Interactive Simulation of the Process of Glottal Wave Generation using a GPU

Kazuhiko Yamamoto
Graduate School of Design / ADCDU and CDS
Kyushu University, Japan
e-mail: yamotulp@gmail.com

Abstract

In this study, the process of glottal wave generation by vocal fold vibration of human during normal phonation is simulated, where the problem is formulated mathmatically as a complex coupled dynamics between vocal folds that is elastic solid and glottal flow that is fluid. The vocal folds are asummed to be anisotropic elastic body, and they are discretaized by MPS(Moving Particles Semi-Implicit) Method on the basis of the a Lagrangian formulation. Glottal airflow is modeled as compressible and thermal fluid and it's governing equations are discretized by FDLB(Finite Difference Lattice Boltzmann) Method on the basis of an Eularian formulation. These two dynamical methods are then coupled. Most of the numerical calculation was implemented on a GPU(Graphics Processing Unit) as shader and on a CPU simultaneously in parallel efficiently. As a result, elastc solid-fluid complex coupled dynamics of the process of glottal wave generation and the process of sound generation from fluid dynamics are successfully simulated directly in 2D. Futhermore, the numerical calculation can be excuted at interactive rate using a usual PC.

Keywords: GPGPU, MultiPhysics, Fluid-Elastic Solid Coupled Dynamics, Aerodynamic Sound, FDLB-MPS Hybrid Method, Glottal Wave

1 Introduction

The glottal wave is the sound source for voiced speech of human. It is generated by the vibration of both vocal folds. The generation process of the glottal wave is a complex coupled dynamics between vocal folds which is elastic solid and glottal airflow which is fluid flow. During phonation, vocal folds deforms largely and both folds collide with each other. According to their vibratory motion, the glottal flow channel also deforms dynamically and even be closed. Futhermore, glottal flow is a high Reynolds number flow. So it is difficult to treat its large deformation using conventional methods like Finite Element Method for elastic solid regime, and in fluid regime, to keep the computational stability and handle the moving and curved boundary conditions in conventional techniques like Finite Difference Method. Therefore, it can be said this is one example of the most difficult multiphysics problem.

As is the case of glottal wave genertion problem, the simulation of elastic solid-fluid coupled dynamics itself is in general one of the most difficult physical problems. In the previous approaches, only a number of methods are established in the graphics field as the particle system, which has generally used like as SPH(Smoothed Particle Hydrodynamics) Method[1] to simulate deformable body and fluid interaction. Chenranes et al.[2] proposed a method that solves such a coupled dynamics problem as strong coupling to set large time step. However, a method is still seached which has physical flexibility with numerical accuracy for complex problems.

As a result of the vocal fold vibration, glottal sound wave are generated as a fluid dynamic sound. Generally, prediction fluid dynamic sounds has been performed by so called analytical hybrid method using Tightthill's equations. It uses the computational result of the fluid dynamics with the assumption of the incompressibility. But recentry, owing to the increase of computational power, direct simulation of fluid dynamic sounds have been tried by solving the compressible Navier Stokes equations using "brute force" approach. However, the computational cost of these calculations are still too expensive, so they should be performed using super computer or PC cluster.

In this study, to simulate the process of glottal wave generation directly, vocal folds are modeled as an anisotropic elastic body including the viscosity friction, and its equation of motion is discretalized by MPS(Moving Particles Semi-Implicit) Method. Glottal airflow is modeled as compressible and thermal fluid, and its governing equations are discretalized by FDLB(Finite Difference Lattice Boltzmann) Method, which can simulate directly fluid dynamic sounds. Both methods are then coupled. To perform the simulation using a usual PC at interactive rate, most of these calculations are implemented on GPU by Shader(NVIDIA cg, GLSL, and NVIDIA CUDA) and in addition, on CPU side simultaneously in parallel using the OS-dependent thread function and OpenMP.

