Mecánica Computacional Vol. XXIII, pp. 1897-1909
G.Buscaglia, E.Dari, O.Zamonsky (Eds.)
Bariloche, Argentina, November 2004

A MASS-CONSERVING SMOOTH METHOD

L.G. Nonato[*], **Norberto Mangiavacchi**[†], **Fabrício S. Sousa**[*], **A. Castelo, J.A. Cuminato**[*]

[*]Instituto de Ciências Matemáticas e de Computação
Departamento de Computação e Estatística
Universidade de São Paulo, USP
Av. Trabalhador São-Carlense, 400
Cxp. 668, Cep:1560-970, So Carlos, SP, Brasil
e-mail: {gnonato,fsimeone,castelo}@icmc.usp.br, web page: http://www.lcad.icmc.usp.br

[†]Faculdade de Engenharia
Departamento de Engenharia Mecânica
Universidade Estadual do Rio de Janeiro, UERJ
e-mail: norberto@uerj.br

Key Words: Smoothing, Mass-conserving, Free-Surface.

Abstract. *Curve/surface smoothing is a problem that appears in different fields, such as computer graphics and computational fluid mechanics. In fluid flow simulation with free surface, particularly when the Reynolds number is high, small undulations may appear at the free surface due to variations in the velocity field from cell to cell. These undulations are frequently much smaller than a cell size and a numerical implementation that acts at cell level cannot take into account these sub-cell undulations, being necessary suppress them. There are several approaches that can be used to smooth these unphysical undulations, such as Gaussian filter. However, in fluid flow simulations it is important that the smoothing process keeps the mass (or volume) unchanged. In this work we present a smoothing technique that suppresses undulations while still conserving the mass. The approach consists in computing a local volume which is preserved during the smoothing process. The results of applying such technique in planar, axisymmetric, and three-dimensional free-surface flows are presented and discussed.*

1 INTRODUCTION

The problem of smoothing a surface (curve) has achieved large popularity over the last two decades. The interest in this subject is motivated by the large number of applications that are dependent on smoothing mechanism, such as visualization and computer graphics.

The literature has presented a range of different strategies devoted to smooth surfaces and curves. A very popular approach is the Laplacian filter,[1] which replaces the position of a vertex by averaging the positions of the neighboring vertices. Although largely employed, Laplacian smoothing presents the undesired effect of shrinking, causing loss of mass. Even more sophisticated versions of Laplacian smoothing[2,3] can not ensure that the volume/mass is preserved.

In some applications, such as incompressible fluid flow simulation with free surface, mass-preserving smoothing becomes a central problem, as undulations may appear at the free surface due to variations in the velocity field from cell to cell, as illustrated in Figure 1. These undulations are frequently much smaller than a cell, and usually they are not present in laboratory experiments because they are physically removed by a combination of surface tension and viscous effects. In computational simulations these undulations are clearly not physical and are only numerical artifacts, resulting from the fact that the numerical methods can not resolve sub-grid phenomena.

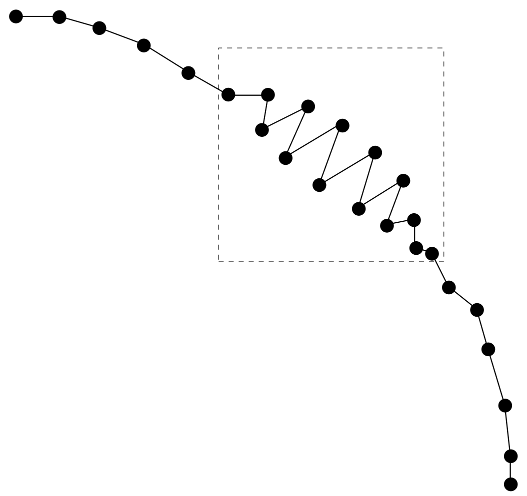

Figure 1: Small high frequency undulations in the free surface.

In this work we present the *Trapezoidal Sub-grid Undulations Removal (TSUR)*, a mass-preserving smoothing technique that aims at changing the position of vertices keeping constant a local volume. Although the idea of controlling a local volume has already been employed in other mass-preserving smoothing techniques,[4,5] TSUR brings out two desirable characteristic, robustness and linearity in terms of computational time.

TSUR has been implemented in three different versions: two-dimensional, axisymmetric, and three-dimensional. The two-dimensional and axisymmetric versions have been incorporated into GENSMAC2D[6] code whereas the three-dimensional version is in GENSMAC3D.[7]

GENSMAC code[8] has been designed for simulating free surface flows and was originally motivated by the need to simulate container filling in the food industry.

Section 2 presents TSUR in the two-dimensional and axisymmetric case. Section 3 describes the three-dimensional version. Some results are presented and discussed in section 4. Conclusions and future works are in section 5.

2 REMOVING UNDULATIONS: TSUR-2D AND TSUR-AXI

In this section we present the details of the TSUR in two-dimensional as well as in axisymmetric case. Although the framework developed here can be applied to any polygonal curve, we will adopt a terminology from fluid mechanics, which is our application target. Some details regarding the implementation into GENSMAC2D will also be discussed.

