A demonstration of parametric resonance of the 2D elastic ring

Eleven (Shiyi) Chen sc7825@nyu.edu

December 16, 2021

Contents

1 Introduction

In this project, the ring is constructed using the zero-length Hookean springs, the ring is immersed in a square domain of fluid. This setup is analogous to a three-dimensional water balloon immersed in water. As for the dimension of the setup, the size of the square is 1 cm times 1 cm, the size of the fingernail. The boundary of the square is periodic, meaning that water flowing up out of the domain will flow into the domain in the bottom again. Similar for the left and right.

Immersed boundary method[3] is used to handle the interaction between the elastic ring and the ring. In the meanwhile, the dynamics of the fluid are solved using the spectral method. In the immersed boundary method, the fluid is in an Eulerian way. For two-dimensional fluid, we use the cartesian coordinate. For example, the velocity of the fluid can be written as $\mathbf{u}(x, y)$. On the other hand, the elastic ring is described in a Lagrangian way. Take a piece of elastic rubber line, for example, the rubber line stays at rest, we use a marker and a ruler to mark the equal-distance points, give each point a parameter, we call it θ. When the structure deforms, the marker signs still stay on the structure. In this way, the velocity of the structure can be written as $\mathbf{U}(\theta)$.

2 Equations

2.1 Equations in fluid

Naiver-Stoke equations for imcompressible fluid in 2d:

$$\begin{aligned}\rho(\partial_t u_x + \mathbf{u}\cdot\nabla u_x) &= -\nabla p + \mu\nabla^2 u_x + f_x(x,y)\\ \rho(\partial_t u_y + \mathbf{u}\cdot\nabla u_x) &= -\nabla p + \mu\nabla^2 u_y + f_y(x,y)\\ \nabla\cdot\mathbf{u} &= 0\end{aligned} \tag{1}$$

where $\mathbf{f} = (f_x, f_y)$ is the external force on the fluid, we will give an formula later, $\mathbf{u} = (u_x, u_y)$ is the velocity of the fluid at each point. p is the pressure in the fluid, ρ is the density of the fluid, and μ is the kinematic viscosity of the fluid.

The first two equations describe the dynamics of the fluid while the third one is the divergence-free condition for the incompressible fluid.

In this project, we will use cgs system (centimeter, gram, second). Also, we will use approximate parameters for water in room temperature, $\rho = 1\mathrm{g}\cdot\mathrm{cm}^{-3}, \mu = 0.01\mathrm{g}\cdot\mathrm{s}^{-1}\mathrm{cm}^{-1}$.

2.2 Equations of the structure(our ring)

We consider our ring as a set of nodes connected with Hookean springs with zero rest length. Consider our ring with location $\mathbf{X}(\theta)$ parameterized by $\theta(\theta\in[0,\pi])$, the tension $\mathbf{T}$ on our spring can be calculated as:

$$\mathbf{T} = T\mathbf{t} = k\frac{d\mathbf{X}(\theta)}{d\theta} \tag{2}$$

where X is the location vector of the structure, and $\mathbf{t}$ is the tangential vector, k is the stiffness of the spring. Due to our spring is massless, the tension on the spring must be balanced by the force of fluid excreting on the boundary(denote it by $-\mathbf{F}$.). By balance of force on a infinitesimal element, we have

$$\partial_\theta\mathbf{T} + (-\mathbf{F}) = 0 \tag{3}$$

Combining Equation 2 and Equation 3, we have:

$$\mathbf{F} = k\frac{\partial^2\mathbf{X}}{\partial\theta^2}. \tag{4}$$

2.3 Equations of fluid-structure interactions

As a convention, we will denote state of the structure by the upper case vector$(\mathbf{U},\mathbf{F},\mathbf{X})$, while the state of the fluid by the lower case vector$(\mathbf{u},\mathbf{f},\mathbf{x})$.

