this section, we suppose two examples of seepage flow under a sheet pile wall. Both problems are symmetric about a vertical line. The horizontal distance to depth ratio is assumed as 2:1. The coefficient of permeability is considered as unity. The boundaries of the first problem lie along the edges of root cell where it is thought that further divisions of cells are only carried out for the cells adjacent to the sheet pile after three iterations. In the second example, four boundary segments of problem domain fall inside the root cell. The iterative divisions of cells are carried out for the cells adjacent the boundaries.

4.1. Sample problem 1

An impervious layer lying 10 m under the ground surface level exists in the seepage problem shown in Fig. 7. Half of the domain is considered in the solutions of differential equations given in Eq. 4 and Eq. 6, because of the symmetry around line *CE*. Boundary conditions for the head and stream functions are taken into account as indicated in the figure. In order to make a comparison between different mesh arrangements, three different triangular meshes are assumed. Type-I mesh is finer at the boundary line *CE* whereas and Type-II has finer



Dirichlet and Neumann type boundary conditions for the head:

$$h = h_1 \text{ on } \Gamma_{BC} \qquad \frac{\partial h}{\partial x} = 0 \text{ on } \Gamma_{AB} \text{ and } \Gamma_{CD}$$

$$h = \frac{h_1 + h_2}{2} \text{ on } \Gamma_{DE} \qquad \frac{\partial h}{\partial y} = 0 \text{ on } \Gamma_{AE}$$

Dirichlet and Neumann type boundary conditions for stream function:

$$y = 0 \text{ on } \Gamma_{AB} \text{ and } \Gamma_{AE} \qquad \frac{\partial y}{\partial x} = 0 \text{ on } \Gamma_{DE}$$

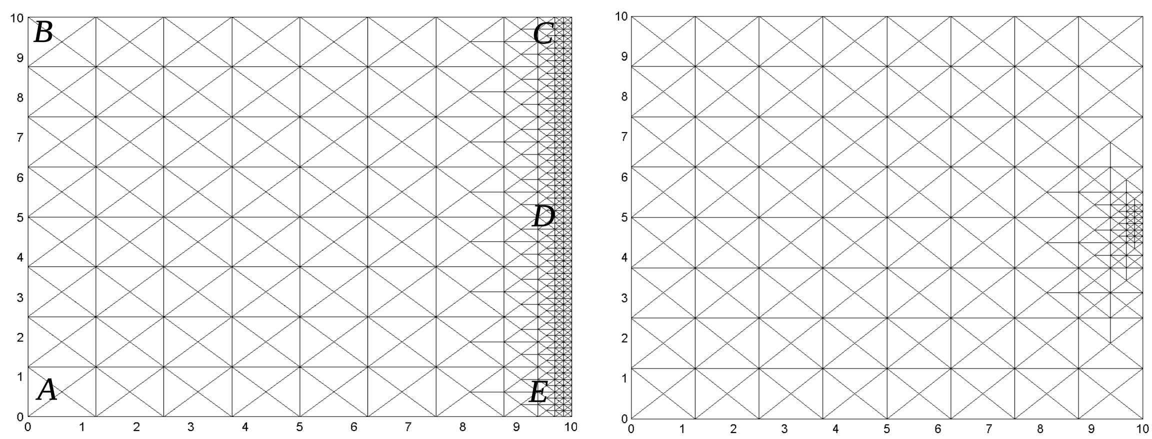
$$y = q_{DE} \text{ on } \Gamma_{CD} \qquad \frac{\partial y}{\partial y} = 0 \text{ on } \Gamma_{BC}$$

Figure 7. Problem geometry and boundary conditions

triangular elements around point *D*. Type-III mesh is a structured mesh which is fine all over the problem domain. Type-I and Type-II meshes shown in Fig. 8 are based on quad-tree grid.