2.1 TSUR-2D

Consider four consecutive particles at a free surface, with given coordinates $\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}$ and $\mathbf{x}_{i+3}$, as shown in figure 2a). Particles at $\mathbf{x}_{i+1}$ and at $\mathbf{x}_{i+2}$ will be repositioned to $\mathbf{x}^*{}_{i+1}$ and to $\mathbf{x}^*{}_{i+2}$ in such a way that $L_1 = L_2 = L_3 = L$, $h_1 = h_2 = h$ (see figure 2b), and the final area A of the trapezium formed by the points $\mathbf{x}_i, \mathbf{x}^*{}_{i+1}, \mathbf{x}^*{}_{i+2}$ and $\mathbf{x}_{i+3}$ be equal to the area of the quadrilateral before modification. The area of the quadrilateral is computed by dividing the quadrilateral into two triangles, computing the area of each triangle and then adding the two contributions to get the total area. The signed area of the triangle defined by the points $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$, for instance, taken in a counter-clockwise direction, is computed from (see Preparata and Shamos (1985)[9] and de Berg et al. (1997)[10]):

$$A_{ijk} = \frac{1}{2}\left((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i)\right)$$

where $\mathbf{x}_i = (x_i, y_i)$. We have that $A_{ijk} = A = 2Lh$, and therefore $h = \frac{A}{2L}$. Note that h may be positive or negative according to the sign of A_{ijk}. Given

$$\boldsymbol{\tau} = (\tau_x, \tau_y)^t = \frac{\mathbf{x}_{i+3} - \mathbf{x}_i}{||\mathbf{x}_{i+3} - \mathbf{x}_i||_2} \ ,$$

the unit vector tangent to $\mathbf{x}_{i+3} - \mathbf{x}_i$, and $\mathbf{n} = (\tau_y, -\tau_x)^t$, the outward unit normal, the new positions are given by $\mathbf{x}^*{}_{i+1} = \mathbf{x}_i + L\boldsymbol{\tau} + h\mathbf{n}$ and $\mathbf{x}^*{}_{i+2} = \mathbf{x}_i + 2L\boldsymbol{\tau} + h\mathbf{n}$.

In GENSMAC2D, the method above is applied to displace all adjacent pairs of points on the free surface. However, particles are allowed to move only when their destination cell are the same as their original cell. The method is applied in successive sweeps across the whole fluid interface. The number of sweeps for optimal performance depends on the problem. Typically we apply one sweep every 5 to 50 time steps.

2.2 TSUR-AXI

The TSUR-2D method as described above is an *area conserving* filtering method. In order to be able to use it in axisymmetric problems we modified it in such a way that the volume of rotation with respect to the z axis is preserved. For instance, when the particles at $\mathbf{x}_{i+1}$ and $\mathbf{x}_{i+2}$ in figure 2a) are repositioned to $\mathbf{x}^*{}_{i+1}$ and $\mathbf{x}^*{}_{i+2}$ to form an isosceles trapezium as in figure 2b), instead of preserving the area, we would like to preserve the volume of rotation V_q of the quadrilateral $\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}$ and $\mathbf{x}_{i+3}$, which is given by

$$V_q = 2\pi \iint r\, dr\, dz$$

where the integrals are taken over the shaded area of figure 2a).

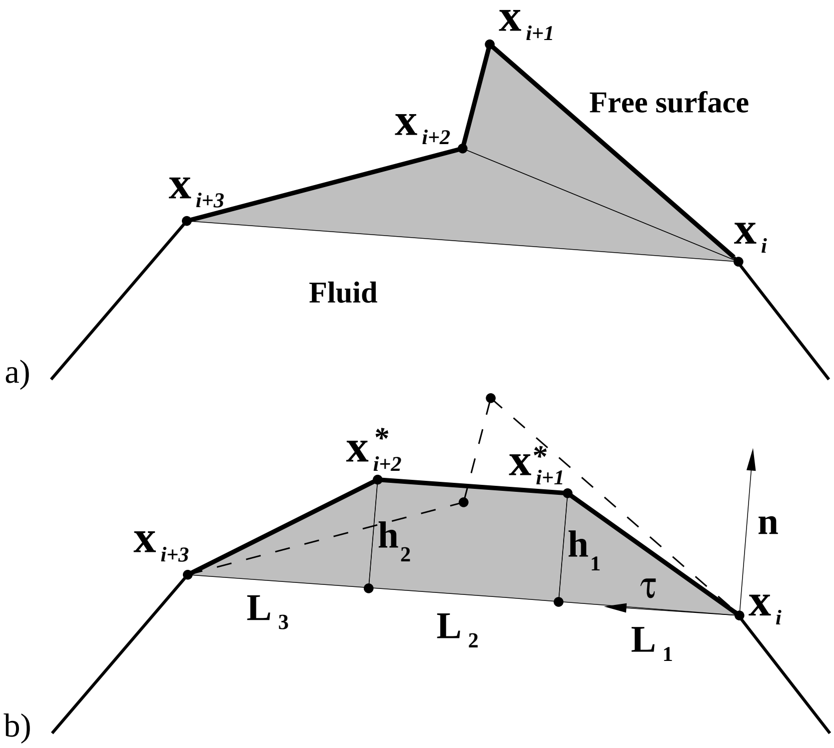

Figure 2: Two-dimensional Trapezoidal Sub-grid Undulations Removal (TSUR-2D) method.