The action from the fluid to the structure is by imposing the non-slip condition, i.e.,

$$\mathbf{U}(\theta,t) = \mathbf{u}(\mathbf{X}(\theta,t),t) \tag{5}$$

Mathematically, it is equivalent to write it as

$$\partial_t\mathbf{X}(\theta,t) = \mathbf{U}(\theta,t) = \mathbf{u}(\mathbf{X}(\theta,t),t) = \int_{[0,L]^2}\mathbf{u}(\mathbf{x},t)\delta(\mathbf{x}-\mathbf{X}(\theta,t))d\mathbf{x}. \tag{6}$$

Where δ is the two dimensional delta function.

The action from the structure to fluid is by spreading the force on the structure ($\mathbf{F}$) back to the fluid ($\mathbf{f}$), this is done also with the same delta function as it is in Equation 6:

$$\mathbf{f}(\mathbf{x},t) = \int_0^{2\pi} \mathbf{F}(\theta,t)\delta(\mathbf{X}(\theta,t)-\mathbf{x})d\theta \tag{7}$$

An intuitive understanding of Equation 7 is to consider an arbitrary fluid domain that contains a piece of the structure. Denote this part of fluid by Ω, the total force this part of fluid feels from the structure can be calculated as:

$$\begin{aligned}\int_\Omega \mathbf{f}(\mathbf{x},t)d\mathbf{x} &= \int_\Omega \int_0^{2\pi} \mathbf{F}(\theta,t)\delta(\mathbf{X}(\theta,t)-\mathbf{x})d\theta d\mathbf{x} \\ &= \int_\Omega \int_{\{\theta|\mathbf{X}(\theta)\in\Omega\}} \mathbf{F}(\theta,t)\delta(\mathbf{X}(\theta,t)-\mathbf{x})d\theta d\mathbf{x}\end{aligned}$$

The above equation tells that for an arbitrary piece of fluid, the total force on the fluid is only determined by the structure within the domain of this piece of fluid.

3 Discretizations

In this section, we will discretize the equations we got in the last section. In practice, we do the discretize space and time sanctimoniously. Here, however, we do it in two steps just be more intuitive.

3.1 Spacial discretization

Suppose the size of the square is L. In each direction, there are N points(including the points on two boundaries). Thus, the point on the grid can be represented as $\mathbf{x} = (x,y) = (ih, jh)$, where $h = \dfrac{L}{N-1}$ is the mesh width. As for the structure, we assign the equal distance nodes at the initialization. Thus, $\theta = k\Delta\theta$. In the computer program, we set $\Delta\theta = 1$.

Now, we want to discretize the Equation 1, and we need to define the discretization to the spacial differential operators in the Navier Stokes equation. The partial derivative D_α on direction α is defined as:

$$D_\alpha \phi(\mathbf{x}) = \frac{\phi(\mathbf{x}+h\hat{e}_\alpha)) - \phi(\mathbf{x}-h\hat{e}_\alpha))}{2h}. \tag{8}$$

The gradient is defined as:

$$\mathbf{D} = (D_1, D_2) \tag{9}$$

The divergence is defined as:

$$\mathbf{D}\cdot\mathbf{\Phi} = D_1\phi_1 + D_2\phi_2 \tag{10}$$

The Laplacian operator L is defined as:

$$(L\mathbf{u})(\mathbf{x}) = \sum_{\alpha=1}^{2} \frac{\mathbf{u}(\mathbf{x}+h\hat{e}_\alpha) + \mathbf{u}(\mathbf{x}-h\hat{e}_\alpha) - 2\mathbf{u}(\mathbf{x}))}{h^2} \tag{11}$$

Also define the skew operator S as:

$$S(\mathbf{u})\phi = \frac{\mathbf{u}\cdot\mathbf{D}\phi + \mathbf{D}\cdot(\mathbf{u}\phi)}{2} \tag{12}$$

The convection term $\mathbf{u}\cdot\nabla\mathbf{u}$ can be approximated by $\mathbf{S}(\mathbf{u})u$ with its α elements defined as:

$$(S(\mathbf{u})\mathbf{u})_\alpha = \frac{\mathbf{u}\cdot\mathbf{D}u_\alpha + \mathbf{D}\cdot(\mathbf{u}u_\alpha)}{2} \tag{13}$$

