Finite element methods for investigating the moving boundary problem in biological development

Cornel Marius Murea
Laboratoire de Mathématiques, Université de Haute-Alsace,
4, rue des Frères Lumière, 68093 Mulhouse, France
C.Murea@uha.fr
http://www.math.uha.fr/~edp/murea/

George Hentschel
Emory University, Department of Physics,
Atlanta, GA 30322 USA
phshgeh@physics.emory.edu

Content

Early Avian Limb Development

The free boundary problem

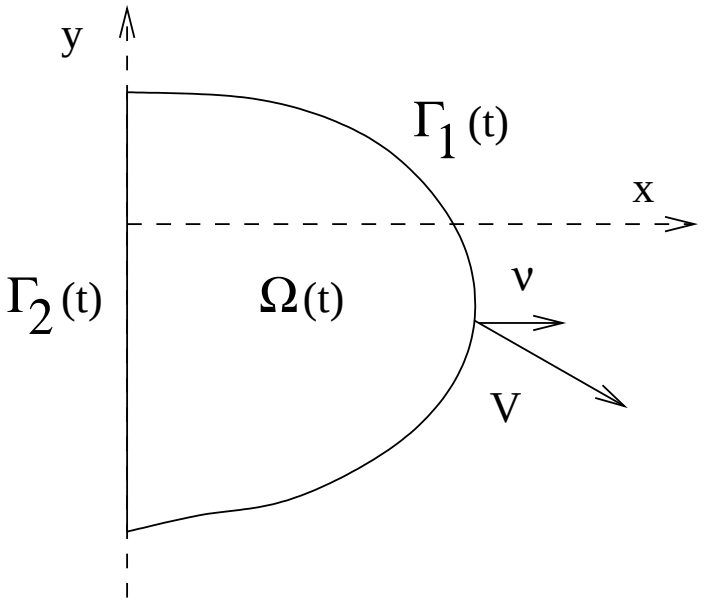
Numerical methods. Two algorithms

Numerical tests. A semicircle and non-convex initial domain

Concluding remarks

Early Avian Limb Development

$\Gamma_1(t)$ a non closed curve and its ends evolve on the Oy ax.
$\Gamma_2(t)$ the segment which has the same ends as $\Gamma_1(t)$.
$t \geq 0$ the time, ν the outer unit normal vector to the boundary.



First mathematical model

Knowing the initial domain $\Omega(0) = \Omega^0$, find the moving domain $\Omega(t)$, the velocity $\mathbf{v}$ and the pressure p such that

$$\begin{aligned} \mathbf{v} &= -\alpha \nabla p, && \text{in } \Omega(t) \\ \nabla \cdot \mathbf{v} &= S, && \text{in } \Omega(t) \\ \mathbf{v} \cdot \nu &= 0, && \text{on } \Gamma_2(t) \\ p &= \gamma \kappa, && \text{on } \Gamma_1(t) \\ V_\nu &= \mathbf{v} \cdot \nu, && \text{on } \Gamma_1(t) \end{aligned}$$

where $\alpha > 0$ is a constant, $S(\mathbf{x}, t)$ is the rate of growth, $\gamma > 0$ is the surface tension, κ is the curvature, V_ν is normal velocity of the boundary $\Gamma_1(t)$.

The pressure formulation

Knowing the initial domain $\Omega(0) = \Omega^0$, find the moving domain $\Omega(t)$ and the pressure p such that

$$\begin{aligned} -\Delta p &= S/\alpha, && \text{in } \Omega(t) \\ \frac{\partial p}{\partial \nu} &= 0, && \text{on } \Gamma_2(t) \\ p &= \gamma\kappa, && \text{on } \Gamma_1(t) \\ V_\nu &= -\alpha\frac{\partial p}{\partial \nu}, && \text{on } \Gamma_1(t) \end{aligned}$$

where $\alpha > 0$ is a constant, $S(\mathbf{x}, t)$ is the rate of growth, $\gamma > 0$ is the surface tension, κ is the curvature, V_ν is normal velocity of the boundary $\Gamma_1(t)$.

