

Numerical simulation of a bubble cluster

Tommaso Sciortino

October 2011

Abstract

We present a scheme for simulating the motion of a collection of 2-dimensional bubbles which touch at 1-dimensional walls. In section 1, we introduce a model where bubble walls are represented as Bezier curves. Using the Euler-Lagrange technique, we derive equations of motion for the control points of the Bezier curves in section 2. Finally, in section 3, we discuss how those equations can be solved numerically to produce an animation.

1 Representation

We represent a bubble cluster as a planar graph which partitions the plane into faces. Each face represents the interior of a bubble. Each edge represents a wall where two bubbles touch. Vertices occur in the graph where multiple bubble walls meet at a point.

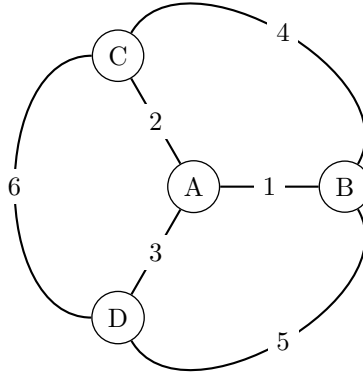


Figure 1: A planar cubic graph

Since more than one wall may connect the same two vertices the graph is a multigraph. Furthermore, since only vertices where three walls meet are physically stable we will constrain the graph to be 3-regular or “cubic”. Similarly, loops are physically unstable and thus disallowed.

1.1 Bubbles as faces

Each bubble face can be defined by the ring of serially connected edges which form its border. Consequently each edge belongs to two faces. In addition to the normal internal faces there is one external face which represents the air outside the bubble cluster.

If we view each edge as a pair of directed edges - one in each direction - then each face can be defined as a list of directed edges where the head of each is the tail of the next. Consequently, each directed edge belongs to exactly one face and (except for the external face) the directed edges of each face all run clockwise. This convention is useful for properly calculating various quantities when we need to integrate around a face in the right order and direction.

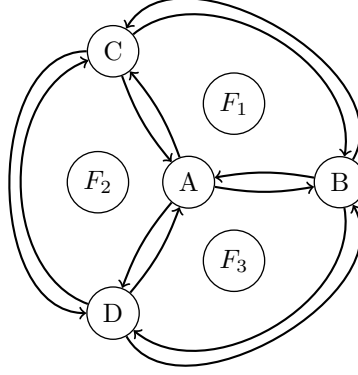


Figure 2: A directed graph representation

1.2 Bubble walls as cubic Bezier curves

The position of a bubble wall is represented by a cubic Bezier curve defined as a parametric equation with four control points $\vec{B}_n(t)$

$$\vec{B}(p, t) = (1 - p)^3 \vec{B}_0(t) + 3p(1 - p)^2 \vec{B}_1(t) + 3p^2(1 - p) \vec{B}_2(t) + p^3 \vec{B}_3(t) \quad (1)$$

The parameter p runs from 0 to 1. We also define

$$\vec{B}(p, t) = B_x(p, t)\hat{x} + B_y(p, t)\hat{y} \quad (2)$$

and

$$B_x(p, t) = (1 - p)^3 x_0(t) + 3p(1 - p)^2 x_1(t) + 3p^2(1 - p) x_2(t) + p^3 x_3(t) \quad (3a)$$

$$B_y(p, t) = (1 - p)^3 y_0(t) + 3p(1 - p)^2 y_1(t) + 3p^2(1 - p) y_2(t) + p^3 y_3(t) \quad (3b)$$

In terms of the directed edge representation, one directed edge traverses the curve as p goes from 0 to 1, the other from 1 to 0.

2 Physics

The position of a bubble cluster's walls at any time t is given by the positions of the control points. We denote these as $q_1(t), q_2(t), \dots, q_n(t)$. To find how the walls change with time we must numerically solve for these functions.

To derive equations of motion we use Lagrangian mechanics. We define the Lagrangian $\mathcal{L}_T = K_T - V_T$ as the total kinetic minus total potential energy of the system. The equations of motion are the path for which the integral of the Lagrangian over time is stationary. To find these we generate the following equations:

$$\frac{d}{dt} \frac{\partial \mathcal{L}_T}{\partial q'_i} = \frac{\partial \mathcal{L}_T}{\partial q_i} \quad \text{for each } q_i \quad (4)$$

Taken together these yield a nonlinear system of ordinary differential equations.

