

# Multi-Domain Finite Element Meshing for Parotid Acinar Cell Modeling and Simulation

John Rugis<sup>a,b</sup>      Nathan Pages<sup>b</sup>      David Yule<sup>c</sup>      James Sneyd<sup>b</sup>

<sup>a</sup>corresponding author - email: j.rugis@auckland.ac.nz, phone: 649-923-2313

<sup>b</sup> Department of Mathematics, University of Auckland, Auckland, New Zealand

<sup>c</sup>School of Medicine and Dentistry, University of Rochester, Rochester, NY, USA

September 6, 2017

## Keywords

multi-domain mesh, mesh refinement, surface curvature, finite element modeling, parotid acinar cells

## Abstract

350 words maximum.

Contribution: Multi-domain mesh surface smoothing using iterative coupled refinement.

Termination of refinement guided by reference target surface curvature.

COMPLETE THIS WHEN EVERYTHING ELSE IS DONE.

## 1 Introduction

The primary role of salivary gland acinar cells is to secrete saliva, the lack of which causes a host of severe medical difficulties [FvdVS<sup>+</sup>85, Mel91]. Thus, an understanding of the mechanisms underlying saliva secretion are vital for the understanding of oral health. The basic mechanism of saliva secretion is well understood [Nau92], and has been previously modeled in detail. In brief, agonist stimulation (by hormones or neurotransmitters) leads to the production of the second messenger, inositol trisphosphate (IP3), which then releases  $\text{Ca}^{2+}$  from the endoplasmic reticulum (ER) through IP3 receptors (IPR) which are also  $\text{Ca}^{2+}$  channels.  $\text{Ca}^{2+}$  released from the ER is then pumped back into the ER by  $\text{Ca}^{2+}$  ATPase pumps (SERCA) or removed from the cell by plasma membrane ATPases.

Feedback between  $\text{Ca}^{2+}$  and IPR, as well as between  $\text{Ca}^{2+}$  and IP3 production and degradation, leads to repetitive cycles of  $\text{Ca}^{2+}$  release and reuptake, and thus to periodic  $\text{Ca}^{2+}$  spikes. Stochastic properties of the IPR are also closely involved in determining the interspike interval. These  $\text{Ca}^{2+}$  spikes activate  $\text{Cl}^-$  channels (in the apical region of the acinar cell) and  $\text{K}^+$  channels (in the basal region of the cell), leading to flow of  $\text{Cl}^-$  and  $\text{K}^+$  out of the cell, but in different directions. These ionic flows, in turn, set up ionic gradients that drive the flow of water through the cell, via osmosis.

A critical feature of salivary gland acinar cells is their spatial organisation [SMZ<sup>+</sup>17]. The apical and basal regions of the cell have different sets of ion channels, and open into different environments. These spatial heterogeneities are the reason that salivary acinar cells can transport water (i.e., produce saliva) in a specified direction. Although much is understood about saliva secretion, important questions, both experimental and theoretical, remain. From the theoretical point of view, one of the most interesting questions is how the structure of the acinar cells affects the properties of the  $\text{Ca}^{2+}$  spikes. IPR are not distributed uniformly through the cell, but are concentrated in the apical region. Furthermore, IP3 is produced only at the basal membrane. There is thus a significant spatial separation between the various components of the feedback system that governs oscillatory  $\text{Ca}^{2+}$  spiking. How these spatial heterogeneities might affect  $\text{Ca}^{2+}$  dynamics is one of the most important current theoretical questions in the study of how  $\text{Ca}^{2+}$  spiking controls saliva secretion.

We have previously studied the effects of spatial heterogeneity in a single parotid acinar cell, with a structure that is determined from experimental measurements [STAB<sup>+</sup>03]. However, acinar cells do not sit in isolation. Each cell is situated in group, called an acinus, that resembles a bunch of grapes. Inside each acinus, each parotid cell is coupled to neighbouring acinar cells by gap junctions, and all secrete water into a common lumen that has a highly branched structure. This means that the properties of the Ca<sup>2+</sup> spikes cannot be properly determined by the consideration of an isolated cell. Furthermore, the rate at which an acinus secretes water (into a common lumen) may be influenced by the structure of the lumen, which determines the network structure of the acinus, as well as which cells lie ‘downstream’ of other cells.

For a proper understanding of saliva secretion, and how it is affected by spatial heterogeneities, all these complications must be taken into account. Thus, here we extend our initial single-cell model to a multicellular model of an acinus of seven cells (the smallest acinus for which we have the detailed anatomical structure). We focus here on the technical details of mesh construction, leaving details of the scientific results for a subsequent publication.

## 2 Design Considerations

We have been using a straightforward implementation of the Finite Element Method (FEM) [Goc06, Hug, Gos05] for our three-dimensional parotid cell simulations and continue doing so in this latest work. In keeping with relatively common FEM practice, we model the surface of each cell with a triangle mesh, then fill-in the cell interiors with tetrahedrons which results in a complete volumetric mesh for each cell. With the meshes that we developed for our prior work, the cells did not physically contact each other i.e. there was a physical gap between the cells.

