

# DARSS: a hybrid mesh smoother for all hexahedral meshes

Dhaval Jani · Anoop Chawla · Sudipto Mukherjee ·  
Raman Khattri

Received: 7 January 2011 / Accepted: 15 June 2011 / Published online: 8 July 2011  
© Springer-Verlag London Limited 2011

**Abstract** A method for smoothing hexahedral meshes has been developed. The method consists of two phases. In the first phase, the nodes are moved based on an explicit formulation. A constraint has also been implemented to prevent the deterioration of elements associated with the node being moved. The second phase of the method is optismoothing based on the Nelder–Mead simplex method. The summation of the Jacobian of all the elements sharing a node has been taken as the function to be maximized. The method has been tested on meshes up to 18,305 hexahedral elements and was found to be stable and improved the mesh in about 112.6 s on an Intel Centrino® 1.6 GHz, 1 GB RAM machine. The method thus has the advantage of being effective as well as being computationally efficient.

**Keywords** Mesh smoothing · Optismoothing · Mesh refinement · Mesh quality improvement

## 1 Introduction

Finite element methods require spatial decomposition of the computational domain into simple geometric elements like triangle or quadrilateral in 2D and tetrahedral or hexahedral in 3D. Algorithms, such as virtual decomposition [1], sweeping [2], whisker weaving [3], grafting [4], H-Morph [5], have been devised to create unstructured hexahedral and hex-dominant meshes. In addition, many automatic mesh generation tools are used for the discretization of the domain. However, for complex geometries, the generated meshes may contain elements with significantly varying shapes and even inverted elements and such elements may also be generated while modifying the geometry of the meshed component. The presence of such elements in the mesh increases solution time and leads to erroneous results [6–8]. Meshes with such elements also affect the efficiency of the simulations. Mesh smoothing techniques repair poor quality meshes by adjusting grid point locations without changing the mesh topology. In general, most of these smoothing techniques adjust the geometric position of each node individually. Mesh smoothing is also known to improve accuracy of the solution and reduce the overall computational effort. Most often, these smoothing methods are used as a final step during mesh generation, to regulate elemental shape variations from an ideal shape. Despite flurry of activity in the field of mesh modification [9–19], this remains a difficult and computationally expensive problem.

The most widely used local smoothing algorithm is Laplacian smoothing [11]. The node of interest is moved to the centroid of the neighbors without any evaluation of the quality of resulting elements. It operates heuristically and works quite well for meshes in convex regions. However, there are instances of it generating elements of worse

---

D. Jani · A. Chawla (✉) · S. Mukherjee  
Department of Mechanical Engineering, Indian Institute  
of Technology Delhi, New Delhi 110016, India  
e-mail: achawla64@gmail.com

D. Jani  
e-mail: jani\_2211@yahoo.co.in

S. Mukherjee  
e-mail: sudipto@mech.iitd.ac.in

R. Khattri  
Department of Computer Science, Indian Institute of Technology  
Delhi, New Delhi 110016, India  
e-mail: raman2805@gmail.com

quality than those a technique to optimize element quality has been reported [13]. The optimization algorithm was based on the steepest descent method and moved a node within the feasible region to improve the quality of associated elements.

A modified Laplacian smoothing method was introduced which moved a node only if the resulting quality of the surrounding elements was improved [16]. However, the method did not move the node if the calculated position of a node did not improve the quality of the mesh even if the surrounding elements were invalid; hence, the method does not ensure that the resulting mesh will have no invalid elements. The isoparametric approach [18] too does not ensure that the mesh has only valid elements [15].

Methods like Winslow smoothing [18] and geometric transformation-based smoothing [19] are limited to two-dimensional unstructured meshes. Adaption of such methods for three-dimensional meshes is not obvious. A criterion for evaluating various surface mesh optimization techniques has been presented in [20]. A method for untangling 3D tetrahedral as well as hexahedral meshes has been presented [21].

In addition to such methods, there also exist optimization-based smoothing methods. Some of these methods like [9, 22] are not suitable for large meshes. A method based on maximizing a variant of the scaled—Jacobian metric has been described in [23]. However, the method does not ensure that the improved mesh will (a) consist only of untangled (valid) elements and (b) improve the shape. They use a quality measure based on the condition number of a set of Jacobian metrics and numerical optimization is performed by a conjugate gradient and line-search method [24]. A method of mesh smoothing, where each node is moved to attain equilibrium as a center of bubbles, was proposed in [25, 26]. An angle-based smoother for the planar meshes was proposed by [27]. Application of a variational functional formulated using a local cell quality metric for mesh improvement has been demonstrated on 2D and 3D meshes in [28] while use of gradient-based smoothing has been demonstrated in [29]. A sequential geometric element transformation method (GETMe) for smoothing of triangular surface meshes has been reported in [30]. Geometric transformations are used to iteratively improve the worst element of the mesh to regular shape element and hence achieve mesh improvement. In [31], this approach has been generalized to a simultaneous approach for triangular/quadrilateral mixed surface meshes in which all mesh elements are transformed simultaneously and node updates are obtained by transformed node averaging. Such regularizing transformations have been shown to exist for polygons with an arbitrary number of nodes [32, 33]. The sequential as well as the simultaneous GETMe approach has been extended

to 3D meshes, i.e. tetrahedral meshes [34] and to hexahedral meshes [35].

A method for improvement of hexahedral solid mesh based on quasi-statistical modeling of mesh quality parameters was presented in [36]. Alternatively, the approach given in [37] is based on space mapping. The method based on implementation of quasi-statistical modeling to produce elements with a Gaussian distribution of the mesh quality parameter values was presented in [38]. The reported mesh smoothing approach was based on signal processing techniques. Other optimization-based methods include those developed by [39–41]. A constrained-based smoothing of boundary meshes has been presented in [42].

In the present study, a smoothing algorithm which we have named *DARSS* has been developed and implemented to smooth meshes with hexahedral elements. The algorithm consists of two phases. The first phase is an explicit step to compute the new position of the nodes while the second phase optimizes the position of the node of interest. The first phase of *DARSS* is enhancement of the parallelogram smoothing algorithm reported in [15], where the parallelogram smoothing method is extended to smooth all hexahedral unstructured meshes. Also, the algorithm is strengthened with constraints to preserve the existing quality of elements. Elements whose quality could not be improved during the first phase were smoothed in the second stage using a technique based on Nelder–Mead simplex optimization algorithm. The second phase of the method can also be used for fine local improvements of all elements.