Existence

The last problem is similar to the Hele-Shaw problem with surface tension, also called Mullins-Sekerka problem ($\Gamma_2(t)$ is empty).

The existence and uniqueness of classical solution for the Hele-Shaw with surface tension problem was proved in *Escher J., Simonett G.*, (1997).

Non normal boundary velocity

The shape of the moving domain is determined solely by its normal velocity. In other words, if the velocity of $\Gamma_1(t)$ has the form

$$\mathbf{V} = V_\nu \cdot \nu + V_\tau \cdot \tau$$

where τ is the unit tangent vector to the boundary, then the movement of the domain and the pressure are the same as in the case $V_\tau = 0$.

We set $V_\tau = -\alpha\,(\nabla p \cdot \tau)$ and then $\mathbf{V} = -\alpha \nabla p$.

The advantage of this choice is that we do not need to evaluate the normal vector to the boundary when we compute the velocity of the boundary.

Numerical methods

To solve numerically the Hele-Shaw problem with surface tension, there exists an efficient approach named $\theta - L$ introduced in *Hou T.Y., Lowengrub J.S. & Shelley M.J.*, (1994).

This method is not appropriate if we replace the linear Darcy's law by the non-linear Navier-Stokes equation with a volume source located at a certain point in the domain

$$\rho_0 \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) - \mu \Delta \mathbf{v} + \nabla p = \mathrm{f} + \frac{\mu}{3} \nabla (\nabla \cdot \mathbf{v})$$

where $\rho_0 > 0$ is the density of the fluid and $\mu > 0$ its viscosity.

A frequented framework used for Navier-Stokes equation in moving domain is Arbitrary Lagrangian Eulerian together with the Finite Element Method, *Hughes T.J.R., Liu W.K., Zimmermann T.K.*, (1981).

Other approaches are Time-Space Finite Elements, *Bansch E.*, (2001) and Immersed Boundary Methods, *Peskin* (2002). The last one was employed to study the avian limb development in *Dillon, Othmer* (1999).

In this paper we present two algorithms which belong to the general framework called "front-tracking methods" *Crank* (1987). The numerical results were produced for the Darcy's law, but these algorithms could be used for the steady Navier-Stokes equation also.

Algorithm 1

We denote by Δt the time step and by $N = T/\Delta t$ the number of time steps.

We approximate $\Gamma_1(n\Delta t)$ by a polygonal line Γ_1^n of vertices (x_i^n, y_i^n) for $i = 0, \ldots, M$. We have $x_0^n = x_M^n = 0$, for all n. We denote by Ω^n the polygonal domain bounded by Γ_1^n and the Oy ax.

Generate a triangular mesh for Ω^0 using **freefem++**.

for each n **from** 0 **to** $N - 1$ **do**

Step 1: Compute the discrete curvature $\kappa^n(x_i, y_i)$ at each vertex (x_i, y_i) of Γ_1^n as the inverse of the ray of the circle passing through the three points (x_{i-1}^n, y_{i-1}^n), (x_i^n, y_i^n) and (x_{i+1}^n, y_{i+1}^n).

Step 2: Compute p^n by the Finite Element Method, P_1

$$\begin{cases} -\Delta p^n(x,y) &= S/\alpha, & \text{in } \Omega^n \\ p^n(x_i,y_i) &= \gamma\,\kappa^n(x_i,y_i), & \forall\,(x_i,y_i) \text{ vertex of } \Gamma_1^n \\ \dfrac{\partial p^n}{\partial \nu}(x,y) &= 0, & \text{on } \Gamma_2^n. \end{cases}$$

Step 3: Compute the discrete gradient of p^n at each vertex of Γ_1^n as a weighted mean of the gradients in the neighborhood triangles.

Step 4: Compute the vertices of Γ_1^{n+1}

$$\begin{aligned} x_i^{n+1} &= x_i^n - \alpha \Delta t \frac{\partial p^n}{\partial x}(x_i^n, y_i^n) \\ y_i^{n+1} &= y_i^n - \alpha \Delta t \frac{\partial p^n}{\partial y}(x_i^n, y_i^n) \end{aligned}$$

then set $x_0^{n+1} = x_M^{n+1} = 0$.

