

# A New Smoothing Algorithm for Quadrilateral and Hexahedral Meshes

Sanjay Kumar Khattri<sup>1</sup>

Department of Mathematics, University of Bergen, Norway  
sanjay@mi.uib.no  
<http://www.mi.uib.no/~sanjay>

**Abstract.** Mesh smoothing (or r-refinement) are used in computer aided design, interpolation, numerical solution of partial differential equations, etc. We derive a new smoothing called parallelogram smoothing. The new smoothing tries to fit a given domain by the parallelograms. We present several numerical examples and compare our results against the traditional Laplacian smoothing. Presented numerical work shows that the new approach is superior to the Laplacian smoothing.

## 1 Introduction

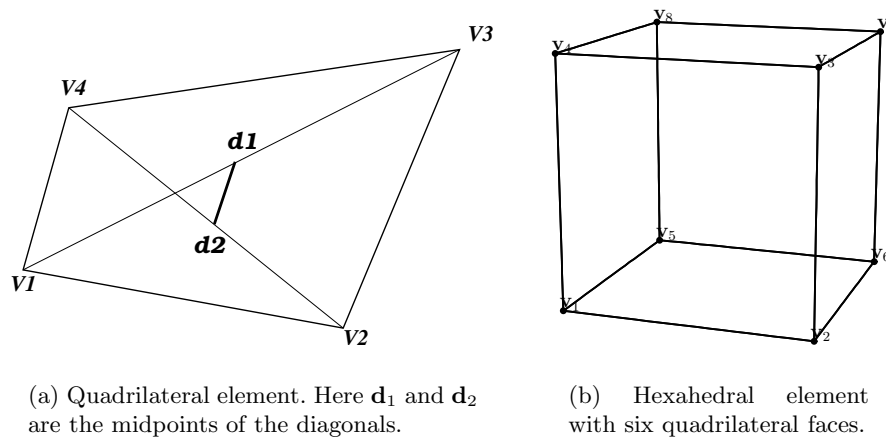
Mesh smoothing such as the Laplacian smoothing [13] and the Winslow smoothing [14] has many applications. For example, computer graphics [7–9, and references therein], interpolation [10, and references therein], numerical solution of partial differential equations [11–14, 17, and references therein], Arbitrary Lagrangian Eulerian Methods [1] and mesh generation [3–5]. It is known (see [2, 13, 17]) that mesh smoothing improves accuracy of the solution and also reduces the overall computational effort. Smoothing is effective in improving distorted and inverted elements in automatically generated meshes [13]. Mesh smoothing adjusts the positions of the interior vertices in the mesh while preserving its topology [13, 17]. Due to simplicity and computational efficiency, the Laplace smoothing is the most popular method for improving mesh quality. It repositions the vertices of the mesh by moving each interior node to the geometric center of its neighbours [13]. However, the Laplacian smoothing can produce inverted elements (See the Section 3) and such meshes cannot be used in applications. We present a new mesh smoothing algorithm called the parallelogram smoothing. Numerical work shows that the new smoothing is performing better than the Laplacian smoothing. Parallelogram smoothing tries to fit a given domain with parallelograms in 2D and parallelepipeds in 3D.

Let us mention advantages of the parallelogram smoothing. The error of the Streamline Methods on parallelogram and parallelepiped mesh is minimum (see the Figures 7, 11 and 15 in [15]). Many discretization methods such as the Multi Point Flux Approximation (MPFA) results in a symmetric discrete system on a parallelogram or parallelepiped mesh [16]. So, efficient solvers such as the Conjugate Gradient can be used. Quadrilateral and hexahedral elements

are expressed by bi-linear and tri-linear mappings from a reference square and cube respectively. These mappings are widely used with discretization techniques such as Finite Element Methods, Finite Volume Methods [16] and Streamline Methods [15]. One of the major problem in numerical computations is ensuring the invertibility of the bi-linear and tri-linear mappings. Since these mappings are not linear, the positivity of the Jacobian at the vertices does not ensure the global invertibility of the mappings (or ensuring the invertibility of the mapping is computationally intensive). One encouraging fact is that for parallelograms and parallelepipeds elements these mappings become linear.

An outline of the article is as follows. In the Section 2, a measure of parallelogram property of quadrilateral and hexahedral elements, and a quadratic parallelogram functional for quadrilateral and hexahedral meshes are presented. Optimization of the functional leads to a new smoothing algorithm. Section 3 presents numerical work and comparison with the Laplacian smoothing. Finally Section 4 concludes the article.

