

Numerical Methodology of Mesh-based Simulation Technique

Adamu Musa Mohammed¹; Ali Musa Mohammed²; Mohammed Dahiru Buhari³; Abdulrahman A. Abdulrasheed⁴

^{1,4}Department of Chemical Engineering, Abubakar Tafawa Balewa University Bauchi, Bauchi State – Nigeria

²Department of Electrical Engineering, University of Birmingham, Birmingham United Kingdom

³Department of Electrical and Electronics Engineering, Abubakar Tafawa Balewa University Bauchi, Bauchi State – Nigeria

Abstract— A clear methodology for mesh-based simulation technique is presented in this study. The principles from creating the model to creating geometry, mesh generation, calculation, and result analysis (post-processing) were highlighted. Some of the results picked from previous works that employed the mesh-based technique were presented and that demonstrates how robust this technique is. It is obvious that today the use of computer simulations, in general, is paramount and that with computer simulations, different solutions can be evaluated numerically, and only promising designs are sent to the laboratory or production line for scale-up production.

Keywords:- Models; simulation; geometry; mesh; mesh generation; Elmer.

I. INTRODUCTION

Traditionally, computer simulations are carried out over system domains made of meshed (grid) surfaces. According to [1], “traditional modelling approaches make use of parametric patches, implicit surface, or subdivision surfaces that have been well integrated into 2D or 3D software”. The procedure for mesh-based simulation includes building a geometry, meshing and solution of the model equations using a given software. The system is first translated into a geometry which is latter discretized into mesh of a given size and finally the model, which is normally a differential equation, is solved. After building the geometry the operation is usually carried out in three stages (mesh generation, solution and postprocessing) and this, sometimes, require three independent software to do the job. For example, in computational study of mass transfer at surfaces with reactive nanocones [2] the geometry and meshing were done by Gmsh software, whereas solution and postprocessing were carried out by Elmer and Paraview, respectively. Each of the stages mentioned above is as important as the other and the error in one stage will significantly lead to an error in the other. However, some software (e.g. COMSOL) were developed in such a way that it can handle all the three operations.

When complex geometries were involved in a quantitative study of a physical phenomenon, numerical methods (such as finite element method) that discretize space are often the best method [3]. The procedure for discretization in Multiphysics is not much different with the numerical simulation that is based on partial differential equation discretization.

II. MODEL, GEOMETRY AND CALCULATION

A. Partial Differential Equations

In numerical studies, a problem is represented in form of differential equations (partial differential equation). The Partial differential equation (PDE) is an equation that relates a function of several variables to its partial derivatives. Many problems in engineering and sciences are presented in form PDE, e.g., wave equation, mass transfer equation, heat transfer equation, Schrödinger equation etc. Mostly, physical phenomena are modelled by equations that relate several partial derivatives of physical quantities such as momentum, velocity, forces, temperature, etc [4]. Equation 1 and 2 shows the example of first and second order partial differential equation in three dimensions (3D).

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} \quad (1)$$

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \quad (2)$$

The typical way to solve PDEs is to discretize them since they constitute a large sparse of matrix problem. However, there are many ways of discretizing PDEs and the simplest is finite difference approximations for partial operators. The finite element method replaces the original function (overall equation of the global domain) with a function (sub-equations) that has some degree of smoothness but is piecewise polynomial on simple cells such as rectangle, squares, or triangles. These triangles or rectangles are called the grid or mesh in 2D or 3D.

With numerical PDEs, the conditions at one or both extremes need to be specified in other to set limit or boundary. Therefore, initial value and boundary value problems are included as constraints equations in the solution of PDEs. Boundary conditions are practically essential for defining computational problems because the resultant quality of computations can critically be decided on how those boundary conditions are numerically treated [5].