Thus, the discrete version of Navier Stokes equations can be written as:

$$\begin{aligned} \rho(\partial_t \mathbf{u} + S(\mathbf{u})\mathbf{u}) &= -\mathbf{D}p + \mu \mathbf{L}\mathbf{u} + \mathbf{f} \\ \mathbf{D} \cdot \mathbf{u} &= 0 \end{aligned} \tag{14}$$

The spacial discretized equation for the spring can be written as:

$$\mathbf{F}(k\Delta\theta) = k\frac{\mathbf{X}((k-1)\Delta\theta) + \mathbf{X}((k+1)\Delta\theta) - 2\mathbf{X}(k\Delta\theta)}{\Delta\theta^2} \tag{15}$$

3.2 Temporal discretization

For accuracy, we will finish updating the velocity of the fluid and the location of the structure in two half steps:

In the first half step:

1. Update $\mathbf{X}^n$ to $\mathbf{X}^{n+1/2}$:

$$\mathbf{X}^{n+1/2}(k\Delta\theta) = \mathbf{X}^n(k\Delta\theta) + \Delta t/2 \sum_{\mathbf{x}\in g_h} \delta_h(\mathbf{x} - \mathbf{X}(k\Delta\theta))\mathbf{u}^n(\mathbf{x})h^2 \tag{16}$$

2. Update the force using the updated structure positions in this step:

$$\mathbf{F}^{n+1/2}(k\Delta\theta) = k\frac{\mathbf{X}^{n+1/2}(k-1)\Delta\theta + \mathbf{X}^{n+1/2}((k+1)\Delta\theta) - 2\mathbf{X}^{n+1/2}(k\Delta\theta)}{\Delta\theta^2} \tag{17}$$

3. Interpolate $\mathbf{F}^{n+1/2}$ to $\mathbf{f}^{n+1/2}$ using kernel δ_h:

$$\mathbf{f}^{n+1/2}(\mathbf{x}) = \sum_{i=0}^{N-1} \mathbf{F}^{n+1/2}(i\theta)\delta_h(\mathbf{X}^{n+1/2}(i\Delta\theta) - \mathbf{x})\Delta\theta \tag{18}$$

4. Calculate $\tilde{p}^{n+1/2}, \mathbf{u}^{n+1/2}$ by:

$$\begin{aligned} \rho(\frac{\mathbf{u}^{n+1/2-\mathbf{u}^n}}{\Delta t/2} + S(\mathbf{u}^n)\mathbf{u}^n) &= -\mathbf{D}\tilde{p}^{n+1/2} + \mu L\mathbf{u}^{n+1/2} + \mathbf{f}^{n+1/2} \\ \mathbf{D} \cdot \mathbf{u}^{n+1/2} &= 0 \end{aligned} \tag{19}$$

In the second half step, we have:

1. update $\mathbf{X}^n$ based on the new fluid velocity in the half step, i.e. $\mathbf{u}^{n+1/2}$:

$$\mathbf{X}^{n+1}(k\Delta\theta) = \mathbf{X}^n(k\Delta\theta) + \Delta t \sum_{\mathbf{x}\in g_h} \delta_h(\mathbf{x} - \mathbf{X}(k\Delta\theta))\mathbf{u}^{n+1/2}(\mathbf{x})h^2 \tag{20}$$

2. Calculate $\tilde{p}^{n+1}, \mathbf{u}^{n+1}$:

$$\begin{aligned} \rho\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + S(\mathbf{u}^{n+1/2})\mathbf{u}^{n+1/2} &= -\mathbf{D}p^{n+1/2} + \mu L(\frac{\mathbf{u}^n + \mathbf{u}^{n+1}}{2}) + \mathbf{f}^{n+1/2} \\ \mathbf{D} \cdot \mathbf{u}^{n+1} &= 0 \end{aligned} \tag{21}$$

4 Numerical methods

4.1 Solving dynamics in fluid: the spectral method

This part focuses on the solving Equation 19 and Equation 21, which are of the form:

$$
\begin{aligned}
(I - \frac{\Delta t \mu}{2\rho} L)\mathbf{u} + \frac{\Delta t}{\rho}\mathbf{D}q = w \\
\mathbf{D} \cdot \mathbf{u} = 0
\end{aligned}
\tag{22}
$$

with $q, \mathbf{u}$ unknown.