To compute V_q, the quadrilateral is divided into two triangles, $\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}$ and $\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+3}$. For a triangle in the $r - z$ plane with coordinates $\mathbf{x}_i = (r_i, z_i)$, $\mathbf{x}_j = (r_j, z_j)$, $\mathbf{x}_k = (r_k, z_k)$, the volume of rotation is given by

$$V_{ijk} = \frac{\pi}{3} \left(r_i + r_j + r_k\right) \left(\left(r_j - r_i\right)\left(z_k - z_i\right) - \left(r_k - r_i\right)\left(z_j - z_i\right)\right) \ .$$

Hence, the volume of rotation for the quadrilateral $\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}$, $\mathbf{x}_{i+3}$ is given by

$$\begin{aligned} V_q = \quad & \frac{\pi}{3} \left(r_i + r_{i+1} + r_{i+2}\right) \left(\left(r_{i+1} - r_i\right)\left(z_{i+2} - z_i\right) - \left(r_{i+2} - r_i\right)\left(z_{i+1} - z_i\right)\right) \ + \\ & \frac{\pi}{3} \left(r_i + r_{i+2} + r_{i+3}\right) \left(\left(r_{i+2} - r_i\right)\left(z_{i+3} - z_i\right) - \left(r_{i+3} - r_i\right)\left(z_{i+2} - z_i\right)\right) \ . \end{aligned} \tag{1}$$

We proceed to construct an isosceles trapezium $\mathbf{x}_i, \mathbf{x}^*{}_{i+1}, \mathbf{x}^*{}_{i+2}$ and $\mathbf{x}_{i+3}$ with the same volume of rotation as the original quadrilateral, $\mathbf{x}_i, \mathbf{x}_{i+1}, \mathbf{x}_{i+2}$ and $\mathbf{x}_{i+3}$ as shown in figure 2b.

Let us define the length $L_1 = L_2 = L_3 = L = \frac{1}{3}||\mathbf{x}_{i+3} - \mathbf{x}_i||_2$,

$$\boldsymbol{\tau} = (\tau_r, \tau_z)^t = \frac{\mathbf{x}_{i+3} - \mathbf{x}_i}{||\mathbf{x}_{i+3} - \mathbf{x}_i||_2} \ ,$$

the unit vector tangent to $\mathbf{x}_{i+3} - \mathbf{x}_i$, and $\mathbf{n} = (\tau_z, -\tau_r)^t$, the outward unit normal.

The volume of rotation of the isosceles trapezium can be shown to be given by

$$V_t = 2\pi L h \left(2r_i + 3L\tau_r + \frac{5}{6} h \tau_z \right) \ .$$

The final volume of rotation V_t has to be equal to the initial one V_q. On one hand, if $\tau_z \neq 0$, h has to satisfy

$$h^2 + \frac{6}{5}\left(2\frac{r_i}{\tau_z} + 3L\frac{\tau_r}{\tau_z} \right) h - \frac{3}{5}\frac{V_q}{\pi L \tau_z} = 0 \ , \quad (\tau_z \neq 0) \ .$$

This expression, when solved for h, produces two solutions, and we select the solution with $r^*_{i+1} \geq 0$ and $r^*_{i+2} \geq 0$, that is, non-negative radial coordinates for the final positions of the two internal points of the stencil.

On the other hand, if $\tau_z = 0$, h has to satisfy

$$h = \frac{V_q}{2\pi} \frac{1}{(2\, r_i\, L \ + 3L^2 \tau_r)} \ .$$

The new positions are given by $\mathbf{x}^*{}_{i+1} = \mathbf{x}_i + L\boldsymbol{\tau} + h\mathbf{n}$ and $\mathbf{x}^*{}_{i+2} = \mathbf{x}_i + 2L\boldsymbol{\tau} + h\mathbf{n}$. As in the case of planar flows, the method is applied in successive sweeps across the whole interface, typically every 5 to 50 time steps.

3 TSUR-3D

Based on the ideas presented above, we develop the TSUR-3D smoothing filter. In GENS-MAC3D, the TSUR-3D is applied to the unstructured grid that defines the free surface, which is represented by a "*half-edge*" data structure.[11] Figure 3 shows a typical vertex $\mathbf{v}$, and its corresponding star, formed by the set of vertices $\mathbf{x}_i$, which are connected by the edges $(\mathbf{v}, \mathbf{x}_i)$ and $(\mathbf{x}_i, \mathbf{x}_{i+1})$, $i = 0, 1, \ldots, n$. The faces $(\mathbf{x}_i, \mathbf{v}, \mathbf{x}_{i+1})$, $i = 0, 1, \ldots, n$ ($\mathbf{x}_{n+1} = \mathbf{x}_0$) complete the definition of the star of $\mathbf{v}$.

As in TSUR-2D and TSUR-AXI, the TSUR-3D aims at preserving a local volume, which is defined as the volume bounded by the faces $(\mathbf{x}_i, \mathbf{v}, \mathbf{x}_{i+1})$ and $(\mathbf{x}_i, \mathbf{p}, \mathbf{x}_{i+1})$, $i = 0, 1, \ldots, n$, where $\mathbf{p}$ is an arbitrary point, as shown in figure 4.