## 2 Measure of Parallelogramness and Parallelogram Functional



**Fig. 1.** Quadrilateral and Hexahedral Elements.

Figure 1(a) shows a quadrilateral element with vertices  $\mathbf{v}_1$ ,  $\mathbf{v}_2$ ,  $\mathbf{v}_3$  and  $\mathbf{v}_4$ . The midpoints of the diagonals  $\mathbf{v}_1\mathbf{v}_3$  and  $\mathbf{v}_2\mathbf{v}_4$  are  $\mathbf{d}_1$  and  $\mathbf{d}_2$  respectively. Lets call this quadrilateral element  $k$ . Euclidean distance between the midpoints  $\mathbf{d}_1$  and  $\mathbf{d}_2$  is

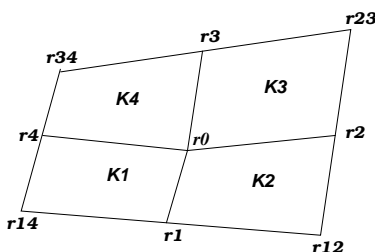
$$Q(k) = \left\| \frac{\mathbf{v}_1 + \mathbf{v}_3}{2} - \frac{\mathbf{v}_2 + \mathbf{v}_4}{2} \right\|_{L_2}. \quad (1)$$

Here  $k$  means that we are referring to the quadrilateral element named  $k$ . Since the mid points of two diagonals of a parallelogram co-incide; i.e.,  $\mathbf{v}_1 + \mathbf{v}_3 = \mathbf{v}_2 + \mathbf{v}_4$ . If  $Q(k)$  approaches zero than the element  $k$  approaches a parallelogram and for  $Q(k)$  equal to zero the element  $k$  will be a parallelogram.  $Q(k)$  will be called a measure of the parallelogramness of the quadrilateral element  $k$ .

Figure 1(b) shows a planar hexahedral element (a hexahedra with planar surfaces) with vertices  $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5, \mathbf{v}_6, \mathbf{v}_7$  and  $\mathbf{v}_8$ . Lets call this hexahedra  $k$ . Let us further denote quadrilateral surfaces of the hexahedra  $k$  by  $s_i$  where  $i = 1 \dots 6$ . The surface  $s_i$  will be a parallelogram if  $Q(s_i)$  is equal to zero. Parallelogramness of the hexahedral element  $k$  can be expressed as a sum of parallelogramness of the six quadrilateral faces

$$Q(k) = \sum_{i=1}^6 \|Q(s_i)\|_{L_2} . \quad (2)$$

A hexahedral element is a parallelogram if the six surfaces are parallelograms. Thus the hexahedral element  $k$  will be a parallelogram if  $Q(k)$  is equal to zero. Figure 2 shows a structured  $2 \times 2$  quadrilateral mesh. The mesh shown in the



**Fig. 2.** A  $2 \times 2$  structured quadrilateral mesh.

Figure 2 consists of four quadrilateral elements lets call them  $k_i$ ,  $i = 1 \dots 4$ . For the  $2 \times 2$  mesh shown in the Figure, 2  $\mathbf{r}_{14}, \mathbf{r}_1, \mathbf{r}_{12}, \mathbf{r}_2, \mathbf{r}_{23}, \mathbf{r}_3, \mathbf{r}_{34}, \mathbf{r}_4$  are the boundary nodes and these nodes are fixed during smoothing process. Let the coordinates of the node  $\mathbf{r}_i$  be  $(x_i, y_i)$ . The Laplacian smoothing moves grid points to the geometric center of incident vertices. The Laplacian smoothed position of the node  $\mathbf{r}_0 = (x_0, y_0)$  is

$$\mathbf{r}_0 = \frac{\mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_3 + \mathbf{r}_4}{4.0} . \quad (3)$$

For deriving an alternative smoothing, let us first define a functional  $f$  for a mesh shown in the Figure 2 as