For the implementation of the *DARSS* algorithm, a program was developed in C++. The first phase of the algorithm has been explicitly coded while, the GNU Scientific Libraries (GSL) were used for implementation of Nelder–Mead minimization. The program also includes an implementation of Laplacian smoothing to compare with the algorithm proposed. The algorithm was tested on finite element models of complex geometries representing anatomical structure of human body. The model consisted of hexa-dominant meshes representing all major bones, flesh, muscles and other soft tissues. The mesh quality parameters (Jacobian, aspect ratio, Warp and Skew) were measured in HyperMesh®, Altair HyperWorks. As the *DARSS* has its genesis in parallelogram smoothing [15], the next section gives brief discussion of parallelogram smoothing.

## 2 Parallelogram smoothing

The parallelogram smoothing [15] technique that forms the basis of our technique is discussed briefly.

Figure 1 shows a quadrilateral element with vertices  $V_1$ ,  $V_2$ ,  $V_3$  and  $V_4$ . The midpoints of the diagonals  $V_1V_3$  and  $V_2V_4$  are  $d_1$  and  $d_2$ , respectively. Let this quadrilateral element be called  $k$ . Euclidean distance between the midpoints  $d_1$  and  $d_2$  is

$$D_k = \left\| \frac{V_1 + V_3}{2} - \frac{V_2 + V_4}{2} \right\| \quad (1)$$

If the mid points of two diagonals of a quadrilateral coincide; i.e.,  $V_1 + V_3 = V_2 + V_4$ ,  $D_k$  approaches zero and the element  $k$  approaches a parallelogram.  $D_k$  is hence a measure of the parallelogramness of the element  $k$ .

In Fig. 2, a set of quadrilateral elements  $k_i$  share a node  $r_0$ . A functional for the mesh can be defined as:

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

Here, index  $i$  is for the four surrounding quadrilaterals around the node  $r_0$  and  $D(k_i)$  is the measurement of the parallelogramness of the quadrilateral element  $k_i$  in the mesh. The minimization of this functional  $f(x_0, y_0)$  provides the new nodal position for the node  $r_0$  and the newly formed surrounding cells around this node will have the preferred geometry.

It can be shown that the independent variable  $r_0(x_0$  and  $y_0)$  that minimizes the functional  $f$  are

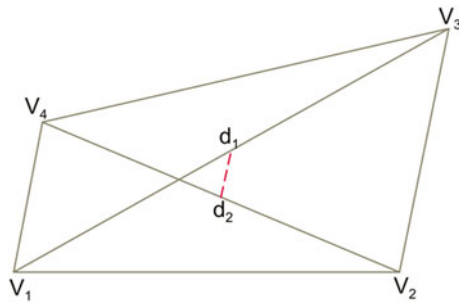


**Fig. 1** A four noded element showing the mid-points ( $d_1$  and  $d_2$ ) of its diagonals (adapted from [15])

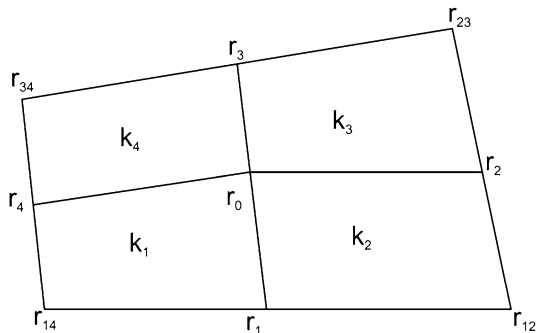


**Fig. 2** Structured quadrilateral mesh (adapted from [15])

$$r_0 = \frac{r_{14} + r_{12} + r_{23} + r_{34}}{4} - \frac{r_1 + r_2 + r_3 + r_4}{2} \quad (3)$$

Equation (3) gives the new coordinates of the node. This formulation was developed in [15] for structured hexahedral meshes, where each internal node is associated with exactly eight elements.

Figure 3 shows a hexahedral element with vertices  $V_1$ ,  $V_2$ ,  $V_3$ ,  $V_4$ ,  $V_5$ ,  $V_6$ ,  $V_7$  and  $V_8$ . This hexahedra is denoted as  $k$ . Further, quadrilateral faces of the hexahedra  $k$  are denoted by  $S_i$ , where  $i = 1-6$ . The face  $S_i$  will be a parallelogram if  $D(S_i)$  is equal to zero. Parallelogramness of the hexahedral element  $k$  can be expressed as a sum of parallelogramness of the six quadrilateral faces.

$$D_k = \sum_{i=1}^6 \|D(S_i)\| \quad (4)$$

Thus, the hexahedral element  $k$  will be a parallelepiped if  $D_k$  is equal to zero.

Figure 4 shows a structured  $2 \times 2 \times 2$  hexahedral mesh. The mesh consists of eight hexahedral elements,  $k_i$ ,  $i = 1-8$ . The nodes in the mesh can be seen to be in three horizontal layers (1, 2 and 3). The node of interest is '14' and its new coordinates are required to be computed. For finding the improved position  $r_i$  of the node  $i$  (here node 14), a functional is defined as follows:

$$f((r_i)_{x,y,z}) = \sum_{i=1}^N D(k_i)^2 \quad (5)$$

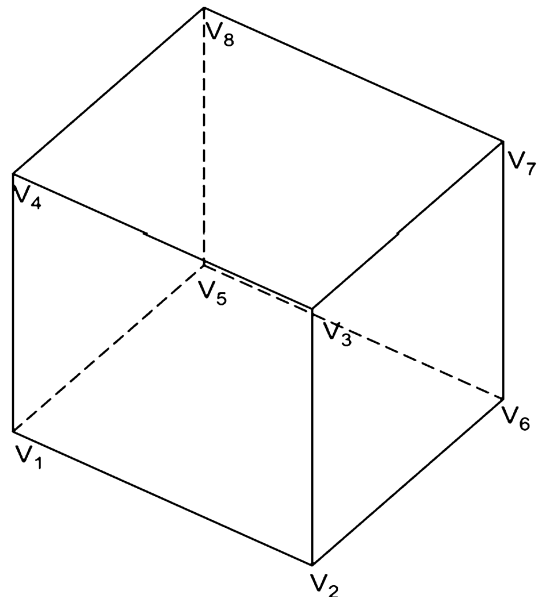


**Fig. 3** Hexahedral element defined with six quadrilateral faces (adapted from [15])

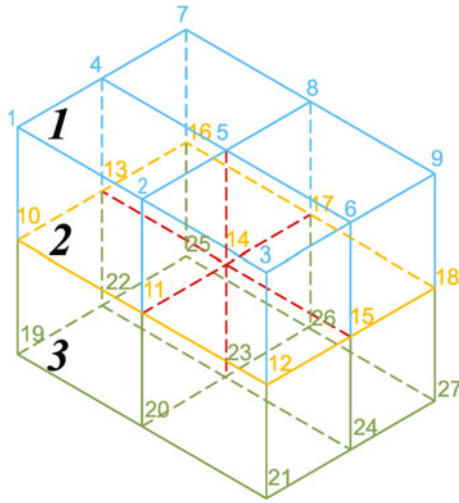


**Fig. 4** Hexahedral  $2 \times 2 \times 2$  mesh

where,  $N$  is the number of elements sharing the node  $i$ . Minimizing the functional  $f$  in Eq. (5) gives the new coordinates  $r_i(x, y, z)$  as follows:

$$r_i = r_{14} = \frac{r_5 + r_{11} + r_{13} + r_{15} + r_{23} + r_{17}}{3} - \frac{r_{16} + r_{12} + r_{24} + r_{10} + r_{18} + r_2 + r_{20} + r_{22} + r_{26} + r_4 + r_6 + r_8}{12} \quad (6)$$

Equation (6) gives the new position of an internal node of a structured hexahedral mesh. As reported in [15], the approach avoids inversion of elements unlike Laplacian smoothing [11].

Some of the observations regarding the technique are as follows:

1. The approach uses explicit form and hence it is as quick as Laplacian algorithm.
2. The approach resulted in much better quality of the mesh than Laplacian smoothing and also avoids generation of inverted elements [15].
3. The application of the explicit form is limited to structured hexahedral meshes. For the unstructured and hexa-dominant meshes, a new form needs to be derived.
4. The degree of parallelogramness used as the objective function, fails to work for polygonal or polyhedral elements with large aspect ratio (for instance, rectangles vs. square for 2D). Hence, the quality of the resulting mesh in such cases is not assured unless external constraints are imposed.
5. So far, the technique has been developed and demonstrated only for structured hexahedral meshes.

In the current work, the parallelogram smoothing technique has been extended to address the above limitations.

### 3 DARSS smoothing

Some of the requirements for getting a smoother mesh can be listed as follows:

- The algorithm should work for all hexahedral unstructured meshes created by any mesh generation and refinement technique.
- It should handle even severely distorted elements.
- It should be efficient and robust.
- It should be work over complex geometries.
- It must not give undue preference to one element shape over other.

If a functional similar to that in Eq. (6) is constructed and minimized for the hexahedral meshes of the type shown in Fig. 5a and b, one may derive the following equations for calculating new coordinates of internal nodes:

$$r_i = r_{11} = \frac{4 \times (r_9 + r_{12} + r_{13})}{9} + \frac{3 \times (r_{18} + r_4)}{9} - \frac{r_{16} + r_{10} + r_{14} + r_{19} + r_2 + r_{20} + r_5 + r_6 + r_8}{9} \quad (7)$$

$$r_i = r_{18} = \frac{5 \times (r_7 + r_{29})}{15} + \frac{4 \times (r_{13} + r_{15} + r_{17} + r_{19} + r_{21})}{15} - \frac{r_2 + r_{12} + r_{10} + r_{14} + r_{16} + r_{20} + r_{22} + r_{24} + r_{26} + r_{28} + r_{30} + r_{32} + r_4 + r_6 + r_8}{15} \quad (8)$$

Analyzing Eqs. (6), (7) and (8) one may write an explicit form for the new coordinates  $r_i$  of an internal node being moved as:

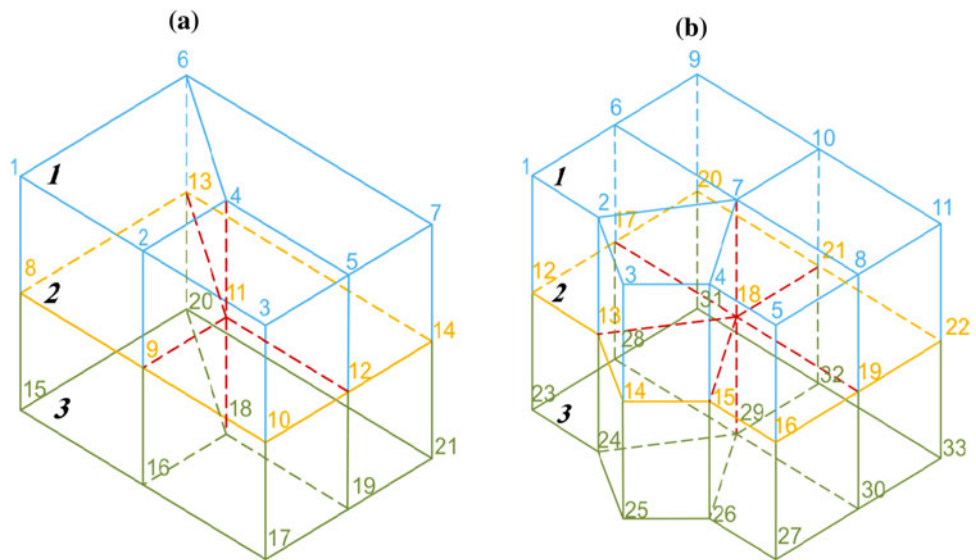
$$r_i = \frac{1}{(3 \times N)} \left[ 8 \times \left( \sum a_j \right) + N \times \left( \sum b_j \right) - 2 \times \left( \sum c_j \right) \right] \quad (9)$$

