Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE
Informatyka w Technologii Materiałów

Vol. 11, 2011, No. 2

SIMULATION OF DROPLET MOTION IN WELDING ARCS AS A CASE STUDY OF REMESHING

PAWEŁ CYBUŁKA*, ALEKSANDER SIWEK, JACEK ROŃDA, KRZYSZTOF BANAŚ

AGH University of Science and Technology, al. Mickiewicza 30, 30-059 Kraków, Poland
**Corresponding author: cybulka@agh.edu.pl*

Abstract

The paper describes re-meshing techniques used for modelling computational domains with moving boundaries. The techniques are applied in a case study of accurate finite element modelling of droplet motion. The simulation takes place in a 3D domain. Locally the mesh is divided into two parts – a droplet and its surroundings. The part of a mesh around the droplet is composed of prismatic elements forming a thin layer to resolve better the boundary layer around the moving droplet. The rest of the domain is split into tetrahedrons using Delaunay triangulation.

Nodes on the surface of the droplet, during the FE simulation, move relatively to the surrounding material creating a moving boundary of the domain. The quality of the mesh deteriorates in the vicinity of the moving boundary. At each stage of numerical simulation the quality of the mesh is monitored and certain measures, like smoothing, are used to improve it. At certain moment it becomes impossible to retain the quality of the mesh without deleting some elements and creating new ones. The re-meshing is first done locally, and if it does not suffice, it is spread over a larger part of the domain.

The meshing algorithm, used in the whole process, is based on the Delaunay triangulation. Some modifications are done using the concept of the layer of prismatic elements around the droplet.

The objective of the newly developed software is to provide proper mesh management for process simulations with moving boundaries and, possibly, boundary layers. A welding process formulated with consideration of energy and mass transfer to the weld pool is analysed as an example. The process modelling requires the proper consideration of a filler material flow to the weld pool and involves transport of filler material droplets through the welding arc, i.e. plasma beam, to the weld pool.

Key words: mesh generation, mesh movement, re-meshing, welding, mass transfer

1. REMESHING ALGORITHM FOR MOVING BOUNDARIES

Usually, moving boundaries result in deformations of elements close to the surface of the computational domain. Deformed elements can be improved by standard smoothing techniques and translation of mesh nodes away from the boundary (Löhner, 2008). The technique used in our case is the Laplacian smoothing with weights that aims at obtaining elements with similar volume.

Mesh modifications in the presented algorithm are performed using two meshes: the original mesh and its copy. The original mesh is called the reference mesh. The copy of the mesh, for which modifications are performed, is called the current mesh. The two meshes play different roles at different stages of the algorithm. There are several reasons for introducing the copy of the mesh. One is that displacement of mesh nodes may cause changes to the solution, hence, at certain moments proper projection of the solution from the reference mesh to the current mesh should be performed. Another reason is that during mesh modifications it is often neces-

ISSN 1641-8581

sary to go back from the current mesh to the previous reference mesh.

Mesh modifications form a part of the simulation algorithm performed in a series of time steps. The whole algorithm consists of the following procedures:

1. Create an initial mesh – a reference mesh for the first time step
2. For each time step from the first to the end of simulations:
   1) Create the current mesh by copying mesh points of the reference mesh.
   2) Refine the current mesh, to achieve assumed mesh quality measures.
   3) Move the boundary of the computational domain according to the simulation algorithm requirements – for the current mesh.
   4) Optimize the current mesh according to the chosen criteria.
   5) Simulate one time step using the current mesh (if necessary performing again steps 3 and 4).
   6) Go back to the reference mesh – performing nodal motion, remeshing (adding and removing nodes and elements) and the projection of the solution from the current mesh.

In the following sections the main procedures of the algorithm are described in detail for the case of a droplet moving within the computational domain, a case study selected for the paper.

