Imperial College London

MSc Individual Project

Imperial College London

Department of Aeronautics

Bifurcation and Oscillation Effects of Gyrotactic Swimming Microorganism Suspension in Vertical Pipe

Author:
Songrui LI

Supervisor:
Dr. Yongyun Hwang

Second Marker:
Dr. Georgios Rigas

October 3, 2022

Abstract

The term "Gyrotactic" represents the swimming behaviour of certain types of microorganisms in suspension subjected to gravity and shear torque. The interesting collective behaviour is shown in the vertical pipe flow, that is, the cells swim toward the centre and the wall in the downward flow and the upward flow, respectively. Instabilities arise from this effect and lead to bifurcations.

In this project, based on the unsteady solver, the bifurcation was studied under two conditions of fixed flow and prescribed pressure gradient. In fixing the flow rate, bifurcations are found matched the previous results made by [Kes86b] and [FBH20]. Regarding the pressure gradient, one bifurcation related to the initial flow conditions is found. And the explanation is given from the perspective of combined influence on the velocity profile of cell distribution and pressure gradient.

The unsteady solver is also utilised to investigate the flow with the time-period pressure gradient. starting from a derived linearized state-space system, the effects of different parameters on the system are studied. And found "diverge gap" in certain Reynolds numbers, and proposed an explanation.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr. Yongyun Hwang and Mr. Lloyd Fung. They arranged a meticulous schedule in advance, checked my progress every week, discussed the problems I encountered and shared their professional ideas and suggestions, gave me confidence in academics during difficult times.

I would like to thank my family for their great support, allowing me to study in such a beautiful place. I would also like to thank my girlfriend Jingyi for encouraging me and staying with me throughout the Masters, especially during the COVID-19 pandemic.

Contents

List of Figures

List of Tables

Chapter 1

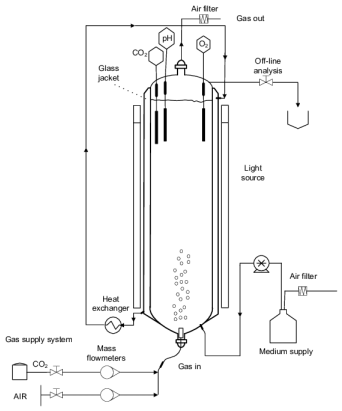
Introduction

1.1 Motivation

In order to resolve the contradiction between increasing energy demand and environmental degradation, sustainable (clean) energy is increasingly attractive. Among them, bioenergy, especially algae fuel, has broad prospects due to its wide availability, carbon neutrality and high yield. Using organic materials such as garbage and manure wastewater, microorganisms (such as Chlamydomonas) can transfer solar energy as a natural part of photosynthesis to biomass oil, with a harvest cycle of 1-10 days. In addition, the U.S. Department of Energy estimates that if the technical difficulties are solved, the production of microalgae biomass oil can replace all petroleum fuel needs of America, with a requirement of 15,000 square miles, only 0.42% of the US land area[GLS+10]

Today, algae cultivation facilities are mainly photobioreactors and open ponds. Although the capital and operating costs are relatively higher, the former method is easier to control and less susceptible to contamination, so it is more suitable for research. Figure1.1a shows a typical outdoor vertical photobioreactor built by researchers[HWZ+18] and Figure1.1b shows the structure of the system [MTB+09]. They are composed of transparent tubes made of glass or plastic, with rigid supports, and equipped with external circulation and thermoregulation devices, and the pH value of the environment is adjusted by CO2 dosage. In each tube, a vast number of swimming microorganisms interact with each other and the aqueous environment. A better understanding of this reciprocal influence has an attractive potential in industry, the energy input of a bioreactor can therefore be considerably minimised[Bee20].

(a) Vertical algae bioreactor system[HWZ+18]



(b) Photobioreactor diagram [MTB+09]

Figure 1.1: Photobioreactor system

The hydrodynamics involved in this interaction are highly related to the orientation mechanics of the microorganisms in suspension. Each type of microorganisms has a specified directed locomotion in respond to external stimuli, known as "taxis". For example, geotaxis refers to the response to gravity; phototaxis, an orientation to light; rheotaxis, to viscous torques, chemotaxis and magnetotaxis to chemical concentration gradients and magnetic field, respectively.

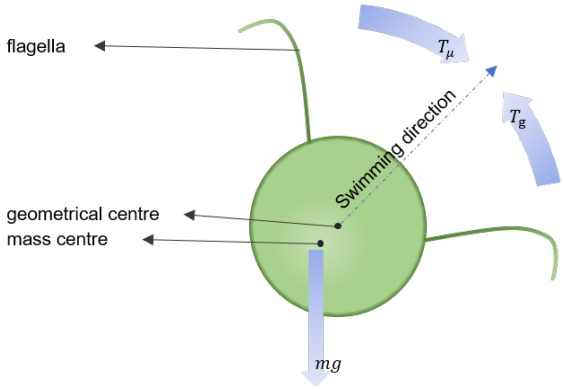
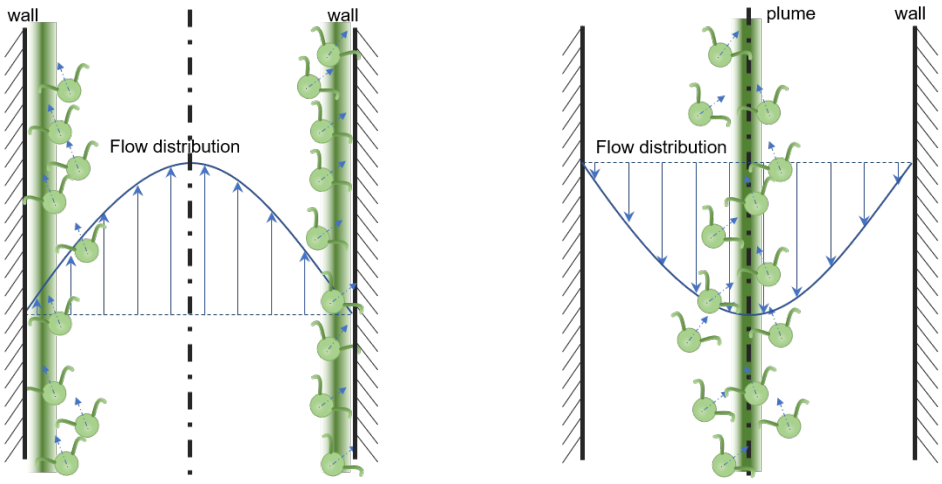


Figure 1.2: Individual behavior due to gyrotaxis effect, T_μ refers to viscous torque and T_g refers to the gravitational torque due to the offset of mass centre that the microorganism experienced. The microorganism is assumed to be a sphere with 2 flagella

Among them, gyrotaxis [Kes84][Kes85a][Kes85b][Kes86a] describes the directional response to both gravity and vortical flow (i.e. the combined motile of geotaxis and rheotaxis). It can be seen in several types of microalgae, such as Chlamydomonas, Dunaliella and Heterosigma, whose centroids deviate from their geometric centres due to their physiological structure. In vertical pipe flow, the resultant gravitational torque is balanced with the horizontal viscous torque it bears. As shown in Figure1.2, this joint effect produces a tilted swimming direction to the vertical. As a result, collective horizontal locomotion manifests itself under specific flow conditions, moving away from upflowing and toward downflowing areas of the fluid. For example, as shown in Figure1.3, a beam of cells called plume is generated in the downflow pipe. On the contrary, the upflow fluid guides the cells to form a ring of high cell concentration near the wall. Furthermore, the non-uniformity of the horizontal concentration caused by the collective motion will cause the inhomogeneous density distribution of the suspension, which will affect the flow velocity distribution.



(a) upflow

(b) downflow

Figure 1.3: Schematic diagram of collective buoyant convection of gyrotaxis micro-algae in upward and downward poiseuille flow

1.2 Literature Review and Objective

The collective locomotion of gyrotaxis suspension in upward flow is first documented by Childress, Lavandowsky & Spiegel [CLS75]. And Kessler documented the downflow circumstance experimentally in [Kes85a]&[Kes86b]. In Kessler's articles, he also developed a set of 1D steady-state equations to describe the system. He first described the swimming capability of the microorganisms by a diffusion term and derived an equation for cell concentration[Kes85a]. And then defined a buoyant constant to describe the cell/fluid coupling intensity and inserted the buoyancy term to

the original steady Naiver-Stokes equation. Then combine the two equations to modelling these collective phenomena [Kes86b]. He also deduced a closed-form theoretical solution for vertical pressure gradient $p_z = 0$. Further steady numerical modelling methods are taken in the basis of it, such as the Fokker-Plank Equation provided by Pedley and Kessler[PK90], which estimates the swimming orientation p.d.f.(probability density function) of microorganisms and Generalised Taylor Dispersion (GTD) theory [HB02] & [MF03], which estimate the collective orientation from the p.d.f. of a single tracer.

As for the instability and bifurcation of the gyrostatic plume emerging in a downward pipe flow. Kessler found that under specific circumstances, the resultant focused beam is prone to instability in a form of regular-spaced high cell concentration regions, referred to blips [Kes85b] & [Kes86b]. In the basis of this, recently, utilising the Fokker-Plank Equation, Hwang and Pedley found a similar blip-like instability in vertical downflow channel flow[HP14]. And Fung, Bearon, and Hwang [FBH20] study the same topic using Fokker-Plank Equation, GTD model and linearised model. By seeking the steady solutions under different Richardson number and basic flow rate, they found different saddle-node bifurcations at low and high flow rates. In their study, the value of central velocity and cell concentration is documented in line charts with Richardson number as the horizontal axis. In high flow rates, a saddle-node bifurcation point is found at $Ri = 59.86$ for GTD model and 57.68 for the linearised model. In low flow rates, a saddle-node bifurcation emerges from $Q = 0$ state.

One of the objectives of this project is to follow Fung, Bearon, and Hwang's bifurcation study[FBH20]. First, develop an unsteady model based on Kessler's 1986 version[Kes86b], find the bifurcation under each fixed flow rate condition and compare the result with L.F.'s work[FBH20]. The second is to modify the model to fix the pressure gradient p_z, to find the bifurcation under each prescribed p_z. Another goal is to use the unsteady solver to simulate the time-periodic flow condition, to investigate the performance of the system under sinusoidal pressure gradient oscillations.

1.3 Outline of the Thesis

In addition to the introduction and conclusion&evaluation of the first and last chapters, this report is divided into three parts. In chapter 2, the modelling method, non-dimensionalisation and numerical scheme are reported as methodology. Chapter 3 discusses the bifurcations when specifying flow and pressure gradient separately. Chapter 4 describes and explains the diverge gap found in pulsating flow.

Chapter 2

Methodology

2.1 The Conservation Law of Cells

2.1.1 Cell Characteristics and Assumptions

In this article, as shown in Figure1.2, we assume that microorganisms are spherical with two sets of flagella on both sides of the axis and have the same radius of a^*. And their centroids have a uniform rearward offset L^* relative to each geometric centre. They swim at an average velocity of V_c^*, along the direction of each cell axis, that is, on the same line as the geometrical and gravity centres.