2.1 Kinetic Energy

To find the kinetic energy we first determine the velocity of every point on a bubble wall. To do this we take the derivative of the position with respect to time.

$$\vec{B}'(p, t) = (1-p)^3 \vec{B}'_0(t) + 3p(1-p)^2 \vec{B}'_1(t) + 3p^2(1-p) \vec{B}'_2(t) + p^3 \vec{B}'_3(t)$$

To have kinetic energy the wall must have mass so we introduce a coefficient m to represent mass per unit interval of the parameter p . This gives us the kinetic energy at the infinitesimal point p

$$K(p, t) = \frac{m}{2} \left| \vec{B}'(p, t) \right|^2$$

To find the total kinetic energy for the bubble wall we integrate over the length of the wall

$$\begin{aligned} K(t) &= \int_0^1 \frac{m}{2} \left| \vec{B}'(p, t) \right|^2 dp \\ &= \frac{m}{2} \int_0^1 \left(\sqrt{B'_x(p, t)^2 + B'_y(p, t)^2} \right)^2 dp \\ &= \frac{m}{2} \int_0^1 B'_x(p, t)^2 + B'_y(p, t)^2 dp \end{aligned}$$

In terms of the derivatives of the coordinates of the 4 control points $x_n(t)$ and $y_n(t)$ we get:

$$\begin{aligned} K(t) &= \frac{m}{2} \int_0^1 \left((1-p)^3 x'_0 + 3p(1-p)^2 x'_1 + 3p^2(1-p) x'_2 + p^3 x'_3 \right)^2 \\ &\quad + \left((1-p)^3 y'_0 + 3p(1-p)^2 y'_1 + 3p^2(1-p) y'_2 + p^3 y'_3 \right)^2 dp \end{aligned}$$

Integrating gives us

$$\begin{aligned}
K(t) = \frac{m}{2} & \left(\frac{x_0'^2}{7} + \frac{x_0'x_1'}{7} + \frac{2x_0'x_2'}{35} + \frac{x_0'x_3'}{70} + \frac{3x_1'^2}{35} \right. \\
& + \frac{9x_1'x_2'}{70} + \frac{2x_1'x_3'}{35} + \frac{3x_2'^2}{35} + \frac{x_2'x_3'}{7} + \frac{x_3'^2}{7} \\
& + \frac{y_0'^2}{7} + \frac{y_0'y_1'}{7} + \frac{2y_0'y_2'}{35} + \frac{y_0'y_3'}{70} + \frac{3y_1'^2}{35} \\
& \left. + \frac{9y_1'y_2'}{70} + \frac{2y_1'y_3'}{35} + \frac{3y_2'^2}{35} + \frac{y_2'y_3'}{7} + \frac{y_3'^2}{7} \right)
\end{aligned}$$

Rephrased more clearly with matrices

$$\begin{aligned}
K(t) = \frac{m}{140} & \left(\begin{bmatrix} x_0' & x_1' & x_2' & x_3' \end{bmatrix} \begin{bmatrix} 10 & 10 & 4 & 1 \\ 0 & 6 & 9 & 4 \\ 0 & 0 & 6 & 10 \\ 0 & 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} x_0' \\ x_1' \\ x_2' \\ x_3' \end{bmatrix} \right. \\
& \left. + \begin{bmatrix} y_0' & y_1' & y_2' & y_3' \end{bmatrix} \begin{bmatrix} 10 & 10 & 4 & 1 \\ 0 & 6 & 9 & 4 \\ 0 & 0 & 6 & 10 \\ 0 & 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} y_0' \\ y_1' \\ y_2' \\ y_3' \end{bmatrix} \right) \quad (5)
\end{aligned}$$

Note that K is just for one bubble wall. To get the total kinetic energy of the system K_T we must sum the kinetic energy of all the walls. Note too that each Bezier curve end point is shared by the other walls meeting at that vertex. This means that some coordinates q_i will appear in more than one wall's kinetic energy equation.

2.2 Potential Energy

Potential energy in our system comes from two sources, air pressure and surface tension. Surface tension potential rises as a wall lengthens. Air pressure potential rises as the air inside a bubble is compressed.

2.2.1 Air Pressure Potential

We use the ideal gas law to estimate the behavior of the gas in the bubble under pressure. Here $P(t)$ and $A(t)$ are pressure and area of the bubble, T the temperature, N the number of particles in the gas, and k is Boltzmann's constant (modified for 2 dimensions). We will assume that all processes are isothermal and that N is constant.

$$P(t) = \frac{NkT}{A(t)} = \frac{\text{constant}}{A(t)}$$

The energy necessary to change a bubble's area from A_1 to A_2 is

$$\Delta E = \int_{A_1}^{A_2} P dA = \int_{A_1}^{A_2} \frac{NkT}{A} dA = NkT \ln \left(\frac{A_2}{A_1} \right)$$

We define the ambient pressure of the outside air as ρ , and A_ρ as the area of the bubble at this pressure. Then the energy stored as potential in a bubble of area $A(t)$ is

$$V_p(t) = -\rho A_\rho \ln \left(\frac{A(t)}{A_\rho} \right) = \rho A_\rho \ln \left(\frac{A_\rho}{A(t)} \right) \quad (6)$$

Though the above formula works for normal bubbles, we must deal separately with the air outside the bubble cluster represented by the external face. Since the area of this bubble is infinite, the formula will not apply directly. However, if we first set the bubble cluster in a large (but finite) room and then take the limit as the room becomes infinitely large, we can get a well-defined formula for the energy.