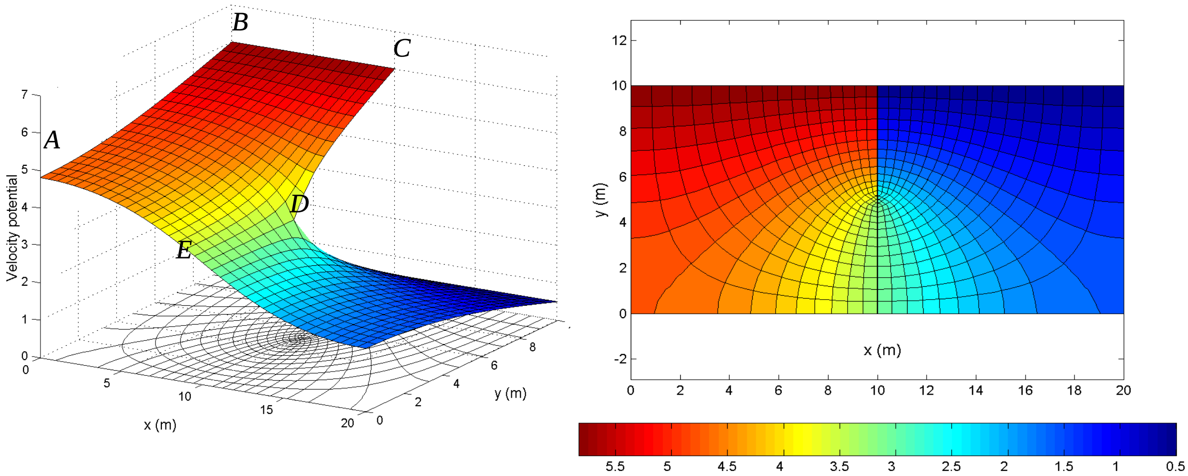
We divide all cells in the first three refinement iterations. After eight refinement iterations in total, Type-I mesh is obtained with 1886 nodes and 3480 triangles whereas Type-II mesh has 255 nodes and 461 triangles. Equipotential and flow lines obtained from the solution of governing differential equation under given boundary conditions are shown in Fig. 9. It is clear that potential function and stream function are conjugate harmonic functions and their constant valued lines intersect at right angles forming an orthogonal flow net as expected.

Plots of flow velocity versus depth at a distance x=5 are shown in Fig. 10. The area under the velocity curve between lines y=0 and y=5 represents the total discharge of the seepage flow which is calculated as 2,41 and 2,40 ($\times k$ m^3/s) for Type-I and Type-II meshes, respectively. A comparison between different mesh configurations and the variation of total discharge with minimum triangular element edge size are shown in Fig. 11.



(a) (b)

Figure 8. a) Type-I and b) Type-II triangular mesh

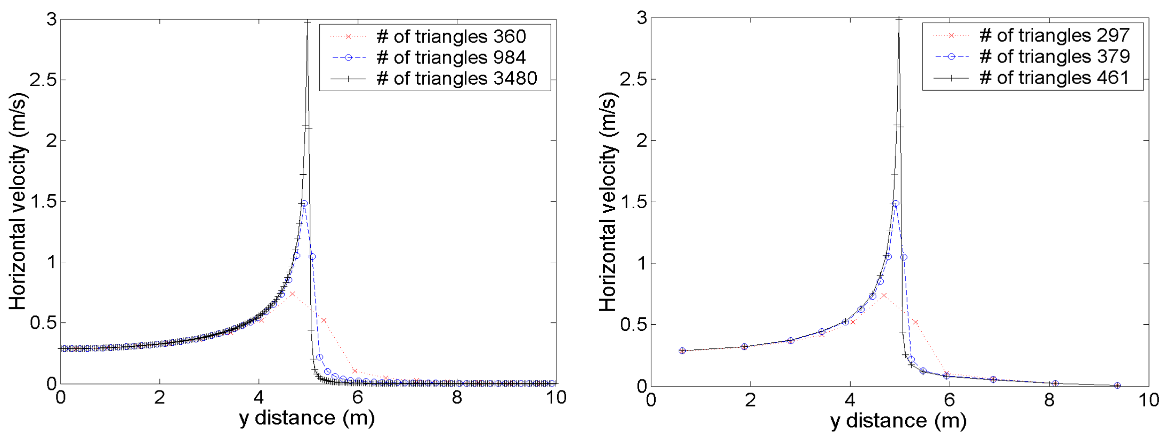


(a) (b)