$$f(x_0, y_0) = \sum_{i=1}^4 Q(k_i)^2 . \quad (4)$$

#### IV

Here, index  $i$  is for the four surrounding quadrilaterals around the node 0 and  $Q(k_i)$  is the measurement of the parallelogramness of the quadrilateral element named  $k_i$  in the mesh. Similar functional can also be defined for unstructured quadrilateral meshes. It should be noted that  $f(x_0, y_0)$  will be zero if  $(x_0, y_0)$  is situated such that the four surrounding quadrilaterals are parallelograms. Functional  $f$  is quadratic in nature and it can be easily shown that it is convex; i.e., the Hessian of the functional  $f$  is positive definite. Figure 3(a) shows the elliptic contours of this functional for a quadrilateral mesh and Figure 3(b) shows the surface plot of the functional  $f$ . Since functional  $f$  is convex so it has a unique minimum. The minimization of this functional  $f$  will provide the proper nodal position for the node  $\mathbf{r}_0$  and the newly formed surrounding cells around this node will be the best parallelogram fit (if squares or rectangles can be fit, the minimization will provide the nodal position for squares or rectangles fit since square and rectangle are a special kind of parallelograms). It can be shown that the independent variables  $x_0$  and  $y_0$  ( $\mathbf{r}_0$ ) that minimizes the functional  $f$  given by the equation (4) are

$$\mathbf{r}_0 = \frac{\mathbf{r}_{14} + \mathbf{r}_{12} + \mathbf{r}_{23} + \mathbf{r}_{34}}{4.0} - \frac{\mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_3 + \mathbf{r}_4}{2.0} . \quad (5)$$

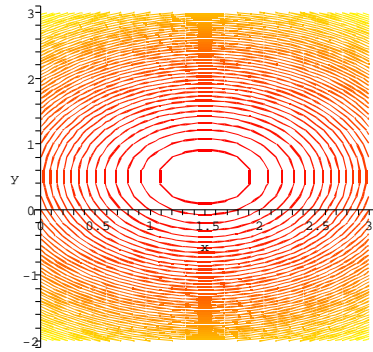
Our new smoothing approach for a structured quadrilateral mesh is given by the equation (5). Thus, our new approach will move internal nodes according to the equation (5). In the next Section several interesting numerical examples are explored for verifying the robustness of the new smoothing given by equation (5) compared to the Laplacian smoothing given by equations (3). Similar smoothing can be derived for structured or unstructured hexahedral meshes. Figure 4 shows a structured  $2 \times 2 \times 2$  hexahedral mesh. The mesh 4 consists of eight hexahedral elements lets call them  $k_i$  with  $i = 1, \dots, 8$ . For the mesh 4, 14 is the internal node while  $1, \dots, 13$  and  $15, \dots, 27$  are the boundary nodes. Boundary nodes are kept fixed during smoothing process. Let the coordinate of a node  $i$  is given as  $\mathbf{r}_i = (x_i, y_i)$ . The Laplacian smoothing moves grid points to the geometric center of the incident vertices. The Laplacian smoothed position of the node 14 is given as

$$\mathbf{r}_{14} = \frac{\mathbf{r}_{11} + \mathbf{r}_{15} + \mathbf{r}_{17} + \mathbf{r}_{13} + \mathbf{r}_5 + \mathbf{r}_{23}}{6.0} , \quad (6)$$

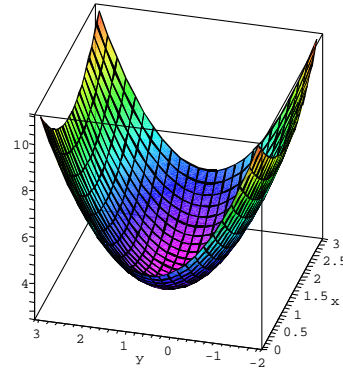
For deriving an alternative smoothed position for the node 14 shown in Figure 4, we define a functional  $f$  as follows

$$f(x_{14}, y_{14}, z_{14}) = \sum_{i=1}^8 Q(k_i) . \quad (7)$$

Here index  $i$  runs over eight surrounding hexahedral elements around the node 14 and  $Q(k_i)$  is measurement of the parallelogramness of the  $i^{th}$  hexahedra defined by equation (2). Similar functional can also be defined for unstructured hexahedral meshes. It can be seen that  $f(x_{14}, y_{14}, z_{14})$  will be zero if the coordinates  $(x_{14}, y_{14}, z_{14})$  of the node 14 are situated such that the surrounding hexahedrals are parallelepiped. Again the functional  $f$  is quadratic and convex with a positive

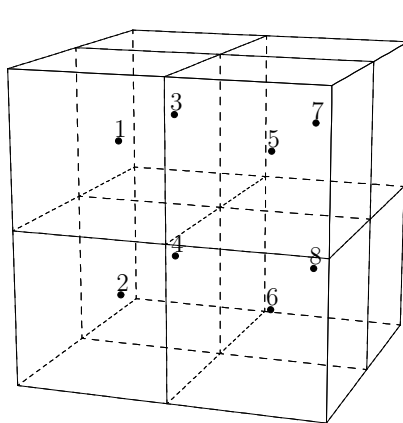


(a) Contours of the functional  $f$ .