Let A_r be the area of the room and A_m be the area of the bubble mass. Then the area of the external bubble is $A_r - A_m$ and the energy required to go from A_r to $A_r - A_m$ is

$$V_p(t) = \lim_{A_r \rightarrow \infty} \rho A_r \ln \left(\frac{A_r}{A_r - A_m(t)} \right) = \rho A_m(t) \quad (7)$$

All that remains is to determine the area of the bubble. This is simply the sum of the signed area under the curves that make it up. The formula for the area under a Bezier curve is

$$\begin{aligned} A_b &= \int_0^1 B_y(p) \frac{\partial B_x(p)}{\partial p} dp \\ &= \int_0^1 ((1-p)^3 y_0 + 3p(1-p)^2 y_1 + 3p^2(1-p) y_2 + p^3 y_3) \\ &\quad (-3(1-p)^2 x_0 + 3(1+p(3p-4))x_1 + 3p(2-3p)x_2 + 3p^2 x_3) dp \\ &= x_0 \left(\frac{-y_0}{2} + \frac{-3y_1}{10} + \frac{-3y_2}{20} + \frac{-y_3}{20} \right) \\ &\quad + x_1 \left(\frac{3y_0}{10} + \frac{-3y_2}{20} + \frac{-3y_3}{20} \right) \\ &\quad + x_2 \left(\frac{3y_0}{20} + \frac{3y_1}{20} + \frac{-3y_3}{10} \right) \\ &\quad + x_3 \left(\frac{y_0}{20} + \frac{3y_1}{20} + \frac{3y_2}{10} + \frac{y_3}{2} \right) \end{aligned}$$

Rephrased more succinctly with matrices:

$$A_b = \frac{1}{20} \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} -10 & -6 & -3 & -1 \\ 6 & 0 & -3 & -3 \\ 3 & 3 & 0 & -6 \\ 1 & 3 & 6 & 10 \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (8)$$

To determine the area of a bubble from the area under the curves that make it up one must be careful to integrate each wall in the right direction. Here, the directed edge view is very useful in avoiding sign errors.

Of course to get the total potential due to pressure we need to sum over all walls of all bubbles in the cluster.

2.2.2 Surface Tension Potential

The potential energy due to surface tension varies linearly with wall length. We will introduce a coefficient τ for energy per unit length.

$$\begin{aligned} V_s(t) &= \tau \text{Length}(B) \\ &= \tau \int_0^1 \left| \frac{d\vec{B}(p, t)}{dp} \right| dp \\ &= \tau \int_0^1 \sqrt{\left(\frac{dB_x(p, t)}{dp} \right)^2 + \left(\frac{dB_y(p, t)}{dp} \right)^2} dp \end{aligned} \tag{9}$$

where

$$\begin{aligned} \frac{dB_x(p, t)}{dp} &= -3(1-p)^2 x_0 + 3(3p-1)(p-1)x_1 + 3p(2-3p)x_2 + 3p^2 x_3 \\ \frac{dB_y(p, t)}{dp} &= -3(1-p)^2 y_0 + 3(3p-1)(p-1)y_1 + 3p(2-3p)y_2 + 3p^2 y_3 \end{aligned}$$

The integral is of the square root of a 4th degree polynomial in p . This is known as an “elliptic integral” and cannot generally be solved in closed form.

2.3 Calculating the equations of motion

Having calculated the kinetic and potential energy of the system we can finally use them to calculate the Lagrangian $\mathcal{L}_T = K_T - V_T$.

Observe that the formula for kinetic energy is defined only in terms of the velocities of the control points q'_1, \dots, q'_n and the potential is defined only in terms of positions of the control points q_1, \dots, q_n . This simplifies the equations of motion greatly:

$$\begin{aligned} \frac{d}{dt} \frac{\partial \mathcal{L}_T}{\partial q'_n} &= \frac{\partial \mathcal{L}_T}{\partial q_n} \\ \frac{d}{dt} \frac{\partial (K_T - V_T)}{\partial q'_n} &= \frac{\partial (K_T - V_T)}{\partial q_n} \\ \frac{d}{dt} \left(\frac{\partial K_T}{\partial q'_n} - \frac{\partial V_T}{\partial q'_n} \right) &= \frac{\partial K_T}{\partial q_n} - \frac{\partial V_T}{\partial q_n} \\ \frac{d}{dt} \frac{\partial K_T}{\partial q'_n} &= - \frac{\partial V_T}{\partial q_n} \\ \frac{d}{dt} \frac{\partial K_T}{\partial q'_n} &= - \left(\frac{\partial V_{Tp}}{\partial q_n} + \frac{\partial V_{Ts}}{\partial q_n} \right) \end{aligned}$$

To get the equations of motion we need to calculate the above equation for every control point coordinate in the system. This means generating 8 equations for every bubble wall in the system. We will only work out 2 such equations here as the rest can be inferred from them by symmetry.