We have recently extended our mathematical modeling to include dynamic interactions between adjacent cells and this imposed additional demands on our mesh construction. With our new design, each cell now touches one or more other cells, sharing some portion of its surface. We have chosen to model adjacent cell surfaces with *conformal faces* in which numerous individual triangle mesh faces are shared in common between adjacent cells. Note that this facilitates a fairly straightforward mathematical implementation of the interactions between cells. The final complete mesh is said to be *multi-domain* (one domain for each cell) in that we still subsequently keep track of what’s going on within the physical confines of each cell independently.

As with our prior work, the source data for our acinar cell mesh construction consists of a calibrated confocal microscopy image stack. We have 31 images at 1024 by 1024 pixels each with a resolution of approximately 0.069 micro-meters per pixel and a stack spacing of 0.80 micro-meters. The lower resolution of the stack spacing means that there may be small scale cell surface detail captured in the image planes that has been missed along the stack acquisition axis, and because of this, the source data can be said to be ‘incomplete’. As will be described later in this paper, we chose a modeling resolution of approximately 0.2 micro-meters i.e. approximately three times lower than the pixel resolution but four times higher than the stack resolution.

In fact we have two image stacks: one highlighting the basal membrane of each cell and another highlighting the apical membrane. For the purposes of mesh construction, we chose the basal membrane stack because it more clearly shows the complete outline of each cell. Refer to our previous publication [SMZ<sup>+</sup>17] for details of the experimental methods used in the image stack creation process, both basal and apical.

Note that, to maximize the transparency and reproducibility of our computational work, we favor the use of open-source software tools. Our work-flow combines existing tools (building on the work of others whenever possible) along with software tools and utilities that we developed ourselves when existing tools did not suffice. Our source code files and data can be found on GitHub<sup>1</sup>.

## 3 Description of Method

We’ve designed an FEM mesh creation work-flow for our purposes that can overall be described as having three sequential steps:

1. Image segmentation

---

<sup>1</sup>[https://github.com/jrugis/cell\\_mesh.git](https://github.com/jrugis/cell_mesh.git)

2. Surface triangulation and refinement
3. Volumetric meshing

### 3.1 Image segmentation

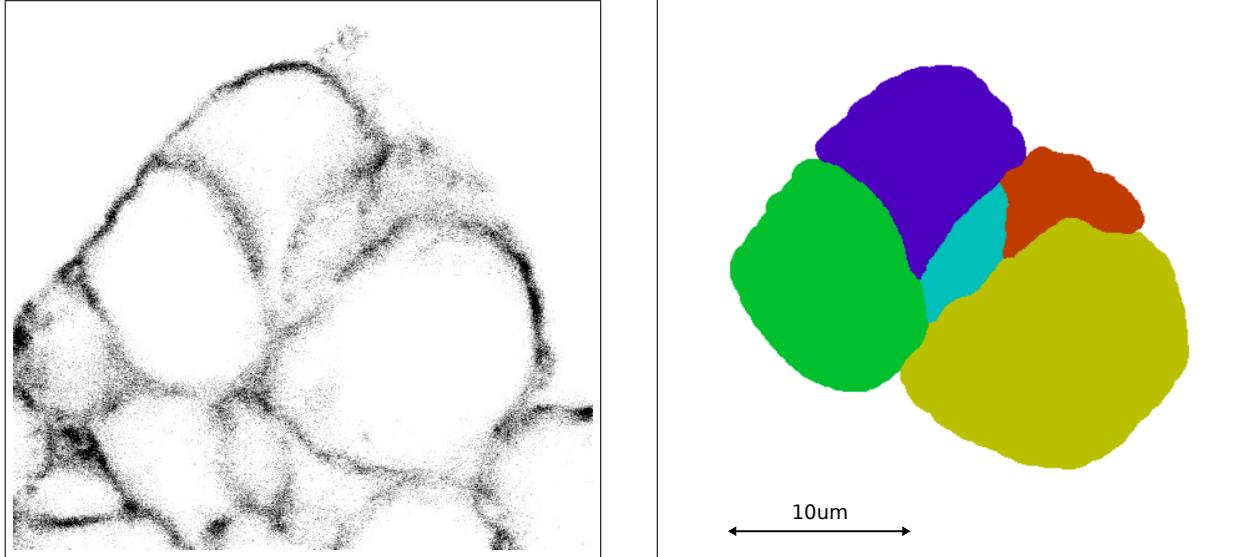


Figure 1: Subsection of a microscopy image slice on the left and its segmentation on the right.

By visual inspection we identified and selected a contiguous clump of seven cells within our source data image stack. This clump spanned thirty of the images and covered a maximum of approximately one-quarter of each image. Segmentation was done manually by tracing the outline of each cell in each image with a distinct color, followed by matched color flood-fill. One-quarter of original image number 16, along with it's segmentation, is shown in Figure 1.