The idea to solve Equation 22 is to take the DFT on both sides of Equation 22 and match the coefficient term by term. The purpose of doing this is that we can term the differential equation into an algebraic equation. But the first step, we need to find out the corresponding differential operators in Fourier space.

Under the definition of our partial derivative as in Equation 8, we have:

$$
\begin{aligned}
(\mathbf{D}_1 f)(\mathbf{x}) &= \frac{f(\mathbf{x} + h\hat{e}_1) - f(\mathbf{x} - h\hat{e}_1)}{2h} \\
&= \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} \left(\frac{\exp(\frac{2\pi i}{L_0} m_1 h) - \exp(-\frac{2\pi i}{L_0} m_1 h)}{2h} \right) \exp(\frac{2\pi i}{L_0}(m_1 x + m_2 y)) \hat{f}_{m_1,m_2} \\
&= \sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} \left(\frac{i}{h} \sin(\frac{2\pi h}{L_0} m_1) \right) \exp(\frac{2\pi i}{L_0}(m_1 x + m_2 y)) \hat{f}_{m_1,m_2}
\end{aligned}
\tag{23}
$$

Since $f(\mathbf{x})$ can be represented as $\sum_{m_1=0}^{N-1} \sum_{m_2=0}^{N-1} \exp(\frac{2\pi i}{L_0}(m_1 x + m_2 y)) \hat{f}_{m_1,m_2}$, the coefficient in front of it is what we are looking for. Thus,

$$
\hat{\mathbf{D}}_1 = \frac{i}{h} \sin(\frac{2\pi h}{L_0} m_1). \tag{24}
$$

Similarly,

$$
\hat{\mathbf{D}}_2 = \frac{i}{h} \sin(\frac{2\pi h}{L_0} m_2). \tag{25}
$$

Similar to the partial derivative, for the Laplacian operator, we have:

$$
\hat{L} = -\sum_{\alpha=1}^{2} \frac{4}{h^2} \left(\sin(\frac{\pi h m_\alpha}{L_0}) \right)^2 \tag{26}
$$

As such, Equation 22 can be written in the Fourier space as:

$$
\begin{aligned}
(1 - \frac{\Delta t \mu}{2\rho} \hat{L})\hat{\mathbf{u}} + \frac{\Delta t}{\rho}\hat{\mathbf{D}}\hat{q} = \hat{\mathbf{w}} \\
\hat{D} \cdot \hat{\mathbf{u}} = 0
\end{aligned}
\tag{27}
$$

Taking $\mathbf{D}\cdot$ on the both side of the first equation, by the div zero condition, we have:

$$
\hat{q} = \frac{\hat{\mathbf{D}} \cdot \hat{\mathbf{w}}}{\frac{\Delta t}{\rho} \hat{\mathbf{D}} \cdot \hat{\mathbf{D}}} \tag{28}
$$

Bring Equation 28 back to Equation 27, we have:

$$
\hat{\mathbf{u}} = \left(\hat{\mathbf{w}} - \frac{\hat{\mathbf{D}}(\hat{\mathbf{D}} \cdot \hat{\mathbf{w}})}{\hat{\mathbf{D}} \cdot \hat{\mathbf{D}}} \right) / (1 - \frac{\Delta t \mu}{2\rho} \hat{L}) \tag{29}
$$

In this project, we only need to compute $\hat{\mathbf{u}}$.

