International Journal of Advanced Geosciences, 1 (1) (2013) 1-12
©Science Publishing Corporation
www.sciencepubco.com/index.php/IJAG

The dispersion algorithm: consideration of bathymetry in 2D mesh generation in ocean modeling

Mohammad A. Feizi Chekab, Parviz Ghadimi*, Hassan Ghassemi

Department of Marine Technology, Amirkabir University of Technology, Tehran, Iran
**Corresponding author E-mail: pghadimi@aut.ac.ir*

Abstract

The Dispersion Algorithm has been further developed and improved to control locally the distribution of nodes in the computational domain. This improvement allows the user to take into consideration the effect of the bathymetry and other physical factors while maintaining the previous global capability of the original Dispersion Algorithm. It has been demonstrated that, by a simple adjustment of the value of the key parameter r(n) resulting in a new parameter R(n, x, y), any physical factor could be taken into account for the application of the proposed Algorithm in different fields of engineering. This concept has been explored to particularly include the bathymetry of the geometrical domain as an important physical factor in ocean modeling. The applicability and efficiency of the proposed algorithm has been demonstrated using various actual geometries. The bathymetry of the Persian Gulf, the Caspian Sea and the Gulf of Mexico have been successfully modeled, analyzed, and meshed by the algorithm.

Keywords: Dispersion Algorithm; Bathymetry; Ocean modeling; 2D mesh generation; Influence coefficient; Node distribution.

1 Introduction

The two dimensional Dispersion Algorithm [1] has been recently introduced by the present authors as an algorithm for two dimensional node distribution which has shown high performance in resolving complex horizontal boundaries. However, in dealing with Ocean Engineering problems, other important physical factors such as bathymetry and wave characteristics should be considered in the node positioning.

The characteristics of water flows are highly dependent on the depth of the water. Therefore, bathymetry, which represents the distribution of underwater depth of lakes or oceans, is one of the most important physical factors to be considered in the ocean engineering problems.

Several researchers in the last decade have discussed the effect of different physical factors on the nodes distribution and mesh resolution. Legrand et al. [2], in 2000, presented an incremental method to generate automatically boundary-fitted Delaunay triangulations of the global ocean. Although, his method does not directly consider the bathymetry in the mesh generation, but it allows local mesh refinement in order to resolve topological or dynamical features. Hagen et al. [3], in 2001, introduced a wave-length based finite element mesh generation and then discussed the influence of bathymetry, especially the steepness of the topography on the mesh resolution. In 2005, Bilgili et al. [4] introduced a Matlab public-domain two-dimensional mesh generation package named BatTri, which considers the bathymetric conditions of the domain in the mesh generation process. Also, in the same year, Jones and Davies [5] used the finite element model (TELEMAC) and demonstrated how a water-depth dependent criterion for determining element size gives an optimal distribution over the majority of the region.

Also, in a very important work in 2007, Greenberg et al. [6] discussed many factors influencing the mesh resolution needed in ocean modeling such as the variations in coastline and bottom topography, the forcing meteorology, the latitudinal dependence of the Coriolis force and the Rossby Radius of deformation. After presenting these factors, the authors of this inspiring paper attempted to qualify and quantify the influence of each factor on the horizontal mesh resolution needed, for a more accurate modeling of the oceans.

In recent years, the concept of relating the elements sizes to different physical factors has been widely used by many researchers [7-23]. Among the most important ones are works done by Robin et al., Jones, Horsburgh, Wolf, Parrish et al. and White et al. Robins et al. [7, 8], used this concept to simulate the water flows and sediments transport in coastal models. Jones [9-13] analyzed the flows and water behavior of the west coasts of Britain. Wolf [16] achieved a coastal flooding analysis by use of this concept. Horsburgh [17] analyzed the impact of tsunami in the UK coastline using an unstructured mesh considering the bathymetry. Parrish et al. [19] incorporated the bottom stress and Coriolis force into 2D mesh generation for shallow water modeling. White et al. [21] developed a three dimensional unstructured mesh where the effect of the bathymetry is taken into consideration.

What can be concluded from the mentioned works is that consideration of the effect of tides and the bathymetry is of prime importance in mesh generation in the field of ocean modeling.

The two dimensional "Dispersion Algorithm" [1] developed earlier by the present authors did not consider the three-dimensional effect or physical factors other than boundary shapes in the distribution of the nodes. However, the present paper does indeed inculde physical factors in general and bathymetry in partiuclar. The current study is an effort to show the simplicity of implementing such physical factors in the Dispersion Algorithm. The proposed idea is then applied to various complex geometries and real applications in ocean engineering.

2 The dispersion algorithm

Since all definitions in the current algorithm will be used subsequently and repeatedly refered to, the Dispersion Algorithm is briefly described first. In Dispersion Algorithm [1], the nodes are distributed on lines called "Emulators". These lines are sketched parallel to the boundaries of the computational domain and the intervals between these lines are described as follows:

$$\begin{aligned} r(n) &= A.\tan(\alpha) \\ \alpha &= B.\frac{n}{N+1}.\pi/2 \end{aligned} \qquad \begin{cases} A \in R^m \\ 0 < B < 1 \\ n = 1, 2, ..., N \end{cases} \tag{1}$$

where r(n) is the distance of the nth emulator from the boundaries and N is the total number of emulators needed by the user. *A* and *B* are the distribution coefficients. Fig.1 illustrates the concept of emulators.