Note that with our image stack the ratio of stack spacing to pixel resolution is 11.6 (being  $0.069/0.80$ ). To bring this closer to a ratio of one-to-one, we reduced the X and Y dimensions in the segmented image stack by a factor of four using nearest neighbor interpolation (to retain the distinct coloring of each cell) resulting in a pixel spacing of approximately 0.28 micro-meters. The reduced images were then combined into a single XYZ TIFF stack for convenience.

#### 3.1.1 Curvature

We also wanted to extract some information from the segmented image stack about the smoothness of the surface of each cell. We chose line curvature as a surface smoothness characteristic indicator. Both line and surface curvature have been used successfully by one of us (Rugis) in other work to characterize the surface of objects [Rug05, RK06b, RK06a]. (A description of how we used characteristic line curvature information as a reference for guidance in a cell surface curvature smoothing process in this current work will be given in following sections.)

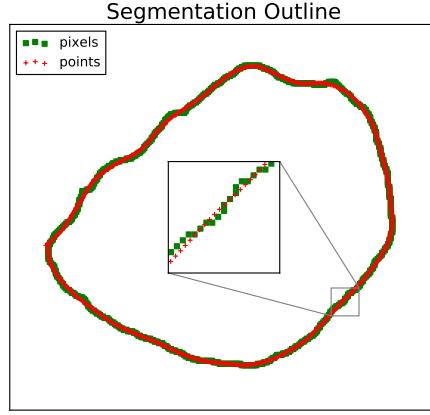


Figure 2: A sample segmentation pixel outline in green with smoothed node points in red.

For curvature calculation purposes, we went back to the 1024 by 1024 segmented images and extracted pixels associated with the closed curve boundary outline for each cell in each image, then selected the outline containing the maximum number of pixels for each cell as being the closest to a “great arc” slice through that cell<sup>2</sup>.

Next, considering that fact that image pixels are all located on a regular rectangular grid (not very useful for calculating actual local curvature!), a smoothed version of each cell outline was created using a 2D Savitzky-Golay (least-squares) fitting filter [SG64]. See Figure 2 for a sample cell outline showing the original pixel locations as well as the smoothing process result.

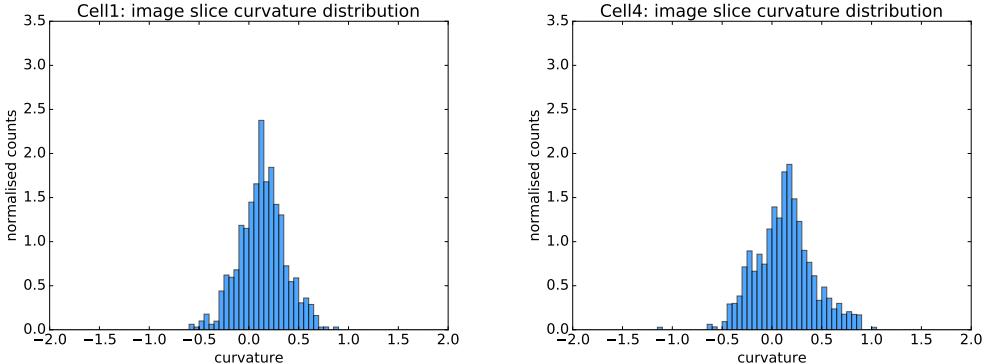


Figure 3: Reference curvature histograms for two of the cells (in  $\mu\text{m}^{-1}$ ). The distributions are “near normal”.

Planar line curvature was calculated at each point on the smoothed outlines using a local estimator as given in [Rug08]. We used weighted histograms to visualise the curvature distribution for each of the seven cells. Two of the histograms are shown in Figure 3. Note that in both cases the curvature distribution is biasing towards the positive, as would be expected with any closed curve from the outside using the convention that positive curvature is associated with convex line segments.

cell	curvature std ( $\mu\text{m}^{-1}$ )
1	0.2305
2	0.3567
3	0.3864
4	0.2832
5	0.6868
6	0.3627
7	0.4226

Table 1: The standard deviation of curvature values for seven cells extracted from the microscopy image stack.

<sup>2</sup>This great arc criterion was based on the fact that only the curvature associated with a great arc of a sphere is equal to the sphere surface mean curvature.

To simplify our characterisation of cell surface shape we chose the weighted standard deviation of the curvatures for each cell as a single (although imperfect<sup>3</sup>) measure that characterizes the surface smoothness and, in that sense, the shape, of each cell. The weighted curvature standard deviation for each of the seven cells is shown in Table 1.

### 3.2 Surface triangulation and refinement

For the surface triangulation and refinement process we started with the reduced XYZ TIFF image stack described in the previous section. We now thought of this reduced stack as essentially a 256x256x31 solid voxel block within which each voxel is labeled with a color associated with the cell that it belongs to.