Figure 9. a) 3-D surface plot and b) 2-D contour plot of function f and y

4.2. Sample problem 2

The second example is the seepage flow into a sheet pile cofferdam shown in Fig. 12. The problem is symmetric around a vertical line passing through *EF*. Total hydraulic head on Γ_{AB} considered as 10m. All boundaries are considered in the process of quad-tree refinement. The quad-tree grid obtained after 6-level refinement is shown in Fig. 13-a. Triangles based on the quad-tree grid are smoothed and triangles and nodes out of the boundaries are cleared to obtain the final mesh shown in Fig. 13-c.



(a) (b)

Figure 10. Horizontal component of flow velocity under the sheet pile, a) Type I mesh and b) Type II mesh

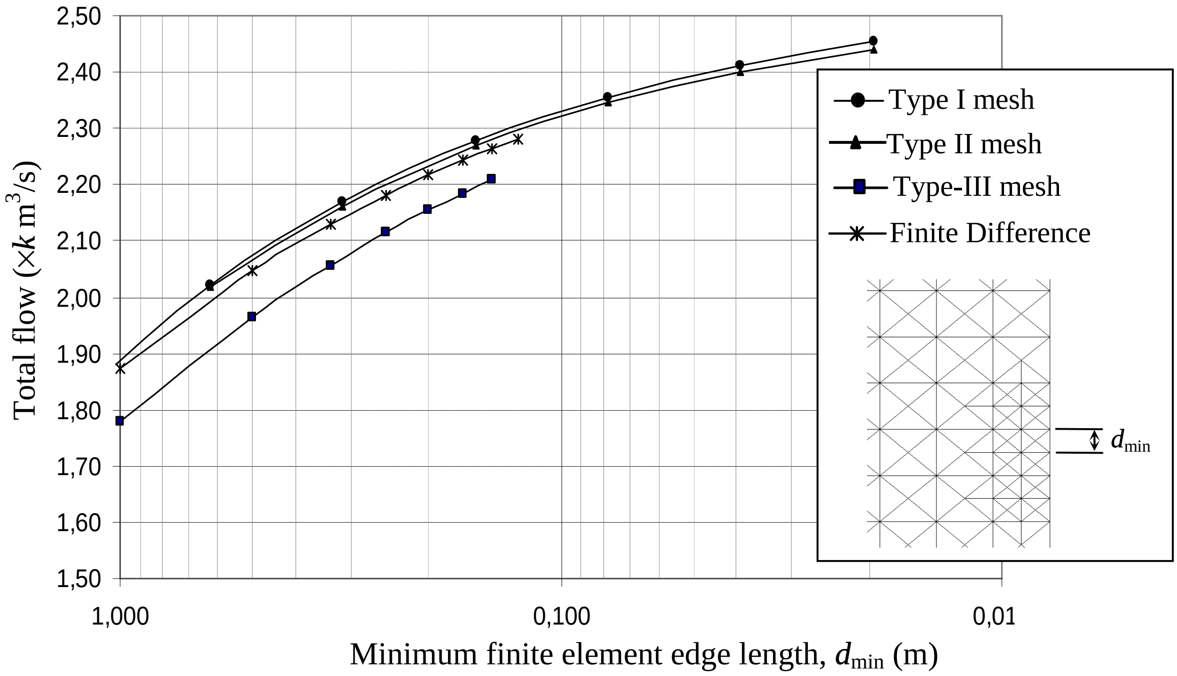
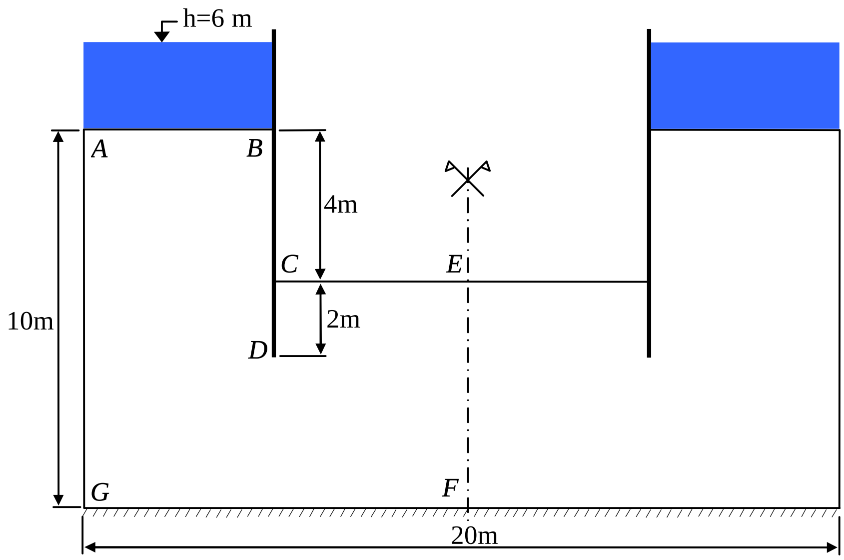