➤ Boundary value and initial value problems

When a solution and derivatives value of a system of differential equations (ordinary or partial) is specified at more than one point, it is said to be a boundary value problem (BVP). The points at which these derivatives are specified are called the boundaries, which they occur at the extreme. Since differential equation represent problems in engineering, it is very important to know the boundary condition for such problems so as to proffer solutions. For instance, in mass transfer problems on the surface of a block [2], the initial concentration at surface ‘A’ and the final concentration at the

opposite surface 'B' must be specify. This knowledge will help in understanding what happens in-between at a particular time and the whole profile can be figured out.

In the order hand, the initial value problem (IVP) is a differential equation along with a specified condition at initial state or with an appropriate number of initial conditions. The IVP does not require to specify the value at boundaries rather it needs the values at initial condition. This usually apply for dynamic system that is changing over time. In summary, in IVP the conditions at one extreme of the interval are known whereas in BVP the conditions at both extremes of the interval are known and that is why all PDEs are BVP and IVP. For instance, we need the initial position and velocity of a particle in motion to be able to solve for its position under differential equation at time t .

➤ *Discretization*

There are two major approaches to discretisation namely, Lagrangian and Eulerian. Where the former has it's coordinate and properties defined for every local field, the latter has it's coordinate and properties defined for the overall system.

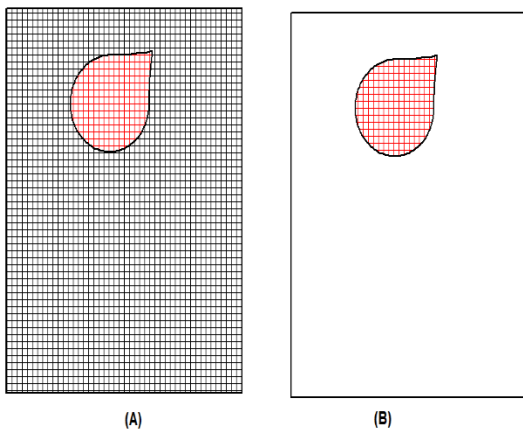


Fig. 1: Structured mesh (A) and unstructured mesh (B)

In Eulerian approach, discretisation is completed in a fixed space and no movement of nodes and elements is required (Fig. 1A). This approach is suitable for simple geometries and for fast calculation, it is however, not suitable for high deformable and complex geometries. In the other hand, the Lagrangian approach has it's nodes associated with material particles thereby representing free surfaces (Fig. 1B), this helps to evaluate the deformation and define the interfaces of different materials [6]. Therefore, Lagrangian approach can handle high deformation model due to non-restriction of a closed domain [7].

There is also an approach called Arbitrary Lagrangian-Eulerian (ALE) that combines the two approaches and harnessed the advantage of both Lagrangian mesh and the Eulerian mesh. In this approach, the nodes in the mesh can be flexible and move as the material deforms like the Lagrangian mesh or fixed in a space and eliminate the issue of mesh distortion as in Eulerian mesh or it can move in some arbitrary manner [8]. This flexibility offered by ALE has a great advantage of handling large distortion problems.

B. *Geometry and mesh generation*

A geometry of a computational domain represents a specific problem of interest by its physical features which include all details of its shape, size, and dimension. The problem may be presented in two or three dimensions depending on the precision requirement and the cost of simulation. After creating (building) the geometry, the next step is geometric discretization, in other word, meshing or grid formation. Meshing is usually performed so as to divide the models into cells and the mesh is generated inform of polygons or polyhedral that connect in a series of lines and points. There are varieties of software for geometry and mesh generation, and these includes open-sourced software such as Gmsh, NETGEN, GridPro, MeshGenC++, etc and commercial software like ANSYS, Gambit, Grid Pro, CFD, etc. The mesh generated could be structured or unstructured.