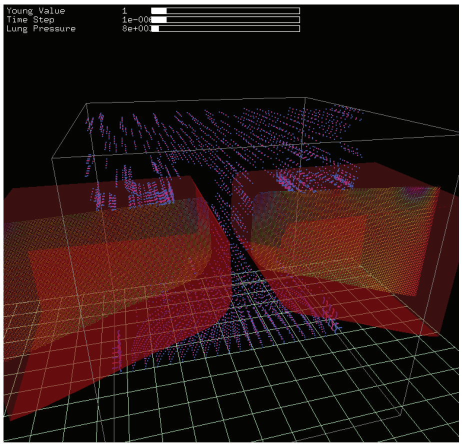


Figure 1: *Screen Shot.*

In this work, the numerical method is presented in two spatial dimension noting that three dimensional extensions are straightforward.

2 The Contributions of This Sketch

The main contributions of this report are:

(a): I propose FDLB-MPS coupling Method and applly it to the direct simulation of the process of glottal wave generation.
(b): In MPS Method, I present the formulation of an anisotropic elastics, and it is implemented on GPU using CUDA efficiently.

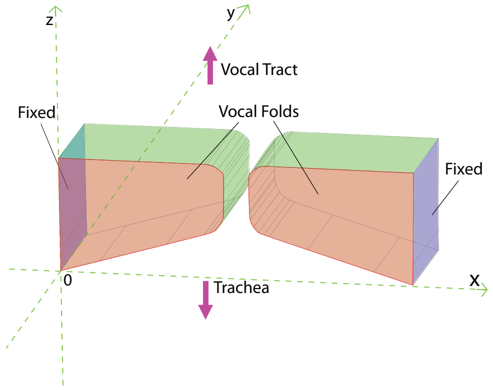
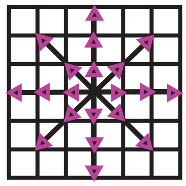


Figure 2: *The coordinate difinition for vocal folds.*



i	Velocity Vector	|c|
0	(0,0)	0
1~4	(1,0), (0,1), (-1,0), (0,-1)	1
5~8	(2,0), (0,2), (-2,0), (0,-2)	2
9~12	(3,0), (0,3), (-3,0), (0,-3)	3
13~16	(1,1), (-1,1), (-1,-1), (1,-1)	$\sqrt{2}$
17~20	(2,2), (-2,2), (-2,-2), (2,-2)	$2\sqrt{2}$

Figure 3: *Velocity set in D2Q21 model.*

(d): In FDLB Method, I simulated glottal flow as compressible and thermal fluid, allowing the direct simulation of fluid dynamic sounds. I also expand Mei et al.'s boundary treatment[3] and apply it to this model.
(e): The complex dynamics coupled between elastic solid, that deforms largelly, and fluid flow, that has high Reynolds number, are successfully simulated.
(f): Futhermore, I implemented these method on GPU and CPU simultaneously in parallel, and archieved an amazing framerate: that is the computational speed increased more than 400 times.

3 Elastic Object

MPS Method which is developed by Kosizuka et al. for the analysis of incompressible fluids originally[4]. It is a comparatively up-to-date particle based Method for the numerical analysis using a continuum model. Later, this method has been expanded for the problem of an isotropic elastics by Song et al.[5]. We further expand this method for an anisotropic elastics just like the real materials and apply it to represent the motion of vocal fold tissues, which are assumed to be isotropic in the x-z plane and anisotropic along the y axis. The coodinate difinition for vocal fold is shown in Figure 2. Their anisotropic elastics formulation is shown below in Eq.(1) as,

$$\rho\frac{\partial^2}{\partial t^2}\vec{\Psi} = \frac{E}{2(1+\nu)}\Delta\vec{\Psi} + \frac{E}{2(1-\nu)}\nabla(\nabla\cdot\vec{\Psi}) + \mu_y\frac{\partial^2}{\partial y^2}\vec{\Psi} + \eta\Delta\frac{\partial}{\partial t}\vec{\Psi} \tag{1}$$

where $\vec{\Psi}$ indicate the displacement, ρ, E, μ_z, and η denote the dencity, Young value, Poison rate along the y axis, and viscosity coefficient, respectively. This equation is discretized by replacing each differential operator using the corresponding model of the MPS method, which represent the interaction of the neighbering particles. These models are shown below.