Step 5: Γ_2^{n+1} is the segment of ends $\left(x_M^{n+1}, y_M^{n+1}\right)$ and $\left(x_0^{n+1}, y_0^{n+1}\right)$.

Step 6: Compute a triangular dynamic mesh for Ω^{n+1}, where $\partial\Omega^{n+1} = \Gamma_1^{n+1} \cup \Gamma_2^{n+1}$ using a relaxation algorithm.

endfor;

Algorithm 2

for each n **from** 0 **to** $N-1$ **do**

Step 1: Let (x_i^n, y_i^n) for $i = 0, \ldots, M$ be points on $\Gamma_1(n\Delta t)$. Compute the interpolating cubic spline functions

$$\Gamma_1^n = \{(x(s), y(s)),\ s \in [a, b]\}$$

Step 2: Compute the curvature $\kappa^n(x_i, y_i)$ at each vertex $\left(x_i^n, y_i^n\right)$ using

$$k(s) = \frac{x'(s)y''(s) - x''(s)y'(s)}{\left((x'(s))^2 + (y'(s))^2\right)^{3/2}}$$

Step 3: Generate a triangular mesh for Ω^n using **freefem++** where $\partial\Omega^n = \Gamma_1^n \cup \Gamma_2^n$ and Γ_2^n is the segment of ends $\left(x_M^n, y_M^n\right)$ and $\left(x_0^n, y_0^n\right)$.

Step 4: Step 5: Step 6: Compute p^n, ∇p^n and (x_i^{n+1}, y_i^{n+1}) as in the Algorithm 1.

endfor;

Numerical tests. The initial domain is a semicircle

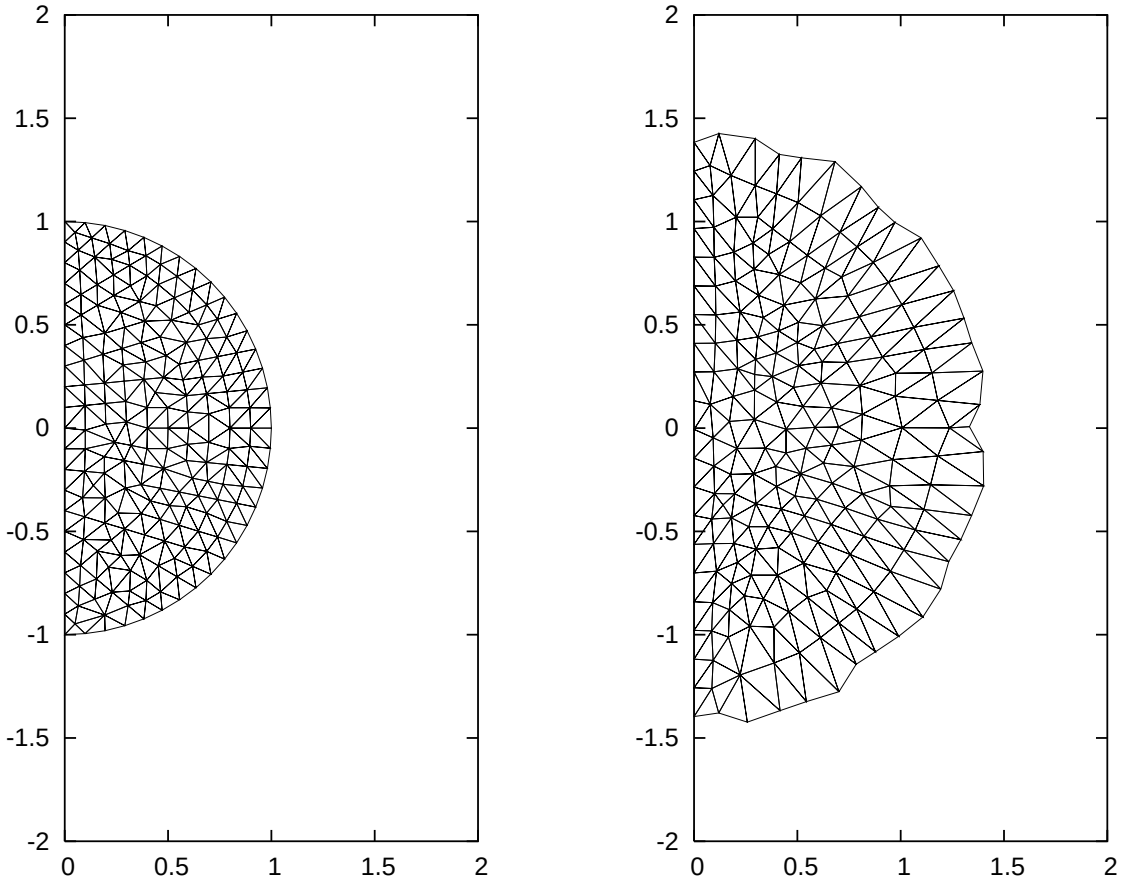
The algorithms have been implemented using the programming language C++ and the Finite Element classes of F. Hecht.