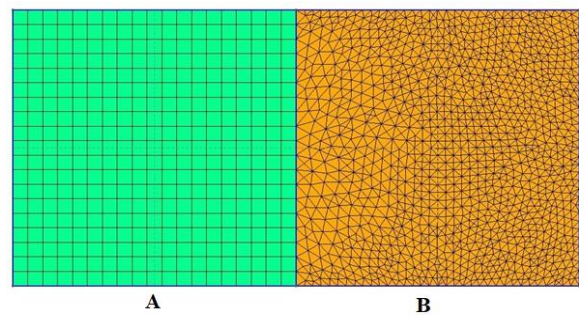


Fig. 2: Structured mesh (A) and unstructured mesh (B)

The structured meshes are meshes with implicit connectivity whose structure allows for easy identification of elements and nodes (Fig. 2A). In the other hand unstructured meshes (Fig. 2B) has general connectivity with an arbitrary structure and the connectivity of its elements must be defined and stored [9]. Structured mesh requires less memory compared to the unstructured mesh. The choice depends on many factors such as memory requirement, numerical accuracy, ease of generation, computation time, among others. Moreover, it is very difficult or impossible to compute structure mesh for a complex geometric domain or a geometry that requires high resolution mesh at a localized region.

➤ *Gmsh*

Gmsh is the software used for mesh generation in one of the case studies reported in this thesis. Gmsh is a three-dimensional finite element grid generator, it has a built-in CAD engine and post-processor. It is a user-friendly mesh tool that has an advanced visualisation capability (Gmsh manual). Basically, Gmsh is built around four modules: geometry, mesh, solver and post-processing. In building a geometry the constituents are points, lines, surfaces and bodies. These constituents form a complete solid model where lines are defined from points, surfaces from lines, and bodies from surfaces. The mesh generation is performed in similar way as the geometry: lines are discretized first; the mesh of the line is then used to mesh the surfaces; then the mesh of the surfaces is used to mesh the volumes. Fig. 3 is an example of solid geometry (cube) with unstructured mesh.

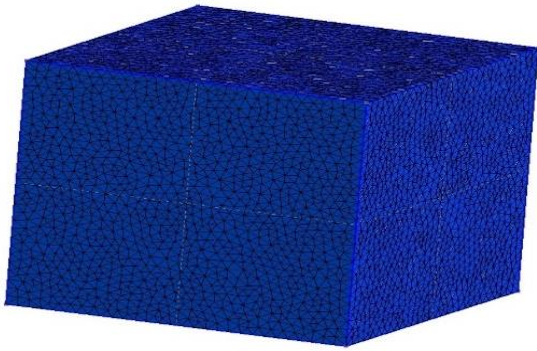


Fig. 3: A cubic geometry with unstructured mesh

Gmsh software is robust in building and meshing any kind of geometry which can be exported to finite element solver like Elmer for calculation and post-processing. Detail guide on how Gmsh is use in building geometry and meshing could be found in the user manual [10].

C. Calculation and post-processing

After building the geometry of a particular problem and generating the mesh, the project file is then opened/imported to a solver for calculations where the model has to be setup and appropriate equation defined. In this work Elmer Graphical User Interface solver (ElmerGUI) is used for the solutions of the mesh-based approach problems and a Visualization Toolkit ParaView is used for post-processing.

➤ **Elmer**

Elmer is an open-source finite element multiphysical simulation software. The software has a graphical user interface (ElmerGUI) which is capable of importing finite element mesh files in various formats [11]. It can also set up PDE systems for solution and export the model data and results for Elmer Solver. The stages in running the simulation with Elmer are:

- Model setup
- Add equation
- Add material
- Add body force
- Set initial conditions
- Set boundary conditions

To setup the model, the simulation type that is either steady-state or unsteady-state need to be specified. Maximum iteration, maximum output level, and time-stepping method also need to be specified. In addition to the simulation parameters, some constants, e.g gravity, vacuum permittivity, Stefan Boltzmann, and unit charge, are being specified. There are handful of equations to be used depending on the type of problem one has. In our case, ‘model PDE’ and ‘linear elasticity’ equations were used for mass transfer and linear elastic problem respectively.

The idea behind choosing a material in the material library is to use the specific properties of the material in question e.g density, heat capacity, specific heat capacity etc for accurate computation and results. Initial and boundary conditions are set, and the simulation is run after saving the model. The result file is then open with a post-processing software ParaView in our case.