$$\Delta\phi_i = \frac{2d}{\kappa n_0}\sum_{j\neq i}\Big((\phi_j-\phi_i)w(|\mathbf{r}_j-\mathbf{r}_i|)\Big) \tag{2}$$

$$\nabla\phi_i = \frac{d}{n_0}\sum_{j\neq i}\left(\frac{\phi_j-\phi_i}{|\mathbf{r}_j-\mathbf{r}_i|^2}(\mathbf{r}_j-\mathbf{r}_i)w(|\mathbf{r}_j-\mathbf{r}_i|)\right) \tag{3}$$

$$\nabla\cdot\mathbf{u}_i = \frac{d}{n_0}\sum_{j\neq i}\left(\frac{(\mathbf{u}_j-\mathbf{u}_i)\cdot(\mathbf{r}_j-\mathbf{r}_i)}{|\mathbf{r}_j-\mathbf{r}_i|^2}w(|\mathbf{r}_j-\mathbf{r}_i|)\right) \tag{4}$$

where κ, d, and n_0 denote a constant, spatial dimension, and initial dencity of particles, respectively. w is so called weight function.

$$w(r) = \begin{cases} \dfrac{r_e}{r}-1 & (0\leq r\leq r_e) \\ 0 & (r_e<r) \end{cases} \tag{5}$$

where r is the distance between two particles i and j, and r_e is the threshold distance to consider the interaction between two particles. Each particles are updated its position and angular value, linear velocity and angular velocity at every time step in order to satisfy the law of conservation of momentum.

4 Compressible and Thermal Fluid

On the other hand, concerning about the glottal airflow, we use the compressible and thermal fluid model of FDLBM(Finite Difference Lattice Boltzmann Mehod)-D2Q21Model. In the original LBM, the fluid is considered as a collection of particles that are represented by a particle velocity distribution function at each discrete lattice node. Particles collide with each other and properties associated with the lattice nodes are updated at discrete time steps. The rules governing the collisions are designed such that the time-average motion of the particles is consistent with the Navier-Stokes equations. FDLBM is a method of introducing the finite difference scheme into LBM, and here we use the second-order-accurate Runge-Kutta scheme for time marching, the third-order-accurate up-winding scheme for spatial derivation of the lattice BGK equation. This has some merits: 1, unstructed mesh can be used also, 2, Increased computational stability can be obtained for a high Reynolds number flow , 3, arbitrary velocity vectors can be selected, etc. In the case of the compressible and thermal fluid, multiple kinds of velocity vector are needed. The velocity vector is shown in Figure 2. In this study, I apply the distrete BGK equation as a governing equation of the FDLBM. The discrete BGK equation represents evolution of a velocity distribution function of particles $f_i(t,r)$ as

$$\begin{aligned}\frac{\partial f_i(t,r)}{\partial t} + \frac{\partial}{\partial r_\alpha}c_{i\alpha}f_i(t,r) - \frac{Ac_{i\alpha}}{\tau}\frac{\partial(f_i-f_i^{(0)})}{\partial r_\alpha} \\ = -\frac{1}{\tau}[f_i(t,r)-f_i^{(0)}(t,r)]\end{aligned} \tag{6}$$

where t and r indicate the time and the space, and A is a constant. Subscript i and α denote the directions of particle motion and the space directions. and $c_{i\alpha}$ represents a particle velocity that is shown in Figure 3. τ is called a relaxation time and $f_i^{(0)}(t,r)$ is the equilibrium distribution function, that is determained so as to recover the corresponding fluid dynamics eauations:Compressible Navier Stokes Equations. The third term on the left hand side of Eq.(6) means negative viscosity. We can set large time increment for high Reynolds number flows due to this additional term[6]. The right hand side of Eq.(6) is a collision term. It represents that a particle distribution approaches an equilibrium state by the collisions amoung particles. The equilibrium distribution function is given by a following polynomial of the flow velocity up to the third order.