In short words, the method employs a balance procedure that involves a plane π defined by the normal computed from the average of the normals of the vertices $\mathbf{x}_i$, $i = 0, 1, \ldots, n$. The vertex $\mathbf{v}$ and the vertices $\mathbf{x}_i$, $i = 0, 1, \ldots, n$, are projected onto π and $\mathbf{v}$ is slid over π to the

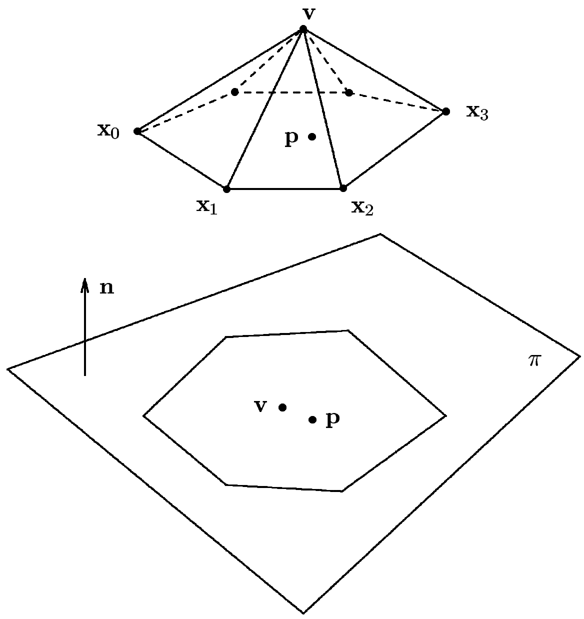

Figure 3: A typical vertex and its star projected into the plane π.

centroid of these projections. Finally, $\mathbf{v}$ is displaced in the direction of the normal of π in a such way that the local volume is preserved (see figure 3).

This procedure smoothes small undulations and keeps the volume of the fluid unchanged. In the following we present the details.

3.1 Normal vector calculation

As mentioned above, the smoothing procedure requires a normal vector, which is obtained using information from the neighboring vertices $\mathbf{x}_i$, $i = 0, 1, \ldots, n$. Let $\mathbf{v}$ be a vertex on the surface, and let $S(\mathbf{v})$ be the star of $\mathbf{v}$. Let $\mathbf{p}$ be the average of the vertices in $S(\mathbf{v})$, given by

$$\mathbf{p} = \frac{1}{n} \sum_{i=0}^{n} \mathbf{x}_i \tag{2}$$

where $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n \in S(\mathbf{v})$. The normal vector can be computed by averaging the normal vectors of the faces $(\mathbf{x}_i, \mathbf{p}, \mathbf{x_{i+1}})$ ($i = 0, 1, \ldots, n$). More precisely,

$$\mathbf{n} = \frac{\sum_{i=0}^{n-1} (\mathbf{x}_i - \mathbf{p}) \times (\mathbf{x}_{i+1} - \mathbf{p}) + (\mathbf{x}_n - \mathbf{p}) \times (\mathbf{x}_0 - \mathbf{p})}{|| \sum_{i=0}^{n-1} (\mathbf{x}_i - \mathbf{p}) \times (\mathbf{x}_{i+1} - \mathbf{p}) + (\mathbf{x}_n - \mathbf{p}) \times (\mathbf{x}_0 - \mathbf{p}) ||}. \tag{3}$$

where $\times$ denotes the vector product. Note this is a weighted average: normals of faces with greater area have a greater weight.

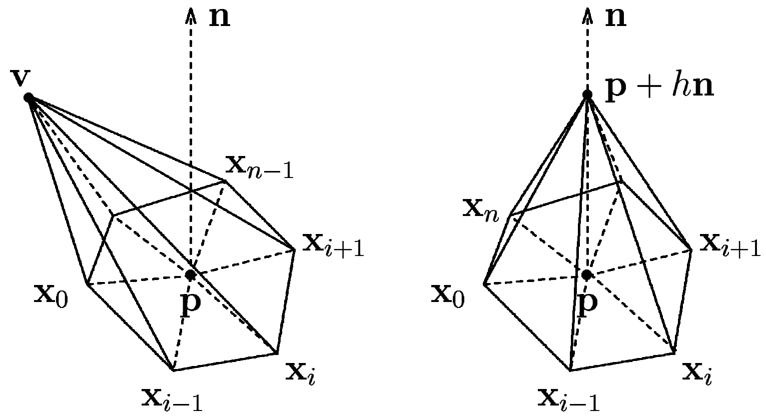

Figure 4: A vertex with its star, before and after the vertex balance procedure has been applied.