The first simulation was performed using the **Algorithm 1** for $R_0 = 1$, $S = 2$, $\gamma = 1$, $\alpha = 0.5$, $\Delta t = 0.05$. The number of time steps is $N = 10$.

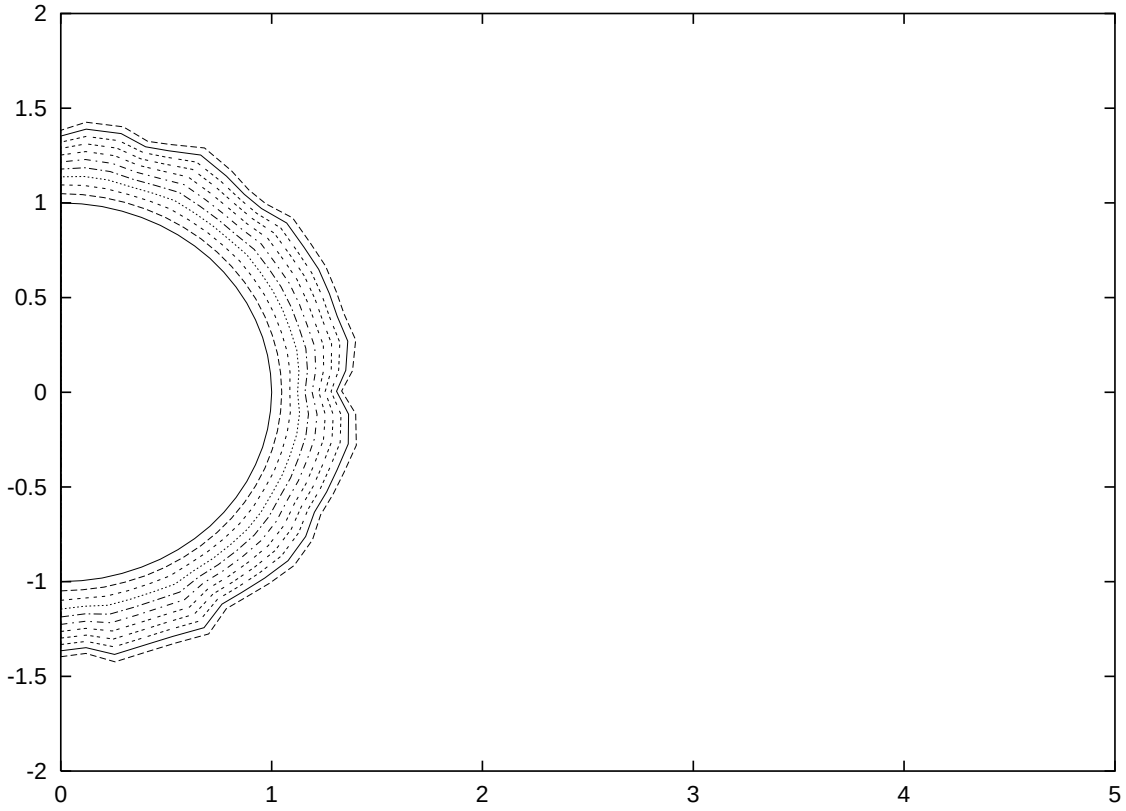
A dynamic mesh technique is used in order to generate a triangular mesh at each time step. The initial mesh has: 208 vertices, 362 triangles, $M = 32$ (the number of vertices on the boundary Γ_1), $h = 0.155178$ (the mesh size).

