A continuous model for bone formation

David Stojanovski
Supervisor: Dr. Pascal Buenzli
Monash University

February 2014

1 Introduction

Bone tissue is replaced via the regulatory action of two families of cells, osteoblasts and osteoclasts. When bone tissue ages, it is eventually resorbed by what are called osteoclasts. Naturally the removal of bone tissue will result in a cavity being left behind, which must then be refilled with new soft tissue. This soft tissue eventually mineralises to form new bone. The cells responsible for production of this soft tissue are osteoblasts [4].

The process of resorbing old tissue and laying down new tissue is a continuous process. This results in bone that, on a small enough scale, has a microstructure that evolves continuously with time. The aim of this paper is to develop a model for this evolution in order to gain insight on how bone microstructure varies during this regulatory process, in particular in people with physiological bone disorders such as osteoporosis wherein cortical porosity is greater. In general, this problem is known as interface evolution, a very general field that spans across many sciences. The currently most efficient models for tracking the evolution of interfaces are recent, dating back to the seminal work by Sethian and Osher in 1988. The level set method is the computational basis of our model.

2 The equation for interface evolution

Suppose we have some curve $\phi(x, y; t) = 0$ that is a boundary, or *interface*, separating the xy-plane into two regions. Let us assume, for now, that we can write ϕ as

$$\phi(x, y; t) = y - h(x; t)$$

Such that $y = h(x; t)$ forms the interface. Assuming every point on the interface moves at a uniform velocity v normal to the interface, the partial differential equation for this motion is

$$\frac{\partial h}{\partial t} = v|\nabla h| = v\sqrt{1 + \left(\frac{\partial h}{\partial x}\right)^2} \tag{1}$$

We seek some solution $h(x, t)$ that describes the evolution of the interface's shape as time varies. The equation (1) is a Hamilton-Jacobi equation with an initial condition $h(x, 0) = h_0(x)$ and periodic boundary conditions. For now we shall assume that the velocity v is constant everywhere along the interface, and later we will develop more sophisticated models of this velocity. In particular, the velocity of each point on the interface will depend upon its local curvature.

2.1 A finite difference solution

We can make an early attempt at solving equation (1) using a basic finite difference method. If we take a forward difference for the time derivative of h and central difference for the space derivative of h, we obtain

$$\frac{h_j^{n+1} - h_j^n}{\Delta t} = v\sqrt{1 + \left(\frac{h_{j+1}^n - h_{j-1}^n}{\Delta x}\right)^2}$$

Hence we have an equation for the curve at time $t + \Delta t$ given by

$$h_j^{n+1} = v\Delta t\sqrt{1 + \left(\frac{h_{j+1}^n - h_{j-1}^n}{\Delta x}\right)^2} \tag{2}$$

A simple MATLAB script implements the finite difference method given in equation (2) on the preceding page with the initial condition $h(x,0) = \sin(x)$. The velocity $v = 1$ and the interface is tracked from $t = 0$ to $t = 10$ between $x = 0$ and $x = 4\pi$. We choose 2000 time steps and 500 space steps, as shown in Figure 1.