3.2 Vertex balance procedure

The vertex balance procedure is applied at each vertex of the fluid in order to move it to a new position on a line passing through $\mathbf{p}$, where $\mathbf{p}$ is the average (2), with the direction $\mathbf{n}$ given by (3), in such a way that the local volume is preserved. Let π be the plane with normal vector $\mathbf{n}$ containing $\mathbf{p}$. Notice that $\mathbf{p}$ together with $S(\mathbf{v})$ gives rise to a polyhedron that defines the local volume which will be preserved, as illustrated in figure 4. If $\mathbf{p}$ is inside the projection of $S(\mathbf{v})$ onto π, we compute the new position of $\mathbf{v}$ by

$$\mathbf{v}_{\text{new}} = \mathbf{p} + h\mathbf{n} \tag{4}$$

where h is computed to ensure that the local volume is preserved. This is guaranteed if

$$V = hV_1 \quad \text{or} \quad h = \frac{V}{V_1} \tag{5}$$

where V is the volume of the polyhedron $(\mathbf{v}, \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{p})$ and V_1 is the volume of the unitary polyhedron $(\mathbf{p} + \mathbf{n}, \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{p})$. This is because the volume of the polyhedron $(\mathbf{p} + h\mathbf{n}, \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{p})$ is equal to h times the volume of the unitary polyhedron $(\mathbf{p} + \mathbf{n}, \mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{p})$.

From the volume of a tetrahedron $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$ given by

$$V = \frac{1}{6} \cdot \det \begin{vmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1 \end{vmatrix}. \tag{6}$$

it is possible to compute the volume of the polyhedron that defines the local volume.

As we can see in figure 4 this procedure does not remove any undulations, but just moves the vertex to a new position, that is better centered in its star, improving the quality of the surface mesh, while preserving local volume. This improves the robustness of the undulation removal procedure presented below.

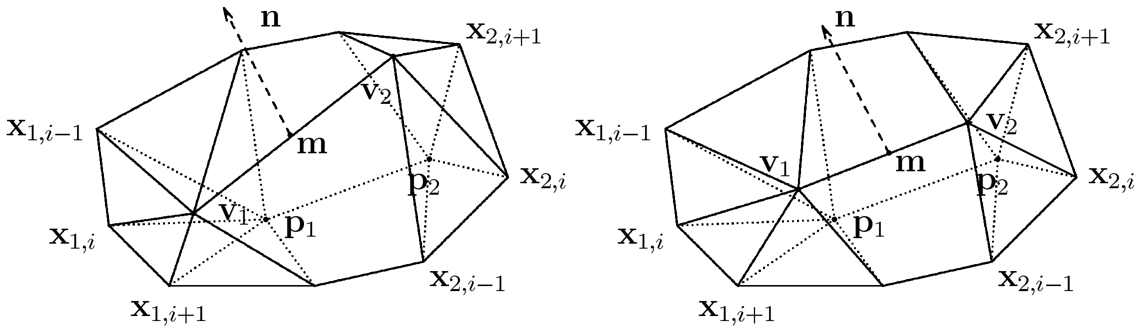

Figure 5: Undulation removal procedure, showing an edge and its star.

3.3 Undulation removal procedure

The vertex balance procedure is designed to produce a more homogeneous mesh that represents either a free surface or an interface. However, it cannot remove the sub-grid undulations, requiring a further step to smooth the undulations.

Let $\mathbf{e}$ be an edge of the surface, and $\mathbf{v}_1$, $\mathbf{v}_2$ its vertices, as shown in figure 5. The smoothing procedure changes the position of $\mathbf{v}_1$ and $\mathbf{v}_2$ simultaneously, trying again to preserve a local volume, which can be defined as follows.

Let $\mathbf{n}_1$ and $\mathbf{n}_2$ be the normal vectors computed by (3) at the vertices $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively. Averaging the normal vector $\mathbf{n}_1$ and $\mathbf{n}_2$ we obtain a normal $\mathbf{n}$ for the edge $\mathbf{e}$, i.e.,

$$\mathbf{n} = \frac{\mathbf{n}_1 + \mathbf{n}_2}{||\mathbf{n}_1 + \mathbf{n}_2||}. \tag{7}$$

Let

$$\mathbf{m} = \frac{1}{2}(\mathbf{v}_1 + \mathbf{v}_2) \tag{8}$$

be the average of the vertices in $\mathbf{e}$. We determine the heights h_1 and h_2 in the direction of $\mathbf{n}$ as

$$h_1 = \langle \mathbf{v}_1 - \mathbf{m}, \mathbf{n} \rangle \quad \text{and} \quad h_2 = \langle \mathbf{v}_2 - \mathbf{m}, \mathbf{n} \rangle \tag{9}$$

where $\langle \; , \; \rangle$ denotes the inner product. Moreover, we compute the points $\mathbf{p}_1$ and $\mathbf{p}_2$ by

$$\mathbf{p}_1 = \mathbf{v}_1 - h_1\mathbf{n} \quad \text{and} \quad \mathbf{p}_2 = \mathbf{v}_2 - h_2\mathbf{n}. \tag{10}$$

Thus, we have two polyhedra, $(\mathbf{v}_1, \mathbf{x}_{1,0}, \dots, \mathbf{x}_{1,n}, \mathbf{p}_1)$ with volume V_1 and $(\mathbf{v}_2, \mathbf{x}_{2,0}, \dots, \mathbf{x}_{2,m}, \mathbf{p}_2)$ with volume V_2, where $\mathbf{x}_{1,0}, \dots, \mathbf{x}_{1,n} \in S(\mathbf{v}_1)$ and $\mathbf{x}_{2,0}, \dots, \mathbf{x}_{2,m} \in S(\mathbf{v}_2)$ (see figure 5). We have to determine a new height h in the direction of $\mathbf{n}$, such that the local volume is preserved.