1.1. The creation of the initial mesh

As a first step a rectangular triangulation domain is created. Within the domain the points on the surface of the body are generated. These points are stationary during the smoothing process. In the following step the whole domain is filled with random points, approximately equidistributed in space, having the specified distance from its nearest neighbours and the boundary points.

The process of adding new points consists in generating a random location and checking whether there are any other points within a distance r. The distance r is determined by the distance of the location from the nearest boundary point R, using some specified function, $Fun(R)$. If a point is found within the radius r, the location is rejected, otherwise a new point is created in the location. The whole process is illustrated in figure 1. The 2D solution to this problem is described in (Madej at al., 2009).

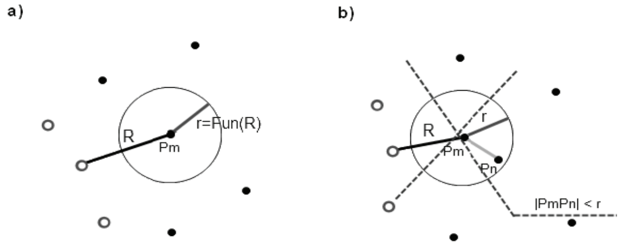


Fig. 1. *Adding a new point to the triangulation. A newly generated point Pm is accepted in case a) and rejected in case b), because of the non-existence or existence of another internal point within the radius r = Fun(R) from Pm. Circles denote boundary points and dots internal points.*

The result of the two previous steps, for the case of a droplet inside a rectangular computational domain is shown in figure 2. To the left a part of the computational domain is shown with points used for mesh generation, to the right a cross-section of the part of the domain with faces of tetrahedral elements and the droplet boundary clearly visible.

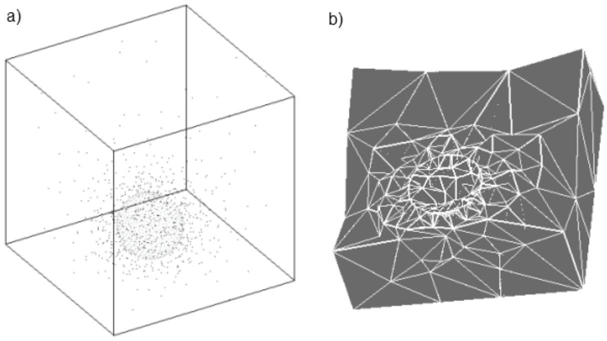


Fig. 2. *a) Discretization domain filled with internal points and points situated on the boundary of a droplet. b) Cross-section of the domain after triangulation.*

The next step is to identify the interior of a droplet and the droplet boundary. A droplet boundary is composed of faces containing only points on the boundary of the droplet. In further computations it is assumed that the droplet boundary is a set of three point triads. A droplet interior is composed of elements containing only interior and boundary faces. In our case study, the result of droplet identification is shown in figure 3a.

The last step in creation of the initial mesh is a preliminary correction of elements' quality by appropriate smoothing and face swapping. The smoothing algorithm aims either at equalizing volume of elements or generating the optimal shape according to the standard Laplace smoothing. Using proper weights, the algorithm will use both equalizing volume of elements and saving desired shape of elements. The problem of smoothing and swapping is analyzed in the works by Löhner (2008) and Jurczyk (2007).

In the case of using weights, the algorithm of the Laplacian smoothing performs the following steps for each point within the domain. First of all, weights are computed for each of its neighbouring points. The weight is a function of distances between the point for which the weight is computed and other neighbouring points of the considered point. Weights are normalized using the mean of weights for all neighbouring points. Finally a new position of the considered point is computed using a special formula with an additional free parameter. The formula used in the process for 2D case (for example) is presented next to figure 4.

The final result of smoothing for our test case of the mesh around the droplet is presented in figure 3b. The improvement as compared to the mesh in figure 3a is visible.