$$\begin{aligned}f_i^{(0)} = F_i\rho(1-2Bc_{i\alpha}u_\alpha+2B^2c_{i\alpha}c_{i\beta}u_\alpha u_\beta+Bu^2- \\ 2B^2c_{i\alpha}u_\alpha u^2-\frac{4}{3}B^3c_{i\alpha}c_{i\beta}c_{i\gamma}u_\alpha u_\beta u_\gamma)\end{aligned} \tag{7}$$

where ρ, u and e denote the density,the flow velocity, and the internal energy, respectively. They are defined by the particle velocity and the distribution function. The coefficients F and B depend on the internal enagy. They are determined to satisfy some constraints in order to recover the compressible Navier Stokes equations.

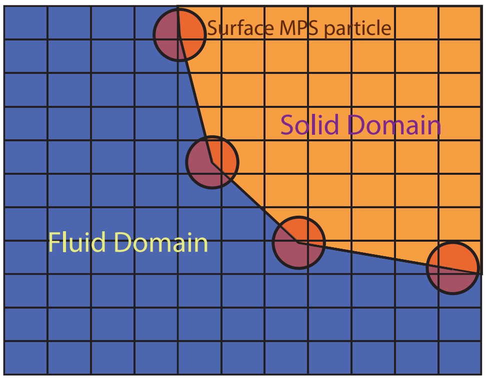


Figure 4: *The surface of the solid geometry .*

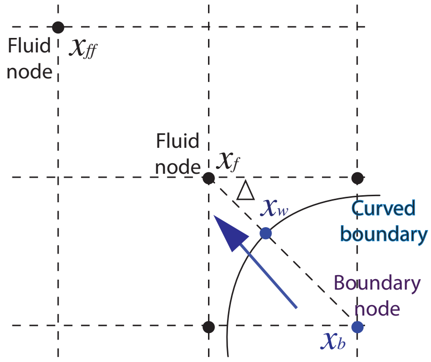


Figure 5: *Moving and curved boundaries.*

5 Elastics and Fluid Interaction

The fluid dynamics of the glottal flow is influenced by the presence of vocal folds, and in turn, vocal folds is diriven by air(Fluid)-induced force. My approch to solve the problem is coupling both FDLB and MPS methods.

Here, we define surface particles of the MPS particles located along the outside of the geometry. Thus, surface of vocal folds is defined connecting adjacent particles by the straight line as indicated in Figure 4. This surface becomes the interface between FDLBM and MPSM regime. In LBM, for boundaries involving complex geometries or those that can move, the bounce-back rule has been substantially improved by Mei et al.[3]. To handle complex moving and curved boundaries, we apply this method to compressible and thermal FDLBM by adding a few modifications, in which the boundary does not necessarily lies on the lattice nodes, and can deform during the simulation. As indicated in Figure 5, the packet distribution for a node, x_f, in the fluid adjacent to the boundary, is streamed from it neighbers. Thus, a fictitious packet ditribution, $f_i(t, x_b)$, is defined on the node x_b which lies just inside the object boudary. The fraction of the link that is intersected by the boundary in the fluid region is denoted by $\triangle = |x_f - x_w|/|x_f - x_b|$. The post-collision value of $f_i(t, x_b)$ is given by

$$f_i(t, x_b) = (1 - z)f_i(t, x_f) + zf_i^*(x_b) + 2B_q\rho e_i \cdot u_\omega \tag{8}$$