From the linearity between the volumes of the polyhedra and the heights we have

$$V_1 + V_2 = V_1(h/h_1) + V_2(h/h_2) \tag{11}$$

or

$$h = \frac{V_1 + V_2}{V_1/h_1 + V_2/h_2}. \tag{12}$$

The new positions of the vertices $\mathbf{v}_1$ and $\mathbf{v}_2$ are thus given by

$$\mathbf{v}_1 = \mathbf{p}_1 + h\mathbf{n} \quad \text{and} \quad \mathbf{v}_2 = \mathbf{p}_2 + h\mathbf{n}. \tag{13}$$

This undulation removal procedure is applied periodically to all the edges of the surface mesh. The frequency of applications is dependent on the problem, the grid resolution, and the time step.

4 RESULTS

As mentioned above, in simulation with of higher Reynolds number and in regions with strong surface area reduction, sub-grid undulation may occur in the free surface, causing a degradation of the overall numerical precision. In this section we present the results of applying TSUR in order to sort out such a problem. We illustrate the performance of TSUR in planar, axisymmetric and three-dimensional free surface flow simulation.

4.1 Planar Flow

Our first example concerns a free surface flow simulation of the filling of a container. The domain is discretized using a uniform 50×60 cell mesh.

Figure 6 shows a comparison of two simulations, one without TSUR-2D (dashed line), and the other with TSUR-2D (dotted line). It can be observed (figure 6 in the right) that the free surface is undulated without TSUR-2D, thus errors are introduced by these undulations in the computation of the surface tension. TSUR has removed the undulations altogether, making the results more reliable.

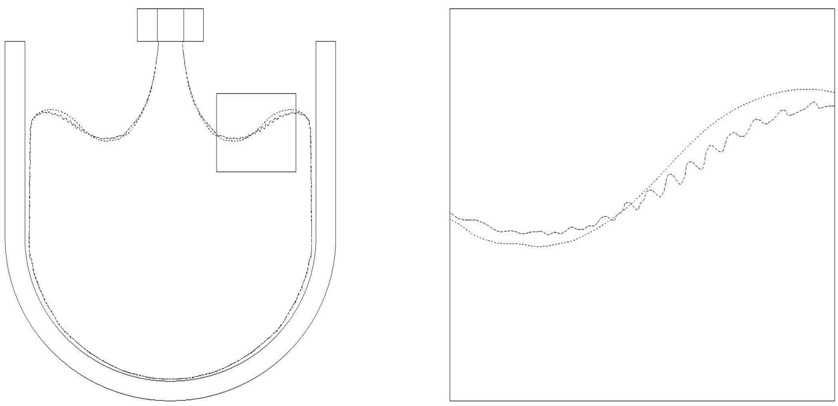

Figure 6: Comparison of a simulation without TSUR (dashed line) against a simulation with TSUR (dotted line)

4.2 Axisymmetric flow

In order to illustrate TSUR-AXI we simulate a sessile drop with surface tension. Figures 7a) shows the free-surface of the drop without TSUR-AXI. In figure 7b) we can observe a smoother curve. Notice that the undulations in the top and in the right bottom of figure 7a) were removed by TSUR-AXI.

Figures 7c) and 7d) present a 3D view of the sessile drop. It is clear from figure 7d) that TSUR-AXI has smoothed the surface properly. In this simulation TSUR-AXI has been applied each 100 cycles.