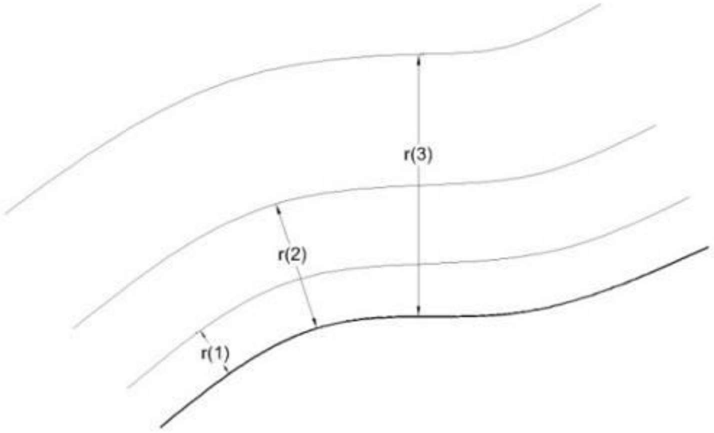


Fig.1: The emulators and the implementation of function $r(n)$ on a sample boundary.

Despite its high performance in manipulating the nodes placement toward the boundaries of the domain, its flexibility in controlling the effect of other physical factors on node distribution has not been demonstrated yet which is an incentive for the current study.

Here, a method is proposed in which the nodes distribution can be manipulated according to the bathymetry of the domain and the extent that the user wishes to take this important physical factor into consideration. To explain the proposed algorithrm, it is necessary to describe the concepts of local and global control.

3 Local versus global control

In the "simple" or original Dispersion Algorithm [1], where the only physical factor to be considered was the geometry of the boundary, the fundamental variable which controls the nodes distribution is r(n), defined by Eq. 1. The role of this variable in the Dispersion Algorithm is to define the distance of a node on the nth emulator to the nearest node. Regarding this crucial role, it is obvious that the node distribution can be locally or globally manipulated by controlling the value of r (n) either locally or globally. When the values of r(n) in different regions are changed proportionally in the entire domain, a global change has occured. However, when r(n) is modified independently in different areas of the domain, a local control has been exercised.

In the Dispersion Algorithm previously explained [1], the quantity of r (n) was controlled globally in the entire domain by changing the coefficient A or B in Eq. 1. Accordingly, a single r (n) was needed to define the nth emulator. However, to take into consideration the effect of a local property such as bathymetry or the wave characteristics, a local control on r(n) is needed. For example, to satisfy the requirements of the bathymetry in ocean engineering applications, r (n) should be greater whereever the domain is deeper, and smaller in the shallower areas. To explain how this local control could be implemented, the Bathymetry as a local physical factor is chosen. However, this concept can be used for any other local property of the domain.

4 Bathymetry in dispersion algorithm

To implement the concept of bathymetry, the definition of r (n) remains the same, but a new variable, R(n,x,y), is defined as follows:

$$R(n,x,y) = E(x,y) * \ r(n) \qquad (2)$$

In which r(n) is defined by Eq. 1 and E(x,y) is a coefficient which represents the elevation of the domain and is intruduced by the equation

$$E(x,y) = \underbrace{1 + c * \left|\frac{Z_{max} - Z(x,y)}{Z_{max} - Z_{min}}\right|}_{\text{Local Propertyof the domain}} \qquad (3)$$

where Z_{max} and Z_{min} are the maximum and minimum elevation of the domain. Z(x,y) is the elevation of the domain at the point (x,y). Also, “c” is a non negative constant coefficient. This constant is the key controller of the level of influence of the bathymetry in the nodes distribution. Accordingly, from now on, it will be called the “influence coefficient”. The influence coefficient is globally used which basically means that it does not change in the entire domain. If the influence coefficient “c” is zero, then the bathymetry does not affect the nodes distribution. This goes to show that as value of “c” increases, the effect of bathmetry becomes more apparent and more significant. It must be noted that, it is only necessary to replace the related part in the Eq. 3 (i.e. replace $\frac{Z_{max}-Z(x,y)}{Z_{max}-Z_{min}}$) to make this method work for another local property.

To demonstrate the application of this concept, a sample geometry is chosen which is depicted in Fig.2.