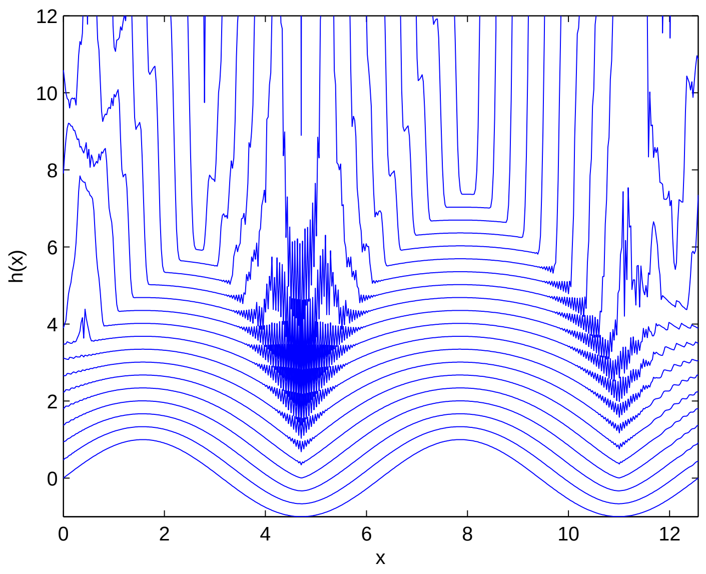


Figure 1: Using a basic finite difference method to track the interface

Clearly using a naive finite difference method is met with failure even with a smooth initial condition. The reason for this is that cusps form around the local mimina of $\sin(x)$ as the curve's shape evolves. This results in large computational errors of the gradient around these cusps, which eventually dominate the solution.

2.2 The method of artificial viscosity

One method of stabilising the solution of the Hamilton-Jacobi equation is to introduce a *viscosity* term to the differential equation, which 'smooths' the equation somewhat and reduces the formation of cusps.

$$\frac{\partial h}{\partial t} = v|\nabla h| + \epsilon \nabla^2 h = v\sqrt{1 + \left(\frac{\partial h}{\partial x}\right)^2} + \epsilon \frac{\partial^2 h}{\partial x^2} \tag{3}$$

The parameter ϵ in equation (3) on the previous page can be tuned to reduce the magnitude of the added viscosity term. Using a finite difference approximation of the second derivative, we have

$$h_j^{n+1} \approx v\Delta t \sqrt{1 + \left(\frac{h_{j+1}^n - h_{j-1}^n}{\Delta x}\right)^2} + \epsilon \Delta t \left[\frac{h_{j+1}^n - 2h_j^n + h_{j-1}^n}{(\Delta x)^2}\right] + h_j^n \tag{4}$$

Computationally, this is hardly more difficult than using a basic finite difference method. We tune the diffusion parameter ϵ to be proportional to Δx, so we never have more diffusion than is necessary for the solution to remain stable. The revised MATLAB script uses the finite difference method given by equation (4), choosing $\epsilon = \Delta x/5$ and otherwise identical parameters. The result is shown in Figure 2.

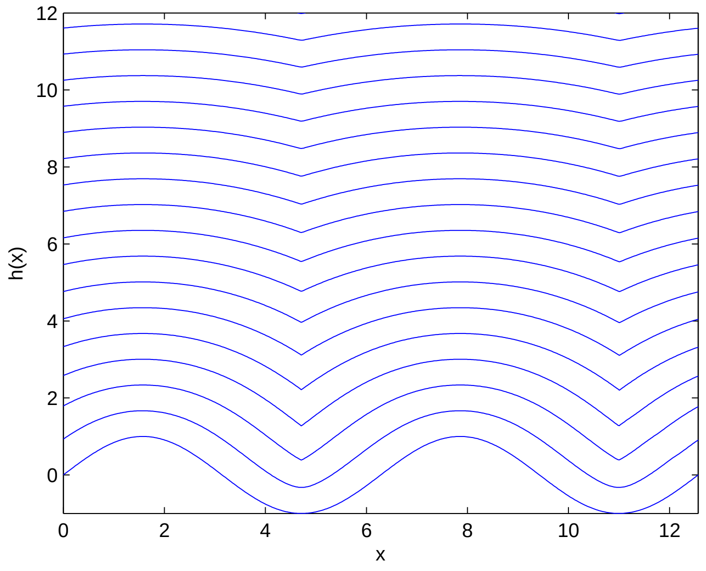


Figure 2: The method of artificial viscosity

The solution is now stable, albeit the solution is somewhat *smeared* by the presence of the viscosity term. Naturally we would have to reduce the size of the space step in order to keep the viscosity to a minimum. However, the amount we reduce the space step is limited by the Courant-Friedrichs-Lewy condition, and we are also limited by the additional computational effort required to have such a small time step.