According to Kessler's estimation[Kes86b], $a^* \approx 5 \times 10^{-4} cm$, $L^* \approx 10^{-5} cm$, $V_c^* \approx 10^{-2} cm\ s^{-1}$. Besides,the density of the microalgaes ρ_c^* is estimated 5% greater than the flow density ρ_w^*.

2.1.2 Gyrotaxis Equations

The direction of a gyrotaxis cell is determined by the balance between the gravitational and shear moments it subjected in a flow. Based on the spherical assumption, the balanced equation can be described as:

$$\mathbf{T}_\mathbf{g}^* + \mathbf{T}_\mu^* = m^* g L^* + 4\pi\mu a^* 3\left(\nabla \times \mathbf{U}^* a^* - 2\mathbf{\Omega}^*\right) = 0, \tag{2.1}$$

with $\mathbf{T}_\mathbf{g}^*$ and $\mathbf{T}_\mu^*$ as the gravitational and shear torques respectively, μ as viscosity coefficient, $\mathbf{U}$ as flow velocity, $\mathbf{\Omega}^*$ as flow vorticity.

As a result, in a vertical flow of velocity of u^*. With the radial coordinate $\mathbf{r}^*$, the radial velocity of swimmers $\mathbf{V}_\mathbf{cr}^*$ can be described as:

$$\mathbf{V}_\mathbf{cr}^* = \beta^* \frac{d\mathbf{u}^*}{d\mathbf{r}^*}, \tag{2.2}$$

where

$$\beta^* = -\frac{4\pi\mu a^* 3}{m^* g L^*} \tag{2.3}$$

The coefficient β^* is introduced by Kessler [Kes86b] as a length scale modelling the gyrotactic behaviour of swimmers.

The random swimming behaviour is summarised as diffusion coefficient D^*. The radial cell flux J^* is then described as:

$$J^* = \mathbf{n}^* \mathbf{V}_\mathbf{cr}^* - D^* \frac{\partial \mathbf{n}^*}{\partial \mathbf{r}^*}, \tag{2.4}$$

The balanced cell conservation equation is shown below, and there is a detailed derivation process in Appendix A. n^* represents the cell concentration.

$$\frac{\partial n^*}{\partial t} + \nabla \cdot (n^* U^*) = 0 \tag{2.5}$$

On the basis of the continuity equation of incompressible flow $\nabla \cdot \mathbf{U}^* = 0$, plus the flux term, the cell conservation law can be described as:

$$\frac{\partial \mathbf{n}^*}{\partial t} + \mathbf{U}^* \nabla \cdot \mathbf{n}^* = -\frac{1}{\mathbf{r}^*} \frac{\partial}{\partial \mathbf{r}^*} \left[\mathbf{r}^* \left(J^* \right) \right] \tag{2.6}$$

In a vertical flow, that is, $\mathbf{U}^* = \mathbf{u}^*(\mathbf{r})$, $\mathbf{U}^* \nabla \cdot \mathbf{n}^*(\mathbf{r}, \mathbf{t})$ term is eliminated, and replace J with Equation2.4, the cell conservation equation is:

$$\frac{\partial \mathbf{n}^*}{\partial t} = -\frac{1}{\mathbf{r}^*} \frac{\partial}{\partial \mathbf{r}^*} \left[\mathbf{r}^* \left(\beta^* \mathbf{n}^* \frac{\mathbf{du}^*}{\mathbf{dr}^*} - D^* \frac{\partial \mathbf{n}^*}{\partial \mathbf{r}^*} \right) \right] \tag{2.7}$$

In practice, a compensation term is plugged in to improve the cell number conservation against the numerical cell number lost.

$$\frac{\partial \mathbf{n}^*}{\partial t} = -\frac{1}{\mathbf{r}^*} \frac{\partial}{\partial \mathbf{r}^*} \left[\mathbf{r}^* \left(\beta^* \mathbf{n}^* \frac{\mathbf{du}^*}{\mathbf{dr}^*} - D^* \frac{\partial \mathbf{n}^*}{\partial \mathbf{r}^*} \right) \right] + K_p (N_0^* - N^*) n^* \tag{2.8}$$

where the total number of cells is:

$$N^* = \int_0^1 \mathbf{n}^*(\mathbf{r}) \mathbf{r} d\mathbf{r} \tag{2.9}$$

2.2 Momentum Equation with Buoyant Term

In a vertical pipe flow, the Naiver-Stokes equations can be simplified as below in a cylindrical coordinate:

$$\mu \frac{\partial \mathbf{u}^*}{\partial t} = -p_z^* + \rho_w^* g + \mu \left[\frac{1}{\mathbf{r}^*} \frac{\partial}{\partial \mathbf{r}^*} \left(\mathbf{r}^* \frac{\partial \mathbf{u}^*}{\partial \mathbf{r}^*} \right) \right] \tag{2.10}$$

The density of the suspension ρ^* is:

$$\rho^* = \rho_c^* n^* v^* + \rho_w^* (1 - n^* v^*) = \Delta \rho^* n^* v^* + \rho_w^* \tag{2.11}$$

with v^* as the volume of a single cell and the density difference $\Delta \rho^* = \rho_c^* - \rho_w^*$.
Replace ρ_w^* in Eq.2.10 with Eq.2.11, the momentum equation is then become:

$$\mu \frac{\partial \mathbf{u}^*}{\partial t} = \mu \left[\frac{1}{\mathbf{r}^*} \frac{\partial}{\partial \mathbf{r}^*} \left(\mathbf{r}^* \frac{\partial \mathbf{u}^*}{\partial \mathbf{r}^*} \right) \right] - (p_z^* - (\rho^* - \Delta \rho^* n^* v^*) g) \tag{2.12}$$

Define $P_z^* = p_z^* + \rho^* g$, Eq.2.12 can be then written as:

$$\frac{\partial \mathbf{u}^*}{\partial t} = \frac{1}{\mathbf{r}^*} \frac{\partial}{\partial \mathbf{r}^*} \left(\mathbf{r}^* \frac{\partial \mathbf{u}^*}{\partial \mathbf{r}^*} \right) - \left(\frac{P_z^*}{\mu} + \frac{\Delta \rho^* v^* g}{\mu} n^* \right) \tag{2.13}$$

Kessler concludes the cell-concentration/fluid-velocity coupling coefficient as the buoyant constant α^* before n^*. With

$$\alpha^* = -\frac{\Delta \rho^* v^* g}{\mu} \tag{2.14}$$

The momentum equation of vertical pipe flow in cylindrical coordinate is finally:

$$\frac{\partial \mathbf{u}^*}{\partial t} = -\frac{P_z^*}{\mu} + \alpha^* n^* + \frac{1}{\mathbf{r}^*} \frac{\partial}{\partial \mathbf{r}^*} \left(\mathbf{r}^* \frac{\partial \mathbf{u}^*}{\partial \mathbf{r}^*} \right) \tag{2.15}$$

2.3 Mesh and Boundary Conditions

The mesh is set to be one dimensional, equal-spaced, from the center of the pipe to the pipe wall covering half of the pipe. Symmetric boundary conditions are deployed at the centre. And for the wall boundary conditions, an impenetrable boundary condition is deployed for flow velocity, and the no-flux condition is set for that of cell concentration.

As a result, for cell concentration law,

$$\left(\frac{\partial \mathbf{n}^*}{\partial \mathbf{r}^*} \right)_{r^*=0} = 0, \quad \textit{at centre} \tag{2.16}$$

$$(J^*)_{r^*=R} = -\mathbf{n}^*\beta\left(-\frac{d\mathbf{u}^*}{d\mathbf{r}^*}\right)_{r^*=R} - D\left(\frac{\partial \mathbf{n}^*}{\partial \mathbf{r}^*}\right)_{r^*=R} = 0, \quad at\ wall \tag{2.17}$$

and for momentum equation:

$$\left(\frac{\partial \mathbf{u}^*}{\partial \mathbf{r}^*}\right)_{r^*=0} = 0, \quad at\ centre \tag{2.18}$$

$$(\mathbf{u}^*)_{r^*=R} = 0\ , \quad at\ wall \tag{2.19}$$

2.4 Parameters and Non-dimensionalisation

Symbol	Name	Scale	Dimensional	Non-Dimensional	
R	Radius of the tube	$\mathbf{L}$	$R^* = 1\ cm$	$R = R^*/R^*$	1
V_c	Velocity of swimming cells	$\mathbf{L/t}$	$10^{-2}\ cm/s$	V_c^*/V_c^*	1
ρ_w	Density of water	$\mathbf{m/L^3}$	$1\ \ gm/cm^3$	ρ_w^*/ρ_w^*	1
n_1	Average concentration	$\mathbf{N/L^3}$	average cell concentration $cells/\mathrm{cm}^{-3}$	n_1^*/n_1^*	1

Table 2.1: Characteristic parameters

Symbol	Name	Scale	Dimensional	Non-Dimensional	
v	Volume per cell	$\mathbf{L^3}$	$5\times10^{-10}cm^3$	$\frac{v^*}{R^{*3}}$	5×10^{-10}
μ	Dynamic viscosity	$\mathbf{m/Lt}$	$10^{-2}\ gm/(cm\ s)$	$\frac{\mu^*}{\rho^* R^* V_c^*}$	5×10^{-10}
β	Gyrotactic Constant	$\mathbf{L}$	$\frac{C\mu V_c}{\rho_C g L} = 5\times10^{-2}\ cm$	$\frac{\beta^*}{R^*}$	5×10^{-2}
α	Negative Buoyancy	$\mathbf{L^4/t^2}$	$\frac{\Delta\rho}{\rho}vg = 2.5\times10^{-8}\ cm^4/s^2$	$\frac{\alpha^*}{R^{*2}V_c^{*2}}$	2.5×10^{-4}
D	Diffusion	$\mathbf{L^2/t}$	$5\times10^{-4}\ cm^2/s$	$\frac{D^*}{R^* V_c^*}$	5×10^{-2}
ν	Kinematic viscosity	$\mathbf{L^2/t}$	$\frac{\mu}{\rho} = 10^{-2}cm^2/s$	$\frac{\nu^*}{R^* V_c^*}$	1
Re	Reynolds number	1	$\frac{\rho \mathrm{R} \mathrm{V}_c}{\mu} = 1$	Re	1
γ	Self-focusing scale	$\mathbf{L}$	$\frac{Re\ \alpha\beta}{8D} = 3\times10^{-5}\ cm$	$\frac{\gamma^*}{R^*}$	3×10^{-5}
Q_0	flow rate	$\mathbf{L^3/t}$	$10^{-2}\ cm^3/s$	$\frac{Q0^*}{R^{*3}/t^*}$	1

Table 2.2: Non-dimensionalized parameters

Symbol	Name	Scale	Definition
t	Time	$\mathbf{t}$	$\frac{t^*}{R^*/V_c^*}$
u	Velocity	$\mathbf{L/t}$	$\frac{u^*}{V_c^*}$
$r(R)$	Radius	$\mathbf{L}$	$\frac{r(R)}{R^*}$
N	Cell number	$\mathbf{n}$	$\frac{N^*}{n_1^* R^{*3}}$
n	Concentration	$\mathbf{n/L^3}$	$\frac{n^*}{n_1^*}$
P	Pressure	$\mathbf{m/t^2L}$	$\frac{P^*}{\rho^* V_c^{*2}}$

Table 2.3: Non-dimensionalized variables

The parameters are non-dimensionalized by characteristic parameters shown in table2.1. And other non-dimensionalized parameters and variables are shown in table2.2and 2.3.