Here, the difinition of f_i^* is different from Mei2000[3] as,

$$\begin{aligned} f_i^* = F_i\rho(A_q + B_q e_i \cdot u_{bf} + C_q(e_i \cdot u_f)^2 + D_q(u_f)^2 \\ +E_q(e_i \cdot u_{bf})(u_f)^2 + F_q(e_i \cdot u_f)^3). \end{aligned} \tag{9}$$

A_q through F_q is the coefficient same as Eq.(7), and the third term of right hand side of Eq.(8) is an additional collision term, and for $\triangle > 1/2$,

$$u_{bf} = (1 - \frac{3}{2\triangle})u_f + \frac{3}{2\triangle}u_\omega \quad and \quad z = \frac{2\triangle - 1}{\tau + 1/2} \tag{10}$$

and for $\triangle < 1/2$,

$$u_{bf} = u_{ff} \quad and \quad z = \frac{2\triangle - 1}{\tau - 2} \tag{11}$$

Note that u_{bf} represents the virtual speed of the boundary node x_b in term of the fluid velocity, u_f at x_f, u_{ff} at node x_{ff}, and the boundary velocity, u_w at x_w. Here, as regards the velocity vectors that move more than one lattice node a time step, we can apply this method by dividing time step so that these vectors move by one lattice node.

On the other hand, to calculate the force of the fluid F_i acting on a MPS surface particle, we consider two separate sets of forces, those arising from hydrostatic pressure and those arising from dynamic forces due to fluid momentum. The forces due to hydrostatic pressure act normal to the surface and are generated by the incoherent motions of the fluid molecules against the surface. The dynamic forces are generated by the coherent motion of continuous fluid and can be divided into a force normal to the surface of the object and a tangential shearing force. Here, we neglect the tangential shearing force becouse it is negligible in comparision to the force due to hydrostatic pressure. The total force acting on the surface particle n are obtained by distributing linearly fluid pressure forces $P_T()$ multiplied by surface area and summing up the both side's as

$$F_T = -L_g(P_T(n, n-1)\vec{n}_{right} + P_T(n, n+1)\vec{n}_{left}) \tag{12}$$

in the case of two dimensional situation, where L_g denotes the length of perpendicular direction to the simulation plane of the geometry that is the length of vocal fold here, and $\vec{n}$ is the outward surface normal. The fluid pressure force is represented as

$$P_T(a, b) = \int_a^b (1 - \frac{|\vec{r}_s - \vec{r}_a|}{|\vec{r}_b - \vec{r}_a|})(p(r_s) - p_0 + \frac{1}{2}\rho(r_s)(\vec{v}_{rel}(r_s) \cdot \vec{n})^2)ds \tag{13}$$

where v_{rel}, $p()$, $\rho()$, p_0, and r is the velocity of fluid relative to the surface, fluid pressure and dencity on the surface, fluid static pressure and position, respectively. To obtain the value of each parameter on an arbitrary point on the surface, the two-dimensional Lagrangian interpolated polynomials formed by the Eularian mesh points are used. Thus, the parameter ϕ on the surface can be approximated by

$$\phi(s) = \sum_{ij}(\prod_{k=1, k\neq i}^{imax} \frac{X - x_{kj}}{x_{ij} - x_{kj}})(\prod_{m=1, m\neq j}^{jmax} \frac{Y - y_{im}}{y_{ij} - y_{im}})\phi(r_{ij}), \tag{14}$$

where $imax$ and $jmax$ are the maximum numbers of the mesh points in the x- and y-directions respectively, and $\phi(r_{ij})$, x_{ij}, y_{ij} are the parameter value, x-,y-position for the FDLBM Eularian mesh point (i, j), respectively. X, Y are the position of the point on the surface. It should be stressed that the above polynomials can be either centered around the boundary or ended at boundary. By adding F_T to surface particles as external forces, surface particles are updated at each time step.