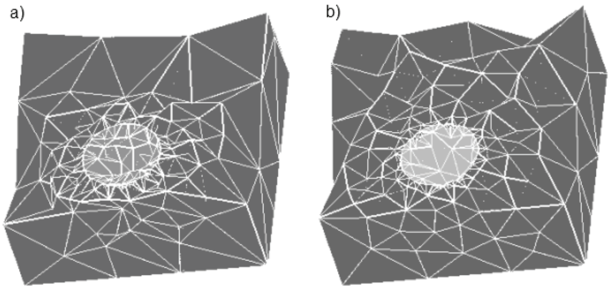
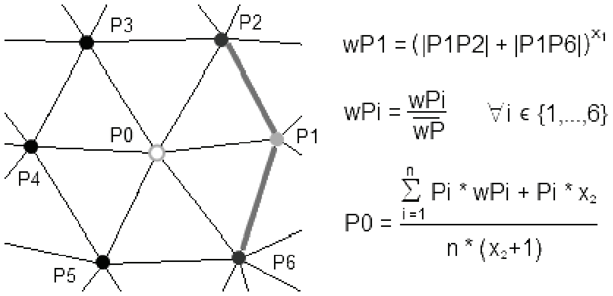


Fig. 3. a) Identification of a droplet and a surrounding domain, b) The mesh around a droplet after smoothing mesh with proper weights.



$$wP1 = (|P1P2| + |P1P6|)^{x_1}$$

$$wPi = \frac{wPi}{\overline{wP}} \quad \forall i \in \{1,...,6\}$$

$$P0 = \frac{\sum_{i=1}^{n} Pi * wPi + Pi * x_2}{n * (x_2 + 1)}$$

***Fig. 4.** The mesh and the scheme for evaluation of the new position for the point P0 in the weighted Laplacian smoothing algorithm. Weights wPi are computed at points Pi, i=1,...,n=6. Free parameters X_1 and X_2 are chosen to control the influence of subsequent weights.*

1.2. Mesh modifications

As it was already mentioned, it is assumed that in the actual computations two meshes are necessary at each time step: the reference mesh with the solution from the previous time step, and the current mesh obtained by suitable modifications of the reference mesh, for which the solution at the current time step is computed.

Both meshes undergo modifications during calculations:

- motion of nodes, in the case of the current mesh due to some prescribed or computed boundary displacement and in the case of the reference mesh in order to accommodate, at the end of a time step, the changes in the current mesh,
- optimization of a mesh according to some chosen criteria, considering a mesh motion as well as adding and deleting new nodes and elements. In the case of the current mesh, it leads to produce the best mesh for computations and in the case of the reference mesh, it helps to generate the optimal mesh at the end of each time step. Optimality is understood as a proper balance between the computational cost of handling the mesh and the best quality measures together with the degree of refinement

The three steps of mesh modifications are illustrated in figure 5 for our test case of droplet motion. The cross-section of the domain shows faces of elements with droplet differentiated using a different colour. The first modifications (figure 5a) take into account all points situated inside a droplet. Then, the mesh deforms according to the droplet motion (figure 5b) and the algorithm of weighted Laplacian smoothing is applied. First, weights are chosen in such a way that they equalize the volume of elements. This step pushes the points in elements close to a droplet boundary towards or away from a boundary, as it is necessary (figure 5c). Next, several smoothing steps are performed with weights selected in order to improve elements' quality measures. The final result is shown in figure 5d.

1.3. Insertion of new points

Sometimes, a motion of nodes does not suffice and it becomes necessary to insert new points within the sub-domain. Methods for adding new points are described in (Jurczyk, 2007). This process is performed after calculations of the simulation algorithm and is followed by suitable projection of the solution from the old to the new enriched mesh.