As a result, Equation2.8 and 2.13 are dimensionlessly converted to:

$$\frac{\partial \mathbf{u}}{\partial t} = P_z + \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left(\mathbf{r}\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right)\right] + Ri\mathbf{n} \tag{2.20}$$

$$\frac{\partial \mathbf{n}}{\partial t} = -\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left[\mathbf{r}\left(-\beta\mathbf{n}\left(-\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right) - D\frac{\partial \mathbf{n}}{\partial \mathbf{r}}\right)\right] + K_p(N_0 - N)\mathbf{n} \tag{2.21}$$

where $Ri = \mu\alpha$, referred as Richardson number [Hun98].

2.5 Constant Flow Rate Equations

In this project, in addition to fixing the pressure gradient like Kessler's work in[Kes86b], a prescribing flow rate approach is also used like Fung's work[FBH20]. In order the determine the flow rate Q, after multiplying r on both sides of Eq.2.20, integrate :

$$\int_0^R \mathbf{r}\frac{\partial \mathbf{u}}{\partial t} = \frac{1}{2}R^2 P_z + \frac{1}{Re}\int_0^R \frac{\partial}{\partial \mathbf{r}}\left(\mathbf{r}\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right)d\mathbf{r} + \int_0^R Ri\mathbf{r}\mathbf{n}dr \tag{2.22}$$

Insert the boundary conditions Eq.2.18, Eq.2.22 becomes:

$$\int_0^R \frac{\partial \mathbf{r}\mathbf{u}}{\partial t}dr = \frac{1}{2}R^2 P_z + \frac{1}{Re}R\left[\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right]_{\mathbf{r}=R} + Ri\int_0^R \mathbf{r}\mathbf{n}d\mathbf{r} \tag{2.23}$$

The flow rate Q can be determined by the integration of velocity u:

$$Q = \int_0^R u\,(r)\,rdr = Constant, \tag{2.24}$$

which means the LHS of the integrated equation Eq.2.23 is 0:

$$\frac{1}{2}R^2\frac{\partial p}{\partial z} + \frac{1}{Re}R\left[\frac{\partial u}{\partial r}\right]_{r=R} + Ri\int_0^R \mathbf{r}\mathbf{n}dr = 0, \tag{2.25}$$

with the definition of total cell number

$$N = \int_0^R \mathbf{r}\mathbf{n}dr, \tag{2.26}$$

being plugged into Eq.2.25, the NS equation for constant flow rate is then written into:

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{2}{Re}\frac{1}{R}\left[\frac{\partial u}{\partial r}\right]_{r=R} - \frac{2Ri}{R^2}N + \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left(\mathbf{r}\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right)\right] + Ri\mathbf{n} \tag{2.27}$$

2.6 Numerical Method

2.6.1 Discrete Scheme for Fixing Flow Rate

Discretized Control Equations

Let us recall the equations for the fixed flow rate Eq.2.27 Eq.2.8:

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{2}{Re}\frac{1}{R}\left[\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right]_{r=R} - \frac{2Ri}{R^2}N + \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left(\mathbf{r}\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right)\right] + Ri\mathbf{n} \tag{2.27}$$

$$\frac{\partial \mathbf{n}}{\partial t} = -\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left[\mathbf{r}\left(-\beta\mathbf{n}\left(-\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right) - D\frac{\partial \mathbf{n}}{\partial \mathbf{r}}\right)\right] + K_p(N_0 - N)\mathbf{n} \tag{2.21}$$

The discrete scheme of each function is:

$$\frac{\mathbf{u}^{j+1} - \mathbf{u}^j}{\Delta t} = -\frac{2}{Re}\frac{1}{R}\mathbf{K}\mathbf{u}^{j+1} - \frac{2Ri}{R^2}N^j + \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\mathbf{K}_1\ \left((\mathbf{K_3}\mathbf{r})\mathbf{K_2}\mathbf{u}^{j+1}\right)\right] + Ri\mathbf{n}^j \tag{2.28}$$

$$\frac{\mathbf{n}^{j+1} - \mathbf{n}^j}{\Delta t} = -\frac{1}{\mathbf{r}}\mathbf{K_1}\ \beta(\mathbf{K_3}\mathbf{r})(\mathbf{K_3}\mathbf{n}^j)(\mathbf{K_2}\mathbf{u}) + \frac{1}{\mathbf{r}}\mathbf{K_1}\ D(\mathbf{K_3}\mathbf{r})\mathbf{K_2}\mathbf{n}^{j+1} + K_P(N_0 - N)\mathbf{n}, \tag{2.29}$$

where,

$$\mathbf{K} = \frac{1}{\Delta r}\begin{bmatrix} 0 & \cdots & \frac{1}{2} & -2 & \frac{3}{2} \\ \vdots & \cdots & \vdots & \vdots & \vdots \end{bmatrix}, \quad \mathbf{K_3} = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ & & & & 2 \end{bmatrix} \tag{2.30}$$

$$\mathbf{K_1} = \frac{1}{\Delta r}\begin{bmatrix} -1 & 1 & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix}, \quad \mathbf{K_2} = \frac{1}{\Delta r}\begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & -1 & 1 \end{bmatrix} \tag{2.31}$$

Matrix $\mathbf{K}$ estimates the gradient of velocity at the wall $\left[\frac{\partial u}{\partial r}\right]_{r=R}$ and $\mathbf{K_3}$ estimates the first-order middle point interpolation values, i.e. $r_{i+1/2} = r_i + r_{i+1}/2$ for variable r. Matrix $\mathbf{K_2}$ and $\mathbf{K_3}$ stands for first-order backward and forward discritization.

As a result, the scheme can be concluded as:

$$\mathbf{G}\mathbf{u}^{j+1} = \mathbf{u}^j + Ri\mathbf{n}^j - \frac{2Ri}{R^2}N^j \tag{2.32}$$

$$\mathbf{H}\mathbf{n}^{j+1} = \mathbf{n}^j + \mathbf{n}'^j + \mathbf{K_{comp}}\mathbf{n}^j \tag{2.33}$$

where

$$\mathbf{G} = \mathbf{I} + \Delta t\frac{2}{Re}\frac{1}{R}\mathbf{K} - \Delta t\frac{1}{Re}\frac{1}{\mathbf{r}}\mathbf{K}_1(\mathbf{K_3 r})\mathbf{K_2} \tag{2.34}$$

$$\mathbf{H} = \mathbf{I} - \Delta t\frac{1}{\mathbf{r}}\mathbf{K_1}\ D\left(\mathbf{K_3 r}\right)\mathbf{K_2} \tag{2.35}$$

$$\mathbf{n}'^j = -\Delta t\frac{1}{\mathbf{r}}\mathbf{K_1}\ \beta(\mathbf{K_3 r})\left(\mathbf{K_3}\mathbf{n}^j\right)\left(\mathbf{K_2 u}\right) \tag{2.36}$$

$$\mathbf{K_{comp}} = \Delta t K_p(N_0 - N) \tag{2.37}$$

Numerical Integration

The total cell number and flow rate are calculated by:

$$N = \sum_{i=1}^{end-1}(\mathbf{K_3}N_i)(\mathbf{K_3}r_i)[\mathbf{dr}, \mathbf{0}] \tag{2.38}$$

$$Q = \sum_{i=1}^{end-1}(\mathbf{K_3}u_i)(\mathbf{K_3}r_i)[\mathbf{dr}, \mathbf{0}] \tag{2.39}$$

Add 0 at the end of $\mathbf{dr}$ because the last node is not needed in this scheme.

Discretized Boundary Conditions

Recall the boundary conditions Eq.2.16, 2.17, 2.18, and 2.19. Embedded in the matrices, replace the rows of $\mathbf{G}$ and $\mathbf{H}$ matrix as:

$$\mathbf{G}\ (1) = \begin{bmatrix} -3/2 & 2 & -1/2 & 0 \cdots \end{bmatrix} \tag{2.40}$$

$$\mathbf{G}\ (end) = \begin{bmatrix} \cdots & \cdots 0 & 1 \end{bmatrix} \tag{2.41}$$

$$\mathbf{H}\ (1) = \begin{bmatrix} -\frac{137}{60} & 5 & -5 & \frac{10}{3} & -\frac{5}{4} & \frac{1}{5} & 0 \cdots \end{bmatrix} \tag{2.42}$$

$$\mathbf{H}\ (end) = \begin{bmatrix} \cdots 0 & -\frac{1}{5} & \frac{5}{4} & -\frac{10}{3} & 5 & -5 & \frac{137}{60} + \frac{\beta\Delta r}{D}\mathbf{K_4 u} \end{bmatrix}, \tag{2.43}$$

$$where\ \mathbf{K_4} = \frac{1}{\Delta r}\begin{bmatrix} 0 & \cdots & 1/2 & -2 & 3/2 \end{bmatrix} \tag{2.44}$$

Initial Conditions

The initial condition of the system is set as:

$$\mathbf{u^0} = \frac{Q_0\pi^2}{2\pi - 4}cos(\frac{\pi\mathbf{r}}{2}) \tag{2.45}$$

$$\mathbf{n^0} = 1 \tag{2.46}$$

The Eq.2.44 aims to control $\mathbf{u_0}$ in a cosinusoidal shape and so that initial flow rate is equal to Q_0.

