V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

VORTICES FORMATION FOR MEDUSA-LIKE OBJECTS (ECCOMAS CFD 2010)

Vladimir Lazunin*, Vladimir Savchenko†

*†Hosei University,
3-7-2 Kajinocho, Koganei-shi, Tokyo 184-8584, Japan
e-mail: lazunin@gmail.com
e-mail: vsavchen@hosei.ac.jp

Key words: Fluid Dynamics, Medusa, Vortex, Elasticity, Fluid-Solid Interaction

Abstract. *Jellyfish are the earliest known animals to achieve locomotion using muscle power. Medusa-like marine animals use jet propulsion principles for their swimming. As some of the researches indicate, smaller prolate medusae create strong jets during their bell contraction stage, while bigger oblate species produce substantially less distinct jets and broad vortices at the bell margins during their contraction and expansion. While some researches state that the bigger medusae are unable to create jets strong enough for their locomotion because of their morphological limitations, and vortices therefore play the most important role in their predominantly rowing-based swimming model, there also studies which show the contrary - that even for the biggest medusae the jet propulsion model provides more accurate prediction of the animal movements than the rowing model. The role of vortices in jellyfish swimming is therefore unclear. Development of the generalized dynamical model of medusan swimming is of interest to biologists as well as engineers.*

In this paper we propose a method for modeling vortices occuring during the hydromedusan swimming cycle. Because of the radial symmetry, we use a 2-d model (a cross-section) with the surface represented by two ellipsoidal bowls. A simplified approach based on nonlinear deformations of a geometric object is used to model bell contraction-expansion cycle. We use a particle-gridless hybrid method for the analysis of incompressible flows, with averaging velocities field by a Shepard's method (partition of unity).

1 INTRODUCTION

Jellyfish are the earliest known animals to use muscle power for swimming[4]. They swim by contracting and expanding their mesogleal bell. The swimming muscles contract to expel a portion of water rearward out of the subumbrellar cavity, thus generating a thrust force to move the animal forward. Unlike some other marine animals, using the same principles of locomotion, jellyfish doesn't have a muscle-antagoinst for refilling the bell. The bell is therefore refilled when it restores the shape after deformation it received during the thrust phase. The bell consists of a fibre-reinforced composite material called "mesoglea". The elastic characteristics of the mesogleal tissue are studied, for example, in a work of Megill et al.[7]

The contractile muscle fibers of medusae are only one cell layer thick, so the forces that they can produce do not scale favorably with increasing medusa size. For a medusa with the bell of diameter D, the mass of water that needs to be expelled from within the bell scales as D^3, while the muscle force only scales as D^1. Therefore the force required for jet propulsion increases with animal size more rapidly than the available physiological force[4]. Thus, swimming performance may change dramatically with the increase of the medusan body size, and it is impossible to predict optimal swimming parameters based on geometric and kinematic similarity.

Experimental researches, including dye injection, filming and analyzing the resulting flow, indicate, that smaller prolate medusae create strong jets during their bell contraction stage. Bigger oblate medusae, however, produce substantially less distinct jets and broad vortices at the bell margins. A hypothesis proposed by Colin and Costello[2,4,9,10] is that oblate species are using their bell's margins as "paddles", thus utilizing a paddling, or rowing, mode of swimming. According to the model presented by Dabiri et al.[4], big oblate medusae are not capable of swimming via jet propulsion. There is, however, a study of McHenry and Jed[6] which suggests that the jetting model still provides more accurate approximation of swimming in oblate jellyfish.

The flow generated by oblate medusa's pulsatile jets consists mostly of radially symmetric rotating currents called vortex rings. Researches using mechanical jet generators demonstrate that there is a physical limit – called the "vortex formation number" – for the maximum size of the vortex rings. Once this number is reached, no bigger vortex formation is possible, and the extra water creates a trailing current behind the vortex. The energy cost for generating this current is higher than that of creating the vortex ring, so it is optimal to generate the largest possible vortex without any trailing current[9]. Both thrust and efficiency increase in direct proportion with vortex ring volume[10].