2.3.1 Equation of motion for x_1

We will generate the Lagrangian for the x coordinate of the second control point x_1 of the bubble wall as this case is simpler than the first coordinate which is shared by three walls. Our equation is

$$\frac{d}{dt} \frac{\partial K_T}{\partial x'_1} = - \left(\frac{\partial V_{Ts}}{\partial x_1} + \frac{\partial V_{Tp}}{\partial x_1} \right)$$

First observe that since x_1 only appears in the kinetic energy formula for one wall the quantity $\frac{\partial K_T}{\partial x'_1} = \frac{\partial K}{\partial x'_1}$ so the left-hand side of the equation is

$$\begin{aligned} \frac{d}{dt} \frac{\partial K_T}{\partial x'_1} &= \frac{d}{dt} \frac{\partial K}{\partial x'_1} \\ &= \frac{d}{dt} \frac{\partial}{\partial x'_1} \frac{m}{140} \left([x'_0 \quad x'_1 \quad x'_2 \quad x'_3] \begin{bmatrix} 10 & 10 & 4 & 1 \\ 0 & 6 & 9 & 4 \\ 0 & 0 & 6 & 10 \\ 0 & 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} \right. \\ &\quad \left. + [y'_0 \quad y'_1 \quad y'_2 \quad y'_3] \begin{bmatrix} 10 & 10 & 4 & 1 \\ 0 & 6 & 9 & 4 \\ 0 & 0 & 6 & 10 \\ 0 & 0 & 0 & 10 \end{bmatrix} \begin{bmatrix} y'_0 \\ y'_1 \\ y'_2 \\ y'_3 \end{bmatrix} \right) \\ &= \frac{m}{140} \frac{d}{dt} (10x'_0 + 12x'_1 + 9x'_2 + 4x'_3) \\ &= \frac{m}{140} (10x''_0 + 12x''_1 + 9x''_2 + 4x''_3) \end{aligned}$$

The right-hand side has two parts. One for pressure and one for surface tension. We calculate the pressure first.

Observe that x_1 only appears in the formula for the areas of the bubbles A_1 and A_2 on either side of the bubble wall. Let $A_{1\rho}$ and $A_{2\rho}$ be their area at ambient pressure.

$$\begin{aligned} \frac{\partial V_{Tp}}{\partial x_1} &= \frac{\partial}{\partial x_1} \left(\rho A_{1\rho} \ln \left(\frac{A_{1\rho}}{A_1} \right) + \rho A_{2\rho} \ln \left(\frac{A_{2\rho}}{A_2} \right) \right) \\ &= \rho \left(\frac{A_{1\rho}}{A_1} \frac{\partial A_1}{\partial x_1} + \frac{A_{2\rho}}{A_2} \frac{\partial A_2}{\partial x_1} \right) \end{aligned}$$

Note that for one bubble the wall corresponds to a directed edge where p runs from 0 to 1, for the other it runs 1 to 0. This introduces a sign change in the

formula for area. Let us arbitrarily choose A_2 to represent the reversed sign bubble.

$$\begin{aligned}\frac{\partial V_{Tp}}{\partial x_1} &= \rho \left(\frac{A_{1\rho}}{A_1} \frac{3}{20} (2y_0 - y_2 - y_3) - \frac{A_{2\rho}}{A_2} \frac{3}{20} (2y_0 - y_2 - y_3) \right) \\ &= \frac{3\rho}{20} (2y_0 - y_2 - y_3) \left(\frac{A_{1\rho}}{A_1} - \frac{A_{2\rho}}{A_2} \right)\end{aligned}$$

Lastly, observe that for the external face the factor $\frac{A_\rho}{A}$ is instead 1.

Finally we get to the surface tension. Unfortunately this expression is irreducible.

$$\begin{aligned}\frac{\partial V_s(t)}{\partial x_1} &= \frac{\partial}{\partial x_1} \tau \int_0^1 \sqrt{\left(\frac{dB_x(p,t)}{dp} \right)^2 + \left(\frac{dB_y(p,t)}{dp} \right)^2} dp \\ &= \tau \int_0^1 \frac{\left(\frac{\partial}{\partial x_1} \frac{dB_x(p,t)}{dp} \right) \frac{dB_x(p,t)}{dp}}{\sqrt{\left(\frac{dB_x(p,t)}{dp} \right)^2 + \left(\frac{dB_y(p,t)}{dp} \right)^2}} dp \\ &= \tau \int_0^1 \frac{3(3p-1)(p-1) \frac{dB_x(p,t)}{dp}}{\sqrt{\left(\frac{dB_x(p,t)}{dp} \right)^2 + \left(\frac{dB_y(p,t)}{dp} \right)^2}} dp\end{aligned}$$

This value must be estimated by numerical integration.