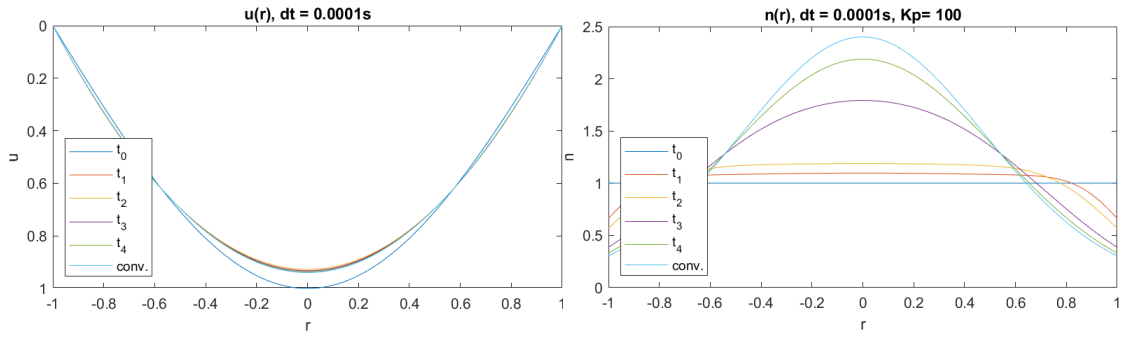
Iteration Steps

The iteration steps are shown below:

Calculate $\mathbf{u^{j+1}}$	$\mathbf{G u^{j+1}} = \mathbf{u^j} + \Delta t \mathbf{n^j} - \Delta t \frac{\mathbf{2Ri\alpha}}{\mathbf{R^2}} \mathbf{N^j}$
Update matrix $\mathbf{H^{j+1}}$	$\mathbf{H}^{j+1}\ (end) = \begin{bmatrix} \cdots 0 & -\frac{1}{5} & \frac{5}{4} & -\frac{10}{3} & 5 & -5 & \frac{137}{60} + \frac{\beta \Delta r}{D} \mathbf{K_4 u}^{j+1} \end{bmatrix}$
Update nonlinear $\mathbf{n\prime}$	$\mathbf{n'}^j = -\Delta t \frac{1}{\mathbf{r}} \mathbf{K_1}\ \beta (\mathbf{K_3 r})(\mathbf{K_3 n}^j)(\mathbf{K_2 u})$
Update $\mathbf{K_{comp}}$	$\mathbf{K_{comp}} = \Delta t K_P\ (N_0 - N)$
Calculate $\mathbf{n}$	$\mathbf{H n}^{j+1} = \mathbf{n}^j + \mathbf{n'}^j + \mathbf{K_{comp}} \mathbf{n}^j$

Table 2.4: iteration steps

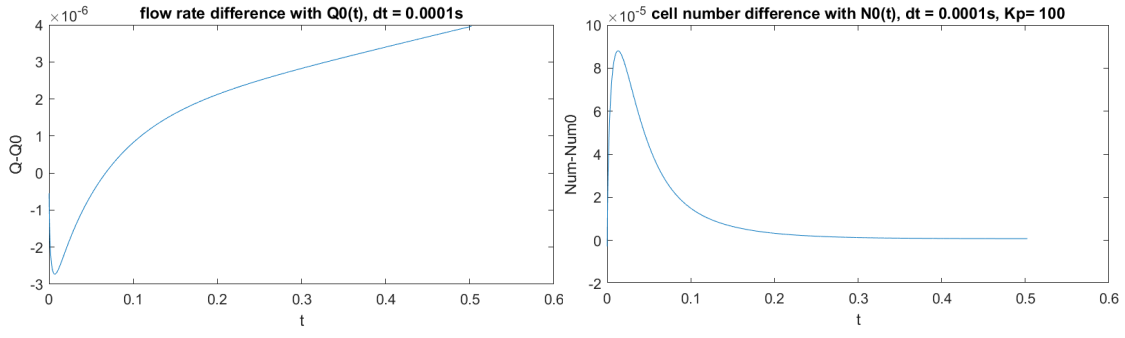
Validation of Code Conservation



(a) Cell concentration distribution

(b) Velocity distribution

Figure 2.1: Typical result of cell concentration and velocity distribution in down flow



(a) Cell number difference

(b) Total flow rate difference

Figure 2.2: Total flow rate and cell number differences with those for the initial state

In order to proof whether the integrated flow rate and cell number change with iteration, parameters are set consistent with that in table2.2 that is, $Ri = 2.5 \times 10^{-4}$, $D = 5 \times 10^{-2}$, $Re = 1$, $Q_0 = 1$. Additionally, set the time step $dt = 1 \times 10^{-4} s$, nodes number of the mesh $n_n = 200$ and compensation coefficient $K_p = 100$. Converged result and snapshots of the flow and cell profile are shown in Figure2.1a and 2.1b.

It can be seen from the snapshots that the plume effect appears gradually and the velocity profile converges faster to the state before the cell number, which is consistent with the theory.

Besides, the difference between the flow rate and the Q_0, and the difference between the number of cells and the initial number of cells n_0 with the iteration steps are also recorded in Figure2.2a and 2.2b.

It is clear to see the difference of cell number has a tendency to stabilise at a low value in the order of 10^{-5}, after a minor escalating then decreasing. And for the flow rate, the difference reach a negative peak before gradually flattening out. As a result, the code is conserved with time within an acceptable error range.

2.6.2 Discritization Scheme for Fixing Pressure Gradient

Discritized Control Equations

In order to include the cell-flow interaction effect while specifying the pressure gradient, relax the $-\frac{2}{Re}\frac{1}{R}\left[\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right]_{r=R}$ term in Eq.2.20 to be P_z and recall the Eq.2.21,

$$\frac{\partial \mathbf{u}}{\partial t} = P_z - \frac{2Ri}{R^2}N + \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left(\mathbf{r}\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right)\right] + Ri\mathbf{n} \tag{2.47}$$

$$\frac{\partial \mathbf{n}}{\partial t} = -\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left[\mathbf{r}\left(-\beta\mathbf{n}\left(-\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right) - D\frac{\partial \mathbf{n}}{\partial \mathbf{r}}\right)\right] + K_p(N_0 - N)\mathbf{n} \tag{2.21}$$

Use the same definition of $\mathbf{K}$ matrices in Eq.2.30 2.31, the discrete scheme is then:

$$\frac{\mathbf{u}^{j+1} - \mathbf{u}^j}{\Delta t} = P_z - \frac{2Ri}{R^2}N + \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\mathbf{K_1}\ \left((\mathbf{K_3 r})\mathbf{K_2}\mathbf{u}^{j+1}\right)\right] + Ri\mathbf{n}^j \tag{2.48}$$

$$\frac{\mathbf{n}^{j+1} - \mathbf{n}^j}{\Delta t} = -\frac{1}{\mathbf{r}}\mathbf{K_1}\ \beta(\mathbf{K_3 r})(\mathbf{K_3}\mathbf{n}^j)(\mathbf{K_2 u}) + \frac{1}{\mathbf{r}}\mathbf{K_1}\ D(\mathbf{K_3 r})\mathbf{K_2}\mathbf{n}^{j+1} + K_P(N_0 - N)\mathbf{n}, \tag{2.29}$$

Take the same boundary conditions as last section.

Initial Conditions

The initial conditions are set as:

$$\mathbf{u^0} = cos\left(\frac{\pi r}{2}\right) \tag{2.49}$$

$$\mathbf{n^0} = 1 \tag{2.50}$$

$\mathbf{u^0}$ is cosinusoidal and $\mathbf{n^0}$ is uniform.

2.6.3 Discrete scheme for Time-period Pressure gradient

Discritized Control Equations

Choose different frequency and amplitude. Choose initial condition $Q_0 = 0$ Set different alpha.

$$P_z = \hat{p}e^{i\omega t}, where\ \omega = \frac{2\pi}{T} \tag{2.51}$$

where A is the amplitude ω and T is the frequency and period respectively. Replace in 2.20:

$$\frac{\partial \mathbf{u}}{\partial t} = \hat{p}e^{i\omega t} - \frac{2Ri}{R^2}N + \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left(\mathbf{r}\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right)\right] + Ri\mathbf{n} \tag{2.52}$$

$$\frac{\partial \mathbf{n}}{\partial t} = -\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left[\mathbf{r}\left(-\beta\mathbf{n}\left(-\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right) - D\frac{\partial \mathbf{n}}{\partial \mathbf{r}}\right)\right] + K_p(N_0 - N)\mathbf{n} \tag{2.21}$$

The same process of discritization is deployed as former sections.

Initial Conditions

The Q_0 is set at low magnitude, try to have a minor push of the system to stabilise the performance.

$$\mathbf{u^0} = \frac{Q_0\pi^2}{2\pi - 4}cos(\frac{\pi \mathbf{r}}{2}),\quad where\ Q_0 = 10^{-4} \tag{2.53}$$

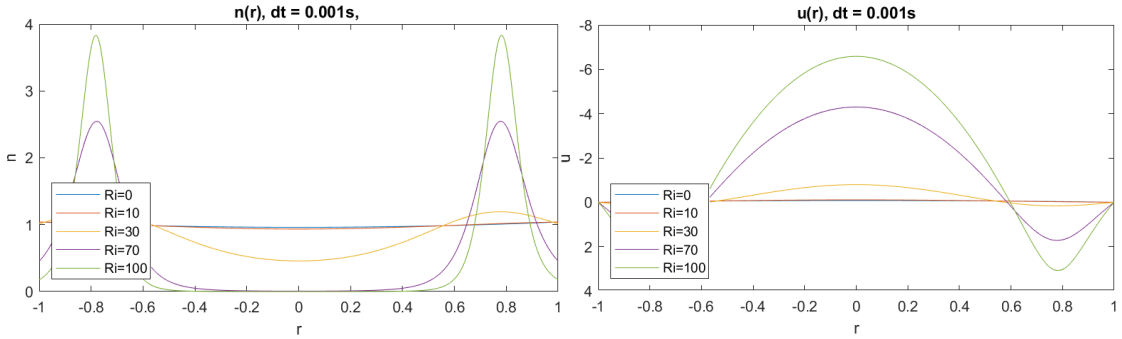
$$\mathbf{n^0} = 1 \tag{2.54}$$

Chapter 3

Bifurcation and Instability of Downflow and Upflow

3.1 Prescribing Flow Rate

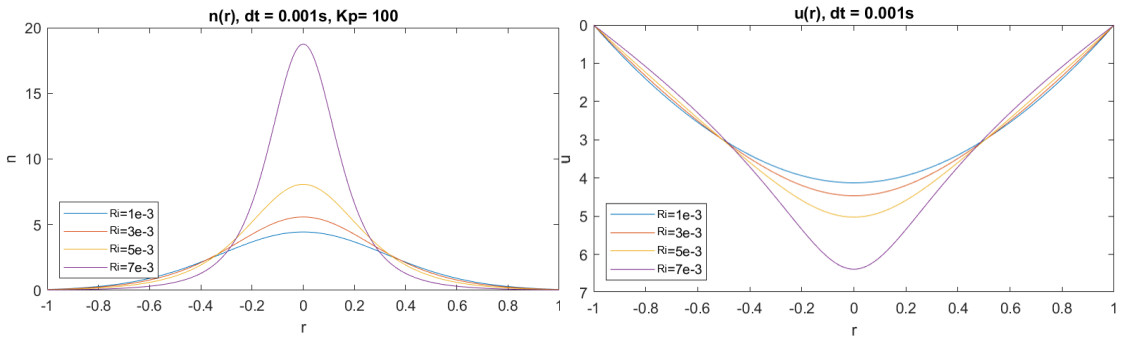
Change the Richardson number Ri and plot the the concentration and velocity distribution in Figure3.1a, 3.1b, 3.2a and 3.2b. It can be seen that in the downward flow, the higher Ri, the higher the central concentration and velocity. And in upward flow, the position of the high concentration peak coincide with the location of the peak reverse flow velocity and doesn't change with the Ri. In either case, when Ri increases to the critical Ri, the code blows up with the profiles becoming considerable sharp, which will be explained further.



(a) Cell concentration with Ri. As Ri increases, more cells move to the wall

(b) Velocity distribution with Ri. As Ri increases, reverse flow near the wall becomes more intense

Figure 3.1: Cell concentration and velocity distribution with Ri when $Q_0 = -2e - 2$.