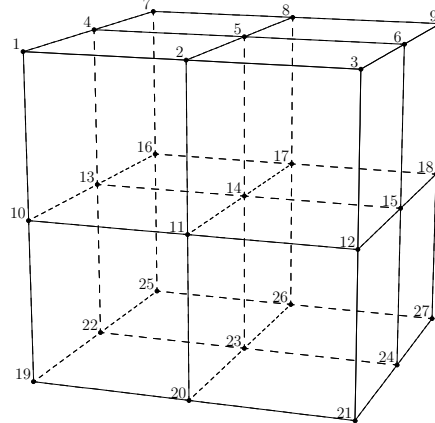


(b) Surface plot the functional  $f$ .

**Fig. 3.** Convex nature of the functional  $f$ .



(a) Central node is surrounded by 8 hexahedrals.



(b) Mesh consists of eight hexahedrals.

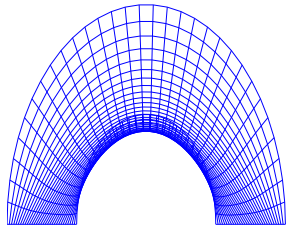
**Fig. 4.** A  $2 \times 2 \times 2$  structured hexahedral mesh.

definite Hessian. It can be shown that independent variables  $(x_{14}, y_{14}, z_{14})$  that minimizes (7) are given as (solution of the equations  $\partial f/\partial x_{14} = 0$ ,  $\partial f/\partial y_{14} = 0$  and  $\partial f/\partial z_{14} = 0$ )

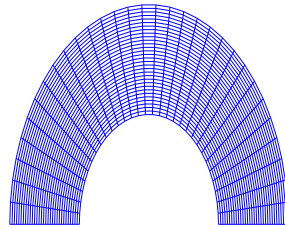
$$\mathbf{r}_{14} = \frac{\mathbf{r}_{11} + \mathbf{r}_{15} + \mathbf{r}_{17} + \mathbf{r}_{13} + \mathbf{r}_5 + \mathbf{r}_{23}}{3.0} - \frac{\mathbf{r}_{10} + \mathbf{r}_{12} + \mathbf{r}_{18} + \mathbf{r}_{16}}{24.0} - \frac{\mathbf{r}_{24} + \mathbf{r}_{26} + \mathbf{r}_2 + \mathbf{r}_6 + \mathbf{r}_8 + \mathbf{r}_4}{48.0}. \quad (8)$$

Equation (8) forms a new smoothing algorithm for structured hexahedral mesh. Similar formulas can be derived for unstructured hexahedral meshes.

### 3 Numerical Experiments



(a) Folded Mesh by the Laplacian Smoothing.



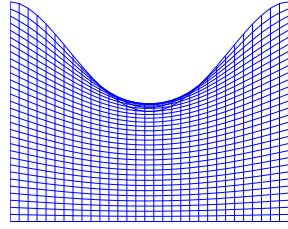
(b) Unfolded Mesh by the New Method.

**Fig. 5.** Comparison of Laplacian and New Method.

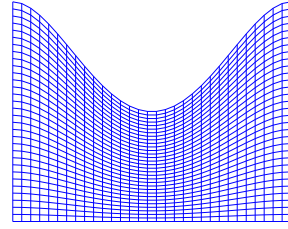
We are using both the new approach and the Laplacian method for quadrilateral mesh smoothing. Figures 7 and 5 report outcome of our numerical experiments. Initial grids are generated by Transfinite Interpolation. It is clear from the Figures that for 2-D geometries the Laplacian smoothing can generate folded grids and the resulting mesh can not be used in numerical simulations. It can be seen in the Figures 7 and 5 that the Laplacian smoothing move nodes outside the boundary and it results in inverted elements. Numerical experiments demonstrate robust nature of the new approach.

### 4 Conclusions

A simple alternative to the traditional Laplacian smoothing is presented. Reported numerical work shows that the new approach is superior to the Lapla-

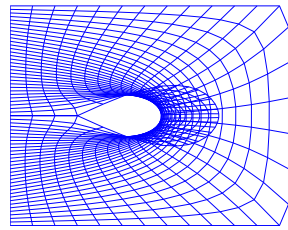


(a) Folded Mesh by the Laplacian Smoothing.

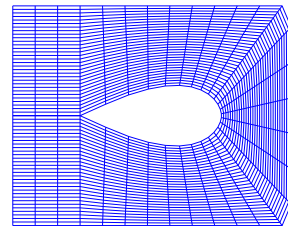


(b) Unfolded Mesh by the New Method.