#### 3.2.1 Surface triangulation

Significant prior work in extracting triangle surface meshes from labelled voxel blocks has been done by Boltcheva et al.[BYB09] and we used their technique. Sample computer code implementing this technique was found on the Computational Geometry Algorithms Library (CGAL) web site<sup>4</sup>. Note that we needed to convert our voxel block data to Inria (.inr) format<sup>5</sup> before passing it to the CGAL code.

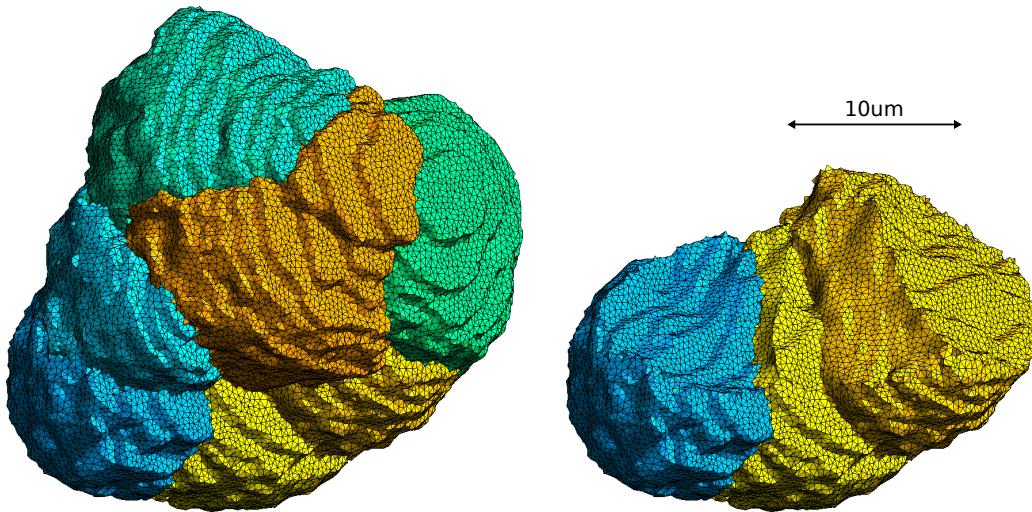


Figure 4: Rough multi-domain surface mesh: all seven cells (left) and three exposed cells (right).

The output from the CGAL code was a multi-domain triangle surface mesh, as shown in Figure 4, where each of the cell faces are labeled in color by cell . Unfortunately however, given our data and subsequent intentions, there are two problems with this mesh: there are “steps” as a result of the relatively limited stack resolution and, on close inspection, the surfaces are rather rough (as can be seen more clearly in the cell labeled “no smoothing” in the top of Figure 5). This lack of surface smoothness has the net effect of increasing the surface area of each cell to a value larger than what it most likely is in the real cells.

#### 3.2.2 Surface smoothing

To reduce the surface area closer to what it should be, we created a constrained surface smoothing process which is the primary contribution of this work. Note that the smoothing process should: 1) minimise the surface area of each cell while 2) maintain the volume of each cell and 3) keep the shared conformal faces between cells as being shared.

A pseudo-code outline for our iterative smoothing process is as follows:

```
Main Loop
{
    Cell Loop (for each cell)
    {

```

<sup>3</sup>Standard deviation is only an unambiguous characterisation given normally distributed data.

<sup>4</sup>[www.cgal.org](http://www.cgal.org)

<sup>5</sup>[www.inria.fr](http://www.inria.fr)

```

Smooth the cell.
Restore volume of cell.
Calculate the cell surface curvature.
}
Have all cells reached the target curvature?
Yes: DONE, exit loop.
No: Go back to top of Main Loop.
}