(a) Cell concentration with Ri. As Ri increases, plume effect is more obvious

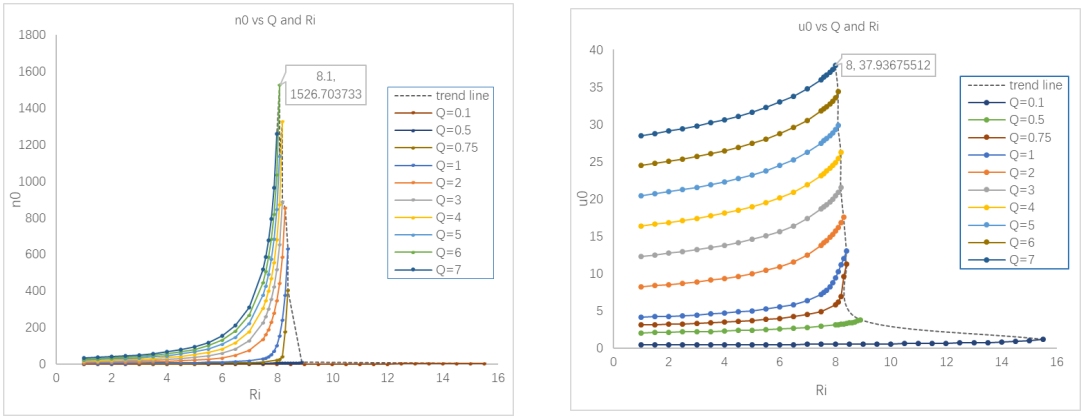
(b) Velocity distribution with Ri. As Ri increases, central velocity increases.

Figure 3.2: Cell concentration and velocity distribution with Ri in downward flow at $Q_0 = 1e - 1$.

3.1.1 Result of High Flow Rates

To take into account of the flow rate influence to the system, the central cell concentration $n(0)$ and velocity $u(0)$ for different Richardson number Ri and flow rate Q are documented and plotted with horizontal axis Ri in Figure3.4. Different lines stand for different flow rates Q.

As mentioned earlier, each condition(flow rate) has a critical Ri after which the system diverges. It can be also be seen from Figure3.3a that the critical Ri of each line has a saddle-noddle bifurcation at $Ri_{critical} = 8$. Take the definition of $\gamma = \frac{Re\ Ri\ \beta}{8D}$ in Kessler's article[Kes86b]. Through analysis, if $\gamma \to 1$, the system will analytically become unstable. The detailed derivation is in Appendix A. Replace $Ri_{critical} = 8$, the $\gamma_{critical} = 1$, which is consistent with Kessler's result.



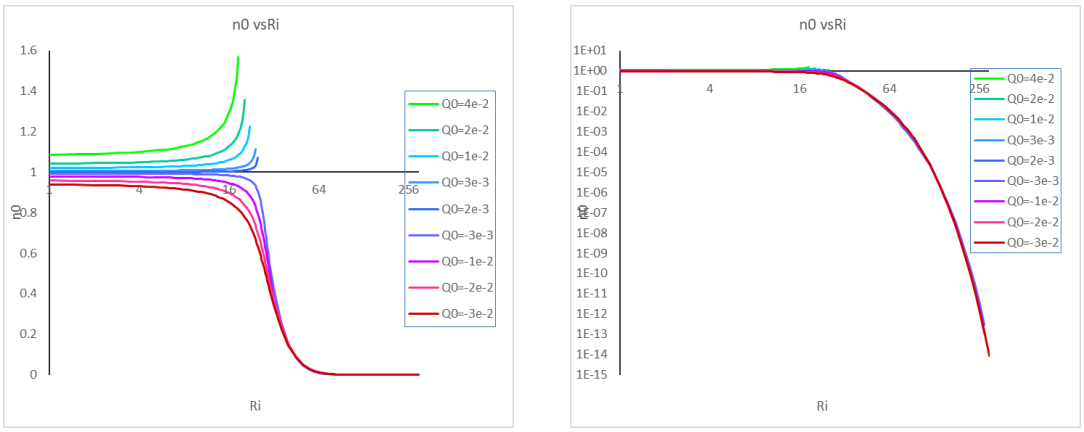
(a) Central cell concentration diagram

(b) Central velocity diagram

Figure 3.3: Central cell concentration and velocity with Ri and Q

3.1.2 Result of Low Negative and Positive Flow Rates

In order to explore the bifurcation at $Q = 0$. In the case of low flow rate, the same parameters are used except the flow rate range is set by $[-3 \times 10^{-2}, 4 \times 10^{-2}]$. Figure3.4 and 3.5 display the centreline cell concentration and velocity. It can be seen that the bifurcation Ri is around 25, which is consistent with Fung's work in [FBH20].



(a) Vertical normal axis

(b) Vertical logarithmic axis.

Figure 3.4: Central cell concentration with Ri

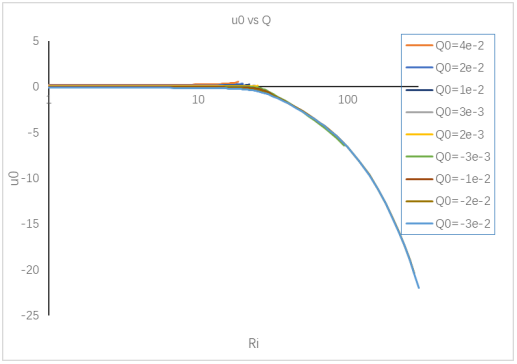


Figure 3.5: Central velocity concentration with Ri

3.2 Prescribing Pressure Gradient

3.2.1 Combine All Data into One Chart

In this section, the pressure gradient is fixed instead of the flow rate. Use different sets of initial flow rate and pressure gradients. 35 sets of data are combined. As shown in the Figure3.6 and 3.7, different marker colours represent different initial flow rate Q_0, and different line colours stand for different pressure gradients p_z. The "*" markers represent the end or start of each line.

Can be seen that

- all lines with a same p_z coincide with each other no matter what Q_0 is;
- for $p_z > 0$ $Q_0 \leq 0$, data always converge.
- for $p_z > 0$ $Q_0 > 0$, data diverge at one critical Ri. The critical Ri becomes bigger with smaller magnitude of Q_0;
- for $p_z < 0$ $Q_0 \geq 0$, data diverge at one critical Ri.
- for $p_z < 0$ $Q_0 < 0$, data diverge then come back to convergence, the divergence gap extended with bigger magnitude of p_z and smaller magnitude of Q_0;

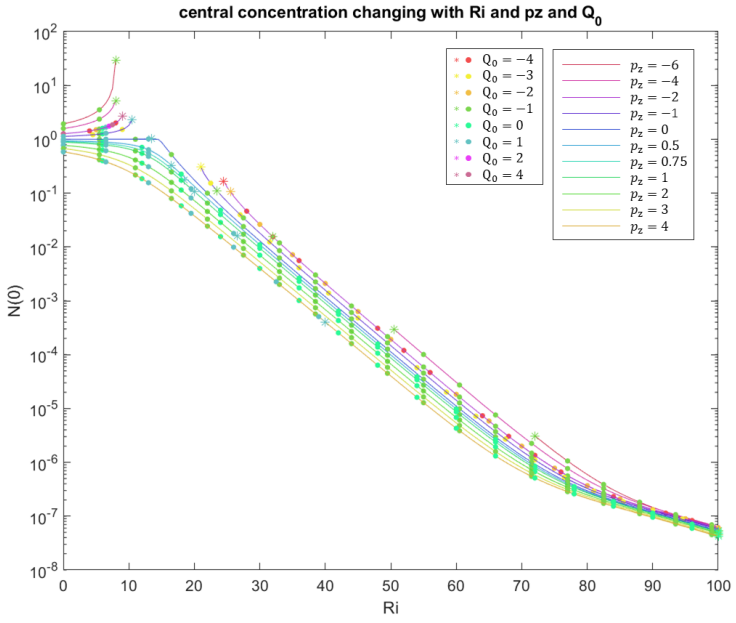


Figure 3.6: Centerline cell concentration changing with different Ri, pressure gradient and initial flow rate, with vertical axis in logarithmic axis.

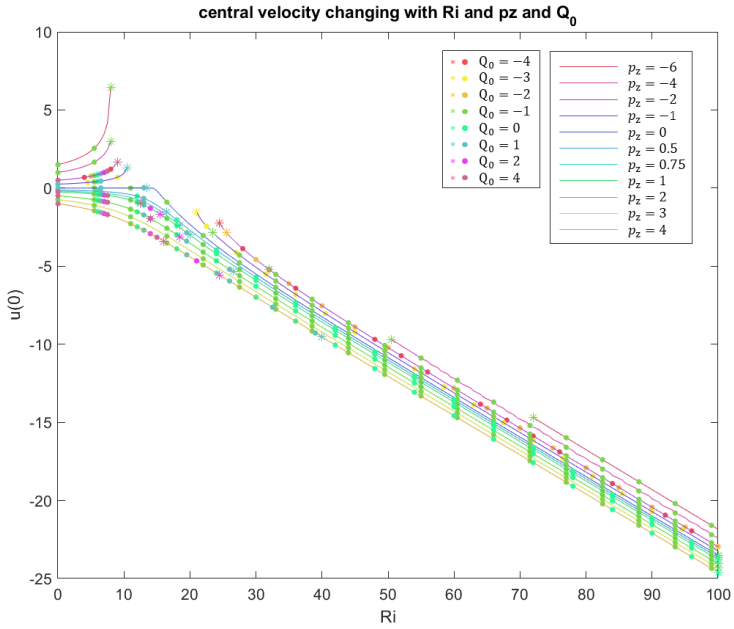


Figure 3.7: Central velocity changing with different Ri, pressure gradient and initial flow rate, with vertical axis in logarithmic axis

3.2.2 Look into How the Initial Condition Affect the Convergence Performance for Negative Pressure Gradient

Cell and Velocity Distribution Varying with Time

Take $p_z = -2$ and $Ri = 30$, set different value of Q_0, plot the cell concentration and velocity distribution in snapshots Figure3.8 to 3.13, 4 initial flow rate conditions are used, $Q_0 = -0.5, -1, -2, -4$.

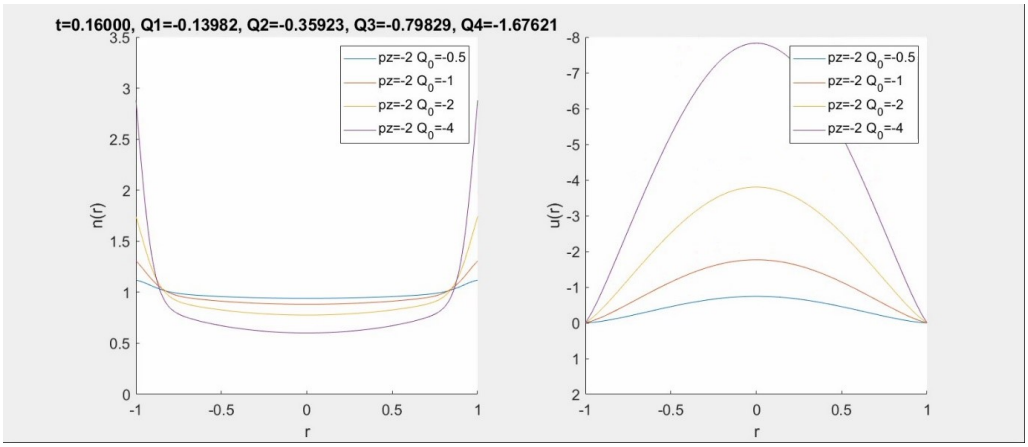


Figure 3.8: Snapshot1: In all cases, the cells immediately move to the wall. The higher the gradient on the wall, the greater the initial flow Q_0. Central velocities start to drop from the highest point related to the initial conditions.