The algorithm for a point insertion works best locally. Otherwise, it can take much too long for large meshes. A local version of the point insertion has the following steps:

- Cut out a local sub-domain associated with the deformed part of the mesh,

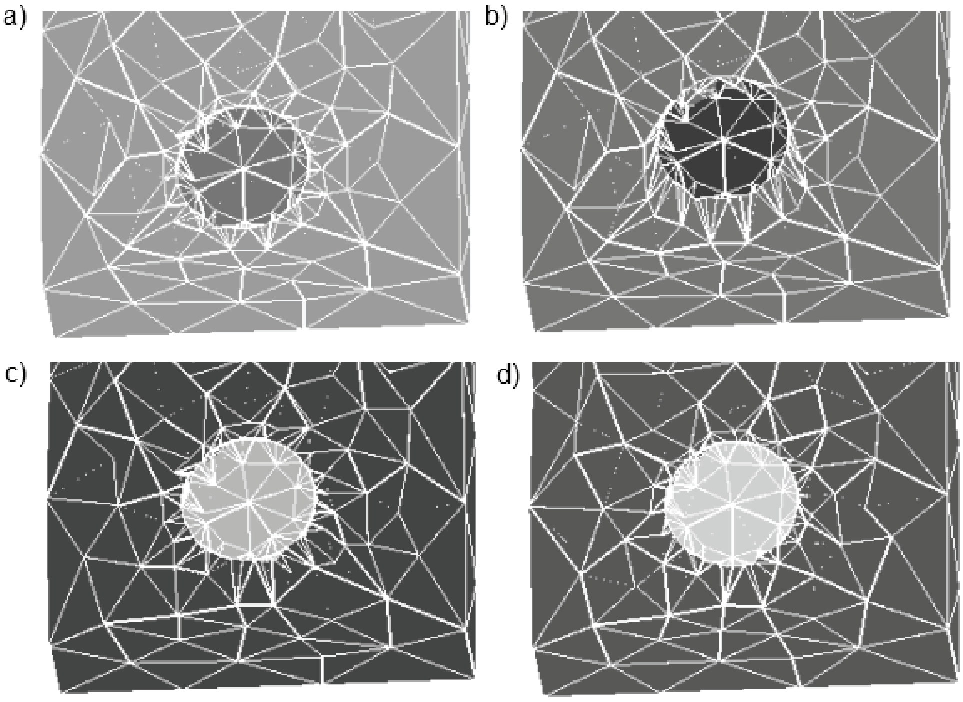


Fig. 5. The example 3D mesh: a) before the droplet motion, b) after the droplet motion, c) after smoothing aimed at equalizing volume of elements, d) after smoothing aimed at improving elements' quality measures.

- Create a set of invariant boundary points consisting of points adjacent to the points inside the local sub-domain,
- Perform the new triangulation of a sub-domain,
- Unite the new local triangulation with the old large mesh.

1.4. Boundary layer creation and management

In order to better resolve a flow around a droplet, a boundary layer, which consists of thin prismatic elements, is created using points on the droplet surface. In the case of simple shapes like our example droplet it suffices to perform the following steps:

- a boundary surface is identified, a surface composed of triangular faces between the droplet and its surroundings
- for each vertex on the boundary surface, an approximate smooth surface is obtained by interpolation of positions of neighbouring vertices
- for each vertex a vector normal to the approximate surface is created using the gradient of the functional expression for the approximate surface
- new points are added in the direction of the normal vector, the number of points determined by the selected number of elements across the boundary layer thickness,
- new elements are created on the basis on the new points.

During computations, the introduction of the boundary layer does not change the mesh motion algorithm. The only difference is that instead of the droplet alone, the droplet and the boundary layer are treated as moving within the tetrahedral mesh.