To better understand the vortex formation and their effect on swimming performance, numerous experimental studies of real live jellyfish were performed[2,4,6,8,9,10]. In this paper we investigate the possibility of modeling and studying the vortices from the mathematical and computer graphics standpoint.

2 RELATED WORK

Müller at al. proposed a particle-based method for interaction of fluids with deformable solids[1]. In their method they model the exchange of momentum between Lagrangian particle-based fluid model and solids represented by polygonal meshes with virtual boundary particles to model the solid-fluid interaction.

Yoon et al. presented a particle-gridless hybrid method for the analysis of incompressible flows[13]. Their numerical scheme included Lagrangian and Eulerian phases. The moving-particle semi-implicit method (MPS) was used for the Lagrangian phase, and a convection scheme based on a flow directional local grid was developed for the Eulerian phase.

Chentanez et al. presented a method for simulating the two-way interaction between fluids and deformable solids[5]. The fluids were simulated using an incompressible Eulerian formulation where a linear pressure projection on the fluid velocities enforces mass conservation, whereas elastic solids were simulated using a semi-implicit integrator implemented as a linear operator applied to the forces acting on the nodes in Lagrangian formulation.

Hirato et al. proposed a method for generating animations of jellyfish with tentacles[11]. They used a simplified computational model based on the MPS method to simulate the fluid.

3 METHODOLOGY AND IMPLEMENTATION

3.1 Simulating the bell contraction-expansion cycle

To simulate the bell contraction-expansion cycle we use a simplified approach based on non-linear deformations of a geometric object. Because the model of jellyfish has radial symmetry, we use a 2D model (cross-section) with the surface of the bell represented by two hemi-ellipsoidal curves – the upper and the lower. For the model representation shown in Figure 1 we used a piece-wise linear approximation with the initial number of nodes equal 40. Space mapping techniques based on radial basis functions (RBFs) were used for non-linear approximation of shape deformations, see, for instance, [14]. Space mapping in R^n defines a relationship between each pair of points in the original model and the model after geometric modification. Let an n-dimensional region $\Omega \subset R^n$ of an arbitrary configuration be given, and let Ω contain a set of arbitrary control points $\{q_i = (q_1^i, q_2^i, ..., q_n^i) : i = 1, 2, ..., N\}$ for the non-deformed object, and $\{d_i = (d_1^i, d_2^i, ..., d_n^i) : i = 1, 2, ..., N\}$ for the deformed object. By assumption, the points q_i and d_i are distinct and given on or near the surface of each of two objects. The goal of the construction of the deformed object is to find a smooth mapping function that approximately describes the spatial transformation. The inverse mapping function can be given in the form

$$q_i = f(d_i) + d_i, \tag{1}$$

where the components of the vector$f(d_i)$ are volume splines interpolating displacements of initial points q_i.

We consider a mapping function as a thin-plate interpolation[15]. For an arbitrary area Ω, the thin-plate interpolation is a variational solution that defines a linear operator T when the following minimum condition is used:

$$\int_{\Omega} \sum_{|\alpha|=m} m!/\alpha!(D^{\alpha}f)^2 d\Omega \to min, \tag{2}$$

where m is a parameter of the variational function and α is a multi-index. It is equivalent to using the radial basis functions $\phi(r) = rlog(r)$ or r^3 for $m = 2$ and 3 respectively, where r is the Euclidean distance between two points.

The volume spline $f(P)$ having values h_i at N points P_i is the function

$$f(P) = \sum_{j=1}^{N} \xi_i \phi(|P - P_j|) + p(P), \tag{3}$$

where $p = \nu_0 + \nu_1 x + \nu_2 y + \nu_3 z$ is a degree-one polynomial. To solve for the weights ξ_j we have to satisfy the constraints h_i by substituting the right part of Equation (3), which gives

$$h_i = \sum_{j=1}^{N} \xi_j \phi(|P_i - P_j|) + p(P_i). \tag{4}$$

For 2D and 3D cases we call $f(P)$ a volume spline. To improve the calculation precision we use subdivision of linear segments to increase the number of boundary points.