```

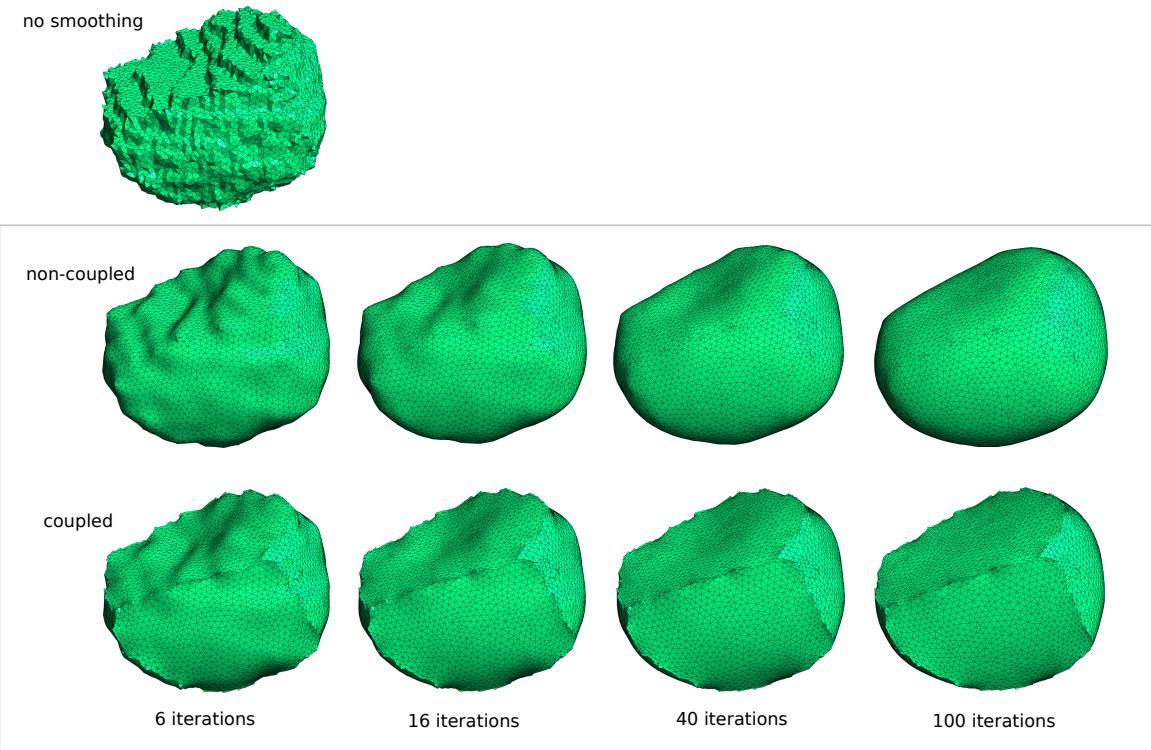


Figure 5: Cell smoothing evolution, with and without inter-cell coupling.

For each cell we used the curvature-flow based smoothing operation described in [DMSB99]. Note that the cell smoothing process works by slightly moving the position of the surface mesh vertices<sup>6</sup> for each cell in turn. Therefore, vertices associated with shared faces get moved twice in each pass through the Main Loop, i.e. once for each cell in every adjacent pair of cells. In this sense, the smoothing is a coupled process. With our approach, if the vertices associated with shared faces were allowed to split from each other, the result would be non-coupled as shown in the middle row of Figure 5, which is not what we want.

Smoothing results after each of each of 6, 16, 40 and 100 iterations are rendered as images in Figure 5. The question remains: After how many iterations should we stop? This is where we found the weighted curvature standard deviation values from Table 1 useful as will be described in the next section.

---

<sup>6</sup>Also, no new faces are added, nor are any faces removed.

### 3.2.3 Iteration termination

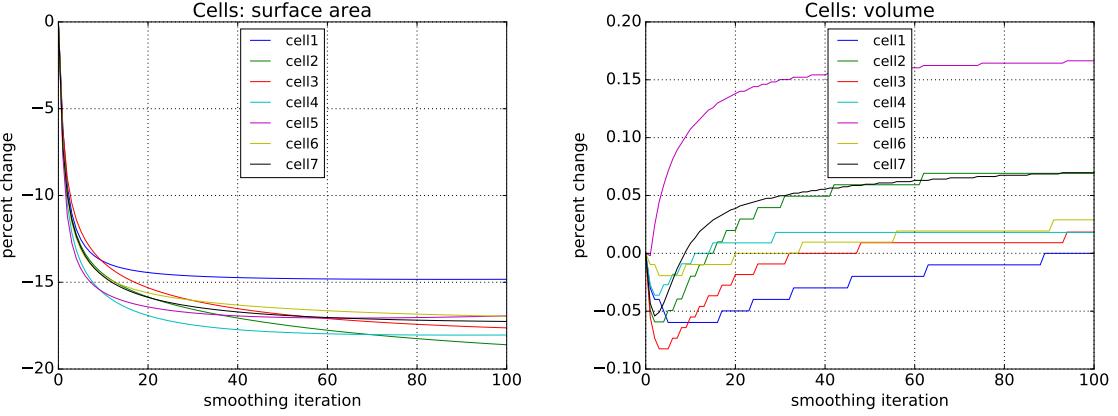


Figure 6: Cell surface area and volume evolution over one hundred coupled smoothing iterations. (Note the much smaller scale range with the plot on the right.)

Recall from the previous section that reducing cell surface area while at the same time maintaining cell volume was an important consideration. The smoothing step in our process has the effect of reducing surface area as shown for one hundred iterations in the left-hand side of Figure 6. Cell volume is explicitly restored in our process, so doesn't vary much, as can be seen in the right-hand side of Figure 6. (The raw data for these plots is given in Appendix Tables 3 and 4.) These plots indicate that we've met the surface area and volume consideration after ten or so iterations, but offer no further guidance as to when to terminate the iterative process. For this we looked to surface curvature.