where  $a_j$  is the coordinates of those immediate neighbors of  $i$  which are shared exactly by two of the elements associated with  $i$  (for instance on plane 2, nodes 9, 12 and 13 in Fig. 5a and nodes 13, 15, 17, 19 and 21 in Fig. 5b),  $b_j$  the coordinates of those immediate neighbors of  $i$  which are shared by exactly  $N/2$  elements associated with  $i$  (for instance on plane 1 and plane 3, nodes 4 and 18 in Fig. 5a and nodes 7 and 29 in Fig. 5b),  $c_j$  the coordinates of nodes on the faces sharing node  $i$  and not included in  $a_j$  or  $b_j$  (for instance, nodes 2, 5, 6, 8, 10, 14, 16, 19 and 20 in Fig. 5a and nodes 2, 4, 6, 8, 10, 12, 14, 16, 20, 22, 24, 26, 28, 30 and 32 in Fig. 5b),  $N$  is the number of elements sharing the node being moved (6 in Fig. 5a and 10 in Fig. 5b).

Equation (9) gives an explicit expression for node positions for smoothing hexa-dominant meshes with variable connectivity.

The robustness of the algorithm can be further enhanced if the node is not allowed to move to such places where it

**Fig. 5** Hexahedral mesh **a** 6 elements (2 layers of 3 elements each) sharing a node ( $i = 11$ ), **b** 10 elements (2 layers of 5 elements each) sharing a node ( $i = 18$ )



may reduce specific element quality below a threshold value. To prevent the deterioration of specific elements associated with the node being moved, a constraint was included in the algorithm that the node is moved only if, quality of all the associated elements either improves or remained the same.

The nodes that could not be moved to a new position due to the above condition are then perturbed using an optimization-based technique so as to improve the quality of the elements in that region.

### 3.1 Optsmoothing

To maintain the overall computational efficiency of the smoothing algorithm, optsmoothing needs to be computationally efficient. It was, therefore, decided to use algebraic operations rather than gradient computations. The Nelder–Mead simplex optimization technique [43] chosen is based on the simplex transformations. A simplex is a special polytope of  $E + 1$  vertices in  $E$  dimensions. For two variables, a simplex is a triangle, and the method is a pattern search that compares function values at the three vertices of a triangle. The worst vertex, where  $f(x, y, z)$  is largest, is rejected and replaced with a new vertex. A new triangle is formed and the search is continued. The process generates a sequence of triangles (which might have different shapes), for which the function values at the vertices get progressively smaller. The size of the triangles is reduced and the coordinates of the minimum point are found.

As the method involves only geometric transformations (Reflect, Contract, Expand and Shrink) and does not require evaluation of the differential, it is effective and computationally compact [44], especially when compared with gradient-based methods.

The Jacobian matrix is the fundamental quantity that describes all the first-order mesh qualities (length, areas and angles). It is appropriate to focus the building of objective functions based on the Jacobian matrix or the associated metric tensor [45].

In *DARSS* also, the function to be maximized at nodes is the sum of the Jacobians of the elements of which the node is a part. The coordinates which maximize the function give the new position of that node. A check is introduced to avoid deterioration of associated elements below the user-defined level. The implementation of optsmoothing based on the Nelder–Mead simplex optimization technique at a node is done as follows:

1. List the set of elements associated with node  $i$ ,  $S_i$
2. Define  $J_i(s, x, y, z)$  as Jacobian of element  $s$  with node  $i$  having coordinates  $x, y, z$ .
3. Compute  $f_i(x, y, z) = \sum_{s \in S_i} J_i(s, x, y, z)$ .
4. Obtain  $(x_o, y_o, z_o)$  so as to maximize  $f_i(x, y, z)$  using NM simplex method.

The complete *DARSS* algorithm can be summarized as follows:

Let the Jacobian of element  $j$  sharing a node  $i$  be  $J_{i,j}$

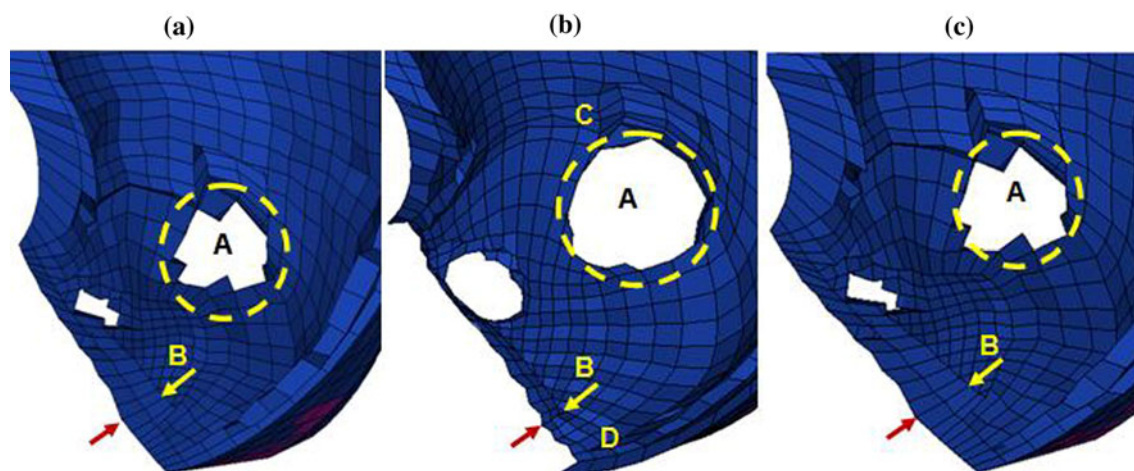


**Fig. 6** Effect of smoothing on mesh representing flesh of pelvic region. **a** Original mesh with distorted elements, **b** mesh after Laplacian smoothing, **c** mesh after smoothing with *DARSS*

Smoothing at a Node  $i_{\text{orig}}$  shared by  $N$  elements would be accomplished as follows:

1. Compute Jacobians of all  $N$  elements at node  $i_{\text{orig}}$ , as  $J_{\text{orig},j}$ , where,  $j = [1, N]$
2. Compute new position of the node  $i_{\text{new}}$   
 Compute the new Jacobian  $J_{\text{new},j}$  for all  $j$   
 if for all  $J_{\text{new},j} \geq J_{\text{orig},j}$ ,  
     move the node  $i_{\text{orig}}$  to  $i_{\text{new}}$   
 else  
     node  $i_{\text{orig}}$  is not moved.
3. Step 1 and 2 are repeated for all the nodes in the mesh.
4. Mesh is subjected to NM smoothing.