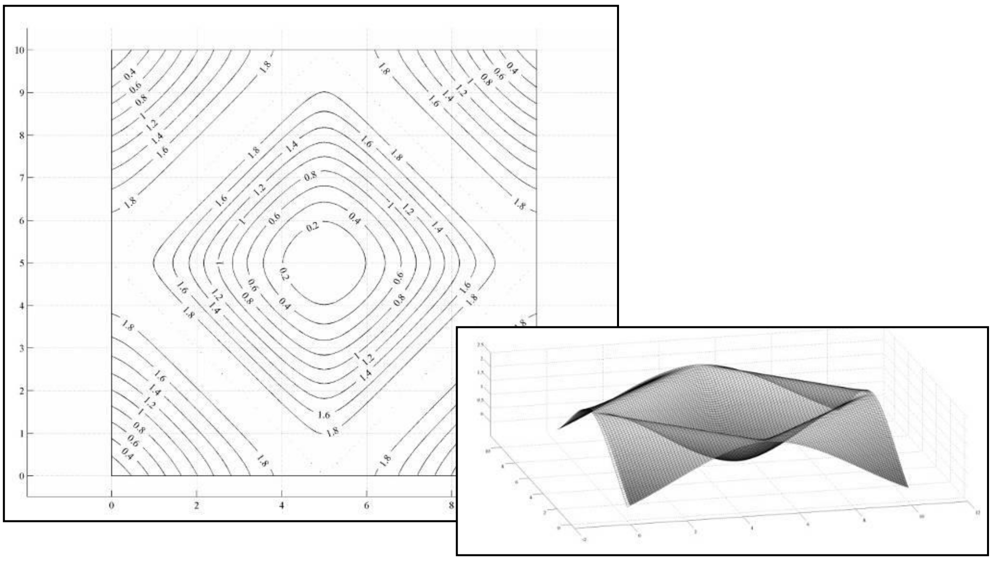
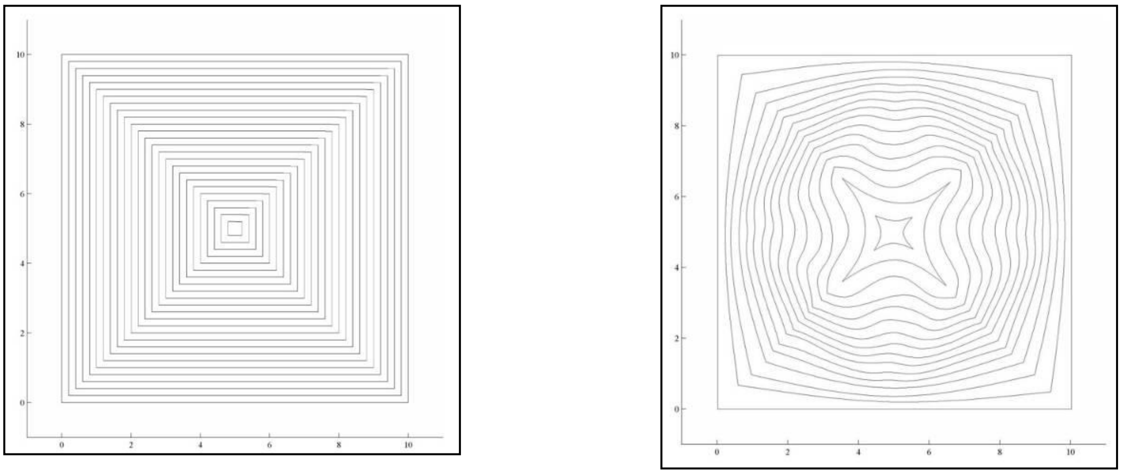


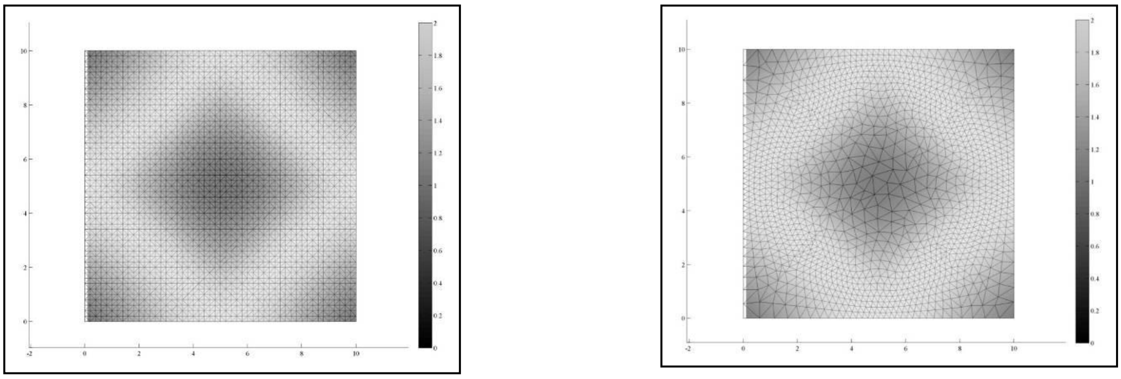
Fig.2: The sample geometry used for illustration.

In the meantime, Fig.3 shows the generated emulators for the sample geometry presented in Fig.2, with and without the influence of bathymetry. The bathymetry effect in the generated mesh is shown in Fig.4.



a b

Fig.3: The generated emulators: a) Without bathymetry (c = 0). b) With bathymetry (c = 1.5).



a b

Fig.4: 2D mesh of the sample geometry:

a) Without bathymetry (c = 0). b) With bathymetry (c = 1.5).

As evidenced in Fig.3, E(x,y) is defined in a way that the minimum size of the elements remains completely under the user’s control as E(x,y) is always greater or equal to 1. So, by this method, the mesh is not refined in the shallow areas but roughened (resulting in coarse mesh) where the domain is deeper.

It can also be seen in Fig.4 that E(x,y) does not affect the elements’ quality significantly. This is so because the emulators are still somehow the replica of the boundaries and the aspect ratio of the elements still follow the same pattern as in the original Dispersion Algorithm.

Because the sample geometry used here is symmetric, to ensure that the applicability and efficiency of the newly proposed algorithm is independent of the geometry, the bathymetry of the same domain has been changed randomly and re-meshed with the same parameters as the above (same number of emulators, same depth coefficient, and same r(n)). The results are then presented in Fig.5. It can be clearly observed that the quality of the generated mesh is not affected by different bathymetries.