Figure 11. Change of total flow with size of element on Γ_{CE} boundary

5. Conclusion

Unstructured mesh usage is an inevitable tool to describe complex geometries. It is also achievable to get higher accuracy with this type of mesh in case of sharp gradients exist. In this paper, as an unstructured mesh creation technique, both developing the algorithm and formation of the quad-tree grid over the problem domain are found to be straightforward. However, it should be noted herein that if triangular mesh generation technique based on quad-tree grid is employed, 2:1 ratio must be kept between two adjacent cells. Otherwise a gradual transition from large to small triangular elements could not be obtained. This rule is also a requirement for the synchronization of finite difference method and quad-tree algorithm.



$h = h_1$ on Γ_{AB}

$h = 0$ on Γ_{CE}

$\dfrac{\partial h}{\partial x} = 0$ on Γ_{AG}, Γ_{BD}, and Γ_{EF}

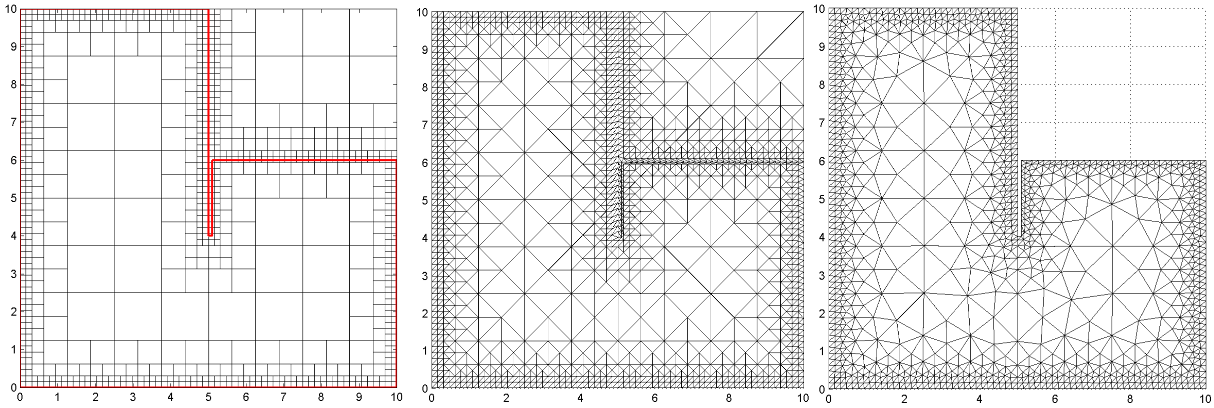
$\dfrac{\partial h}{\partial y} = 0$ on Γ_{GF}

$y = 0$ on Γ_{AG}, Γ_{GF} and Γ_{EF}

$y = q$ on Γ_{BD}

$\dfrac{\partial y}{\partial y} = 0$ on Γ_{AB}, and Γ_{CE}

Figure 12. Problem geometry and boundary conditions



(a) (b) (c)

Figure 13. a) Quad-tree grid, b) triangulation based on quad-tree, c) mesh after post processing

One major difficulty in mesh generation from a quad-tree grid is the formation of triangles at the problem boundaries. In this sense, we have presented a method, which uses the Delaunay criterion, to triangulate a cell containing the boundary line. It is found fairly simple and an efficient way of getting the boundary triangles.