3.2 Fluid simulation

Particle-based methods became a de-facto standard for a class of problems where high precision is not required. Moreover, solving Navier-Stokes equations with moving boundary is a hard problem. For modeling we use almost the same scheme as proposed by Yoon, Koshizuka and Oka[13]. They proposed a particle-gridless hybrid method for the analysis of incompressible flows, where tracing of virtual moving particles is used instead of solving nonlinear equations of velocity field. A particle interacts with other particles according to a weight function $w(r)$, where r is the distance between two particles. The weight function used by Koshizuka et al. is

$$w(r) = \begin{cases} -(2r/r_e)^2 + 2 & (0 \le r < 0.5r_e) \\ (2r/r_e - 2)^2 & (0.5r_e \le r < r_e) \\ 0 & (r_e \le r) \end{cases} \tag{5}$$

Density for a particle is calculated as the sum of weights of its interactions with the other particles (all interaction happens only within the radius r_e):

$$\langle n \rangle_i = \sum_i w(|r_i - r_j|). \tag{6}$$

A gradiend vector between two particles i and j possessing scalar quantities ϕ_i and ϕ_j at coordinates r_i and r_j is equal to $(\phi_j - \phi_i)(r_j - r_i)/|r_j - r_i|^2$. The gradient vector at the particle i is given as the weighted average of these gradient vectors:

$$\langle \nabla \phi \rangle_i = \frac{d}{n^0} \sum_{j \neq i} \left[\frac{\phi_j - \phi_i}{|r_j - r_i|^2} (r_j - r_i) w(|r_j - r_i|) \right], \tag{7}$$

where d is the number of space dimensions and n^0 is the particle number density.

Diffusion is modeled by distribution of a quantity from a particle to its neighbors using the weight function:

$$\langle \nabla^2 \phi \rangle_i = \frac{2d}{\lambda n^0} \sum_{j \neq i} [(\phi_j - \phi_i) w(r_j - r_i)], \tag{8}$$

This model is conservative, because the quantity lost by the particle i is obtained by the particle j.

The continuity equation for incompressible fluid can be written as follows:

$$\frac{D\rho}{Dt} = -\rho(\nabla \cdot u) = 0. \tag{9}$$

The velocity divergence at the particle i is given by:

$$\langle \nabla \cdot u \rangle = \frac{d}{n^0} \sum_{j \neq i} \frac{(u_j - u_i) \cdot (r_j - r_i)}{|r_j - r_i|^2} w(|r_j - r_i|). \tag{10}$$

Then the pressure is calculated as:

$$\frac{u_i^{**} - u_i^*}{\Delta t} = -\frac{1}{\rho} \langle \nabla P^{n+1} \rangle_i, \tag{11}$$

$$\langle \nabla^2 P^{n+1} \rangle_i = \frac{\rho}{\Delta t} \langle \nabla \cdot u^* \rangle_i, \tag{12}$$

where u^* is the temporal velocity obtained from the explicit calculation and u_i^{**} is the new-time velocity. The left side of (12) is calculated using the Laplacian model (8). The right side is the velocity divergence, calculated by (10). It gives a system of linear equations represented by an unsymmetric matrix, which is solved by an unsymmetric-pattern multifrontal method[17]. UMFPACK library is used for solving the system[18].

Instead of using a higher-order gridless convection scheme as it was proposed by Yoon et al.[13] to approximate flow directions, we apply averaging of the velocities field by a simple scheme, based on Shepard's method (partition of unity)[16].

3.3 Fluid and solid interaction

Usually in particle-based methods boundary particles are used to approximate the no-penetration condition[1]. Repulsion and adhesion forces between the particles are used to simulate the no-penetration, no-slip and actio = reactio conditions on the boundary of the solid.