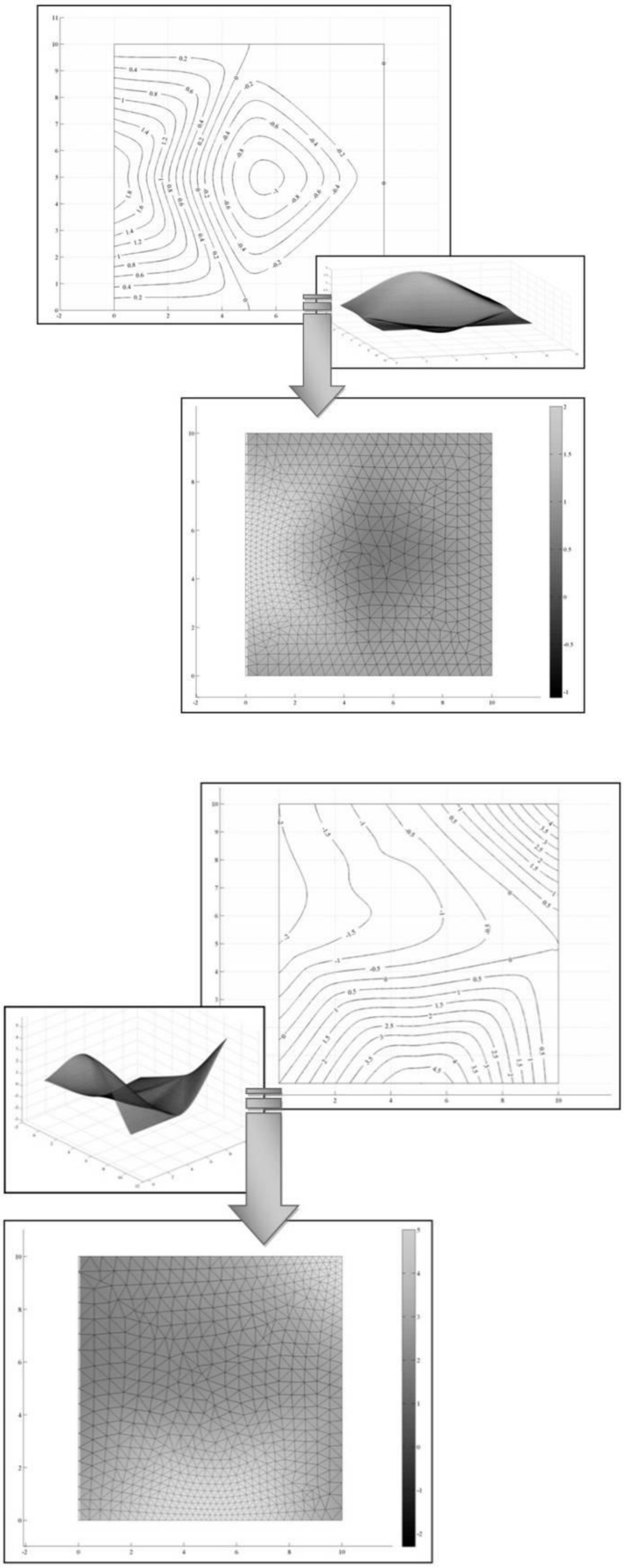


Fig.5: Meshing different bathymetries on the sample domain.

The described method shows that the Dispersion Algorithm can be easily adapted for different fields requiring geometrical modeling as a pre-requisite for numerical computations. Therefore, one can easily claim that this concept of global and local control of r(n), can be used for consideration of the effect of any other physical factors in the future.

5 Ocean engineering application

Although, a sample geometry has been used to describe the efficiency of the algorithm and the effect of the proposed modifications, but still actual examples are needed to verily assess the wider applicability of this method. However, before presenting the modeling of actual geometries such as the Persian Gulf, the Caspian Sea and the Gulf of Mexico, step by step evolution of the spatial discretization is explained and displayed in the next section.

6 Evolution of spatial discretization of complex geometries

Although, geometric modeling of Persian Gulf by the current algorithm will be explained and analyzed in much details in the next section, but here the evolution of its spatial discretization as an example of complex geometry is displayed pictorially. Furthermore, steps involved in achieving the final mesh are outlined.
Firstly, the geometry of the domain and its bathymetry are transferred to the code as demonstrated in Fig.6.

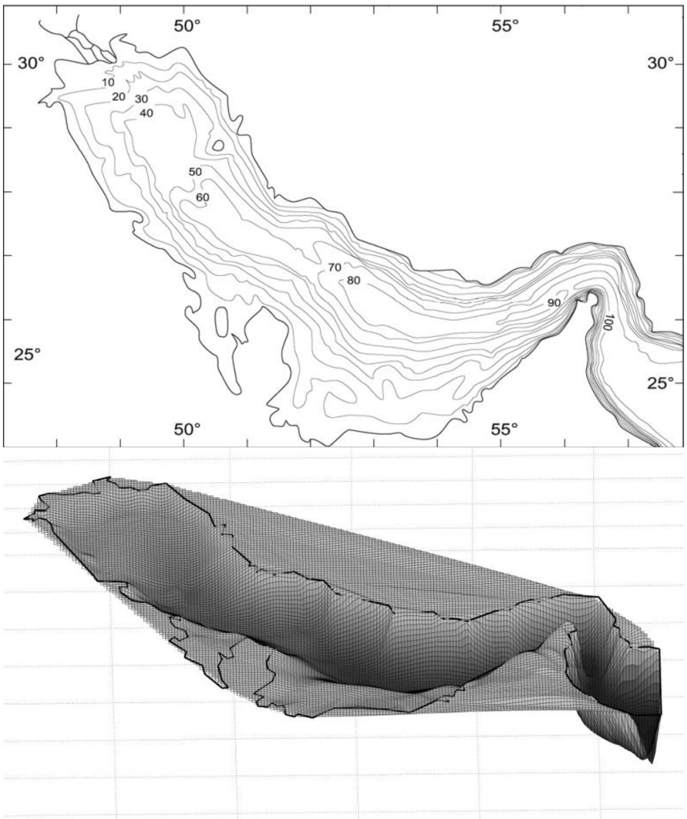


Fig.6: Transferring the geometry and bathymetry [24] to the code.

Then, according to the value of R(n,x,y) defined by Eq. 2, the emulators are generated in the domain. Fig.7 illustrates the emulators for n = 1, 2, 3.