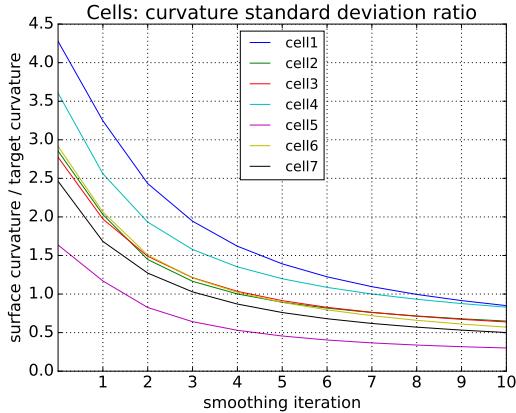


Figure 7: All of the cells hit the target curvature standard deviation ratio of one after nine iterations.

For each cell, in every iteration, we calculated the 2D surface curvature at every mesh vertex using the (local) surface curvature estimator given in [Rug08]. And for each cell, in every iteration, we calculated the weighted standard deviation of those curvatures to get a single smoothness characteristic value. We then compared this value to the associated target standard deviation in Table 1 and expressed it as a ratio as shown in Figure 7. (The raw data for this plot is given in Appendix Table 2.)

With this information in hand, we decided to terminate the iterative process and accept the results after ten iterations by which time all of the cells had more than dropped below their target characteristic curvature.

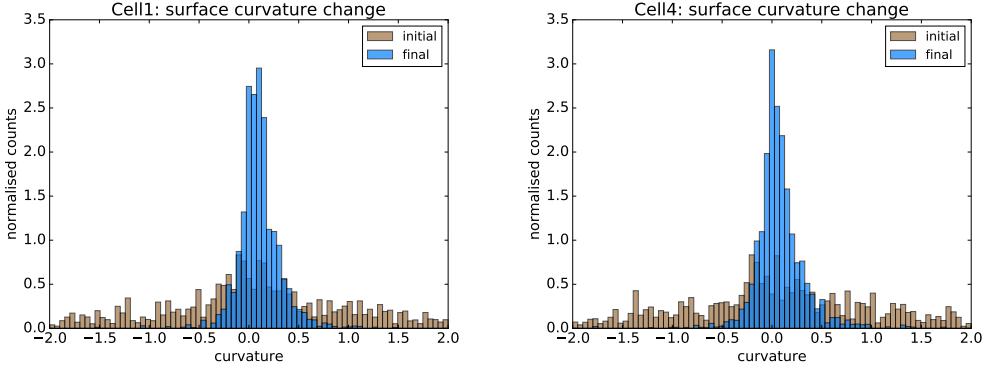


Figure 8: Initial surface curvature distribution in brown and smoothed surface curvature in blue.

For further insight and confirmation of our results we produced initial and final iteration weighted surface curvature histograms for each of the seven cells. Histograms for two of the cells are shown in Figure 8. Note that in all cases, as expected, the initial curvature spread was relatively wide and the final curvature distribution narrower, peaking just to the right (positive) side of zero. The final curvature histograms compared favourably to those associated with the image stacks including those shown earlier in Figure 3.

### 3.3 Volumetric meshing

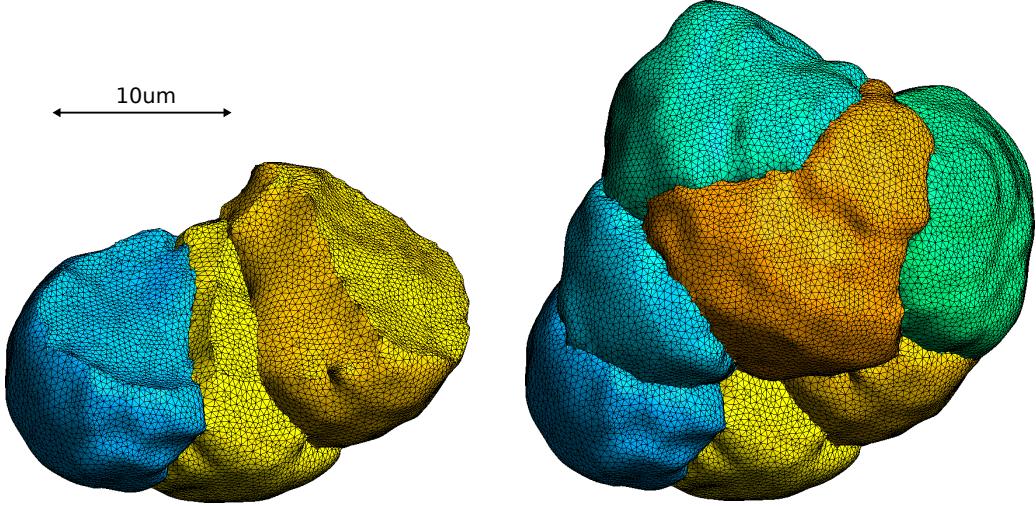


Figure 9: Fully meshed smoothed cells: three exposed cells (left) and all seven cells (right).

In the final step of our process, we filled-in the surface mesh for each cell with tetrahedrons using the Gmsh software tool [GR09]. This volumetric meshing employed a 3-D Delaunay refinement algorithm that reduces the instance of tetrahedral slivers<sup>7</sup> thereby improving the mesh quality from the point of view of the finite element method. Note that the volumetric meshing did not alter the associated surface meshes, insuring that the meshes remained conformal. Fully meshed cells are shown in Figure 9.