Altogether, that gives us

$$\begin{aligned}\frac{m}{140} (10x_0'' + 12x_1'' + 9x_2'' + 4x_3'') &= \frac{3\rho}{20} (2y_0 - y_2 - y_3) \left(\frac{A_{1\rho}}{A_1} - \frac{A_{2\rho}}{A_2} \right) \\ &\quad + \tau \int_0^1 \frac{3(3p-1)(p-1) \frac{dB_x(p,t)}{dp}}{\sqrt{\left(\frac{dB_x(p,t)}{dp} \right)^2 + \left(\frac{dB_y(p,t)}{dp} \right)^2}} dp\end{aligned} \quad (10)$$

2.4 Equation of motion for x_0

We generate the equation of motion for the x coordinate of a vertex x_0 where three walls meet. We will use α , β , and γ to distinguish the three walls.

The left-hand side of the Lagrangian equation is

$$\begin{aligned}\frac{d}{dt} \frac{\partial K_T}{\partial x_0'} &= \frac{m}{140} (20x_0'' + 10x_{1\alpha}'' + 4x_{2\alpha}'' + x_{3\alpha}'') \\ &\quad + \frac{m}{140} (20x_0'' + 10x_{1\beta}'' + 4x_{2\beta}'' + x_{3\beta}'') \\ &\quad + \frac{m}{140} (20x_0'' + 10x_{1\gamma}'' + 4x_{2\gamma}'' + x_{3\gamma}'') \\ &= \frac{m}{140} (60x_0'' + 10(x_{1\alpha}'' + x_{1\beta}'' + x_{1\gamma}'') \\ &\quad + 4(x_{2\alpha}'' + x_{2\beta}'' + x_{2\gamma}'') + (x_{3\alpha}'' + x_{3\beta}'' + x_{3\gamma}''))\end{aligned} \quad (11)$$

For the pressure part of the right-hand side we denote the area of the bubble shared by α and β by $A_{\alpha\beta}$.

$$\frac{\partial V_{Tp}}{\partial x_1} = \rho \left(\frac{A_{\alpha\beta\rho}}{A_{\alpha\beta}} \frac{\partial A_{\alpha\beta}}{\partial x_0} + \frac{A_{\beta\gamma\rho}}{A_{\beta\gamma}} \frac{\partial A_{\beta\gamma}}{\partial x_0} + \frac{A_{\gamma\alpha\rho}}{A_{\gamma\alpha}} \frac{\partial A_{\gamma\alpha}}{\partial x_0} \right)$$

Where

$$\frac{\partial A_{\alpha\beta}}{\partial x_0} = \frac{1}{20} (-10(y_{0\alpha} + y_{0\beta}) - 6(y_{1\alpha} + y_{1\beta}) - 3(y_{2\alpha} + y_{2\beta}) - (y_{3\alpha} + y_{3\beta}))$$

For the tension part of the right-hand side we have

$$\begin{aligned} \frac{\partial V_s(t)}{\partial x_0} = \tau & \left(\int_0^1 \frac{3(1-p)^2 \frac{dB_{\alpha x}}{dp}}{\sqrt{\left(\frac{dB_{\alpha x}}{dp}\right)^2 + \left(\frac{dB_{\alpha y}}{dp}\right)^2}} dp \right. \\ & + \int_0^1 \frac{3(1-p)^2 \frac{dB_{\beta x}}{dp}}{\sqrt{\left(\frac{dB_{\beta x}}{dp}\right)^2 + \left(\frac{dB_{\beta y}}{dp}\right)^2}} dp \\ & \left. + \int_0^1 \frac{3(1-p)^2 \frac{dB_{\gamma x}}{dp}}{\sqrt{\left(\frac{dB_{\gamma x}}{dp}\right)^2 + \left(\frac{dB_{\gamma y}}{dp}\right)^2}} dp \right) \end{aligned}$$

Putting these together to form the equation of motion is left as an exercise to the reader.

2.5 Solving the equations of motion

We are now able to produce the equations of motion; all that is left is to solve them. The equations can't be written in closed form, so it's unsurprising that they can't be solved in closed form either. Instead we'll use a slightly modified form of the forward Euler method for approximating the solution to ODEs.

Observe that for every generalized coordinate q_j we get one equation of the form:

$$\sum_i^n c_{ij} q_i''(t) = F_j(q_1, q_2, \dots)$$

Here c_{ij} is a coefficient determined in equations (10) and (11) and F_j is $-\frac{\partial V(t)}{\partial q_j}$.