4.2 Interaction: constructing delta function by imposing interpolation conditions

One of the most important step for the iteration is to choose an appropriate δ_h during the iteration between the structure and fluid. Let

$$\delta_h(\mathbf{x}) = \frac{1}{h^2}\phi(\frac{x}{h})\phi(\frac{y}{h}) \tag{30}$$

For the normal four points kernel, we impose the following conditions:

1. ϕ is continuous
2. $\phi(r) = 0$ for $|r| \geq 2$
3. $\sum_{i \text{ even}} \phi(r-i) = \sum_{i \text{ odd}} \phi(r-i) = 1/2$ for all r
4. $\sum_i (r-i)\phi(r-i) = 0$ for all r
5. $\sum_i (\phi(r-i))^2 = C$ for all r

For the normal six points kernel, we impose the following conditions:

1. ϕ is continuous
2. $\phi(r) = 0$ for $|r| \geq 3$
3. $\sum_{i \text{ even}} \phi(r-i) = \sum_{i \text{ odd}} \phi(r-i) = 1/2$ for all r
4. $\sum_i (r-i)\phi(r-i) = 0$ for all r
5. $\sum_i (r-i)^2\phi(r-i) = 0$ for all r
6. $\sum_i (r-i)^3\phi(r-i) = 0$ for all r
7. $\sum_i (\phi(r-i))^2 = C$ for all r

The resulting recipe for ϕ is the piece-wise polynomials which satisfy the conditions above. Figure 1 shows the normal four points kernel and the normal six points kernel and their derivatives. As we can see, they are $\mathcal{C}^1$ functions. Apart from the normal kernels, some other kernels outperform in volume conservation and divergence free[1].

5 Implementation

This project uses the implementation from Charles S. Peskin. The original code can be found at https://www.math.nyu.edu/~peskin/ib_lecture_notes/ib_matlab_2D.tar. The main program is listed below:

```
% ib2D.m
% This script is the main program.
global dt Nb N h rho mu ip im a;
global kp km dtheta K;
initialize
init_a
for clock=1:clockmax
  K = 1+sin(clock/20);
  XX=X+(dt/2)*interp(u,X); % update the structure (half step)
  ff=spread(Force(XX),XX); % update the fluid force term (half step)
  [u,uu]=fluid(u,ff);      % update the fluid by solving Navier stoke
  X=X+dt*interp(uu,XX);    % update the stucture using new fluid and new structure

  %animation:
  vorticity=(u(ip,:,2)-u(im,:,2)-u(:,ip,1)+u(:,im,1))/(2*h);
  contour(xgrid,ygrid,vorticity,values)
  hold on
  plot(X(:,1),X(:,2),'ko')
```

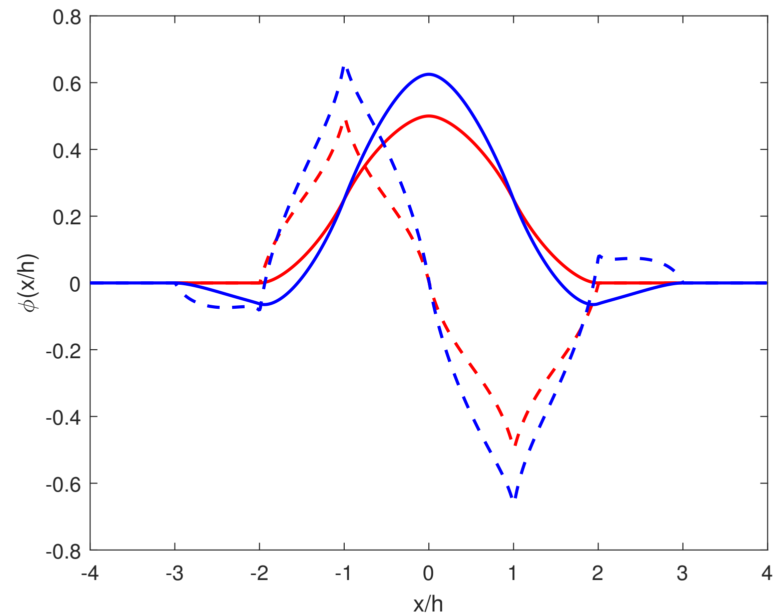

Figure 1: Comparison between normal four points kernel(red) and normal six points kernel(blue)

```
    plot(X([1,11,21,31,41,51],1),X([1,11,21,31,41,51],2),'r.')
    axis([0,L,0,L])
    caxis(valminmax)

    axis equal
    axis manual
    drawnow
    hold off
end
```

In Matlab, the velocity is represented by a three-dimensional array `u`. `u(:,:,1)` is the velocity for the first dimension while `u(:,:,2)` is the velocity for the second dimension. They are both of dimension N by N, where N is the size of the grid. Similar to the spring projects we did during the first half of the semester, the location of the structure is represented by `X`. The size of `X` is `Nb` by 2, where `Nb` is the number of the structure points.