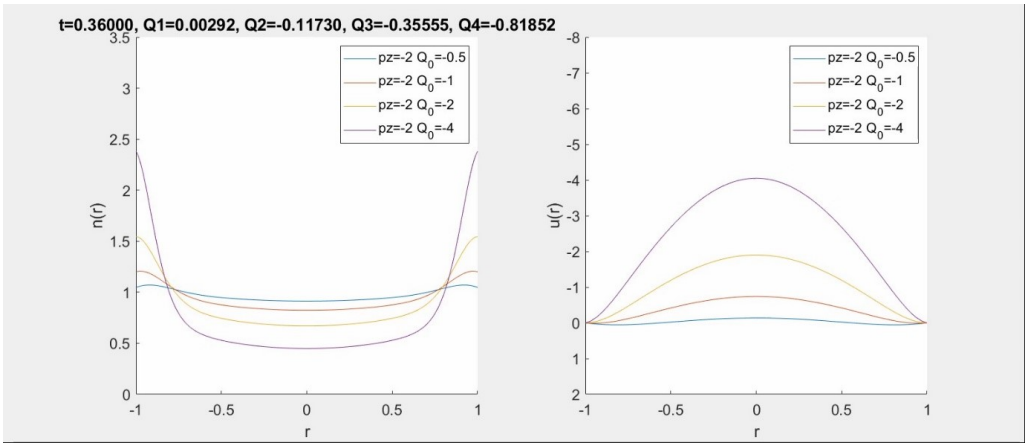


Figure 3.9: Snapshot2: Cells start to move to the centre from the wall; flows keep being pushed down for all lines. Q1 becomes positive (positive direction down for velocity).

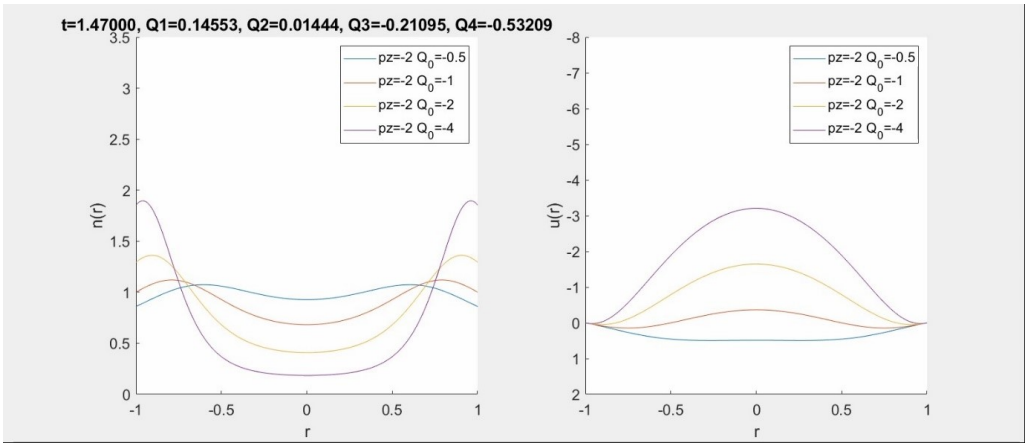


Figure 3.10: Snapshot3: For yellow and purple lines, the cell, velocity profiles and flow rate become relatively stable. For blue and orange lines, cells keep moving to the centre, $Q_1, Q_2 > 0$.

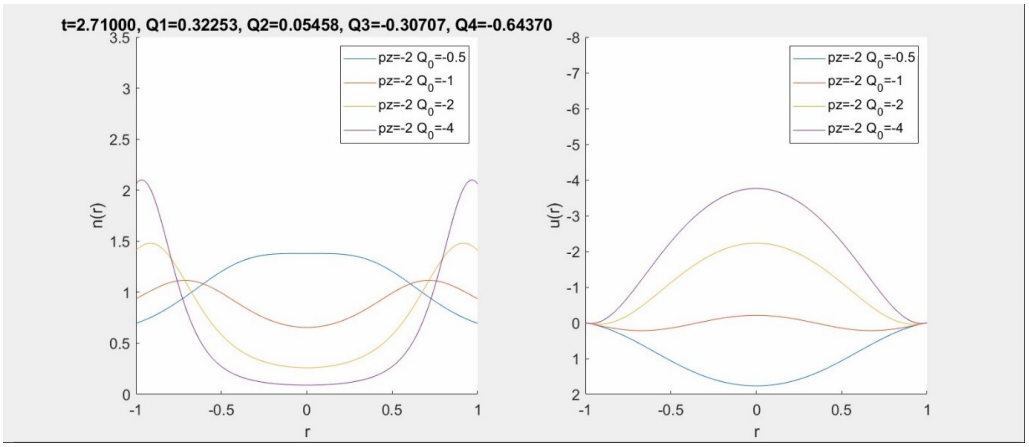


Figure 3.11: Snapshot4: For yellow and purple lines, cells move to the wall gradually to the convergence; velocity profiles go up gradually to stable. For the blue line, cell concentration and velocity distributions become opposite to the beginning curve. The orange line is one step behind the blue line.

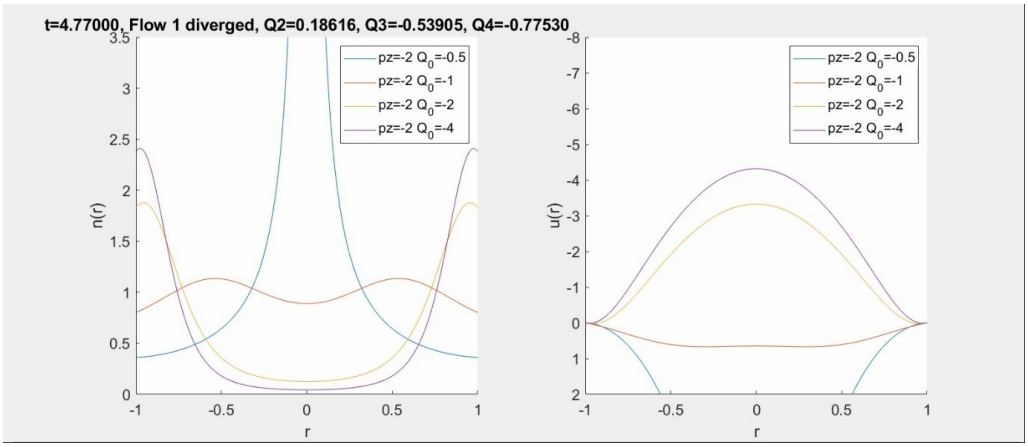


Figure 3.12: Snapshot5: For yellow and purple lines, cells and velocity profiles continue move to the convergence; The blue line is converged. The orange line is one step behind from the blue line.

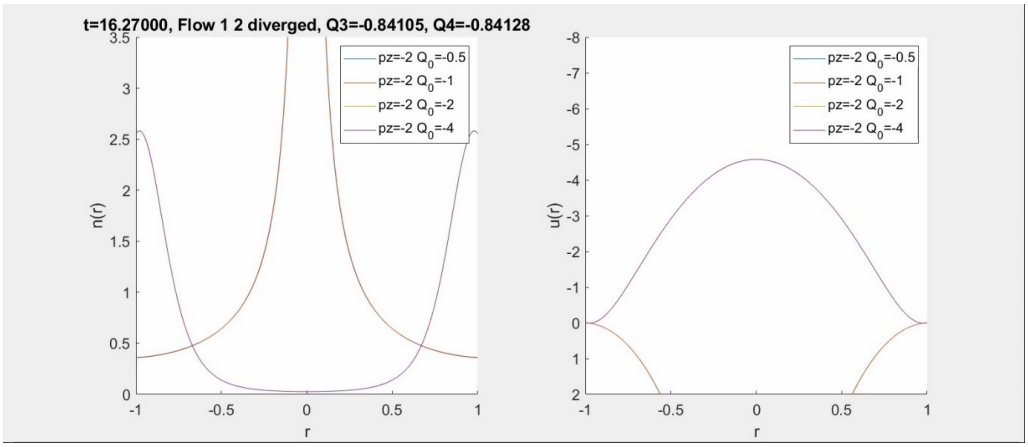


Figure 3.13: Snapshot6: The yellow and purple lines are converged and coincide with each other; the blue and orange lines diverge.

How This Happen

Consider the interactive effect between p_z, cell locomotion and the flow velocity distribution:

- $p_z < 0 \Rightarrow$ the flow is pushed upward;
- $p_z < 0 \Rightarrow$ the flow is pushed downward;
- Cells move to the wall $\Leftrightarrow$ the flow tends to go upward (Q goes negative);
- Cells move to the centre $\Leftrightarrow$ the flow tends to go downward (Q goes positive);

(a) Case 1: unstable

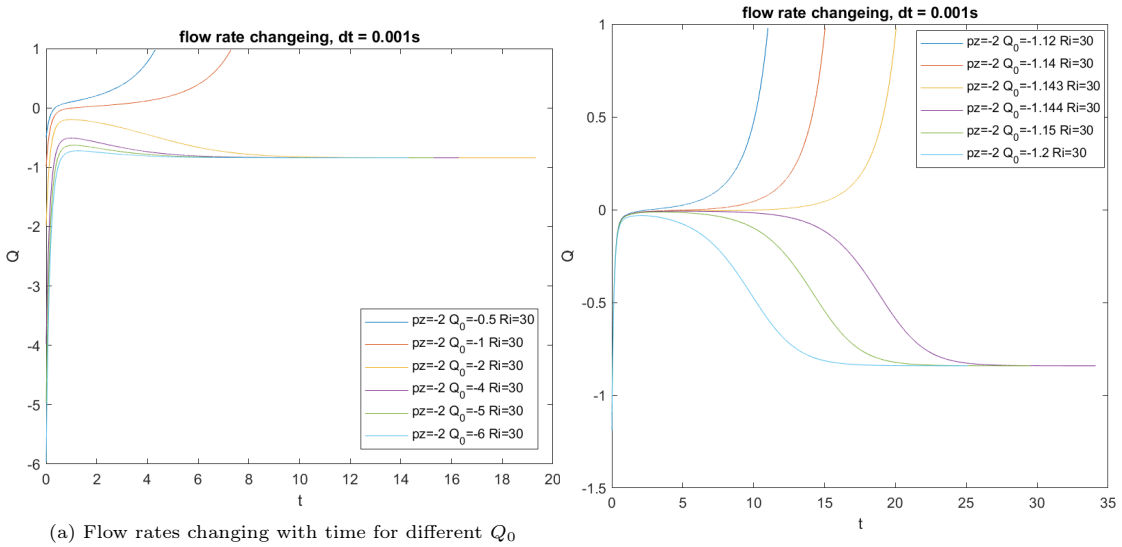
(b) Case 2: stable

Figure 3.14: Schematic diagrams of 2 cases, watermelon line and the blue line stands for velocity and cell concentration profile separately. Case 1 is unstable in a negative p_z environment, and case 2 is stable.