Different mesh configurations show that it is acceptable to get a mesh which is finer only around the tip of the sheet pile, but total flow quantity values from meshes based on the quad-tree grid produce little higher results compared to the structured fine mesh finite difference and finite element method. One major result of this study is that triangular mesh based on the quad-tree grid requires less run time and computer memory compared to the fine meshes over all domain.

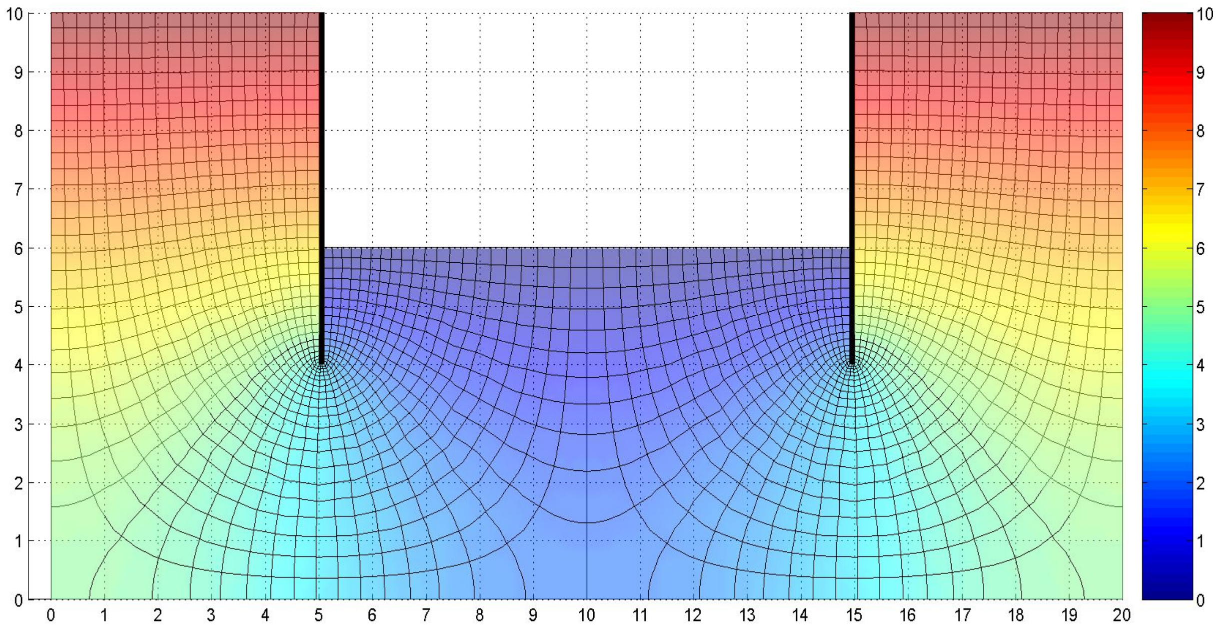


Figure 14. 2-D contour plot of function f and y

In the second test case, it is aimed that the procedure used in this paper has been applied to a different geometry which is slightly complex than first test case. The flow and equipotential lines are shown in Fig. 14. The algorithm used herein is able to easily produce triangular mesh which is finer at the boundaries.

References

1. Basu, P. P., Peano, A., Adaptivity in p-version finite element analysis, *J. Struct. Engng.*, 109, 2310-2324, 1983.

2. Zienkiewicz, O. C., Zhu, J. Z., Gong, N. G., Effective and practical h–p adaptive analysis procedure for the finite element method, *Int. J. Numer. Meth. Engng.*, Cilt 28, 879-891, 1989.

3. Alyavuz, B., Dairesel delikli dikdörtgen levhanin h-tipi sonlu elemanlar ile uyarmali analizi, *Gazi Üniversitesi Müh. Mim. Fak. Dergisi*, 22 (1), 39-46, 2007.