## 4 Results

The algorithm was tested on various meshes with hexahedral elements (structured and unstructured). Test meshes also included a number of degenerated elements formed using coincident nodes. The algorithm worked well on all meshes including those which had a complex physical shape. Two of the meshes from the GM/UVA Human Body Finite Element Model [46, 47] on which the algorithm was run are shown in Figs. 6 and 7 below.

Figure 6 presents the first case and shows the initial mesh (flesh of pelvic region), mesh after Laplacian smoothing and mesh after *DARSS* smoothing. In this case, the mesh had about 4,400 elements and the first phase of *DARSS* smoothing took around 8.6 s. For the same mesh, Laplacian smoothing took about 8 s. The second phase (optismoothing) took around 80 s.

As it can be observed in Fig. 6b, the Laplacian smoothing not only distorted the mesh significantly, but also affected the component geometry. This can be seen as the enlargement of the holes (dimension “A”) present in the mesh and thinning of the component sections

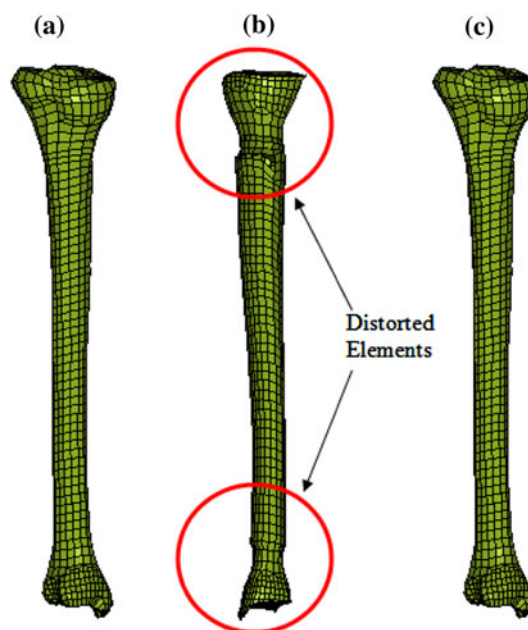
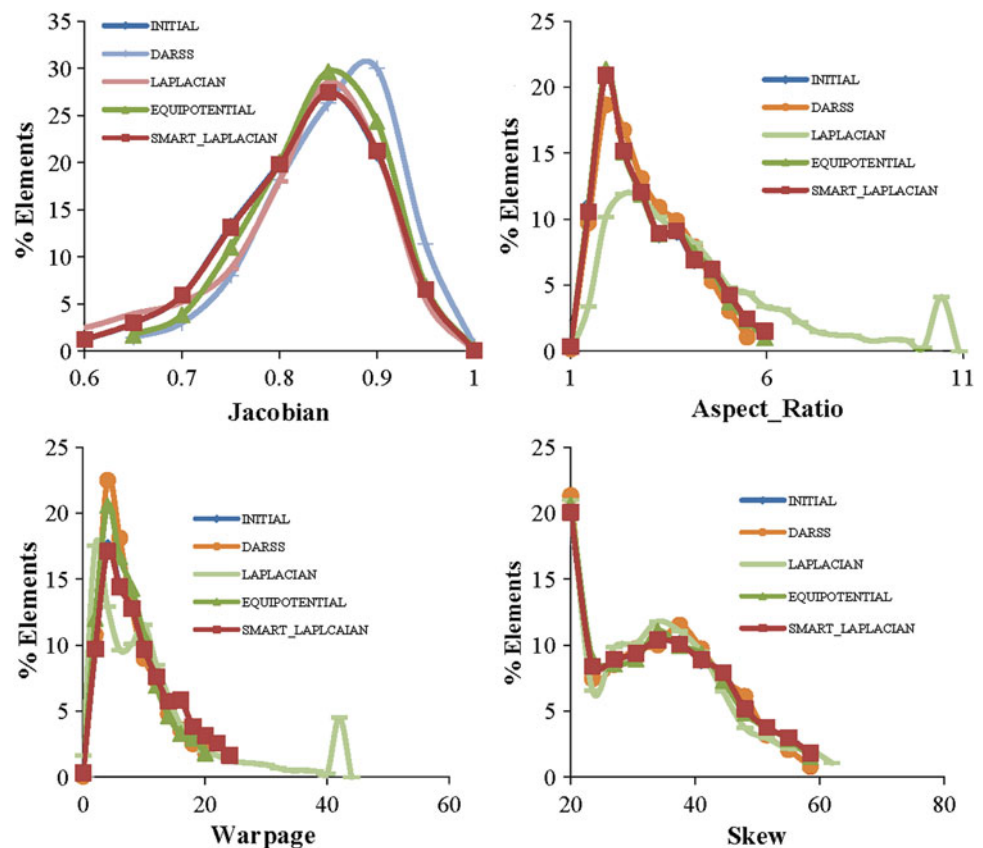


**Fig. 7** Effect of smoothing on mesh representing bone (tibia). **a** Original mesh, **b** mesh after Laplacian smoothing, **c** mesh after *DARSS* smoothing