At the start, with a negative initial Q_0, all lines shape like case2.

From the perspective of the influences of the flow i.e. velocity profile. The negative p_z pushes the flow downward to case1, but the initial cell distribution tends to keep the flow at case2. If p_z is stronger relative to the flow, that is, the magnitude of Q_0 is small, the flow will be pushed to case 1 then diverges. If the initial cell distribution is stronger, the system will eventually become stable in case2.

As a result, as described in Section 3.2.1, the diverge gap widens with a bigger magnitude of p_z and smaller magnitude of Q_0 for $p_z < 0$ $Q_0 < 0$ because that bigger magnitude of p_z and smaller magnitude of Q_0 requires stronger cell profile i.e. bigger Ri to converge.



(a) Flow rates changing with time for different Q_0

(b) Boundary of the bifurcation

Figure 3.15: Flow rates changing with time for $Ri = 30$.

The above figure 3.15a shows the flow rate changes under 6 conditions (including the above 4 conditions).

It can be seen that at the start, flow rates all shoot up to a point (snapshot 1 Figure3.8). Under the converged conditions, the data gradually drop back to the converged result. But for the diverged conditions, the flow rate reaches a turning point, but still rise to divergence (Figure3.11 and 3.13), which is consistence of the snapshots shown in Figure3.8 to 3.13. Increase the resolution to find the critical divergence value of $Q_0 = -1.143$. A obvious trend of bifurcation can be manifested in Figure3.15b.

It seems there exist an edge state flow rate around 0 under this specific condition. And the edge state snapshot for velocity and cell concentration distribution is plotted in Figure3.16.

In order to investigate the edge states value of flow rate Q under different Ri number, with the resolution for each state increased to 10^{-6}, in Figure3.17, the flow rate Q_0 changes with time for each edge condition is plotted.

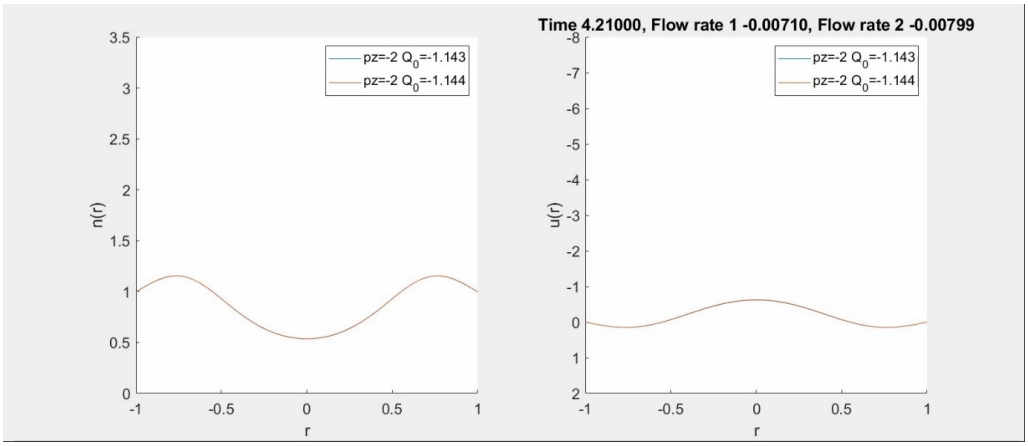


Figure 3.16: Snapshot: edge state between 2 conditions in either sides of the case for $p_z = -2$ $Ri = 30$

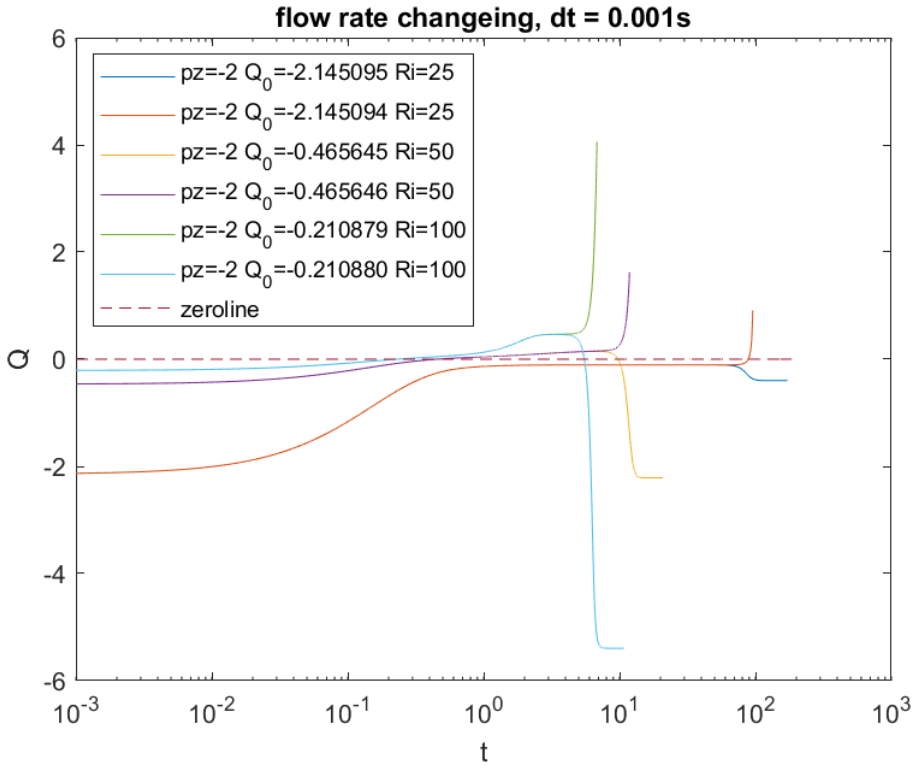


Figure 3.17: Flow rates changing with time for Ri from 25 to 100, the dashed line indicates the horizontal 0 line to show the fact that the system can drop to convergence even the flow rate rise to positive

It is clear that the flow rate at the boundary remains a long time before the bifurcation. In addition, unfortunately, the edge Q exceeds 0, that is, the edge state for this system is not 0. More work needs to be done to find the edge state.

Furthermore, more p_z and Ri are set in order to find a tendency of critical Q_0 and edge state flow rate. The result is shown in Figure3.18.

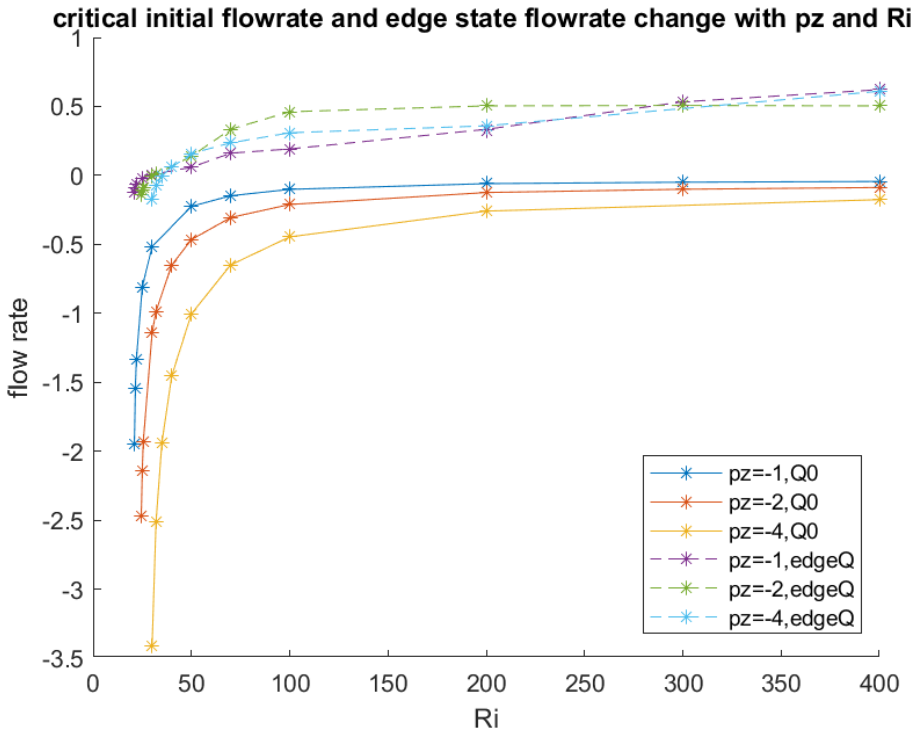
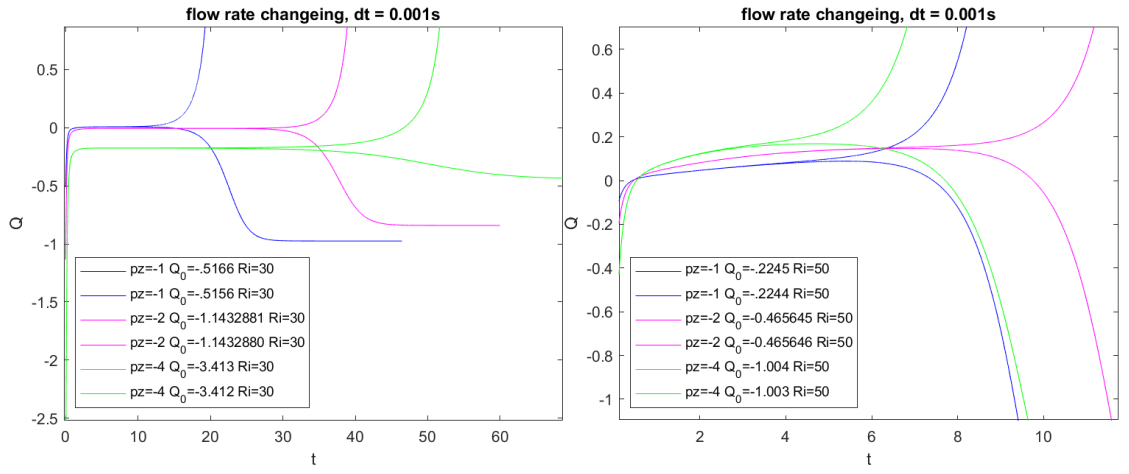


Figure 3.18: Flow rates changing with time for $Ri = 30$.

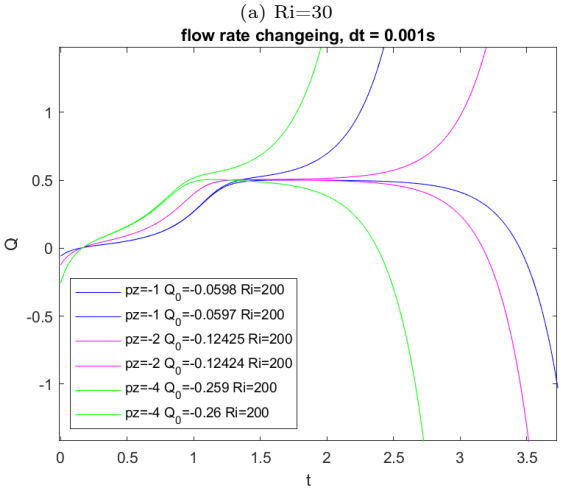
From Figure3.18, the dashed lines stand for the trend of the edge state flow rate. Although it is hard to define an edge state (for instance, it is difficult to locate the edge state flow rate under the condition of $Ri = 100$ in Figure3.17 because of the bump occurring before the edge state, and the bump becomes more obvious for a bigger Ri as shown in Figure3.19d), the lines for different p_z are surely cross with each other despite the error. The solid lines stand for the changes of critical Q_0 with the Richardson number, the regular pattern is not surprising to be logarithmic. The critical initial flow rate Q_0 decrease with the Ri. Besides, there exist a critical Ri, before which the system will not converge. But no critical Ri is found at the other end.