2.3 Upwind schemes

A second, non-diffusive approach to stabilising the solution is to use an upwind scheme, which is generally used to solve hyperbolic partial differential equations [2]. We may use this to solve equation (1) on page 2. Using an upwind scheme, our discretisation becomes

$$h_j^{n+1} \approx v\Delta t\sqrt{1+(\delta_x h)^2} \tag{5}$$
$$\delta_x h = \begin{cases} \frac{h_j^n - h_{j-1}^n}{\Delta x} & h_j^n \leq h_{j-1}^n \\ \frac{h_{j+1}^n - h_j^n}{\Delta x} & h_j^n > h_{j-1}^n \end{cases}$$

Again, trying this in MATLAB, this time using the upwind scheme given in equation (5), we find the results in Figure 3. Notice now that we do not need the diffusive term in order to achieve the same stability. This also means we can keep the space step larger because we need not minimise it to keep the diffusion minimal.

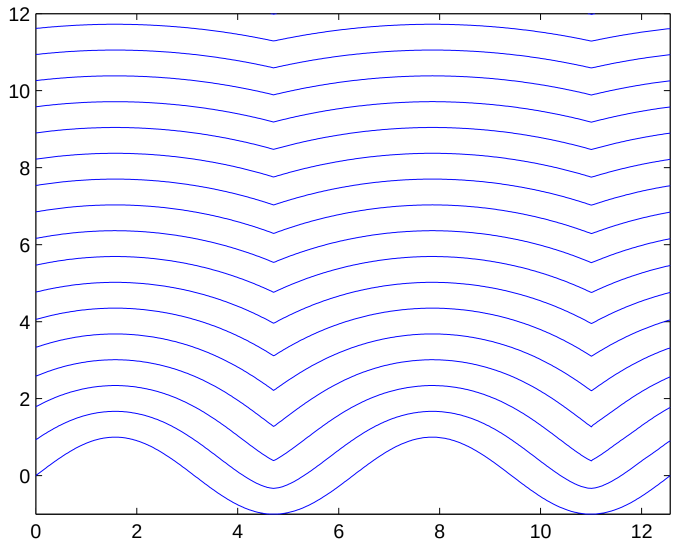


Figure 3: Upwind scheme

3 Cell density dependence in one dimension

In a basic multicellular unit, the rate at which bone is deposited and resorbed depends on the local density of cells on the interface of the BMU. The density depends upon the local curvature of each point along the interface [3].

3.1 Coupled model

We can write this dependence explicitly in the form of a coupled system of partial differential equations. In one dimension,

$$\begin{aligned}\frac{\partial h}{\partial t} &= k_{form}\rho_{OB}\sqrt{1+\left(\frac{\partial h}{\partial x}\right)^2} \\ \frac{\partial \rho_{OB}}{\partial t} &= k_{form}\kappa\rho_{OB}^2\end{aligned} \tag{6}$$

Here k_{form} is the rate at which bone is deposited or resorbed by one cell, ρ_{OB} is the density of cells (in this case, we consider the density of osteoblasts), and κ is the signed curvature at some point x along the (one-dimensional) interface, given by

$$\kappa = \frac{h_{xx}}{(1+h_x^2)^{3/2}}$$

3.2 Solving the coupled model

Given the complex nature of this coupled system of partial differential equations, Mathematica was used to discretise the system using the *method of lines*, given the initial conditions $h(x,0) = \sin(x)$ and $\rho_{OB}(x,0) = \cos^2(x)$. The results of this are shown in Figure 4.