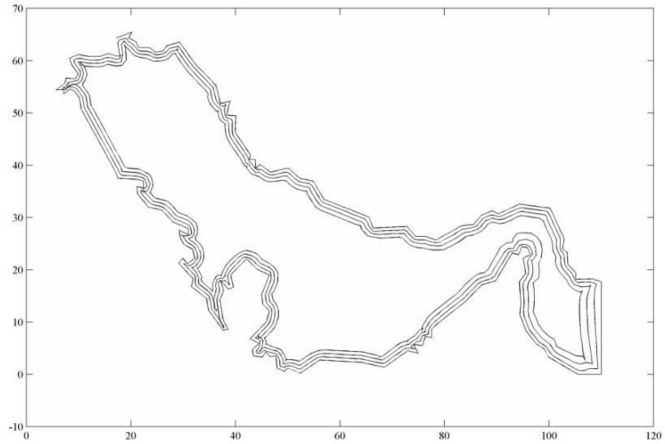


Fig.7: generation of emulators according to R(n,x,y).

Afterward, the nodes are generated on the emulators, dividing the emulators into pieces of length R(n,x,y) according to the number of the emulator, n, and the coordinate of the area in which the nodes are going to be placed. In Fig. 8, the nodes distribution obtained by 15 emulators is displayed.

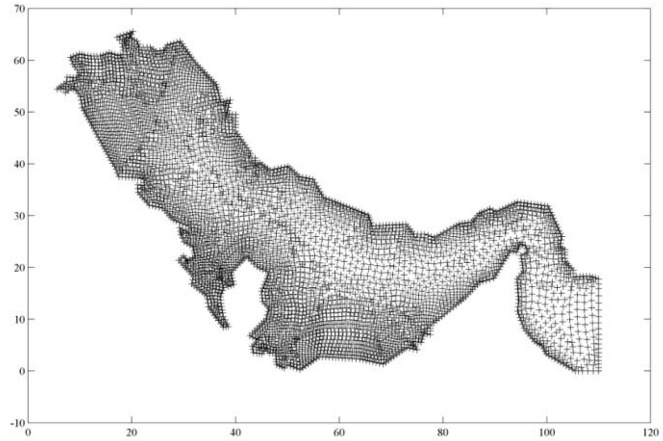


Fig.8. the nodes distribution obtained.

In continuation, by applying the Delaunay Algorithm, the nodes are used to generate a triangular mesh which is illustrated in Fig.9.

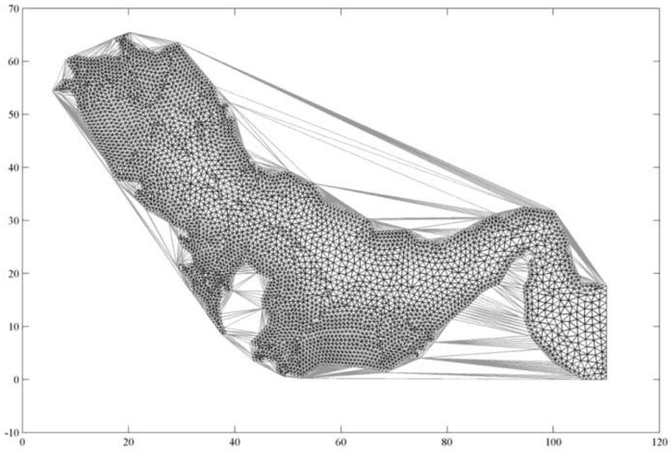


Fig.9: Applying the Delaunay Algorithm.

As observed in Fig.9, there are several undesired triangular elements out of the domain. These elements should be removed to have the final mesh. To accomplish this, the centroid of each element is calculated and the elements whose centers are out of the boundaries are removed from the list of the mesh elements. The resulting mesh, in its final desirable and acceptable form, is illustrated in Fig.10.

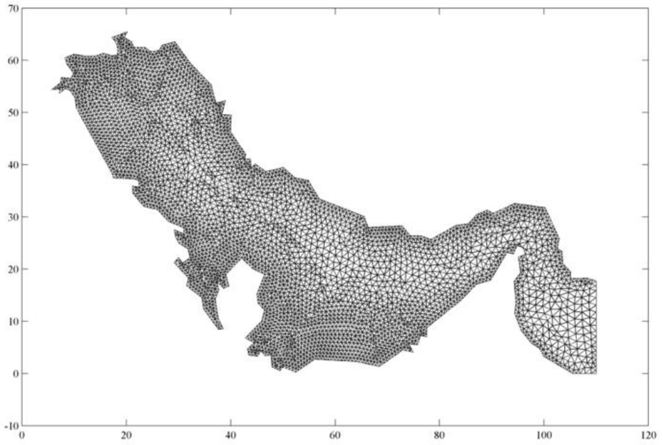
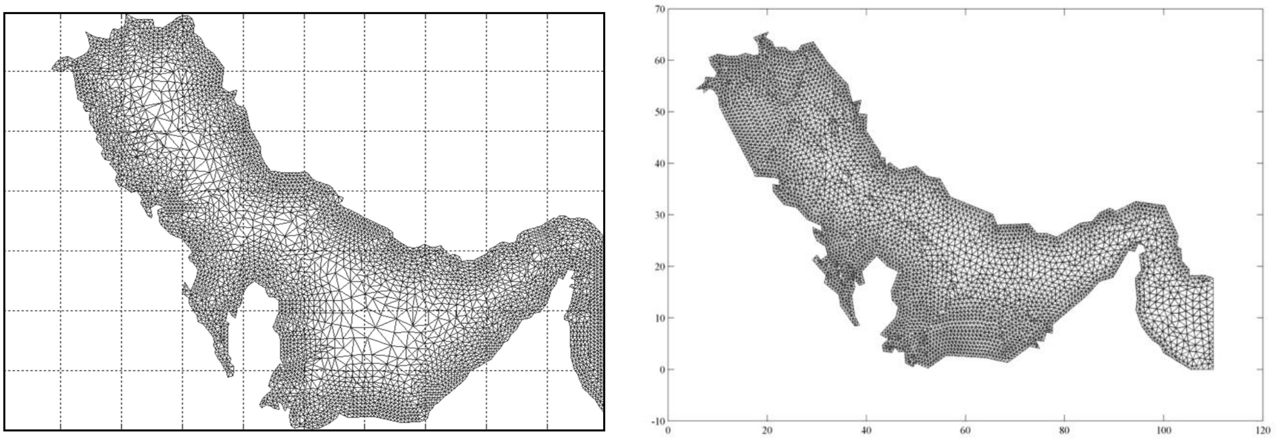


Fig.10: The final mesh of the Persian Gulf.