➤ **ParaView**

This is a post-processing software to read the Elmer output file in the format ‘vtu’. “It is an open-source, multi-platform scientific data analysis and visualization tool that enables analysis and visualisation of extremely large datasets” [12]. ParaView has a collection of libraries for various applications including web visualization, in-situ analysis with catalyst and scripting using python [13]. To provide backbone for visualization and data processing ParaView uses the Visualization Toolkit (VTK). The VTK supports various visualization algorithms which includes vector, scalar, tensor, texture and volumetric methods. It also includes mesh smoothing, contouring, polygon reduction and Delaunay triangulation [14].

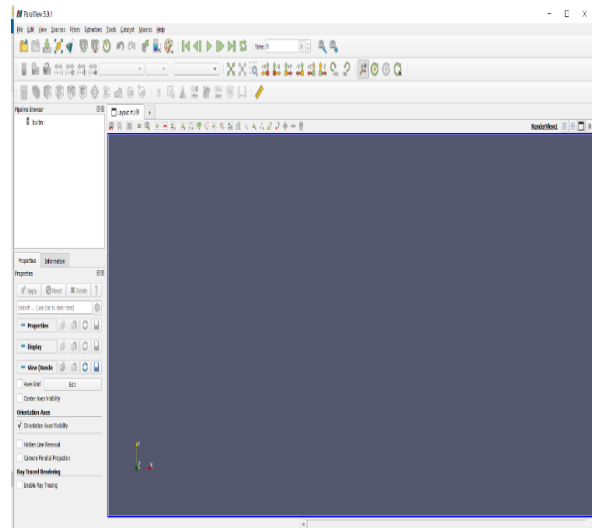


Fig. 4: Para View application window

In this work, ParaView 5.9.1 version was used, and the application window is as shown in Fig. 4. The mass transfer contours, von Mises stresses and displacement were analysis using ParaView. The files are in ‘vtu’ format for both structured and unstructured grids. After uploading, then ‘apply’ button is pressed. Analysis of various properties of the model is done using different filters. Details on how ParaView is used for visualization and post-processing can be found in [14], [12].

III. RESULTS

The numerical methodology for mesh-based simulation is used to solve a variety of problems in engineering and sciences yielding promising and near-perfect results [2], [15]–[18].

For instance, in [2], the mesh-based simulation was used to compute a mass transfer on surfaces structured with reactive nanocones where current densities were computed on the surfaces of the conical electrodes, greatly advancing the field of electrochemistry. In that work, Gmsh software was used to build and mesh the geometry of the conical electrode surface. However, the simulation is done using ELMER and then post-processed with ParaView. Some of the results were shown in Fig. 5 and Fig. 6.

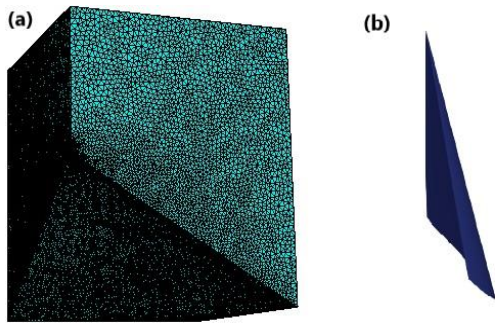


Fig. 5: (a); the computational domain and (b); conical boundary section [2].