6 Implementation

In this work, the calculation of MPS method is implemented on GPU using NVIDIA CUDA(Computed Unified Device Architecture). We had been able to implement this method by the appearance of CUDA. In past GPGPU techniques, the shader program of this method was too difficult to implement. In CUDA, usage of the selection of the various types of memory is the important key for an efficient implementation. Here, during calculation, linear and angular velocity of each particles are stored in the $shared\ memory$ which is $float4$ type, and calculated results are stored in the $global\ memory$ bound to PBO. The quantities of neighbor particles for each specific particles are stored in the $constant\ memory$. The initial distances between a particle and its neighbors, initial weight function, physical properties of each particle i.e., Young's modulus, Poisson rate, kind of the particle etc, neighbor list of all particles are stored in the $texture\ memory$.

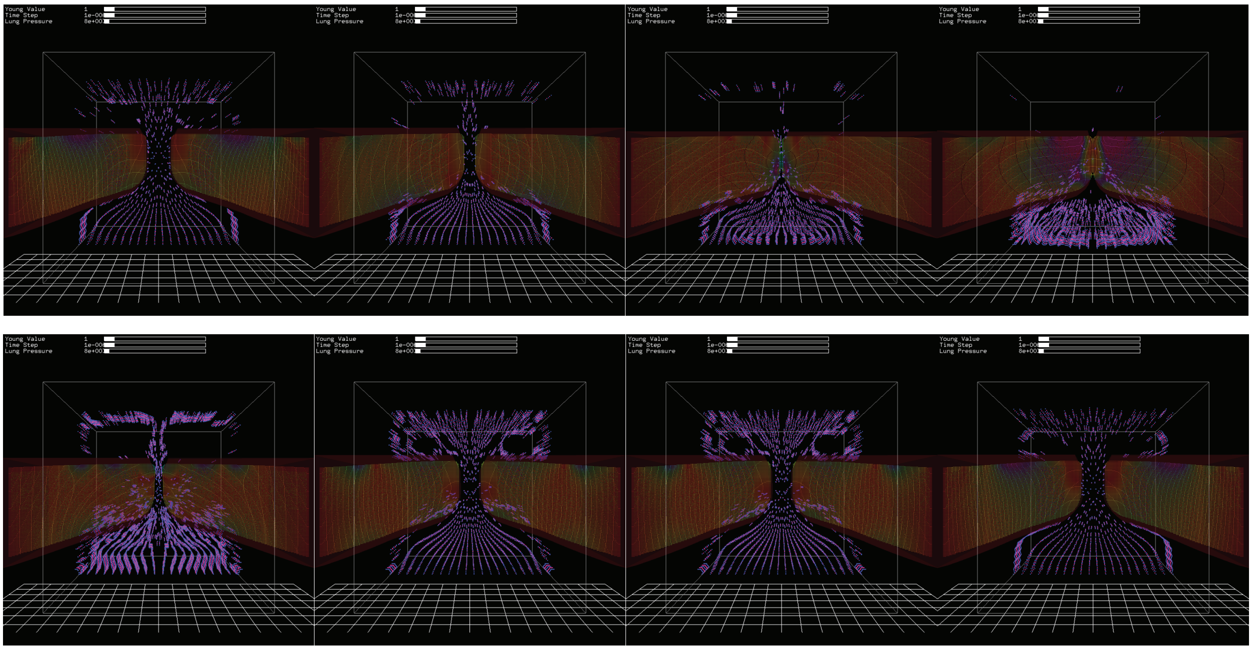


Figure 6: *Result.*

And other parameters are stored in the *global memory*. To calculate the third term of left hand side of Eq.(1), It is necessary to scan all particles two times. So, to syncronize between all particles, it is necessary to devide kernel into two passes. The CUDA kernel simplified pseude-code is shown below.The first pass is

```
//for each particle i
Fetch the parameters of particle i
and store in Shared Memory1&2;

for particles j in all neighbor particles{
   Fetch parameters of particle j;
   Calculate parameters between i and j;
   Calculate  laplacian model
   and add to Shared Memory2;
   Calculate relative distance
   and rotation matrix between i and j;
   Calculate divergence model
   add accel to Shered Memory1;
   Calculate angular accelation
   and store in Shared Memory1;
}

Multiply constant to Shared Memory1;
Calculate viscosity from Shared Memory2
and Add it to Shared Memory1;
Multiply constant to Shared Memory1;
Copy from current value of Shared memory1
to global memory;
```