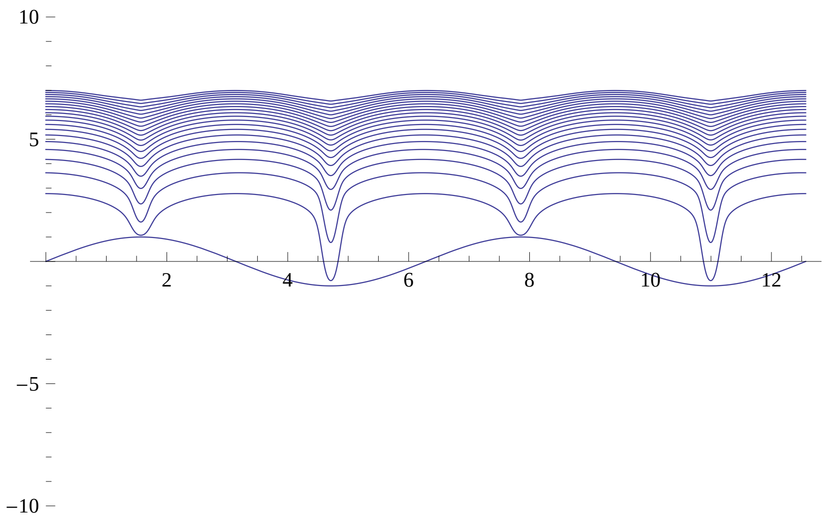


Figure 4: Coupled model in one dimension exhibiting cell density dependence

Using a coupled model, it becomes apparent that the interface reaches a steady state. The general shape of the steady state of the interface, in general, resembles that in the original model that had unbounded growth.

3.3 Caveats of a one-dimensional model

Consider the shape of a cortical BMU, given in Figure 5. There is a point of infinitely steep descent at the very front of the BMU which we cannot discretise in only one spatial dimension. If we generalise the shape of our bone interface to a *plane curve* rather than a one-dimensional function, we can better capture the essence of the shape of a cortical BMU.

4 Extending the model to two dimensions

The level set equation generalises to multiple dimension, giving us a new coupled system of partial differential equations that describes interface evolution in two dimensions.

$$\begin{aligned}\frac{\partial \phi}{\partial t} &= k_{form}\rho_{OB}\sqrt{\left(\frac{\partial \phi}{\partial x}\right)^2 + \left(\frac{\partial \phi}{\partial y}\right)^2} \\ \frac{\partial \rho_{OB}}{\partial t} &= k_{form}\kappa\rho_{OB}^2\end{aligned} \tag{7}$$

All parameters in equation (7) are the same as they were in the one-dimensional case, except curvature is now defined in two spatial dimensions (x and y) rather than only one. Attempting to track the evolution of the interface using a Lagrangian method would prove very difficult because it requires a parameterisation of ϕ. Such a parameterisation is non-trivial because the topology of the zero level set of ϕ can change with time. We can avoid the need for this by using the level set method of Sethian [1].

4.1 The level set method

As the level set method is rather complicated, an existing implementation of the level set method for MATLAB, written by Baris Sumengen [5] was used as it offers a complete implementation of the level set methods proposed by Osher [2]. The toolbox allows for the simulation of interfaces in two dimensions that involve motion in the normal direction, making it suitable for the practical realisation of the two-dimensional level set equation given by equation (7).

5 A level set model of bone tissue renewal

5.1 Modelling the normal motion of the BMU cavity

A cortical BMU with osteoid-depositing osteoblasts and bone-resorbing osteoclasts is shown in Figure 5. As the osteoblasts are depositing new soft tissue, this has the net effect of the cavity closing in a direction normal to the interface between cavity and bone. Normal motion is inwards when the velocity v is negative. Conversely, the action of the osteoclasts near the front of the BMU will cause the cavity to open outward normal to the interface as old bone tissue is resorbed, and so v is positive.

Within a certain band (of defined width) of the front of the interface, the cavity moves outward. To the left of this band, osteoblasts close up the cavity created by the osteoclasts. However, the cavity does not close up completely because the BMU is centred around a blood vessel. There is a certain region called the *Haversian canal*, with radius r, where no bone is deposited [4]. This is accounted for in this simple model.