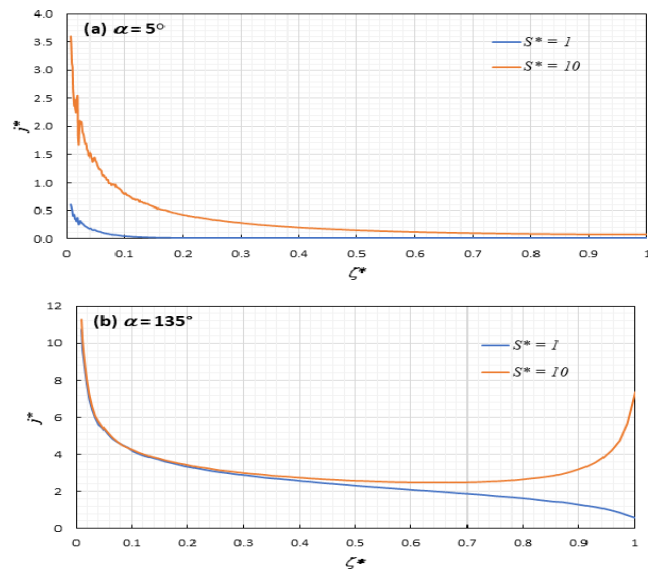


Fig. 6: Local current densities along conical face for (a) $\alpha = 5^\circ$ and (b) $\alpha = 135^\circ$, $S^* = 1$ and $S^* = 10$ [2].

In another research [15] the mesh-based approach was used to compute a linear elasticity problem on a wooden bar that is subjected to a load of 200N at one end and fixed at the other end. The bar geometry was built and meshed, and the stress and displacement were computed. Fig. 7 shows the stress distribution using colour code.

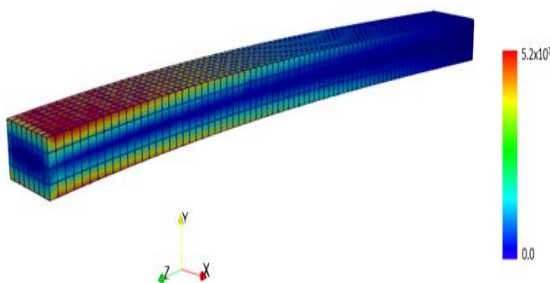


Fig. 7: The von Mises stress on a wooden bar Using mesh-base technique [15].

Additionally, a mesh-based approach can be used to compute flow issues with shifting boundaries and interfaces; however, this approach must account for the distortion caused by the mesh's movement. Tonon et al., 2021 conducted research on a mesh-moving technique without cycle-to-cycle accumulation of distortion and clearly demonstrated how to measure the mesh distortion as well as the quality of the mesh

as it affects the prospective results. Figure 8 shows the extent of the mesh displacement at maximum angle of attack [16].

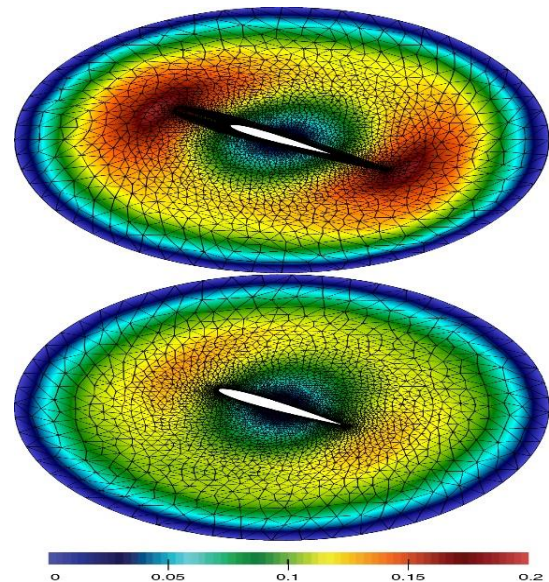


Fig. 8: Mesh displacement magnitude at root palne (top) and tip plane (bottom) [16].

IV. CONCLUSION

In this article, the procedures and ideas of mesh-based simulation are described. The basics of building geometry and meshing with Gmsh (a finite element mesh generator), computing with Elmer (a finite element method solver), and analysing the results obtained with Paraview are described in depth. Several of the outcomes of this methodology were highlighted, and base on that, it is clear that the meh-based technique is reliable and effective for using numerical computation to address a wide range of problems in engineering and the sciences.