**Table 1** Comparison of mesh quality parameters

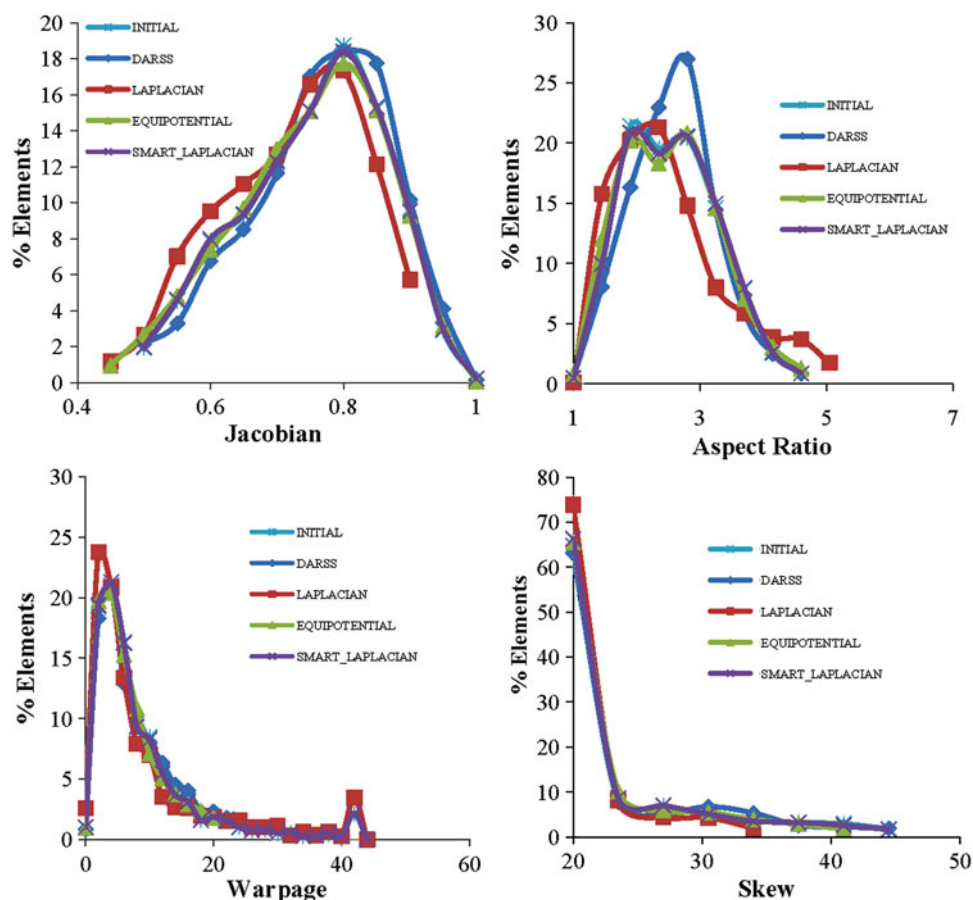
| Smoothing method                                 | Mesh quality metric                     |                                             |                                        |                                     |
|--------------------------------------------------|-----------------------------------------|---------------------------------------------|----------------------------------------|-------------------------------------|
|                                                  | Jacobian worst ( $\bar{X} \pm \sigma$ ) | Aspect ratio worst ( $\bar{X} \pm \sigma$ ) | Warpage worst ( $\bar{X} \pm \sigma$ ) | Skew worst ( $\bar{X} \pm \sigma$ ) |
| Flesh (pelvic region: 4,400 hexahedral elements) |                                         |                                             |                                        |                                     |
| Initial mesh—no smoothing                        | 0.21 (0.83 $\pm$ 0.07)                  | 9.66 (2.9 $\pm$ 1.20)                       | 86.07 (9.2 $\pm$ 4.10)                 | 74.24 (33.6 $\pm$ 10.70)            |
| DARSS                                            | 0.41 (0.84 $\pm$ 0.06)                  | 9.66 (2.85 $\pm$ 1.0)                       | 67.09 (6.9 $\pm$ 4.40)                 | 75.28 (33.2 $\pm$ 10.70)            |
| Laplacian                                        | −8.93 (0.82 $\pm$ 0.09)                 | 25.66 (4.2 $\pm$ 2.20)                      | 176.5 (11.2 $\pm$ 10.10)               | 86.3 (33.4 $\pm$ 10.80)             |
| Smart Laplacian                                  | 0.22 (0.83 $\pm$ 0.07)                  | 9.66 (2.9 $\pm$ 1.20)                       | 86.07 (9.2 $\pm$ 4.10)                 | 74.24 (33.6 $\pm$ 10.70)            |
| Equipotential techniques                         | 0.33 (0.84 $\pm$ 0.06)                  | 9.66 (2.9 $\pm$ 1.2)                        | 67.09 (7.6 $\pm$ 4.50)                 | 74.51 (32.4 $\pm$ 10.20)            |
| Bone (tibia: 2,680 hexahedral elements)          |                                         |                                             |                                        |                                     |
| Initial mesh—no smoothing                        | 0.30 (0.75 $\pm$ 0.11)                  | 9.5 (2.56 $\pm$ 0.73)                       | 44.1 (8.9 $\pm$ 8.5)                   | 65.7 (23.6 $\pm$ 6.3)               |
| DARSS smoothing                                  | 0.38 (0.76 $\pm$ 0.10)                  | 7.07 (2.6 $\pm$ 0.68)                       | 57.68 (7.4 $\pm$ 5.2)                  | 66.91 (23.6 $\pm$ 6.3)              |
| Laplacian                                        | 0.24 (0.72 $\pm$ 0.11)                  | 41.41 (2.5 $\pm$ 0.90)                      | 121.71 (9.2 $\pm$ 9.9)                 | 78.97 (21.6 $\pm$ 3.6)              |
| Smart Laplacian                                  | 0.31 (0.75 $\pm$ 0.11)                  | 8.91 (2.56 $\pm$ 0.73)                      | 101.96 (8.9 $\pm$ 8.5)                 | 59.5 (23.4 $\pm$ 6.1)               |
| Equipotential techniques                         | 0.34 (0.75 $\pm$ 0.11)                  | 9.2 (2.6 $\pm$ 0.76)                        | 44.7 (6.8 $\pm$ 4.5)                   | 65.1 (22.6 $\pm$ 5.3)               |

**Fig. 8** Distribution of Jacobian, aspect ratio, Warpage and Skew in the original mesh, as obtained after smoothing on pelvic region flesh

(dimension “B”). In addition, the resulting mesh has kinks (marked “C”) and severely distorted elements (in the region “D”) with negative Jacobian. The mesh smoothed with *DARSS* smoothing is shown in Fig. 6c. The technique kept the overall component geometry intact while improving the distorted elements.