2. NUMERICAL RESULTS

Figure 6 shows numerical results obtained for FEM evaluations of a droplet vertical motion. A mesh is reconstructed locally when the FE aspect ratio reaches its critical value. A size of a local FEM cluster, i.e. presented fraction of the domain, is chosen to facilitate matching of the local and global meshes after the droplet large relocation. The local FE cluster can be expanded after adding new points during re-meshing in the local cluster. Such re-meshing usually precludes matching of meshes. The droplet vertical motion together with small horizontal oscillations can be seen in figure 6c. In the mesh smoothing algorithm, the FM aspect ration is used in the criterion for mesh quality. The FE aspect ratio is defined as the ratio of a radius of the ball circumscribed around the element to a radius of the ball inscribed in the element. It is assumed that a value of the ratio cannot exceed ten. FE mesh was modified in all ten droplet positions shown in the figure 6c. On each stage, the Laplacian smoothing is performed unless finite elements keep appropriate quality, i.e. FE aspect ration is much less than ten. When the ratio is about ten or more, the algorithm with three steps of mesh modifications, illustrated in figure 5 and described above, is applied. Following this procedure, the mesh quality is improved enough and the Laplacian smoothing technique can be used again.

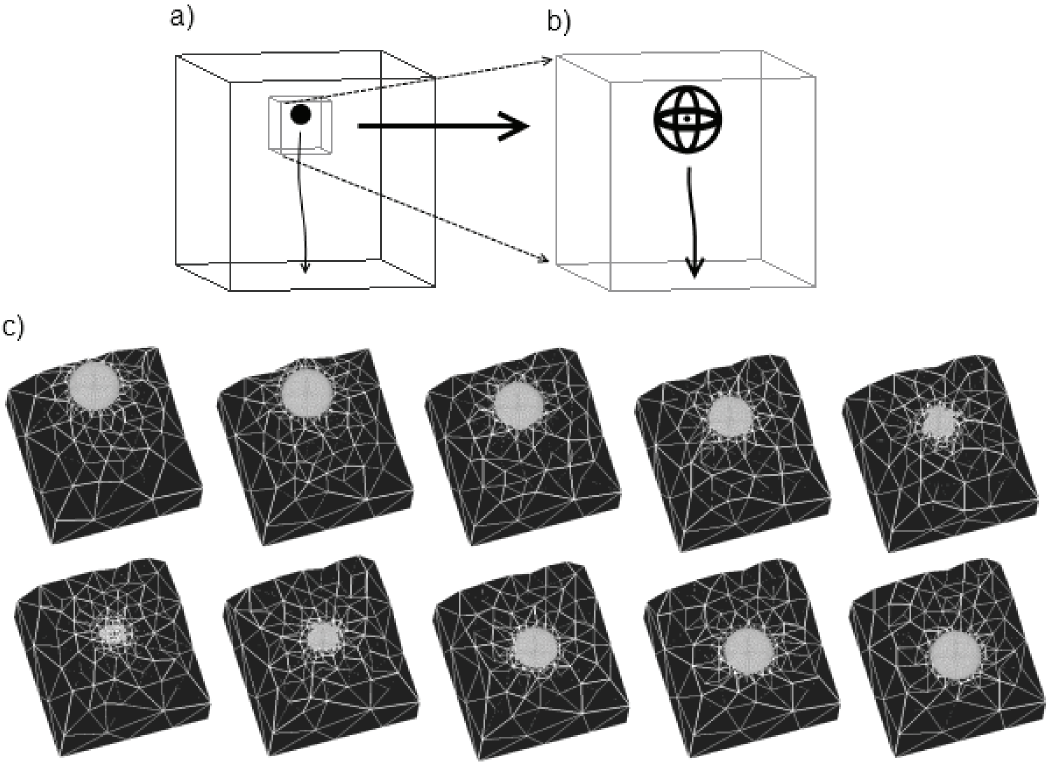


Fig. 6. FEM simulation of a droplet vertical motion with small oscillations in the horizontal direction. a) The computational domain. b) A magnification of the domain fraction containing a droplet. c) 3D mesh cross-section of a droplet neighborhood for subsequent time steps of FEM evaluation. A droplet highest position is shown in the figure upper-left and its lower position can be seen in the figure bottom-right.