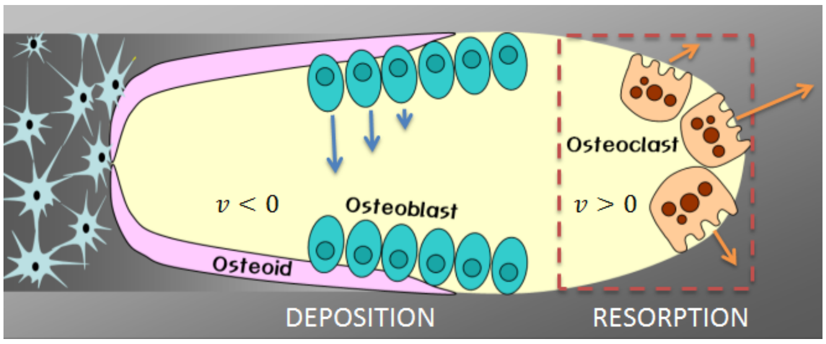
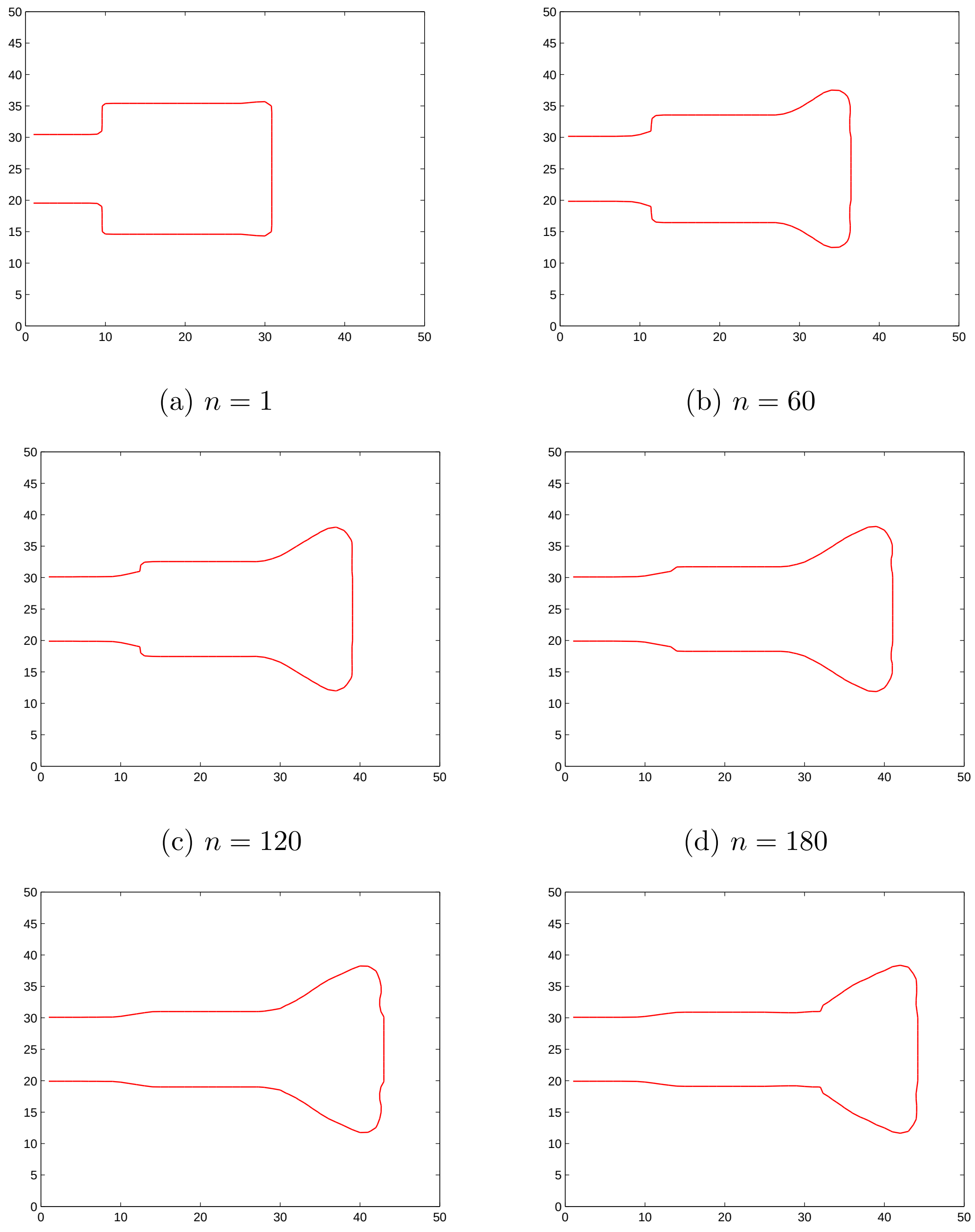