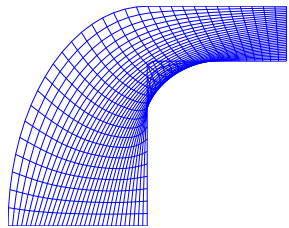
**Fig. 6.** Comparison of Laplacian and New Method.



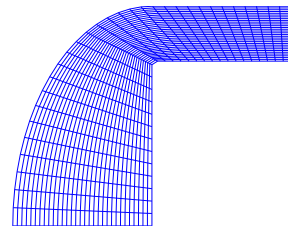
(a) Folded Mesh by the Laplacian Smoothing.



(b) Unfolded Mesh by the New Method.



(c) Folded Mesh by the Laplacian Smoothing.



(d) Unfolded Mesh by the New Method.

**Fig. 7.** Comparison of Laplacian and New Method.

cian smoothing. This method is simple to understand and can be implemented in an existing mesh generation package. Smoothing is derived for the structured quadrilaterals and hexahedral meshes. The real use of this approach can be in the area of hexahedral mesh generation where many meshing algorithms are capable of generating inverted elements.

## References

1. Hermansson, J. and Hansbo, P.: A variable diffusion method for mesh smoothing, *Comm. Numer. Methods Engrg.*, **19**, (2003), 897-908.
2. Daniel, W. and Garry, R.: Improved vector FEM solutions of Maxwell's equations using grid pre-conditioning, *Internat. J. Numer. Methods Engrg.*, **40**, (1997), 3815-3837.
3. Ding, Y., Xia, J., Wang, Y. and Xiao, J.: Fully automatic generation of finite element mesh, *Progr. Natur. Sci. (English Ed.)*, **5**, (1995), 93-98.
4. O'Sullivan, F.: Discretized Laplacian smoothing by Fourier methods, *J. Amer. Statist. Assoc.*, **86**, (1991), 634-642.
5. Hansbo, P.: Generalized Laplacian smoothing of unstructured grids. *Communications in Numerical Methods in Engineering*, **11**, (1995), 455-464.
6. Field, D.A.: Laplacian smoothing and Delaunay triangulations. *Communications in Applied Numerical Methods*. **4**, (1998), 709-712.
7. Floater, M.S.: Mean value coordinates. *Computer Aided Geometric Design*. **20**, (2003), 19-27.
8. Khodakovsky, A., Litke, N., and Schröder, P.: Globally smooth parameterizations with low distortion. *ACM Transactions on Graphics (TOG)*. **22**, (2003).
9. Sheffer, A. and Sturler, E.D.: Smoothing an overlay grid to minimize linear distortion in texture mapping. *ACM Transactions on Graphics (TOG)*. **21**, (2002), 874-890.
10. Mallet, J.L.: Discrete smooth interpolation. *ACM Transactions on Graphics (TOG)*. **8**, (1998), 121-144.
11. Bank, R.E. and Smith, R.K.: Mesh smoothing using a posteriori error estimates. *SIAM J. Numer. Anal.* **34**, (1997), 979-997.
12. Frey, W.H. and Field, D.A.: Mesh relaxation: a new technique for improving triangulations. *Internat. J. Numer. Methods Engrg.* **31**, (1991), 1121-1133.
13. Shontz, S.M. and Vavasis, S.A.: A linear weighted laplacian smoothing framework for warping tetrahedral meshes. Submitted to *SIAM Journal on Scientific Computing* on July 17, 2004 for publication. Available on line at <http://arxiv.org/abs/cs.NA/0410045>.
14. Knupp, P.M.: Winslow smoothing on two-dimensional unstructured meshes. *Proceedings, 7th International Meshing Roundtable, Sandia National Lab.* (1998), 449-457.
15. Hægland, H., Dahle, H.K., Eigestad, G.T., Lie, K.A. and Aavatsmark, I.: Improved Streamlines and Time of Flight for Streamline Simulation on Irregular Grids. Submitted in *Journal*. Available on line at <http://heim.ifi.uio.no/kalie/papers/cvisl.pdf>. October (2005).
16. Khattri, S.K.: Analyzing Finite Volume for Single Phase Flow in Porous Media. *Journal of Porous Media*. Accepted for Publication, (2006).
17. Persson, P.-O. and Strang, G.: A Simple Mesh Generator in MATLAB. *SIAM Review*, 46 (2), (2004), 329-345.