3. CONCLUSION

The algorithms proposed by the Authors can reduce the number of FE mesh modifications. The moving boundary of the domain, i.e. the welding plasma beam, requires only mesh enhancements that can be done by using techniques of smoothing, swapping, deleting and adding points. Another advantage of this strategy is an efficient implementation into the FEM code. The originality of this method consists on the alternative use of the Laplacian smoothing and re-meshing, when it is necessary. Therefore, larger steps of time can be applied in evaluations of droplet relocation. The numerical cost of our method with the weighted smoothing and local re-meshing technique is lower than for traditional Laplacian smoothing/re-meshing techniques.

The numerical technique described here is our first attempt to model welding plasma beam with a droplet transport and considering both energy and mass transfer. Unfortunately, the contact of a droplet with a surface of a weld bath and the instability of energy transfer from a welding plasma beam to a weld pool are not accounted in our FEM model.

ACKNOWLEDGMENTS

The support for this paper by NCBiR project ZPB/33/63903/IT2/10 and by MNiSzW project N N508 583839 is acknowledged.

LITERATURE

Jurczyk, T., 2007, *Efficent Algorithms of Automatic Discretization of Non-Trivial Three-Dimensional Geometries and its Object-Oriented Implementation,* PhD Thesis, AGH University of Science and Technology, Kraków.

Madej, L., Cybułka, G., Perzynski, K., Rauch, L., Pietrzyk, M., 2009, Multi Scale Modeling Based on Digital Material Representation, Proc. X Conf., COMPLAS, Barcelona, CD.

Löhner, R., 2008, *Applied Computational Fluid Dynamics Techniques – An Introduction Based on Finite Element Methods, Second Edition,* John Wiley & Sons, Ltd, Chichester.

SYMULACJA RUCHU KROPLI W ŁUKU SPAWALNICZYM JAKO STUDIUM PRZYPADKU REMESHINGU

Streszczenie

Artykuł przedstawia technikę re-meshingu zastosowaną w modelowaniu trój-wymiarowej przestrzeni z ruchomym brzegiem. Metoda ta znajduje zastosowanie między innymi w dokładnym modelowaniu ruchu spadającej kropli metodą elementów skończonych. W tym celu siatka zostaje lokalnie podzielona na dwie części – kropli i jej otoczenie. Część siatki wokół kropli składa się z elementów pryzmatycznych tworzących cienką warstwę przyścienną. Warstwa pryzmatyczna została utworzona w celu uzyskania dokładnego rozwiązania na granicy przestrzeni kropli a otoczenia. Pozostała część przestrzeni jest podzielona na czworościany za pomocą techniki opartej na triangulacji Delaunay w przestrzeni 3D.

Podczas symulacji MES węzły na powierzchni przesuwanej kropli tworzą ruchomy brzeg względem obszaru otaczającego kroplę, przez co jakość siatki wokół kropli pogarsza się. Na każdym etapie numerycznych symulacji jakość siatki jest monitorowana i pewne środki, takie jak wygładzanie, wykorzystywane są do jej poprawy. W pewnym momencie niemożliwe staje się zachowanie odpowiedniej jakości siatki bez usuwania i tworzenia nowych elementów. Początkowo odpowiedni remeshing siatki wykonywany jest lokalnie, a jeśli to nie wystarczy, jest rozłożony na większą część obszaru.

Celem nowo tworzonego oprogramowania jest zapewnienie właściwego zarządzania siatką podczas symulacji procesów z ruchomym brzegiem oraz ewentualne uwzględnienie warstwy przyściennej. Oprogramowanie zostanie zastosowane między innymi w symulacjach procesu spawania uwzględniających transfer energii i masy do jeziorka. Modelowanie procesu wymaga należytego rozpatrzenia przepływu materiału wypełniającego do jeziorka i obejmuje transport kropli spoiwa poprzez łuk spawalniczy, czyli wiązkę plazmy, do jeziorka.

Received: December 5, 2010
Received in a revised form: December 24, 2010
Accepted: December 24, 2010