After the syncronization between all particles, the second pass is excuted,

```
//for each particle i
Fetch result of first pass
from global memory to Shared Memory;
float2 temp=make_float2(0,0);
for particles j in all neighbor particles{
   Calculate parameters between i and j.
   Calculate gradient model
   and add the result to temp;
}
Multiply constant to temp
and add temp to Shard Memory;
Mutiply constant to Shared Memory;
if(kind of i==SURFACE){
Add external force from global memory
to Shared Memory;
}
update the linear and angular velocity of i;
update the position and angular value of i;
```

On the other hand, the calculation of FDLBM is implemented on GPU by NVIDIA cg to use voxelization techniques. Here, to apply D2Q21Model, six textures are needed. These textures are packed opposite vector to optimize. To create boundary voxel, depth peeling is used. This technique is proposed by Wei2003[7], that uses a generic voxelization algorithem of the boudaries using depth peeling, and extend it to a dynamic boudary generation method that converts any geometric boundary to LBM boundary nodes on-the-fly. We adopt Wei's Method basically. FDLBM simplified pseude-code is

```
Voxelization;

//for each node i
Calculate local energy, dencity
and velocity of node i ;
Calcalate Collision term & Propagetion term;
(update distribution function
 using finite difference scheme)
Apply boundary condition;
```

In FDLBM, Collsion term and Prapagation term are acted in a same shader that is different from LBM. The values of local energy and dencity, velocity are stored in PBO to pass to CUDA kernel.

Collision detection and its force calculation of MPS particle is implemented on CPU side simultaneously in parallel with GPU side using OS-dependent thread function Windows-Thread here. Futhermore, these calculation on CPU is parallelized by OpenMP. After calculatations by both GPU side and CPU side and their synchronization, the resulting data of the CPU side is transfer to GPU.

7 Result

Here, vocal folds ware assumed to be have a structure composed of two layers. that is, the inner layer consisted of the muscle and

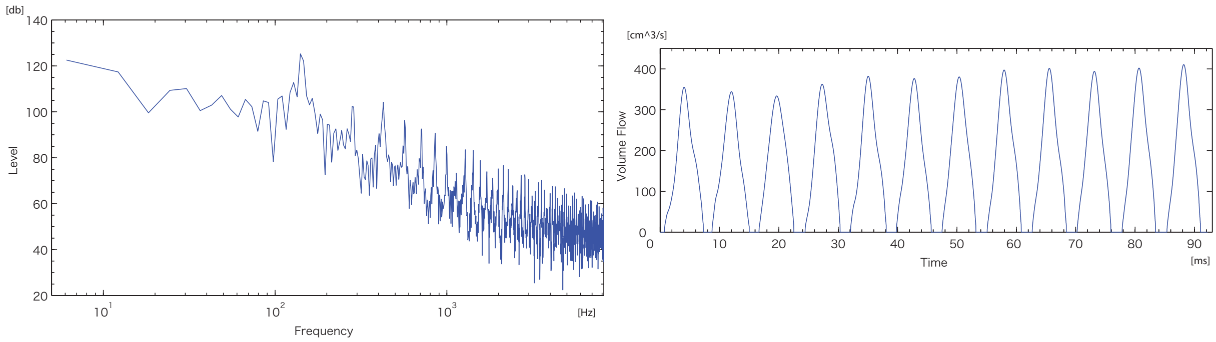


Figure 7: *The generated glottal wave. (Left::Spectrum, Right::Waveform).*

outer one cinsisted of soft tissues[9]. The MPS particles associated to each layer have different values of the Young modulus. These values are $10.0[dyn/cm^2]$ for the inner and $1.0[dyn/cm^2]$ for the outer. The time step is set to $1.0^{-5}[s]$.