Figure 7 presents the second case and shows the effect of smoothing on elements of a bone (tibia) mesh with 2,680 hexahedral elements. The bone model presented in this case was a part of lower extremity model which has 18,305 hexahedral elements. As can be seen in Fig. 7b, the Laplacian smoothing distorted the model geometry and

**Fig. 9** Distribution of Jacobian, aspect ratio, Warpage and Skew in the original mesh, as obtained after smoothing on tibia



produced a surface with visible kinks. Figure 7c shows the model after smoothing with *DARSS*, where it can be seen that the model geometry is not altered.

For both the cases presented, smoothed mesh from *DARSS* were compared to the mesh obtained after smoothing with Smart Laplacian and other mesh improvement schemes (<http://www.cs.sandia.gov/optimization/knupp/Introduction.htm>, <http://www.cubit.sandia.gov/>). The equipotential scheme was found to be the most effective on the present mesh and has hence been reported for comparison. The mesh quality parameters of the initial mesh, mesh after Laplacian smoothing, Smart Laplacian smoothing, equipotential smoothing and after smoothing with *DARSS* has been listed in Table 1, which gives the worst values of quality metrics in the mesh along with the mean and standard deviation ( $\bar{X} \pm \sigma$ ).

The *DARSS* method has improved the Jacobian significantly. In several instances, the minimum Jacobian changed from negative to positive. The maximum aspect ratio and maximum warpage of the elements either improved significantly or remained intact for most of the elements while there was a minimal change in the maximum skew. It was also interesting to note that the improvement in the mesh quality was more tangible in the first phase where the

element Jacobian was significantly changed. The second phase of the smoothing was more effective in the regions where the first phase could not move the node to new position. Similar results were observed when the *DARSS* smoothing was applied to meshes of components of other body parts.

Distributions of the Jacobian, aspect ratio, Warpage and Skew for pelvic flesh case as well as tibia case are shown in Figs. 8, 9, respectively. Quality of mesh before smoothing and after smoothing with various smoothing schemes is compared with that after the *DARSS* smoothing. Figure 8 indicates that before smoothing about 1.5% elements had Jacobian equal to or below 0.6. After Laplacian smoothing, percentage of elements with Jacobian below 0.6 increased to 2% and some elements with Jacobian lower than that in the initial mesh were also observed. The minimum Jacobian in the mesh also dropped to  $-8.9$  from an initial minimum value of 0.21. On the other hand, as a result of *DARSS* smoothing, the minimum Jacobian increased to 0.41 (from 0.21). Overall increase in the number of elements with a higher Jacobian value can also be observed. Although improvement in the aspect ratios and skew of the elements is not as significant (Fig. 8), the proposed method performs better than Laplacian smoothing. From Fig. 8, it

can also be observed that *DARSS* performs better than other methods for improving the mesh in the present cases.

Significant improvement is also observed in the warpage of elements. The initial mesh had elements with warpage as high as 86.07. The maximum warpage in the improved mesh is 67. The histogram shows a significant increase in the number of elements with warpage smaller than 5 (from 17 to 23.5%) with the reduction in number of elements having warpage more than 20 (from 7 to ~0%).

Similar result can be observed in case of the bone mesh (Fig. 9), where *DARSS* can be seen to be the most effective smoothing scheme.

As it can be observed from Figs. 8, 9 and Table 1, in both the cases after smoothing by Smart Laplacian, the worst and the average values of mesh quality parameters remain almost same as that in the initial mesh. On the other hand, the results from the equipotential smoothing were almost as good as that of the *DARSS*.

Eigenvalues for all the cases were positive and change in eigenvalues of respective meshes obtained after different smoothing process was small. Minimum eigenvalue in the initial meshes were found be 0.12 (pelvic flesh) and 0.0299 (bone). The eigenvalues reduced most for the mesh improved with equipotential method (0.11 for pelvic flesh and 0.027 for bone), while for the mesh smoothed with *DARSS* they were (0.117 for pelvic flesh and 0.0296 for bone). Eigenvalues observed for mesh obtained from Laplacian smoothing (0.13 for pelvic flesh and 0.09 for bone) were on the higher side, while those after the smart Laplacian almost remained the same.

## 5 Conclusions

A simple and efficient smoothing technique has been presented. The method was seen to work well with large meshes and can tackle structured as well as unstructured hexahedral meshes. For the cases evaluated, the proposed method performs better than the existing popular alternatives such as Laplacian and smart Laplacian. The strength of the method is in its ability to significantly improve the Jacobian of the elements, in many cases from negative to a significant positive value. Improvement in the mesh quality is primarily in the first explicit phase, where the element Jacobians are significantly changed. Since the method can handle unstructured meshes, it is also suitable for complex meshes as well.