4. Yeh, G. T., Chang, J. R., Cheng, H. P., Sung, C. H., An adaptive local grid refinement based on the exact peak capture and oscillation free scheme to solve transport equations, *Comput. Fluids*, 24 (3), 293-332, 1995.

5. Cruz, L. S., Numerical solution of shallow water equations on quad-tree grids, Ph.D. Thesis, University of Oxford, 1997.

6. Rogers, B., Fujihara, M., Borthwick, A. G. L., Adaptive q-tree Gudunov type scheme for shallow water equations, *Int. J. Numer. Meth. Fluids*, 35, 247-280, 2001.

7. Borthwick, A. G. L., Leon, S. C., Josca, J., Adaptive quad-tree model of shallow-flow hydrodynamics, *J. Hydraul. Res.*, 39 (4), 413-424, 2001.

8. Koçyiğit, Ö., Modelling of water quality and sediment transport in aquatic basins using an unstructured grid system, Ph.D. Thesis, Cardiff University, U.K., 2003.

9. Liang, Q. Du, G., Hall, J. W., Borthwick, A. G. L., Flood inundation modeling with an adaptive quad-tree grid shallow water equation solver, *J. Hydraul. Engng.*, 134 (11), 1603-1610, 2008.

10. Liang, Q., Borthwick, A. G. L., Adaptive quad-tree simulation of shallow flows with wet–dry fronts over complex topography, *Comput. Fluids*, 38, 221 – 234, 2009.

11. Finkel, R. A., Bentley, J. L., Quad-trees: A data structure for retrieval on composite keys *Acta Inform.*, 4 (1), 1-9, 1974.

12. Samet, H., *Applications of spatial data structures*, Addison Wesley Publishing Company, 1990.

13. Yiu, K. F. C., Greaves, D. M., Saalehi, A., Borthwick, A. G. L., Quad-tree grid generation: Information handling, boundary fitting and cfd applications”, *Comput. Fluids*, 25 (8), 759-769, 1996.

14. Wang, Z.J., A quad-tree-based adaptive cartesian/quad grid flow solver for Navier-Stokes equations, *Comput. Fluids*, 27 (4), 529-549, 1998.

15. Greaves, D. M., Borthwick, A. G. L., Hierarchical tree - based finite element mesh generation, *Int. J. Numer. Meth. Engng.*, 45, 447-471, 1999.

16. Bern, M., Eppstein, D., Teng, S.-H., Parallel construction of quad-trees and quality triangulations, *Int. J. Comput. Geom. App.*, 9 (6), pp. 517-532, 1999.

17. Quad-tree-Based Triangular Mesh Generation for Finite Element Analysis of Heterogeneous Spatial Data, ASAE Annual International Meeting, Sacramento, California, USA, 2001.

18. Cedergren, H. R., *Seepage, drainage, and flow nets*, John Wiley & Sons; 3rd edition, 1989.

19. Harr, H.E., *Ground Water and Seepage*, McGraw-Hill, New York, 1962.

20. Wang, H. F., Anderson M. P., *Introduction to groundwater modeling: Finite difference and finite element methods*, Academic Press, 1995.

21. Lo, S. H., A new mesh generation scheme for arbitrary planar domains, *Int. J. Numer. Meth. Engng.*, 21, 1403-1426, 1985.

22. Zhu, Z. Q., Wang, P., Tuo, S. F., Liu, Z., A structured/unstructured grid generation method and its application, *Acta Mech.*, 167, 197-211, 2004.

23. Watson, D.F., Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes, *Comput. J.*, 24, 167-172, 1981.

24. Bowyer, A., Computing Dirichlet tessellations, *Comput. J.*, 24 (2), 162-166 1981.

25. Bern, M., Plassmann, P., *Handbook of Computational Geometry*, Eds: J.R. Sack and J. Urritia, *Elsevier Science*, 303-308, 2000.

26. Field, D. A., Laplacian smoothing and Delaunay triangulations, *Commun. Appl. Numer. Meth.*, 4, 709-712, 1998.

27. Hyun, S., Lindgren, L. E., Smoothing and adaptive remeshing schemes for graded element, *Commun. Numer. Meth. Engng.*, 17 (1), 1-17, 2001.