In this work we model the interaction between fluid particles and the surface of the bell in a form of elastic collision and reflection of the fluid particles off the boundary surface.

3.4 Algorithm

Initially, the 2D boundary is specified as an array of points. Initial deformations are assigned to the bell margin points. Particles are placed on a regular grid inside the bounding box, except the inner area of the bell. Then, the following steps are performed iteratively:

1. The averaged density is calculated for every particle. A ball is generated for every particle, and the density is defined as the volume of the ball divided by the number of particles inside the ball.

2. The bell margin points are moved by a step along the deformation vectors. For the rest of the points their displacement vectors are calculated using RBFs.

3. A cubic spline is fit through the displaced boundary points. Because some segments become too long, they are subdivided by inserting new points.

4. A Poisson equation in a matrix form is solved (about 10000 linear equations), giving new particle densities.

5. Gradient vectors are calculated (7). For every particle, the speed vector is calculated, and the particle is then moved along the vector by the time step Δt.

6. The new values for density of the displaced particles are interpolated back to the nodes of regular grid.

4 RESULTS AND DISCUSSION

We have implemented this method and used it to model and visualize vortex formation for a simple oblate medusa with different swimming parameters. The program was written in C++. A UMFPACK library was used to solve large sparse systems of linear equations. The calculation took approximately 1 sec per step on an Intel Core 2 Quad computer, however, because we have not parallelized the program to use all processor cores on the system, only 1 core was actually used for the calculation. An example of visualization (Figure 1) shows the ability of proposed approach to simulate vortices observed for real medusae.

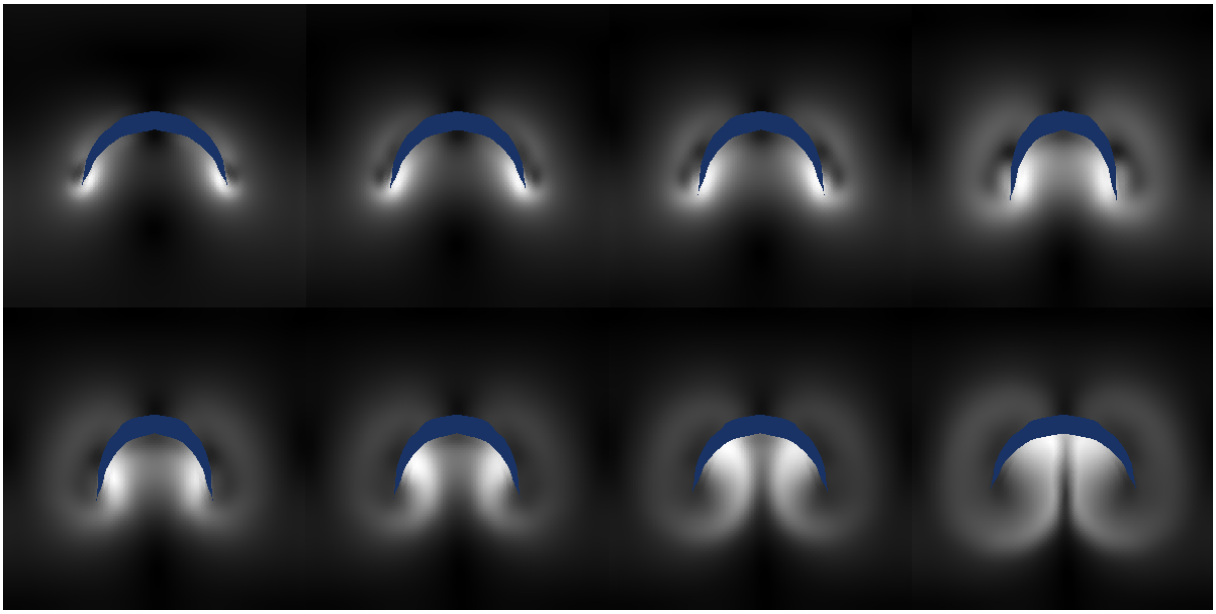

Figure 1: Example of vortex simulation. Contraction (above) and expansion (below) steps.