7 The Persian gulf

Meshing the Persian Gulf with this algorithm can be a perfect example to show the capability of the newly proposed meshing scheme, because it has been meshed previously with the original Dispersion Algorithm without the effect of bathymetry [1]. Therefore, the mesh quality and the computational cost of the modified algorithm are compared with those of the previous version, which are shown in table 1.

Fig.11 illustrates the mesh generated by the Dispersion Algorithm with and without the bathymetry. The influence coefficient of 2 has been assigned for the effect of bathymetry.



a b

Fig.11: The Persian Gulf: a) without bathymetry, b) with bathymetry and an influence coefficient of 2.

To compare the quality of the elements of the mesh, the aspect ratio of the element is defined as follows:

$$q = \frac{4\sqrt{3A}}{h_1^2 + h_2^2 + h_3^2} \qquad (4)$$

where, A is the area and h_i is the side length of the triangle. If q>0.6, the triangle is of acceptable quality. q=1 Corresponds to $h_1 = h_2 = h_3$.

The CPU time and elements quality of the two cases are compared in Table 1.

Table 1: Comparison of the Dispersion Algorithm with and without bathymetry for the Persian Gulf

	Without Bathymetry	With Bathymetry
CPU time	13.7	21.3
Range of element quality (q)	0.78 - 0.93	0.77-0.94

By comparing the two cases, it can be deduced that, the element quality did not predictably change significantly, but the CPU time has increased 7.5 seconds. This increase in time is because of the computational operations added to the code for taking into account the important physical property of bathymetry.

It must be mentioned that this increase in mesh generation time is compensated with a significant improvement in quality needed for the main analysis which may even lead to a lower computational time. Since the nodes distribution has been accomplised in a more appropriate way by the current method, higher density of nodes is achieved where more physical fluctuations is expected and less nodes are generated where the geometry does not exhibit complex features. Based on the above comparisons and the statistics, it can be concluded with certainty that the proposed algorithem is reliable and efficient enough to be further explored and improved in future.

To further ensure the applicability of this technique in ccean engineering areas, two other actual geometries are meshed and discussed below.

8 The Caspian sea

The topography of the Caspian Sea is shown in Fig.12. The map used in this publication originates from the Caspian Environment Program [25].

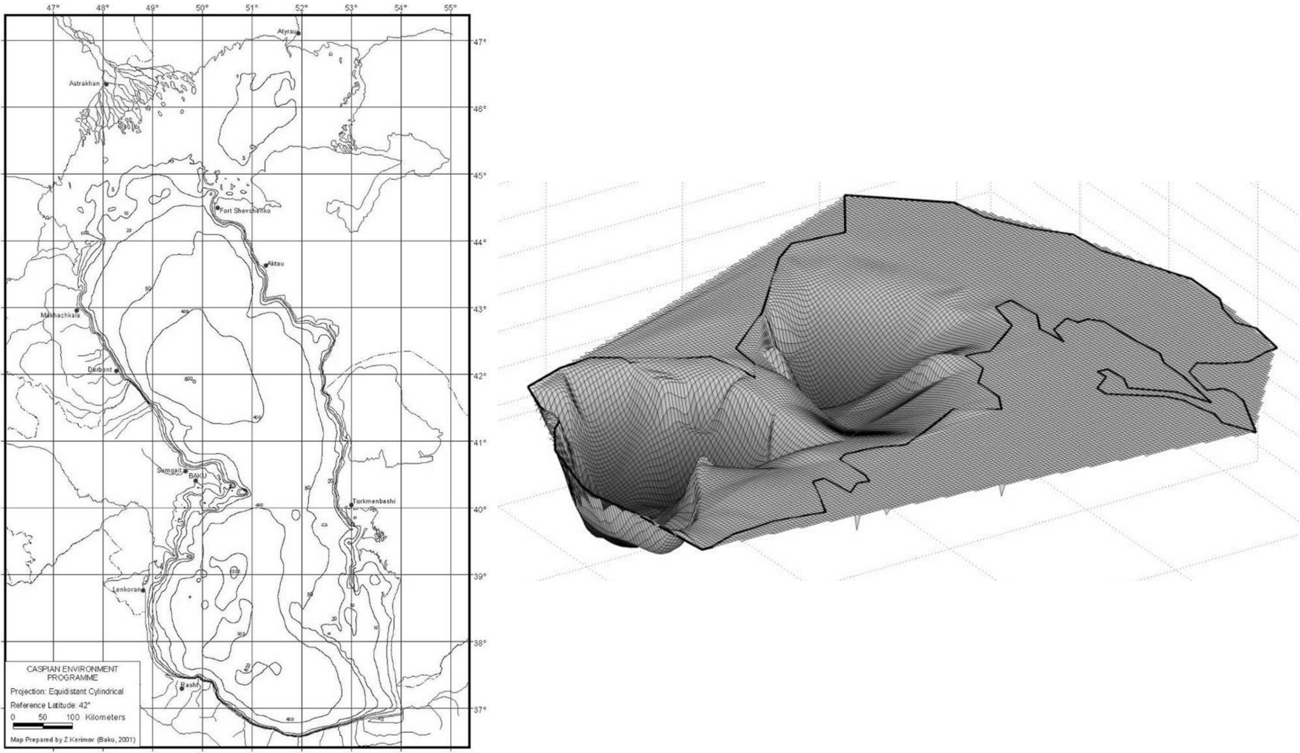


Fig.12: The bathymetry map of the Caspian Sea.