The number of vertices, triangles, boundary edges are the same for the first 10 meshes.

The initial mesh (left) and the mesh after 10 time steps (right)



The evolution of the moving boundary (0–10 time steps)

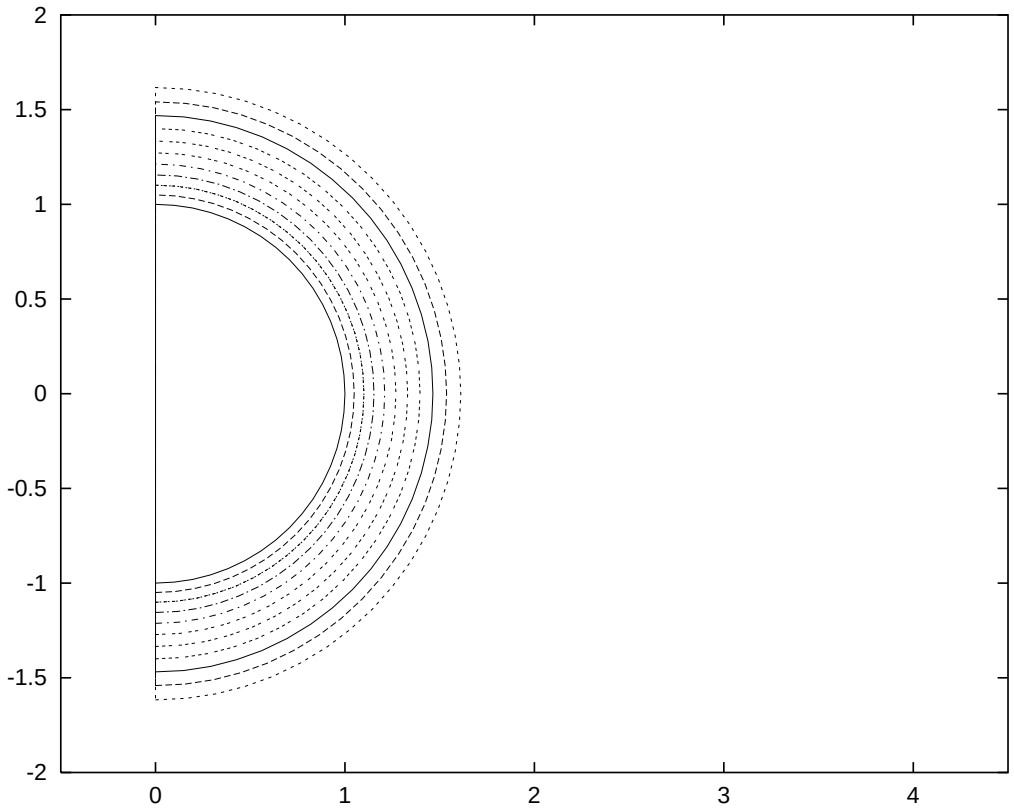


The second simulation was performed using the **Algorithm 2** for $\Delta t = 0.0005$ and $N = 1000$ (the number of time steps). The others parameters are the same as in the first simulation.