Future works may include solving an optimization problem of finding optimal parameters of pulsation and contraction/expansion for different medusa models and different modes of swimming, e. g. fast swimming, slow swimming, the most energetically efficient swimming and so on. These swimming patterns are discussed by Dabiri et al.[4,10]

REFERENCES

[1] M. Müller, S. Schirm, M. Teschner, B. Heidelberger, M. Gross, Interaction of Fluids with Deformable Solids, *Computer Animation and Virtual Worlds*, **15**, 159–171 (2004)

[2] S. P. Colin and J. H. Costello, Morphology, swimming performance and propulsive mode of six co-occuring hydromedusae, *The Journal of Experimental Biology*, **206**, 427–437 (2002).

[3] B. Ataie-Ashtiani and L. Farhadi, A stable moving-particle semi-implicit method for free surface flows, *Fluid Dynamics Research*, **38**, 241–256 (2006)

[4] J. O. Dabiri, S. P. Colin, J. H. Costello, Morphological diversity of medusan lineages constrained by animal-fluid interactions, *The Journal of Experimental Biology*, **210**, 1868–1873 (2007)

[5] N. Chentanez, T. G. Goktekin, B. E. Feldman, J. F. O'Brien, Simultaneous coupling of fluids and deformable bodies, *Eurographics/ ACM SIGGRAPH Symposium on Computer Animation*, 83–89 (2006)

[6] M. J. McHenry, J. Jed, The ontogenetic scaling of hydrodynamics and swimming performance in jellyfish (Aurelia aurita), *The Journal of Experimental Biology*, **206**, 4125–4137 (2003)

[7] W. M. Megill, J. M. Gosline, R. W. Blake, The modulus of elasticity of fibrillin-containing elastic fibres in the mesoglea of the hydromedusa Polyorchis penicillatus, *The Journal of Experimental Biology*, **208**, 3819–3834 (2005)

[8] J. O. Dabiri, M. Gharib, Sensitivity analysis of kinematic approximations in dynamic medusan swimming models, *The Journal of Experimental Biology*, **206**, 3675–3680 (2003)

[9] J. O. Dabiri, S. P. Colin, J. H. Costello, Fast-swimming hydromedusae exploit velar kinematics to form an optimal vortex wake, *The Journal of Experimental Biology*, **206**, 3675–3680 (2003)

[10] J. O. Dabiri, S. P. Colin, J. H. Costello, M. Gharib, Flow patterns generated by oblate medusan jellyfish: field measurements and laboratory analyses, *The Journal of Experimental Biology*, **208**, 1257–1265 (2005)

[11] J. Hirato, Y. Kawaguchi, Calculation model of jellyfish for simulating the propulsive motion and the pulsation of the tentacles, *18th International Conference on Artificial Reality and Telexistence* (2003)

[12] Y. Usami, Theoretical study on the body form and swimming pattern of Anomalocaris based on hydrodynamic simulation, *Journal of Theoretical Biology*, **238**, 11–17 (2006)

[13] H. Y. Yoon, S. Koshizuka, Y. Oka, A particle-gridless hybrid method for incompressible flows, *International Journal for Numerical Methods in Fluids*, **30**, 407–424 (1999)

[14] V. Savchenko, A. Pasko, Transformation of functionally defined shapes by extended space mappings, *The Visual Computer*, **14**, 257–270 (1998)

[15] M. D. Buhmann, Radial basis functions: theory and implementation, *Cambridge University Press* (2003)

[16] D. Shepard, A two-dimensional interpolation function for irregularly spaced data, *Proceedings of the 23th Nat. Conf. of the ACM*, 517–523 (1968)

[17] T. A. Davis, Algorithm 832: UMFPACK, an unsymmetric-pattern multifrontal method, *ACM Transactions on Mathematical Software*, **vol 30, no. 2**, 196–199 (June 2004)

[18] UMFPACK library, http://www.cise.ufl.edu/research/sparse/umfpack/