We generalize F_j to the vector function \vec{F} and express the equation in matrix form:

$$C\vec{q}'' = \vec{F}(\vec{q})$$

Since equations generated for x coordinates only include the second derivative of x coordinates (and vice versa) this equation can be cut into two, one for x and one for y.

Now all that is left is to numerically solve. We start with initial positions and velocities $\vec{q}(t_0)$ and $\vec{q}'(t_0)$. At each step we calculate the state after time Δt , $\vec{q}'(t_{n+1})$ and $\vec{q}(t_{n+1})$ from $\vec{q}(t_n)$ and $\vec{q}'(t_n)$.

$$\begin{aligned} \vec{q}''(t_n) &= \vec{F}(\vec{q}(t_n))/C \\ \vec{q}'(t_{n+1}) &= \vec{q}'(t_n) + \Delta t \vec{q}''(t_n) \\ \vec{q}(t_{n+1}) &= \Delta t \left(\frac{\vec{q}'(t_{n+1}) + \vec{q}'(t_n)}{2} \right) + \vec{q}(t_n) \end{aligned}$$

(Note the matrix division in the first step.)

One final step is to introduce friction. This not only adds realism, but also helps keep the numerical solution stable. Friction will be modeled by a drag friction c_f which is proportional to velocity. It modifies the equations like so:

$$\begin{aligned} \vec{q}'(t_{n+1}) &= \vec{q}'(t_n) + \Delta t \left(\vec{q}''(t_n) - c_f \vec{q}'(t_n) \right) \\ &= (1 - \Delta t c_f) \vec{q}'(t_n) + \Delta t \vec{q}''(t_n) \end{aligned}$$

2.5.1 Simplifying the matrix equation

The most time-consuming step in calculating each frame is to solve the matrix equation:

$$\begin{aligned} C\vec{Q} &= \vec{F} \\ \vec{Q} &= \begin{bmatrix} \vec{Q}_c \\ \vec{Q}_v \end{bmatrix} \\ \vec{F} &= \begin{bmatrix} \vec{F}_c \\ \vec{F}_v \end{bmatrix} \end{aligned}$$

We begin by looking at the structure of C . Here is one example of a 3 bubble

cluster:

$$C = \frac{1}{140} \left(\begin{array}{cccccccc|cccc} 12 & 9 & & & & & & & 4 & 10 & & & \\ 9 & 12 & & & & & & & 10 & 4 & & & \\ & & 12 & 9 & & & & & 4 & & & 10 & \\ & & 9 & 12 & & & & & 10 & & & 4 & \\ & & & & 12 & 9 & & & 4 & & 10 & & \\ & & & & 9 & 12 & & & 10 & & 4 & & \\ & & & & & & 12 & 9 & & & & & \\ & & & & & & 9 & 12 & & & & & \\ & & & & & & & & 12 & 9 & & & \\ & & & & & & & & 9 & 12 & & & \\ & & & & & & & & & & 12 & 9 & \\ & & & & & & & & & & 9 & 12 & \\ \hline 4 & 10 & 4 & 10 & 4 & 10 & & & 60 & 1 & 1 & 1 & \\ 10 & 4 & & & & & 10 & 4 & 1 & 60 & 1 & 1 & \\ & & & & 10 & 4 & 4 & 10 & 1 & 1 & 60 & 1 & \\ & & & & & & & & 4 & 10 & 10 & 4 & \\ & & & & & & & & 4 & 10 & 10 & 4 & \end{array} \right)$$

For convenience the zeros have been left empty and the equations of motion have been ordered with all the inner control points (in matching pairs) before the vertices. Lines have been added to clarify that C is a block matrix of form:

$$C = \left[\begin{array}{c|c} A & B \\ \hline B^\top & D \end{array} \right]$$

Our submatrices have very simple forms:

$$A_{ij} = \begin{cases} \frac{3}{35} & \text{if } i = j \\ \frac{9}{140} & \text{if control points } i \text{ and } j \text{ lie on the same edge} \\ 0 & \text{otherwise} \end{cases}$$

$$B_{ij} = \begin{cases} \frac{1}{14} & \text{if control point } i \text{ is directly connected to vertex } j \\ \frac{1}{35} & \text{if control point } i \text{ is indirectly connected to vertex } j \\ 0 & \text{otherwise} \end{cases}$$

$$D_{ij} = \begin{cases} \frac{3}{7} & \text{if } i = j \\ \frac{n}{140} & \text{if } i \neq j \\ & \text{where } n \text{ is the number of edges shared by vertex } i \text{ and } j \end{cases}$$