The first line of code varies the stiffness periodically. As we will later show that without varying the stiffness, the structure will not vibrate. The following four lines of code are doing the two half steps as proposed in Section 3.2. After updating the fluid and the structure, the following line moves the structure to the center of the frame so that the structure does not slide away.

For the animation part, the contour lines of the vorticity is plotted. In the continuous case, the vorticity ω in 2D is defined as:

$$\omega = \nabla \times \mathbf{u} = \partial_y u_x - \partial_x u_y. \tag{31}$$

The vorticity describes how much the fluid rotates at a certain point. The purpose of plotting vorticity is that it follows the fluid and thus can show the velocity field.

6 Result

6.1 Comparison with the kernels

In this project, both the four points kernel and the six points kernel are implemented and compared. As it turns out, the four points kernel outweighs the six-point one as it is more stable and does a greater job in volume conservation. Thus, the four points kernel is used in the rest of the project. In Figure 2, the two kernels are compared at a time equal to 20 seconds when stiffness does not change periodically and velocity is been added externally to the system. As we can see from the plot, the six points kernel shrinks a lot compared to the four points kernel version.

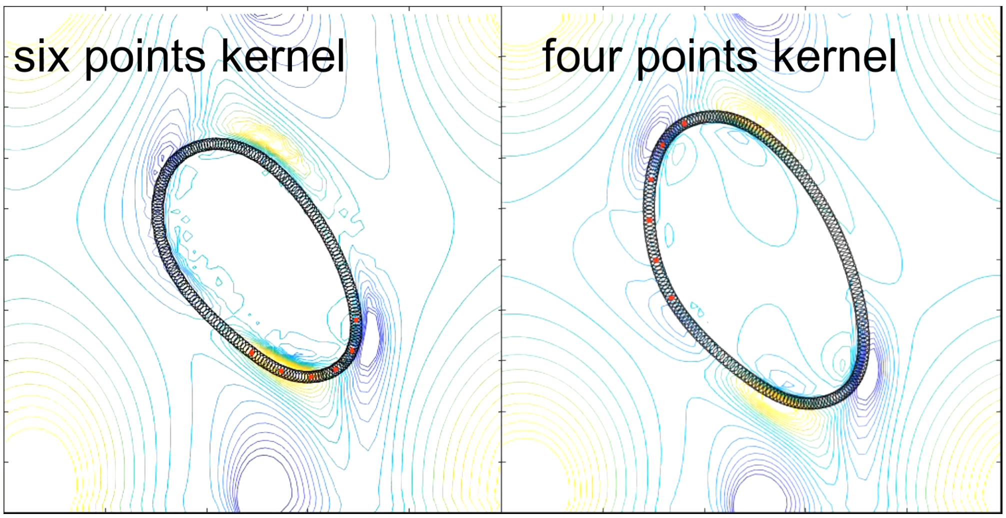

Figure 2: The comparison with the two kernels at time 20 seconds

6.2 Parametric resonance

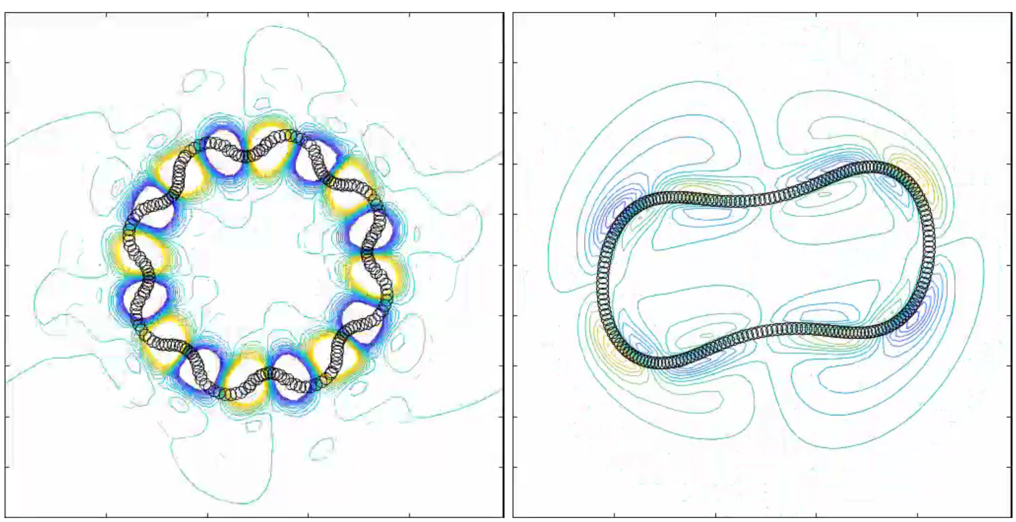

Figure 3: When the period of the stiffness equals to 0.0628 and 0.5027 seconds

Next, the period of the stiffness is changed. For most of the period of the stiffness, the structure stops vibrating as the initial kinetic energy of the structure is consumed by the viscosity.

However, there exist some periods for the stiffness such that the structure keeps vibrating in a certain mode. In Figure 3, two patterns of vibration are shown with stiffness periods corresponding to 0.0628 and 0.5027 seconds. The corresponding videos corresponding to the figure can be found in the Appendix.