## 4 Results

We tested the meshes by running simulations in which adjacent cells interact with each other<sup>8</sup>. The meshes worked well with our solver, converging for each of the cells at every time step.

MORE FROM NATHAN GOES HERE?.

<sup>7</sup>To make the tetrahedrons more equilateral.

<sup>8</sup>Note that full simulation results with coupled cells will be reported in a subsequent publication.

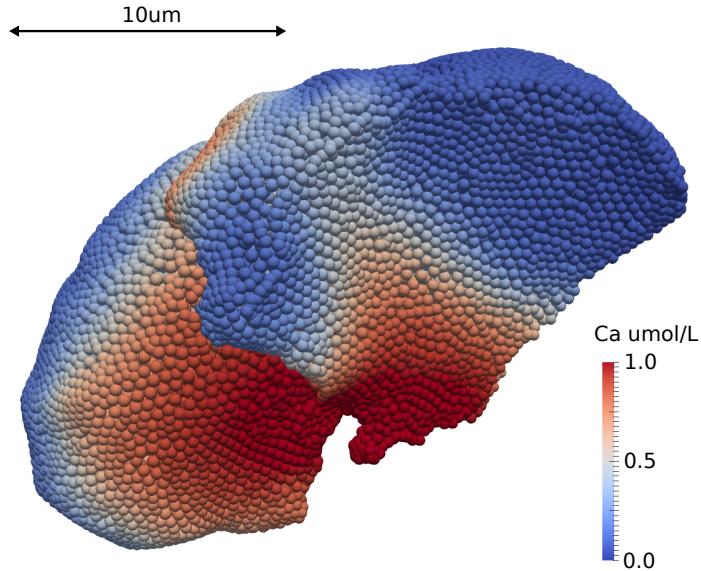


Figure 10: Simulation sanpshot of two interacting adjacent cells.

As an indication of the quality of our results, Figure 10 shows a calcium concentration spike propagating from the apical region (bottom centre of the image) to the basal region (top and left) in two adjacent cells.

## 5 Discussion

Overall the challenge in this latest work could be described as generating a physically accurate mesh model given incomplete data. As described in Section 2, the spacing between source images was relatively large compared to the source image pixel resolution.

The mesh design was guided by, and constrained by, both the practical requirements of finite element modeling and the simplification assumptions made in our mathematical model. We believe that our smoothing process produced meshes that are arguably organic in appearance.

There is a fine point not discussed earlier regarding how we calculated surface curvature in Section 3.2.3. We decided to leave out the curvatures at points along the seams around common faces. These seams are relatively sharp as can be seen in the bottom row of Figure 5. The sharp seams gave unrealistically high curvature values. In the real cells these seams are probably not so sharp as can be inferred from the left hand side image in Figure 1.

In our application, we believe that there is no objective collection of criteria with which to optimally decide precisely when to terminate the iterative smoothing process. However, we do believe using nominally ten iterations is well justified as described earlier in this paper.

## 6 Acknowledgements

This work was supported by the National Institutes of Health grant number 5R01DE019245 and by the Marsden Fund of the Royal Society of New Zealand. Computational resources and support were provided by the New Zealand eScience Infrastructure (NeSI).

## References

- [BYB09] Dobrina Boltcheva, Mariette Yvinec, and Jean-Daniel Boissonnat. Mesh generation from 3D multi-material images. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*, September 2009.

- [DMSB99] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 317–324, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [FvdVS<sup>+</sup>85] P C Fox, P F van der Ven, B C Sonies, J M Weiffenbach, and B J Baum. Xerostomia: evaluation of a symptom with increasing significance. *J Am Dent Assoc*, 110(4):519–25, Apr 1985.
- [Goc06] Mark S Gockenbach. *Understanding and implementing the finite element method*. Siam, 2006.
- [Gos05] Michael R Gosz. *Finite element method: applications in solids, structures, and heat transfer*. CRC Press, 2005.
- [GR09] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
- [Hug] Thomas J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications.
- [Mel91] J E Melvin. Saliva and dental diseases. *Curr Opin Dent*, 1(6):795–801, Dec 1991.
- [Nau92] B Nauntofte. Regulation of electrolyte and fluid secretion in salivary acinar cells. *Am J Physiol*, 263(6 Pt 1):G823–37, Dec 1992.
- [RK06a] J. Rugis and R. Klette. A scale invariant surface curvature estimator. In *LNCS: Advances in Image and Video Technology*, volume 4319/2006, pages 138–147, Berlin / Heidelberg, 2006. Springer.
- [RK06b] J. Rugis and R. Klette. Surface registration markers from range scan data. In *LNCS: Combinatorial Image Analysis*, volume 4040/2006, pages 430–444, Berlin / Heidelberg, 2006. Springer.
- [Rug05] J. Rugis. Surface curvature maps and Michelangelo's David. In *Image and Vision Computing New Zealand 2005*, pages 218–222, 2005.
- [Rug08] J. Rugis. *Digital Surface Curvature*. PhD thesis, University of Auckland, 2008.
- [SG64] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.
- [SMZ<sup>+</sup>17] James Sneyd, Shawn Means, Di Zhu, John Rugis, Jong Hak Won, and David I. Yule. Modeling calcium waves in an anatomically accurate three-dimensional parotid acinar cell. *Journal of Theoretical Biology*, 419:383 – 393, 2017.
- [STAB<sup>+</sup>03] J Sneyd, K Tsaneva-Atanasova, J I E Bruce, S V Straub, D R Giovannucci, and D I Yule. A model of calcium waves in pancreatic and parotid acinar cells. *Biophys J*, 85(3):1392–405, Sep 2003.