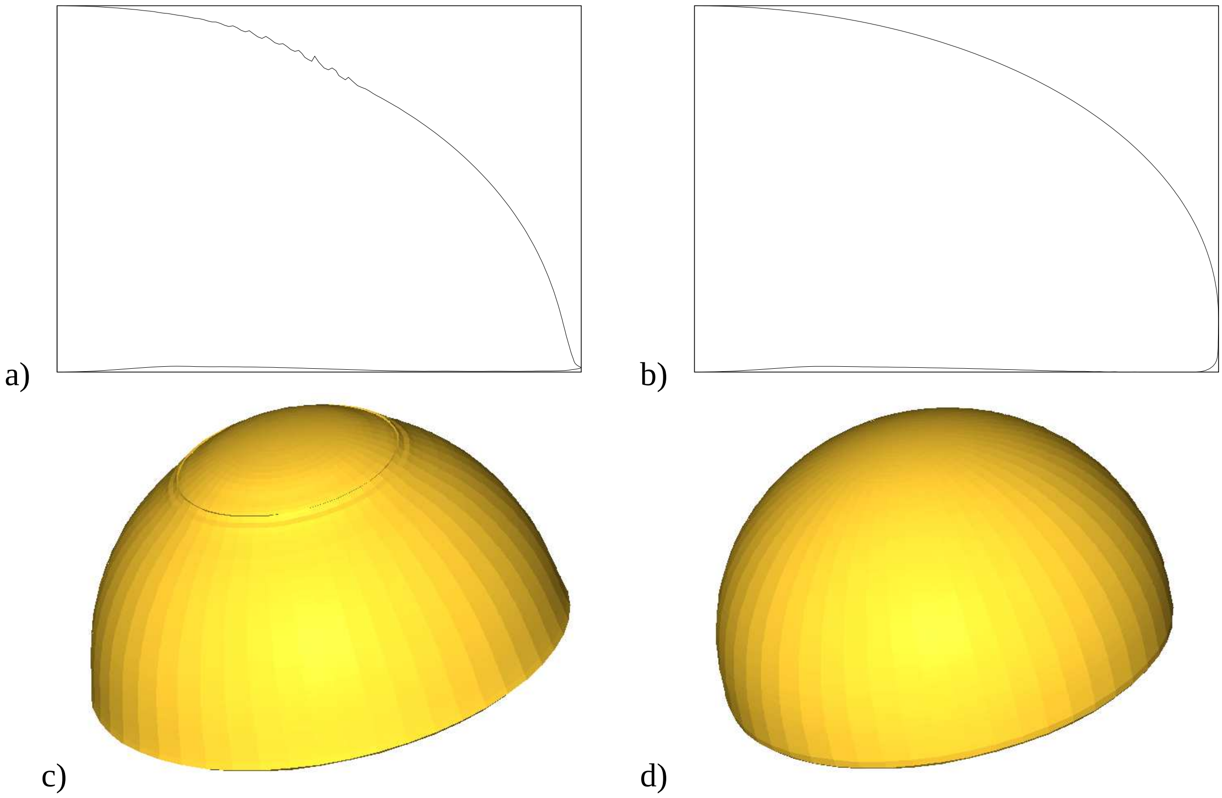

Figure 7: Sessile drop; a) and c) without TSUR; b) and d) with TSUR.

4.3 Three-dimensional flow

The effect of applying TSUR in a three-dimensional flow simulation is illustrated in figure 8. This example shows the simulation of a pendant drop whose parameters are $Re = 0.125$, $Fr = 0.25$, and $We = 0.02$; it is discretized using a uniform $60 \times 60 \times 60$ cell mesh. Figure 8a) shows the simulation without TSUR-3D. Note that the free surface presents large undulations, comprising the precision. Figure 8b) is the same simulation with TSUR-3D, which smooth the surface in a very satisfactory way. In this simulation TSUR-3D were applied after 100 cycles. It is worth comparing the volumes obtained with and without TSUR-3D. The volume of fluid in figure 8a) (without TSUR-3D) is 0.860632457 whereas in figure 8b) (with TSUR-3D) the obtained volume is 0.860760349, given a difference smaller than 10^{-3}, what we consider satisfactory.

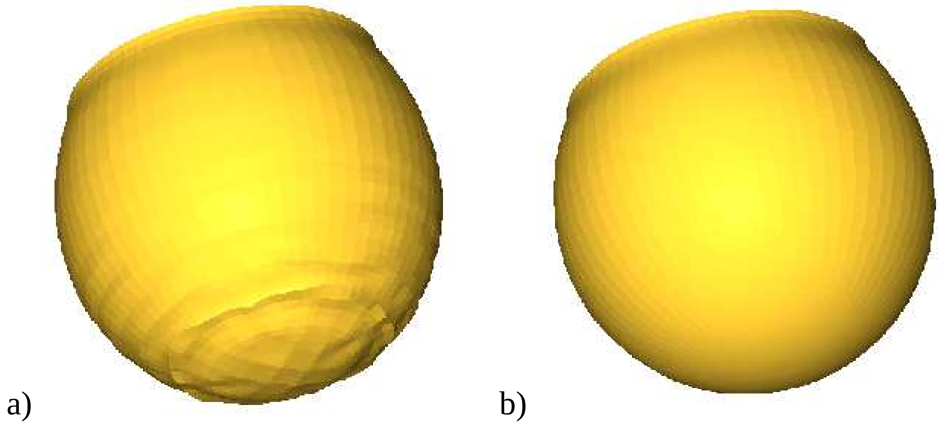

Figure 8: Pendant drop; a) without TSUR; b) with TSUR

We finish this section illustrating the accuracy of TSUR in preserving volumes. Figure 9 shows a sequence of images generated by applying TSUR-3D in the cube of figure 9a). In figure 9b) TSUR-3D where applied 15 times, in figure 9c) 45 times, and in figure 9d) 90 times. The volumes obtained in each example are presented into table 1. The second column is the original volume of the cube in figure 9a). Notice that TSUR increases the volume in a rate of 10^{-5} in the first steps, reducing to 10^{-6} in the final steps.

	0	15	45	90
Volumes	0.504041595	0.504073084	0.50411593	0.504150144

Table 1: Volumes obtained after applying TSUR 15, 45, and 90 times.

5 CONCLUSIONS

In this paper we have presented the trapezoidal sub-grid undulations removal (TSUR) technique, which is capable to smooth curves and surfaces while preserving the volume. Two-dimensional, axisymmetric, and three-dimensional versions of TSUR has been implemented and incorporated into GENSMAC code.