By block LU decomposition we can show:

$$C = \begin{bmatrix} A & B \\ B^\top & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ B^\top A^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & D - B^\top A^{-1} B \end{bmatrix} \begin{bmatrix} I & A^{-1} B \\ 0 & I \end{bmatrix}$$

To solve $C\vec{Q} = \vec{F}$ let us define

$$\begin{bmatrix} \vec{Q}_{2c} \\ \vec{Q}_{2v} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & D - B^\top A^{-1} B \end{bmatrix} \begin{bmatrix} I & A^{-1} B \\ 0 & I \end{bmatrix} \begin{bmatrix} \vec{Q}_c \\ \vec{Q}_v \end{bmatrix}$$

so we get

$$\begin{bmatrix} I & 0 \\ B^\top A^{-1} & I \end{bmatrix} \begin{bmatrix} \vec{Q}_{2c} \\ \vec{Q}_{2v} \end{bmatrix} = \begin{bmatrix} \vec{F}_c \\ \vec{F}_v \end{bmatrix}$$

which when solved gives us

$$\begin{bmatrix} \vec{Q}_{2c} \\ \vec{Q}_{2v} \end{bmatrix} = \begin{bmatrix} \vec{F}_c \\ \vec{F}_v - B^\top A^{-1} \vec{F}_c \end{bmatrix}$$

Now we solve

$$\begin{bmatrix} A & 0 \\ 0 & D - B^\top A^{-1} B \end{bmatrix} \begin{bmatrix} I & A^{-1} B \\ 0 & I \end{bmatrix} \begin{bmatrix} \vec{Q}_c \\ \vec{Q}_v \end{bmatrix} = \begin{bmatrix} \vec{F}_c \\ \vec{F}_v - B^\top A^{-1} \vec{F}_c \end{bmatrix}$$

using the same technique as before

$$\begin{aligned} \begin{bmatrix} \vec{Q}_{1c} \\ \vec{Q}_{1v} \end{bmatrix} &= \begin{bmatrix} I & A^{-1} B \\ 0 & I \end{bmatrix} \begin{bmatrix} \vec{Q}_c \\ \vec{Q}_v \end{bmatrix} \\ \begin{bmatrix} A & 0 \\ 0 & D - B^\top A^{-1} B \end{bmatrix} \begin{bmatrix} \vec{Q}_{1c} \\ \vec{Q}_{1v} \end{bmatrix} &= \begin{bmatrix} \vec{F}_c \\ \vec{F}_v - B^\top A^{-1} \vec{F}_c \end{bmatrix} \\ \begin{bmatrix} \vec{Q}_{1c} \\ \vec{Q}_{1v} \end{bmatrix} &= \begin{bmatrix} A^{-1} \vec{F}_c \\ (\vec{F}_v - B^\top A^{-1} \vec{F}_c) / (D - B^\top A^{-1} B) \end{bmatrix} \end{aligned}$$

Finally we are left with

$$\begin{bmatrix} I & A^{-1} B \\ 0 & I \end{bmatrix} \begin{bmatrix} \vec{Q}_c \\ \vec{Q}_v \end{bmatrix} = \begin{bmatrix} A^{-1} \vec{F}_c \\ (\vec{F}_v - B^\top A^{-1} \vec{F}_c) / (D - B^\top A^{-1} B) \end{bmatrix}$$

Which leaves us with

$$\begin{aligned} \vec{Q}_v &= (\vec{F}_v - B^\top A^{-1} \vec{F}_c) / (D - B^\top A^{-1} B) \\ \vec{Q}_c &= A^{-1} \vec{F}_c - A^{-1} B \vec{Q}_v \end{aligned}$$

To efficiently calculate this requires simple definitions for the matrices used. Matrix A^{-1} can be directly computed:

$$A_{ij}^{-1} = \begin{cases} \frac{80}{3} & \text{if } i = j \\ -20 & \text{if control points } i \text{ and } j \text{ lie on the same edge} \\ 0 & \text{otherwise} \end{cases}$$

We can compute $A^{-1} B$ and $B^\top A^{-1}$ by matrix identities:

$$\begin{aligned} (A^{-1} B)_{ij} &= \sum_k A_{ik}^{-1} B_{kj} \\ &= \sum_k B_{kj} A_{ik}^{-1} \\ &= \sum_k B_{jk}^\top A_{ki}^{-1} \end{aligned}$$

$$(B^\top A^{-1})_{ji} = \frac{80}{3}B_{ij} - 20B_{pj}$$

where p is the control point on the same edge as i .