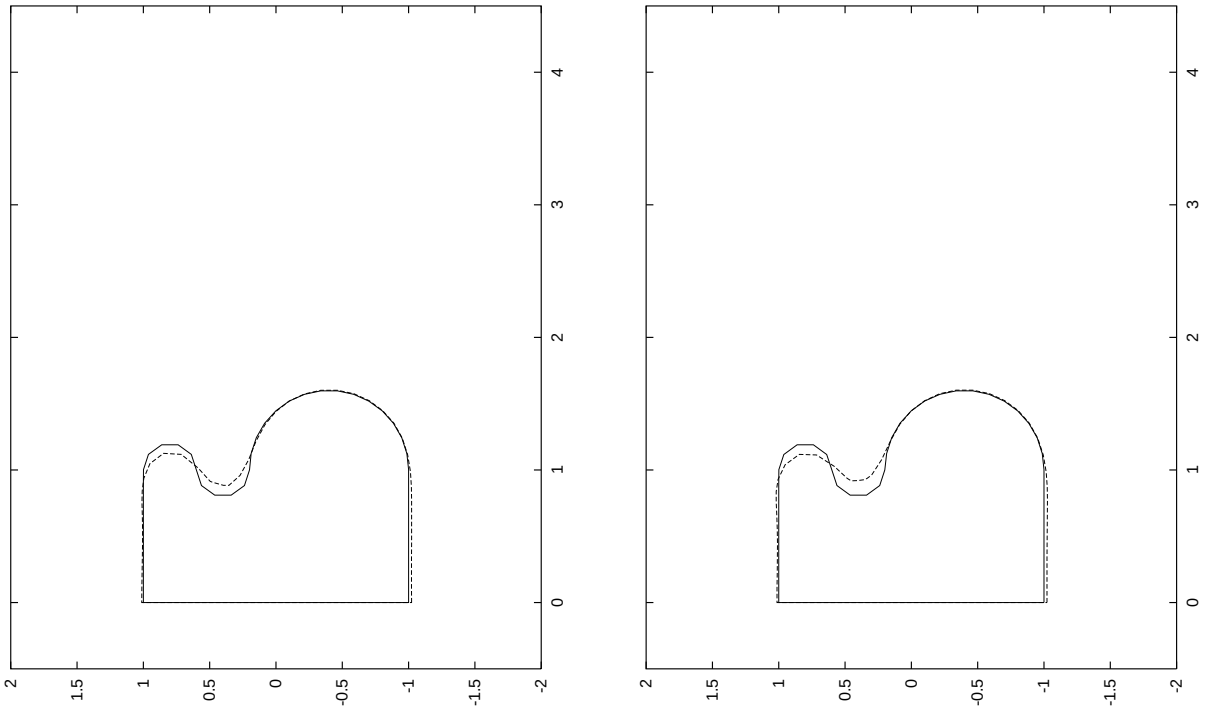
The software `freefem++` was used to generate a triangular mesh at each time step. The numbers of the vertices and of the triangles are not the same for the all meshes. For example, the mesh after 1000 time steps has 205 vertices and 356 triangles.

After 1000 time steps, the domain might be a semicircle of ray $e^{0.5} \approx 1.648721$.

The boundary after 0, 100, . . . , 1000 time steps



The initial (continuous) and the final (dashed) boundaries.
Algorithms 1 (left) and 2 (right)



A non-convex initial domain

The boundary $\Gamma_1(0)$ has two flat parts on the bottom, on the top and three semicircles of rays $r1 = 0.6$, $r2 = 0.2$ and $r3 = 0.2$ respectively.

The simulations ware performed for: $S = 2$, $\alpha = 0.5$, $\gamma = 0.5$, $\Delta t = 0.0001$ and $N = 180$ (the number of time steps).

We have observed that the pressure is almost constant near the two flat parts of $\Gamma_1(t)$ and near the largest semicircle. Consequently, these parts of boundary don't move.

We obtain boundary with self-intersection (like the 8) after 182 time steps using the Algorithm 2 and after 230 time steps using the Algorithm 1.

Concluding remarks

The second algorithm is superior to the first one due in principal to a better approximation of the curvature and a better mesh. We can improve the results by adapting mesh techniques.

In the first numerical test, we have solved the free boundary problem until time $t = 0.5$ and in the second, until $t = 0.018$.

We have to use implicit in time algorithms in order to study this kind of free boundary problem in a long time interval.

The Arbitrary Lagrangian Eulerian framework together with the Finite Element Method will be employed in a future paper. These methods could be employed for the Navier-Stokes equation with surface tension.