The mesh generated for this particular case has been made by 15 emulators and an influence coefficient of 2. Fig.13 shows the generated mesh which includes 15,563 nodes and has been generated in 27.7 seconds.

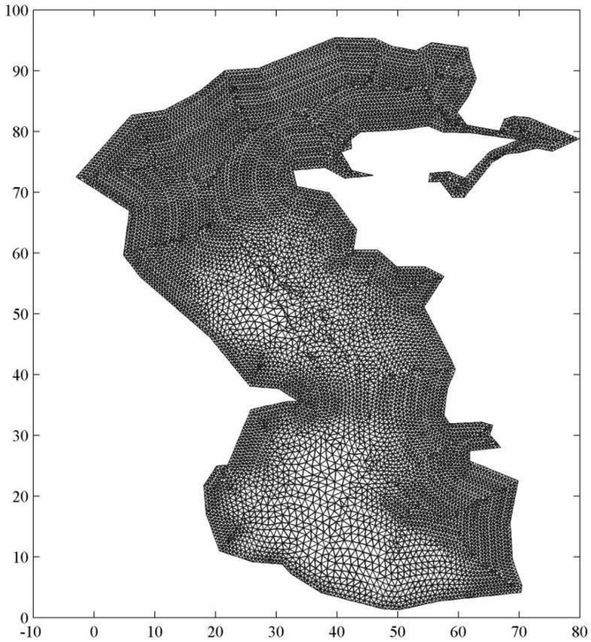


Fig.13: The generated mesh for the Caspian Sea.

9 The gulf of Mexico

The Gulf of Mexico has been previously meshed by Hagen et al. [3]. This was done, based on the wave length to grid size ratio, and also proportional to the truncation error for tidal models. In the current study, this geometry is meshed taking into consideration of this Gulf bathymetry. Fig.14 shows the bathymetry map of the Gulf of Mexico.

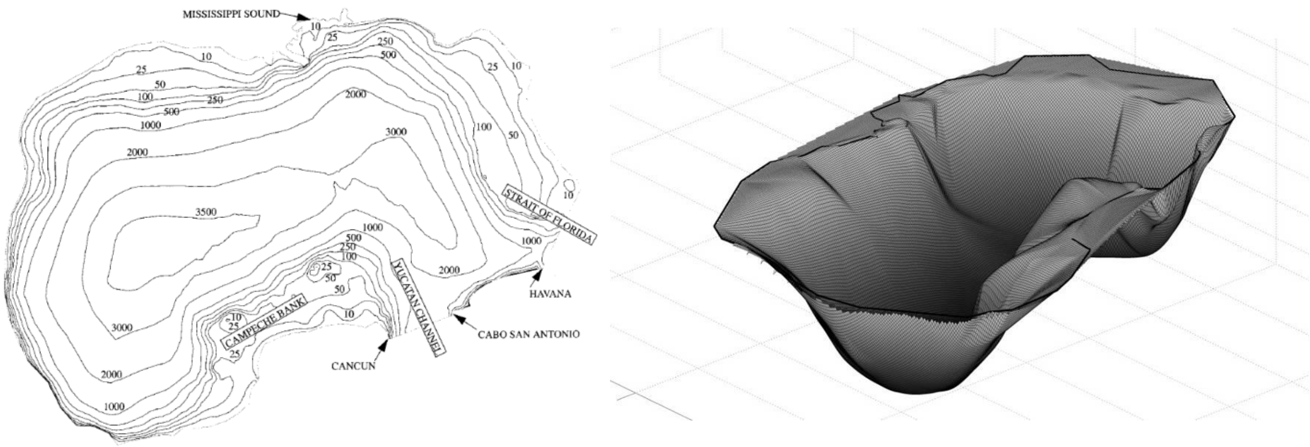


Fig.14: The bathymetry map of the Gulf of Mexico.

The generated mesh for this case is illustrated in Fig.15. This mesh contains 7,124 nodes and has been generated in 15.4 seconds.

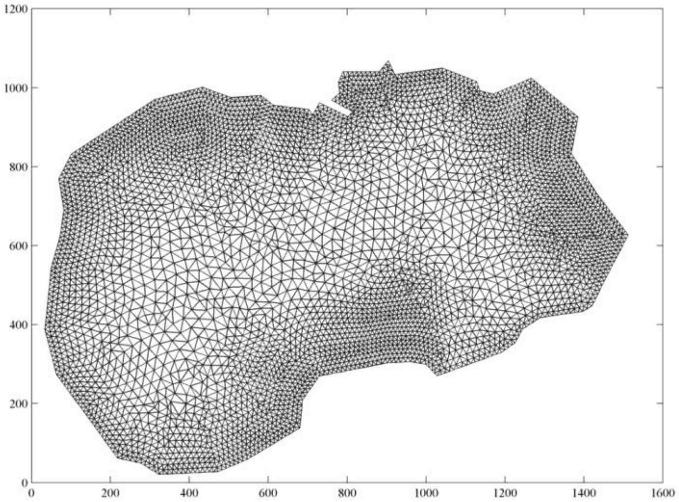


Fig.15: The Gulf of Mexico meshed with an influence coefficient of 2.