REFERENCES

- [1.] Z. Ji, L. Liu, and Y. Wang, "B-Mesh: A Modeling System for Base Meshes of 3D Articulated Shapes," *Computer Graphics Forum*, vol. 29, no. 7, pp. 2169–2177, Sep. 2010, doi: 10.1111/j.1467-8659.2010.01805.x.
- [2.] A. M. Mohammed, M. Iqbal, N. V. Rees, and A. Alexiadis, "Computational study of mass transfer at surfaces structured with reactive nanocones," *Applied Mathematical Modelling*, vol. 74, pp. 373–386, Oct. 2019, doi: 10.1016/j.apm.2019.04.057.
- [3.] T. Fischbacher and H. Fangohr, "Continuum multi-physics modeling with scripting languages: the Nsim simulation compiler prototype for classical field theory," *arXiv:0907.1587 [cond-mat, physics:physics]*, Jul. 2009, Accessed: Apr. 30, 2019. [Online]. Available: <http://arxiv.org/abs/0907.1587>
- [4.] Y. Saad, *Iterative Methods for Sparse Linear Systems*, Second. Society for Industrial and Applied Mathematics, 2003. doi: 10.1137/1.9780898718003.
- [5.] S. Koshizuka, K. Shibata, M. Kondo, and T. Matsunaga, *Moving particle semi-implicit method: a meshfree particle method for fluid dynamics*. London,

- United Kingdom: Academic Press, an imprint of Elsevier, 2018.
- [6.] Z. Bi, *Finite element analysis applications: a systematic and practical approach*. London [England]: Academic Press, 2018.
- [7.] J. J. Monaghan, "A turbulence model for Smoothed Particle Hydrodynamics," *European Journal of Mechanics - B/Fluids*, vol. 30, no. 4, pp. 360–370, Jul. 2011, doi: 10.1016/j.euromechflu.2011.04.002.
- [8.] J. P. Davim, Ed., *Computational methods and production engineering: research and development*. in Woodhead publishing reviews: mechanical engineering series. Duxford, United Kingdom: Elsevier, 2017.
- [9.] M. Bern and P. Plassmann, "Mesh Generation," in *Handbook of Computational Geometry*, Elsevier, 2000, pp. 291–332. doi: 10.1016/B978-044482537-7/50007-3.
- [10.] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, Sep. 2009, doi: 10.1002/nme.2579.
- [11.] Lyly et al, "ELMER - A finite element solver for multiphysics.," 2000, 1999.
- [12.] U. Ayachit, *The ParaView guide: updated for ParaView version 4.3*, Full color version. Clifton Park, NY: Kitware Inc, 2015.
- [13.] P. O'Leary, J. Ahrens, S. Jourdain, S. Wittenburg, D. H. Rogers, and M. Petersen, "Cinema image-based in situ analysis and visualization of MPAS-ocean simulations," *Parallel Computing*, vol. 55, pp. 43–48, Jul. 2016, doi: 10.1016/j.parco.2015.10.005.
- [14.] C. D. Hansen and C. R. Johnson, Eds., *The visualization handbook*. Amsterdam; Boston: Elsevier-Butterworth Heinemann, 2005.
- [15.] A. M. Mohammed, U. D. Hamza, and Abdulrahman A. Abdulrasheed, "Computation of Three-Dimensional Linear Elasticity Problem using Mesh and Mesh-free Particle Modelling Techniques," Jun. 2023, doi: 10.5281/ZENODO.8042289.
- [16.] P. Tonon, R. A. K. Sanches, K. Takizawa, and T. E. Tezduyar, "A linear-elasticity-based mesh moving method with no cycle-to-cycle accumulated distortion," *Comput Mech*, vol. 67, no. 2, pp. 413–434, Feb. 2021, doi: 10.1007/s00466-020-01941-y.
- [17.] D. Violeau and R. Issa, "Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview," *Int. J. Numer. Meth. Fluids*, vol. 53, no. 2, pp. 277–304, Jan. 2007, doi: 10.1002/fld.1292.
- [18.] D. Pearce, S. Fischer, F. Huda, and A. Vahdati, "Applications of Computer Modeling and Simulation in Cartilage Tissue Engineering," *Tissue Eng Regen Med*, vol. 17, no. 1, pp. 1–13, Feb. 2020, doi: 10.1007/s13770-019-00216-9.