The efficacy of TSUR has been illustrated in three distinct cases: planar flow, axisymmetric, and three-dimensional flow. In the case of planar flow, a container filling simulation has been executed. A simulation of a sessile drop was performed to illustrate TSUR in axisymmetric flow. In the three-dimensional case, a pendant drop has been simulated. The efficacy of TSUR in removing undulations has been verified from the results.

A test concerned with the accuracy in preserving the volume has also been performed, showing that TSUR is a reliable tool to be employed in incompressible fluid flow simulation with free surface.

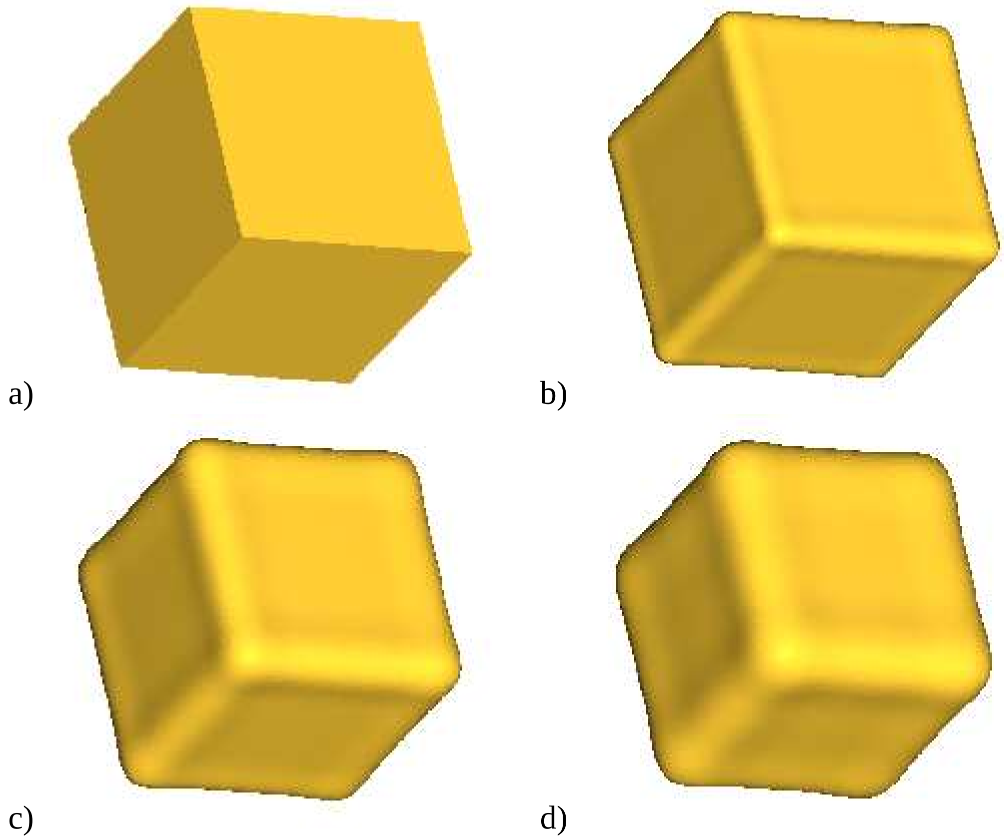

Figure 9: TSUR-3D applied in a cube.

REFERENCES

[1] Freitag L. On combining laplacian and optimization-based mesh smoothing techniques. In *Joint Summer Meeting of American Society of Mechanical Engineers (ASME), American Society of Civil Engineers (ASCE), and Society of Engineers Science (SES)*, pages 37–44, (1997).

[2] Vollmer J., Menel R., and Müller H. Improved laplacian smoothing of noisy surface meshes. *Computer Graphics Forum*, **18**(3), 131–138 (1999).

[3] Taubin G. A signal processing approach to fair surface design. In *SIGGRAPH'95*, pages 351–358, (2000).

[4] Kuprat A., Khamayseh A., George D., and et al. Volume conserving smoothing for piecewise linear curves, surfaces, and triple lines. *Journal of Computational Physics*, **172**(1), 99–118 (2001).

[5] Liu X.G., Bao H.J., Shum H.Y., and et al. A novel volume constrained smoothing method for meshes. *GRAPHICAL MODELS*, **64**(3-4), 169–182 (2002).

[6] Castelo A., Mangiavacchi N., Tomé M.F., Cuminato J.A., Fortuna A.O, Oliveira J., and S. McKee. Surface tension implementation for gensmac2d. *Journal of the Brazilian Society of Mechanical Sciences*, **23**, 523–534 (2001).

[7] Tomé M. F., A. Castelo, Cuminato J. A., Mangiavacchi N., and McKee S. Gensmac3d: A numerical method for solving unsteady three-dimensional free surface flows. *International Journal for Numerical Methods in Fluids*, **37**(7), 747–796 (2001).

[8] Tomé M. F. and McKee S. Gensmac: A computational marker-and-cell method for free surface flows in general domains. *Journal of Computational Physics*, **110**, 171–186 (1994).

[9] Preparata F.P. and Shamos M.I. *Computational Geometry: an introduction*. Springer-Verlag, (1985).

[10] de Berg M., van Kreveld M., Overmars M., and Schwarzkopf O. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, (1997).

[11] Mäntylä M. *An introduction to solid modeling*. Computer Science Press, (1988).