## A Numerical Data

cell	0	1	2	3	4	5	6	7	8	9	10
1	0.9837	0.7466	0.5595	0.4472	0.3726	0.3201	0.2814	0.2520	0.2289	0.2105	0.1955
2	1.0143	0.7216	0.5146	0.4141	0.3559	0.3177	0.2906	0.2702	0.2543	0.2414	0.2309
3	1.0620	0.7552	0.5703	0.4641	0.3963	0.3501	0.3169	0.2921	0.2727	0.2571	0.2443
4	1.0402	0.7387	0.5579	0.4552	0.3903	0.3459	0.3135	0.2888	0.2691	0.2529	0.2394
5	1.1262	0.8054	0.5683	0.4418	0.3649	0.3139	0.2782	0.2524	0.2331	0.2184	0.2070
6	1.0739	0.7598	0.5541	0.4453	0.3766	0.3286	0.2927	0.2649	0.2428	0.2249	0.2102
7	1.0451	0.7137	0.5396	0.4363	0.3692	0.3226	0.2885	0.2626	0.2422	0.2259	0.2126

Table 2: The evolution of standard deviation of mean surface curvature for the seven cells (in  $\mu\text{m}^{-1}$ ) after each of ten coupled smoothing iterations.

cell	0	1	2	3	4	5	6	7	8	9	10
1	628.64	590.17	569.09	559.23	553.53	549.84	547.26	545.37	543.93	542.80	541.90
2	729.06	680.85	657.39	646.44	639.76	635.12	631.65	628.91	626.66	624.77	623.15
3	681.01	638.45	619.64	609.78	603.46	598.93	595.46	592.68	590.37	588.40	586.70
4	712.88	661.36	637.82	626.19	619.08	614.15	610.47	607.57	605.02	603.22	601.52
5	443.18	408.76	393.58	386.72	382.76	380.16	378.30	376.90	375.81	374.92	374.19
6	684.72	636.81	615.75	605.50	599.26	595.00	591.87	589.46	587.55	585.98	584.67
7	627.42	582.15	564.89	555.74	549.94	545.88	542.82	540.42	538.47	536.84	535.46

Table 3: The evolution of surface area for the seven cells (in  $\mu\text{m}^2$ ) at each coupled smoothing iteration.

cell	0	1	2	3	4	5	6	7	8	9	10
1	1004.3	1004.0	1003.9	1003.9	1003.8	1003.7	1003.7	1003.7	1003.7	1003.7	1003.7
2	1012.4	1011.9	1011.8	1011.8	1011.8	1011.9	1011.9	1012.0	1012.0	1012.1	1012.2
3	1090.2	1089.6	1089.4	1089.3	1089.3	1089.3	1089.4	1089.4	1089.5	1089.5	1089.6
4	1105.7	1105.4	1105.3	1105.3	1105.4	1105.4	1105.5	1105.5	1105.6	1105.6	1105.6
5	492.76	492.75	492.88	492.98	493.05	493.11	493.16	493.20	493.23	493.26	493.29
6	1036.3	1036.2	1036.2	1036.1	1036.1	1036.1	1036.1	1036.1	1036.1	1036.2	1036.2
7	903.99	903.60	903.50	903.53	903.60	903.69	903.78	903.87	903.94	904.01	904.07

Table 4: The evolution of volume for the seven cells (in  $\mu\text{m}^3$ ) at each coupled smoothing iteration.

## B Software

Software tools used in this work:

Blender	<a href="http://www.blender.org/">www.blender.org/</a>
CGAL	<a href="http://www.cgal.org/">www.cgal.org/</a>
GitHub	<a href="http://github.com">github.com</a>
Gmsh	<a href="http://gmsh.info/">gmsh.info/</a>
ImageMagick	<a href="http://www.imagemagick.org">www.imagemagick.org</a>
MatLab	<a href="http://www.mathworks.com">www.mathworks.com</a>
ParaView	<a href="http://www.paraview.org">www.paraview.org</a>
Photoshop	<a href="http://www.adobe.com">www.adobe.com</a>
Python	<a href="http://www.python.org">www.python.org</a>