The quality of the elements does not fall bellow 0.76 which means all the elements in the mesh generated by this method are of good quality (q > 0.6).

10 Conclusion

In this paper, the concept of Dispersion Algorithm has been enhanced by applying a new coefficient E(x,y) described in Eq. 3 which locally controls the density of the emulators. This coefficient changes the form of the emulators, but the emulators are still replicas of the boundaries of the domain. This property helps the user to generate node distributions which take into account the boundary shapes, and also whatever the local coefficient E(x,y) represents. In this paper, E(x,y) represents the elevation of the domain which takes into account the importance of the bathymetry in Ocean Engineering problems. This improvement in Dispersion Algorithm re-emphasizes its high potentials for applications in different fields of engineering. The authors believe that the proposed coefficient can easily represent the wave characteristics, the density of the fluid, a turbulence predictor or whatever else important for the user to take locally into account in the mesh generation process.

References

[1] P. Ghadimi, M. A. Feizi Chekab, F. Safarzadeh Maleki. A novel approach to node distribution for 2D mesh generation and its application in marine and ocean engineering, Advances in Engineering Software 41 (2010) 1149–1159.

[2] Legrand. Delaunay mesh generation for an unstructured-grid ocean general circulation model. Ocean Modelling. v2. , 2000 17-28.

[3] S. C. Hagen, J. J. Westerink, R. L. Kolar, O. Horstmann. Two-dimensional, unstructured mesh generation for tidal models. Int. J. Numer. Meth. Fluids 2001; 35: 669–686.

[4] Bilgili A, Smith KW, Lynch DR. BatTri: a two-dimensional bathymetry-based unstructured triangular grid generator for finite element circulation modeling. Comput Geosci 2006, 32(5):632–42.

[5] Jones, J.E., Davies, A.M.. An intercomparison between finite difference and finite element (TELEMAC) approaches to modeling west coast of Britain tides. Ocean Dynamics, 2005, 55, 178–198.

[6] Greenberg, D.A., Dupont, F., Lyard, F.H., Lynch, D.R. and Werner, F.E., 2007. Resolution issues in numerical models of oceanic and coastal circulation. Contonental Shelf Research, 27(9), pp. 1317-1343.

[7] Robins, Peter E. Application of a coastal model to simulate present and future inundation and aid coastal management. Journal of Coastal Conservation, 2010.

[8] Robins, Peter E. Morphological controls in sandy estuaries: the influence of tidal flats and bathymetry on sediment transport. Ocean Dynamics, 2010.

[9] Jones, J. E. Storm surge computations in estuarine and near-coastal regions: the Mersey estuary and Irish Sea area. Ocean Dynamics, 2009

[10] Jones, J. E. An inter-comparison of tidal solutions computed with a range of unstructured grid models of the Irish and Celtic Sea Regions. Ocean Dynamics, 2009.

[11] Jones, J. E. Storm surge computations for the west coast of Britain using a finite element model (TELEMAC). Ocean Dynamics, 2008.

[12] Jones, J. E. Influence of non-linear effects upon surge elevations along the west coast of Britain. Ocean Dynamics, 2007.

[13] Jones, J. E., Davies, A. M. On the modification of tides in shallow water regions by wind effects, J. Geophys., 2008, Res., 113.

[14] Aldridge, J. N. Modelling the distribution and growth of ‘problem’ green seaweed in the Medway estuary, UK. Hydrobiologia, 2009.

[15] Logutov, Oleg G. A multigrid methodology for assimilation of measurements into regional tidal models. Ocean Dynamics, 2008.

[16] Wolf, J. Coastal flooding: impacts of coupled wave–surge–tide models. Natural Hazards., 2008.

[17] Horsburgh, K. J. Impact of a Lisbon-type tsunami on the U.K. coastline and the implications for tsunami propagation over broad continental shelves. Journal of Geophysical Research, 2008, 113(c4).

[18] Le Bars Y., Lyard F., Jeandel C., Dardengo L., The AMANDES tidal model for the Amazon estuary and shelf, Ocean Modelling, Volume 31, Issues 3-4, 2010, Pages 132-149.

[19] Parrish, D. M. and Hagen, S. C. (), Incorporating spatially variable bottom stress and Coriolis force into 2D, a posteriori, unstructured mesh generation for shallow water models. International Journal for Numerical Methods in Fluids, 2009, 60: 237–261. doi: 10.1002/fld.1882.

[20] Hanert E., Walters R. A., Le Roux D. Y., Pietrzak J. D., A tale of two elements: and RT0, Ocean Modelling, Volume 28, Issues 1-3, The Sixth International Workshop on Unstructured Mesh Numerical Modelling of Coastal, Shelf and Ocean Flows, 2009, Pages 24-33.

[21] White L., Deleersnijder E., Legat V., A three-dimensional unstructured mesh finite element shallow-water model, with application to the flows around an island and in a wind-driven, elongated basin, Ocean Modelling, Volume 22, Issues 1-2, 2008, Pages 26-47.

[22] Hall P., Davies A. M. Internal tide modelling and the influence of wind effects, Continental Shelf Research, Volume 27, Issue 9, Recent Developments in Physical Oceanographic Modelling: Part IV, 15 May 2007, Pages 1357-1377.

[23] Davies A. M. , Jones J. E., Xing J. Modeling the influence of small-scale effects upon the larger scale: an oceanographic challenge , Volume 60, Number 4, 921-932.

[24] Kennett, D. J. and Kennett, J. P. Early State Formation in Southern Mesopotamia: Sea Levels, Shorelines, and Climate Change, The Journal of Island and Coastal Archaeology, 1:1, 67 – 99.www.caspianenvironment.org