To perform the simulation, 30000 MPS particles and 25000 regular FDLBM meshes ware used. To visualize, point sprite technique colored by the magnitude of pressure using GLSL is used for MPS particles, and particle trace technique is used for FDLBM fluid .

This program is excuted on a usual PC:
OS:: Microsoft Window XP SP2
CPU:: Intel Core 2 Quad 2.66 GHz
Graphics Card:: NVIDIA Geforce 8800 Ultra
Memory:: DDR2-6400 2Gbyte
Motherboard:: Asustek P5K.

As a result of rendering to the screen once every 10 calculation loops, about 60 frames per sec was obtained. It indicates that the value of each physical parameter can be changed during the simulation interactively. Though a strict bench-mark has not been performed, the computational speed increased more than about 400 times of the speed that is attainable by using CPU alone. The simulated behavior of thumbnails of result is shown in Figure 6. In Figure 6, the bottom and top sides respectively represents lung side and vocal tract side(mouse side).

In Figure 6, the glottal vortex can be captured to appear in the downstream of glottis. When both vocal folds are approached, a jet flow is generated. It is also shown that the vocal folds collide at the bottom side first, and then the collision of their topsides follows. This phenomenon is so called "phase difference" between top and bottom sides, and good agreement was obtained with previous studies.

Moreover, the fluid dynamic sounds is simulated directly. The generated volume flow, which is the source for the glottal wave, in the downstream of glottis is shown in figure 7, and sound pressure at that point is direct proportion to this glottal volume flow. In compressible and thermal FDLBM-D2Q21-Model, sound pressure p_s and sound speed c_s are defined as

$$p_s = \rho e - p_0, \quad c_s = \sqrt{2e}. \tag{15}$$

where p_0 denote the static pressure of air. In Figure 7, the waveform is shaped like triangle which is unique to the glottal flow. This result is good agreement with past expariments.

8 Conclusion and Future

In this work, to simulate the process of glottal wave generation directly, I have presented FDLB-MPS hybrid method by which the complex coupled dynamics between elastics and fluid. Futhermore, I implemented these calculation on GPU as shader and on CPU simultaneously in parallel effeiciently. As a result, flexible simulation method is successfully constructed for the complex coupled dynamics between the elastic body, which entails a large deformation, and fluid flow, which has a high Reynolds number. In addition, the fluid dynamic sounds generated by the vocal fold motion is successfully simulated. Futhermore, the simulation is excuted at interctive rate on a usual PC.

In the future, I will apply this model to the 3D situation, which is more difficult and complicated. It is also necesssaly to examine the numerical accuracy of the proposed method.

Acknowledgements

The work described in this report was fully suppoted by a grant from ADCDU and CDS Kaburagi Laboratry, Kyushu University, Japan.

References

[1]Interaction of fluids with deformable solids, Muller et al., Jornal of Computer Animation and Visual Worlds, Vol.15,2004
[2]Simultaneous coupling of fluids and deformable bodies, Chentanez et al., ACM SIGGRAPH, 2006
[3]Lattice Boltzmann method for 3d-flows with curved bounndary, Mei et al., Journal of Comp. Phys, 2000
[4]Moving-Particles Semi-implicit Method for Fragmentation of Incompressible fluid, Kosizuka et al., Nucl. Sci. Eng., 1996
[5]Dynamic Analysis of Elastic Solids by MPS Method, Song et al., Machine Society of Japan, 2005
[6]Direct simulation of fluid dynamic sounds by the finite difference lattice Boltzmann method, Tshutahara et al, WIT Press, 2007
[7]GPU-Based Flow Simulation with Complex Boundaries, Wei et al., GPU Gem 2, 2003