## References

- White DR, Mingu L, Benzley SE, Sjaardema GD (1995) Automated hexahedral mesh generation by virtual decomposition. In: Proceedings of 4th international meshing roundtable. Sandia National Laboratories, California, pp 165–176
- Knupp PM (1998) Next generation Sweep Tool: a method for generating all hex meshes on a two and one half dimensional geometries. In: Proceedings of 7th international meshing roundtable. Sandia National Laboratory, California, pp 505–513
- Folwell NT, Mitchell SA (1999) Reliable whisker weaving via curve contraction. *Eng Comput* 15:292–302
- Jankovich SR, Benzley SE, Shepherd JF, Mitchell SA (1999) The Graft Tool: an all-hexahedral transition algorithm for creating a multi-directional swept volume mesh. In: 8th international meshing roundtable. S. Lake Tahoe, CA, pp 387–392
- Owen SJ, Saigal S (2000) H-Morph: an indirect approach to advancing from hex meshing. *Int J Numer Methods Eng* 49:289–312
- White D, Rodrigue G (1997) Improved vector FEM solutions of Maxwell's equations using grid pre-conditioning. *Int J Numer Meth Eng* 40:3815–3837
- Shontz SM, Vavasis SA (2004) A linear weighted laplacian smoothing framework for warping tetrahedral meshes. *CS.NA/0410045*. October 2004
- Persson P-O, Strang G (2004) A simple mesh generator in MATLAB. *SIAM Rev* 46(2):329–345
- Canann SA, Stephenson MB, Blacker T (1993) Optsmooth: an optimization-driven approach to mesh smoothing. *Finite Elem Anal Des* 13:185–190
- Cannan SA, Tristano JR, Staten ML (1998) An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes 7th international meshing roundtable, October 26–28, 1998, Dearborn, Michigan, USA
- Hansbo P (1995) Generalized Laplacian smoothing of unstructured grids. *Commun Numer Methods Eng* 11:455–464
- Amenta N, Bern M, Eppstein D (1997) Optimal point placement for mesh smoothing. In 8th ACM–SIAM symposium on discrete algorithms. New Orleans, pp 528–537
- Freitag L (1997) On combining Laplacian and optimization-based mesh smoothing techniques. *Trends in unstructured mesh generation*, July 1997, ASME, pp 37–43
- Xu H, Newman TS (2006) An angle-based optimization approach for 2D finite element mesh smoothing. *Finite Elem Anal Des* 42:1150–1164
- Khattri S (2006) A new smoothing algorithm for quadrilateral and hexahedral meshes, LNCS, ICCS 2006, pp 239–246
- Lee CK, Lo SH (1994) A new scheme for the generation of a graded quadrilateral mesh. *Comput Struct* 52(5):847–857
- Herrmann LR (1976) Laplacian–isoparametric grid generation scheme. *J Eng Mech Div* 102(5):749–756
- Knupp PM (1999) Winslow smoothing on two-dimensional unstructured meshes. *Eng Comput* 15:263–268
- Vartziotis D, Wipper J (2009) The geometric element transformation method for mixed mesh smoothing. *Eng Comput* 25:287–301
- Semonova I, Kozhekin N, Savchenko V, Hagiwara I (2005) A general framework for analysis and comparison of surface mesh optimization techniques. *Eng Comput* 21:91–100
- Knupp PM (2001) Hexahedral and tetrahedral mesh untangling. *Eng Comput* 17:261–268
- Parthasarathy V, Kodiyalam S (1991) A constrained optimization approach to finite element mesh smoothing. *Finite Elem Anal Des* 9:309–320
- Calvo NA, Ideslohn SR (2001) All hexahedral mesh smoothing with a node based measure of mesh quality. *Int J Numer Methods Eng* 50:1957–1967
- Knupp PM (2003) A method for hexahedral mesh shape optimization. *Int J Numer Methods Eng* 58:319–332

25. Shimada K, Gossard DC (1995) Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. *Solid modelling '95* (conference proceedings), pp 409–419
26. Shimada K, Yamada A, Itoh T (2000) Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles. *Int J Comput Geom Appl* 10(4):400–424
27. Zhou T, Shimada K (2000) An angle based approach to two dimensional mesh smoothing. In: *Proceedings of the 9th international meshing roundtable*, pp 373–384
28. Branets L, Carey GF (2005) A local cell quality metric and variational grid smoothing algorithm. *Eng Comput* 21:19–28
29. Kober C, Muller-Hannemann M (2001) A case study in hexahedral mesh generation: simulation of the human mandible. *Eng Comput* 17:249–260
30. Vartziotis D, Athanasiadis T, Goudas I, Wipper J (2008) Mesh smoothing using the geometric element transformation method. *Comput Methods Appl Mech Eng* 197(45–48):3760–3767
31. Vartziotis D, Wipper J (2009) The geometric element transformation method for mixed mesh smoothing. *Eng Comput* 25(3):287–301
32. Vartziotis D, Wipper J (2009) Classification of symmetry generating polygon transformations and geometric prime algorithms. *Math Pannon* 20(2):167–187
33. Vartziotis D, Wipper J (2010) Characteristic parameter sets and limits of circulant Hermitian polygon transformations. *Linear Algebra Appl* 433(5):945–955
34. Vartziotis D, Wipper J, Schwald B (2009) The geometric element transformation method for tetrahedral mesh smoothing. *Comput Methods Appl Mech Eng* 199(1–4):169–182
35. Vartziotis D, Wipper J (2011) A dual element based geometric element transformation method for all-hexahedral mesh smoothing. *Comput Methods Appl Mech Eng* 200:1186–1203
36. Savchenko M, Egorova O, Hagiwara I, Savchenko V (2005) Hexahedral mesh improvement algorithm. *JSME Int J Ser C* 48(2):130–136
37. Savchenko V, Savchenko M, Egorova O, Hagiwara I (2008) Mesh quality improvement radial basis functions approach. *Int J Comput Math* 85(10):1589–1607
38. Taubin G (1995) A signal processing approach to fair surface design. *Proc SIGGRAPH'95* 29:351–358
39. Riccius J, Schweizerhof K, Baumann M (1997) Combination of adaptivity and mesh smoothing for the finite element analysis of shells with intersections. *Int J Numer Meth Eng* 40:2459–2474
40. Bank RE, Smith RK (1997) 1997. Mesh smoothing using a posteriori error estimates *SIAM J Numer Anal*(34):3–979
41. Jacquotte OP, Coussement G (1992) Structured mesh adaption: space accuracy and interpolation methods. *Comput Methods Appl Mech Eng* 101:397–432
42. Yin J, Teodosiu C (2008) Constrained mesh optimization on boundary. *Eng Comput* 24:231–240
43. Lagarias JC, Reeds JA, Wright MH, Wright PE (1998) Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM J Optim* 9(1):112–147
44. Mathews J, Fink KK (2004) *Numerical methods using Matlab*, 4th edn. Prentice Hall Inc., New Jersey, pp 430–433
45. Knupp PM (2000) Hexahedral mesh untangling and algebraic mesh quality metrics. In: *Proceedings of 9th international meshing roundtable*, pp 173–183
46. Deng B (2007) Human model for real-world vehicle–pedestrian impact simulations. Human modeling and simulation in automotive safety. Technical Report, ENSAM de Paris, Paris
47. Untaroiu CD (2005) Development and validation of a finite element model of human lower limb. PhD thesis, UVA