And that gives us:

$$(B^\top A^{-1})_{ji} = (A^{-1}B)_{ij} = \begin{cases} \frac{4}{3} & \text{if control point } i \text{ is directly connected to vertex } j \\ -\frac{2}{3} & \text{if control point } i \text{ is indirectly connected to vertex } j \\ 0 & \text{otherwise} \end{cases}$$

Similarly

$$(B^\top A^{-1}B)_{ij} = \begin{cases} \frac{8}{35} & \text{if } i = j \\ -\frac{n}{105} & \text{if } i \neq j \end{cases}$$

$$(D - B^\top A^{-1}B)_{ij} = \begin{cases} \frac{1}{5} & \text{if } i = j \\ \frac{n}{60} & \text{if } i \neq j \end{cases}$$

Continuing on, the first vector can be defined like so:

$$(\vec{F}_v - B^\top A^{-1}\vec{F}_c)_i = (F_v)_i - \left(\frac{4}{3} \sum_n (F_c)_n + \frac{2}{3} \sum_f (F_c)_f \right)$$

Where the first sum is over the three control points directly connected to vertex i and the second sum is over those indirectly connected to vertex i .

The second vector \vec{Q}_c is defined like so:

$$(Q_c)_i = -\frac{4}{3}(Q_v)_n + \frac{80}{3}(F_c)_i - 20(F_c)_p - \frac{2}{3}(Q_v)_f$$

At this point it's straightforward to calculate each frame by using Cholesky decomposition to quickly solve for Q_v and use that to calculate Q_c .

An even quicker way is to approximate the inverse of the Schur complement matrix $M = D - B^\top A^{-1}B$ using the Neumann series expansion.

$$(5M)^{-1} = \sum_{n=0}^{\infty} (I - 5M)^n$$

$$(M)^{-1} = 5 \sum_{n=0}^{\infty} (I - 5M)^n$$

We use $5M$ to improve convergence speed since this zeros out the diagonals. This reduces the spectral radius as estimated using Gershgorin circle theorem. The spectral radius controls the speed of convergence.

The first three terms of this expansion give us

$$M^{-1} = 5 \sum_{n=0}^{\infty} (I - 5M)^n \approx 5I + 5(I - 5M) + 5(I - 5M)^2$$

All that's left is to compute it:

$$\begin{aligned} (5 \sum_{n=0}^0 (I - 5M)^n)_{ij} &= \begin{cases} 5 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \\ (5 \sum_{n=0}^1 (I - 5M)^n)_{ij} &= \begin{cases} 5 & \text{if } i = j \\ -\frac{5}{12}n & \text{otherwise} \end{cases} \\ (5 \sum_{n=0}^2 (I - 5M)^n)_{ij} &= \begin{cases} 5 + \frac{5}{48} & \text{if } i = j \\ -\frac{5}{12}n + \frac{5}{144}s & \text{otherwise} \end{cases} \end{aligned}$$

Where s is the number of common neighbors shared by i and j . For most vertices in most graphs of the time $s = 0$ so that term can be ignored.

With this simplification each frame can be computed in time linear with respect to the number of vertices.

2.5.2 Configuration parameter consolidation

Lastly, we can use a bit of unit analysis to simplify our configuration parameters. These parameters must be chosen “by hand” and consist of m , τ , ρ , c_f and Δt .

Observe that since the only terms measured in units of time are τ and ρ we can simply choose to rescale time to use units of Δt (whatever that is) without limiting our configuration space. That makes $\Delta t = 1$ (in units of itself) and that allows us to restate the above equation:

$$\vec{q}'(t_{n+1}) = (1 - c_f) \vec{q}'(t_n) + \vec{q}''(t_n)$$

Furthermore, observe that m only appears opposite τ and ρ so we can restate all equations in terms of $\frac{\tau}{m} = \tau_m$, $\frac{\rho}{m} = \rho_m$, and c_f .

Trial and error yields the following values:

$$\rho_m \approx 1 \quad \tau_m \approx 20 \quad c_f \approx .05$$

3 Sources of Error

For the sake of tractability, we represent the bubble wall with a cubic Bezier. Having only 8 degrees of freedom cubic Bezier curves cannot represent all possible configurations for a deformed bubble wall. Even worse, it cannot represent an arc of a circle which is the rest shape of a real bubble wall. This is only noticeable for walls with high curvature, however.

Another source of error occurs in the kinetic energy equation. There we implicitly assume that the mass of the bubble wall is described by the parameter p . But this parameter *does not necessarily traverse the wall at constant speed!* This means that our equations have some parts of the bubble wall heavier than others. This also affects the direction and magnitude of the velocity of each part of the bubble wall.

Another source of error is in approximating the matrix solve when solving for $\vec{q}''(t_{n+1})$. This can be solved exactly but only at prohibitively slow speed.

Further error is introduced by the the numerical integration of the surface tension component.

The most obvious source of error is the *very* basic forward Euler technique used to solve the equations of motion. This technique is known to be very unstable and it is only acceptable here because of friction.

No attempt has been made to estimate the magnitude of these errors. Actual implementations yield satisfactory enough results.