The complete analysis for the parametric resonance can be found at [2]. An understanding of the periodic resonance is that when the stiffness is constant, one can identify different modes of vibrations corresponding to a different set of frequencies(dispersion relationship). This analysis can be done easier if one ignores the convection term and the viscosity term in the NS equation and suppose the tension on the ring is constant(not the case in our project). In the constant tension two-dimensional case, the dispersion relationship is:

$$\omega^2 = \frac{n(n^2-1))F}{2\rho a^3} \quad (32)$$
[1]

$\omega = \frac{2\pi}{T}$, T is the period, F is the constant tension in the ring. n is the wave number and a is the radius of the ring.

An interesting thing happens when we change the stiffness of our ring periodically. When the period we choose for the stiffness coincides with one of the eigenfrequencies of the system as mentioned above, parametric resonance will happen. That is the reason why although there is damping in our system but the ring continues vibrating.

7 Conclusion

In this project, we demonstrate the parametric resonance of a 2D elastic ring using the existing implementation of immersed boundary method. We first introduce our problem immersed boundary method. The procedures of the immersed boundary method are given in the following sections. Finally, the modes of vibration are shown in the result section.

References

[1] Yuanxun Bao, Jason Kaye, and Charles S Peskin. “A Gaussian-like immersed-boundary kernel with three continuous derivatives and improved translational invariance”. In: *Journal of Computational Physics* 316 (2016), pp. 139–144.

[2] Ricardo Cortez et al. “Parametric resonance in immersed elastic boundaries”. In: *SIAM Journal on Applied Mathematics* 65.2 (2004), pp. 494–520.

[3] Charles S Peskin. “The immersed boundary method”. In: *Acta numerica* 11 (2002), pp. 479–517.

Appendices

Comparison between six points kernel and four points kernel: https://www.youtube.com/watch?v=tPh0aOQVk4E
Stiffness = 1+sin(frames/1): https://www.youtube.com/watch?v=7U6c7nruMYc
Stiffness = 1+sin(frames/8): https://www.youtube.com/watch?v=jULYi8TCSbw
Stiffness = 1+sin(frames/20): https://www.youtube.com/watch?v=TdczTUTGHV4

[1]Deduction can be found at https://www.math.nyu.edu/~peskin/heartnotes/CLN-Peskin1975-2.pdf