Figure 5: A simple 2D model for bone cavity evolution

This model was implemented using the MATLAB level set method toolbox, and the results are shown over the course of 300 time steps starting with a rectangular initial shape in Figure 6. We can see that the interface evolves to the right, as expected.



(a) $n = 1$ (b) $n = 60$

(c) $n = 120$ (d) $n = 180$

(e) $n = 240$ (f) $n = 300$

Figure 6: Level set evolution of the basic multicellular unit

5.2 Realising curvature dependence in multiple dimensions

As in the one dimensional model, the rate at which bone tissue is deposited or resorbed will depend on the density of cells, which itself depends on the local curvature at each point on the interface. In order to model this, we use a coupled system of level set equations,

$$\frac{\partial \phi}{\partial t} + V\mathbf{n} \cdot \nabla \phi = 0 \tag{8}$$

$$\frac{\partial V}{\partial t} + V\mathbf{n} \cdot \nabla V = -\kappa V^2 \tag{9}$$

We have introduced a second level set function V that models the normal velocity everywhere in space. This eliminates the need to have a parameterisation of ϕ in order for the velocity at each point along the zero level of ϕ to be defined. We are interested in the velocities at points coinciding with the zero level set of ϕ. Note that

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}$$

We can write the curvature κ in terms of the level set function ϕ.

$$\kappa = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \tag{10}$$

Substituting equation (10) into equation (9) gives

$$\frac{\partial V}{\partial t} + V\mathbf{n} \cdot \nabla V = V^2 \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \tag{11}$$

We solve the system given by equation (8) and equation (11) in order to introduce curvature dependence in our two-dimensional model.

6 Conclusion

In one dimension, the level set equation is not directly solvable using a basic finite difference, but can be solved by adding viscosity to the equation or by using an upwind scheme. Cell density dependence can be modelled in terms of the local curvature along each point of a bone's surface, so it is readily coupled with the level set equation. The level set equation extends simply to more than one dimension, but is more readily solved using the level set method of Sethian rather than using an ad-hoc discretisation of the bone interface. In multiple dimensions, cell density dependence is introduced through coupling with a second level set equation that represents the velocity at each point along the interface.

References

[1] J.A. Sethian. (1999) *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.* Vol 3. Cambridge University Press.

[2] R. Osher and S. Fedkiw (2003) *Level set methods and dynamic implicit surfaces.*

[3] C.M. Biden et al, (2012) *How linear tension converts to curvature: geometric control of bone tissue growth.* PloS one 7.5 e36336

[4] P.R. Buenzli et al, (2014) *Bone refilling in cortical basic multicellular units: insights into tetracycline double labelling from a computational model.* doi 10.1007/51023-013-0495-y Biomechanics and modeling in mechanobiology, 13(1), 185-203.

[5] B. Sumengen, (2005) *A Matlab toolbox implementing Level Set Methods* ⟨`http://barissumengen.com/level_set_methods/`⟩