Look deeper on how the diverged edge state flow rate varies with different p_z and Ri. From the Figure3.19 below, A bump occurs when Ri is big enough and when it's hard to define the edge state flow rate. As a result, the edge state is at least a function of Ri.

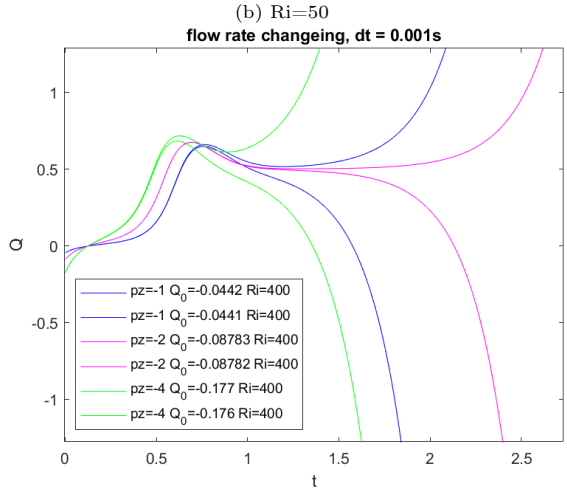


(a) Ri=30

(b) Ri=50



(c) Ri=200



(d) Ri=400

Figure 3.19: Flow rates changes with time for different p_z and Ri.

Chapter 4

Pulsatile Flow

4.1 State Space System

Non-linear State Space System

Recall the control function of the system Eq.2.48 and Eq.2.21.

$$\frac{\partial \mathbf{u}}{\partial t} = P_z - \frac{2Ri}{R^2}N + \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left(\mathbf{r}\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right)\right] + Ri\mathbf{n} \tag{2.47}$$

$$\frac{\partial \mathbf{n}}{\partial t} = -\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left[\mathbf{r}\left(-\beta\mathbf{n}\left(-\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right) - D\frac{\partial \mathbf{n}}{\partial \mathbf{r}}\right)\right] + K_p(N_0 - N)\mathbf{n} \tag{2.21}$$

where

$$P_z = \hat{p}e^{i\omega t}, where\ \omega = \frac{2\pi}{T} \tag{2.51}$$

The discretised form is:

$$\dot{\mathbf{u}} = \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\mathbf{K_1}\ ((\mathbf{K_3 r})\mathbf{K_2})\right]\mathbf{u} + Ri\mathbf{n} - \frac{2Ri}{R^2}N + P_z \tag{4.1}$$

$$\dot{\mathbf{n}} = \frac{1}{\mathbf{r}}\mathbf{K_1}\ D(\mathbf{K_3 r})\mathbf{K_2 n} - \frac{1}{\mathbf{r}}\mathbf{K_1}\ \beta(\mathbf{K_3 r})(\mathbf{K_3 n})(\mathbf{K_2 u}) \tag{4.2}$$

where the pressure gradient is sinusoidal related to the time as:

$$\mathbf{K_1} = \frac{1}{\Delta r}\begin{bmatrix} -1 & 1 & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{bmatrix},\quad \mathbf{K_2} = \frac{1}{\Delta r}\begin{bmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ & & -1 & 1 \end{bmatrix}, \tag{2.31}$$

$$\mathbf{K_3} = \frac{1}{2}\begin{bmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \\ & & & 2 \end{bmatrix} \tag{2.30}$$

Same as mentioned before in Chapter2, $\mathbf{K_1}$ and $\mathbf{K_2}$ represent the first order forward and backward discretization respectively. $\mathbf{K_3}$ is for estimating the middle point values.

The system can then be expressed in a form of long vector combining velocity and concentration values:

$$\begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{n}} \end{bmatrix} = \begin{bmatrix} \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\mathbf{K_1}\ ((\mathbf{K_3 r})\mathbf{K_2})\right] & Ri\mathbf{I} \\ 0 & \frac{1}{\mathbf{r}}\mathbf{K_1}\ D(\mathbf{K_3 r})\mathbf{K_2} \end{bmatrix}\begin{bmatrix} \mathbf{u} \\ \mathbf{n} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{\mathbf{r}}\mathbf{K_1}\ \beta(\mathbf{K_3 r})\mathbf{K_2 u}\mathbf{K_3 n} \end{bmatrix} \tag{4.3}$$

$$+ \begin{bmatrix} -2RiN/R^2 \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} P_z \tag{4.4}$$

Linearized State Space System

Define new matrices to simplify the process in form:

$$\begin{bmatrix}\dot{\mathbf{u}}\\ \dot{\mathbf{n}}\end{bmatrix} = \begin{bmatrix}\mathbf{K_1'} & Ri\mathbf{I}\\ 0 & \mathbf{K_2'}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{n}\end{bmatrix} + \begin{bmatrix}0\\ \mathbf{K_3'}\left(\mathbf{K_2 u}\right)\left(\mathbf{K_3 n}\right)\end{bmatrix} + \begin{bmatrix}-2RiN/R^2\\ 0\end{bmatrix} + \begin{bmatrix}\mathbf{1}\\ 0\end{bmatrix} P_z \tag{4.5}$$

Separate **u**,**n** and $\mathbf{P_z}$ into base state and vibrating state:

$$\begin{aligned}\mathbf{u} &= \mathbf{u_0} + \epsilon\mathbf{u_1} + \dots\\ \mathbf{n} &= \mathbf{n_0} + \epsilon\mathbf{n_1} + \dots\\ P_z &= P_0 + \epsilon P_1 + \dots\end{aligned} \tag{4.6}$$

The non-linear term can be expressed as(ignore the minimal value):

$$\begin{aligned}&\left(\mathbf{K_2 u}\right)\left(\mathbf{K_3 n}\right) = \mathbf{K_2}(\mathbf{u_0} + \epsilon\mathbf{u_1} + \dots)\mathbf{K_3}(\mathbf{n_0} + \epsilon\mathbf{n_1} + \dots)\\ &= (\mathbf{K_2 u_0})(\mathbf{K_3 n_0}) + (\mathbf{K_2}\epsilon\mathbf{u_1})(\mathbf{K_3 n_0}) + (\mathbf{K_2 u_0})(\mathbf{K_3}\epsilon\mathbf{n_1}) + \cancel{\epsilon^2\mathbf{K_2 u_1 K_3 n_1}}\dots\end{aligned} \tag{4.7}$$

The Eq.4.5 become:

$$\begin{aligned}&\begin{bmatrix}\dot{\mathbf{u_0}}\\ \dot{\mathbf{n_0}}\end{bmatrix} + \begin{bmatrix}\dot{\epsilon\mathbf{u_1}}\\ \dot{\epsilon\mathbf{n_1}}\end{bmatrix} = \begin{bmatrix}\mathbf{K_1\prime} & Ri\\ 0 & \mathbf{K_2\prime}\end{bmatrix}\left(\begin{bmatrix}\mathbf{u_0}\\ \mathbf{n_0}\end{bmatrix} + \begin{bmatrix}\epsilon\mathbf{u_1}\\ \epsilon\mathbf{n_1}\end{bmatrix}\right) + \begin{bmatrix}0\\ \mathbf{K_3'}\left(\mathbf{K_2 u_0}\right)\left(\mathbf{K_3 n_0}\right)\end{bmatrix}\\ &+ \begin{bmatrix}0 & 0\\ \mathbf{K_3'}\left(\mathbf{K_3 n_0}\right)\mathbf{K_2} & \mathbf{K_3'}\left(\mathbf{K_2 u_0}\right)\mathbf{K_3}\end{bmatrix}\begin{bmatrix}\epsilon\mathbf{u_1}\\ \epsilon\mathbf{n_1}\end{bmatrix} + \begin{bmatrix}-\frac{2RiN}{R^2}\\ 0\end{bmatrix} + \begin{bmatrix}1\\ \vdots\\ 0\\ \vdots\end{bmatrix}(P_0 + \epsilon P_1)\end{aligned} \tag{4.8}$$

In this project, $u_0 = \mathbf{0}$, $n_0 = \mathbf{1}$, substitute in Eq.4.8 with non-linear terms cancelled out:

$$\begin{bmatrix}\dot{\mathbf{u_1}}\\ \dot{\mathbf{n_1}}\end{bmatrix} = \begin{bmatrix}\mathbf{K_1\prime} & Ri\\ \mathbf{K_3'K_2} & \mathbf{K_2\prime}\end{bmatrix}\begin{bmatrix}\mathbf{u_1}\\ \mathbf{n_1}\end{bmatrix} + \begin{bmatrix}1\\ \vdots\\ 0\\ \vdots\end{bmatrix} P_1 \tag{4.9}$$

The linearised state space system then can be written as with the output as the centreline velocity u(0):

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}p_1\\ y &= \mathbf{C}\mathbf{x} + \mathbf{D}p_1\end{aligned} \tag{4.10}$$

where:

$$\mathbf{x} = \begin{bmatrix}\mathbf{u_1}\\ \mathbf{n_1}\end{bmatrix}, \quad y = \epsilon u\left(0\right)$$

$$\mathbf{A} = \begin{bmatrix}\mathbf{K_1'} & Ri\mathbf{I}\\ \mathbf{K_3'K_2} & \mathbf{K_2'}\end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix}1\\ \vdots\\ 0\\ \vdots\end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix}0\dots0 & 1\ 0\dots0\end{bmatrix}, \quad \mathbf{D} = [0] \tag{4.11}$$

$$\mathbf{K_1'} = \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\mathbf{K}_1\ \left((\mathbf{K_3 r})\mathbf{K_2}\right)\right], \quad \mathbf{K_2'} = \frac{1}{\mathbf{r}}\mathbf{K_1}\ D\left(\mathbf{K_3 r}\right)\mathbf{K_2}, \quad \mathbf{K_3'} = -\frac{1}{\mathbf{r}}\mathbf{K_1}\ \beta(\mathbf{K_3 r})$$

4.2 Effects of Different Parameters

From the linearized state space model, the performance of the system can be influenced by Re, D and β. In this project, the effect of Reynolds number are mainly discussed.

Reynolds Number Effect

In Figure4.1, the bode diagram for the amplitude of the centreline velocity is shown with different Re numbers. It can be seen that the data diverge at $RE = 3$ and $Re = 4$ before a critical frequency ω. It is interesting because when set the pressure gradient as steady, which is equivalent to the case of $\omega = 0$, the system converges to a steady value and not diverge. If the system diverge in the steady state, it is not surprising that it diverges before a certain critical frequency.

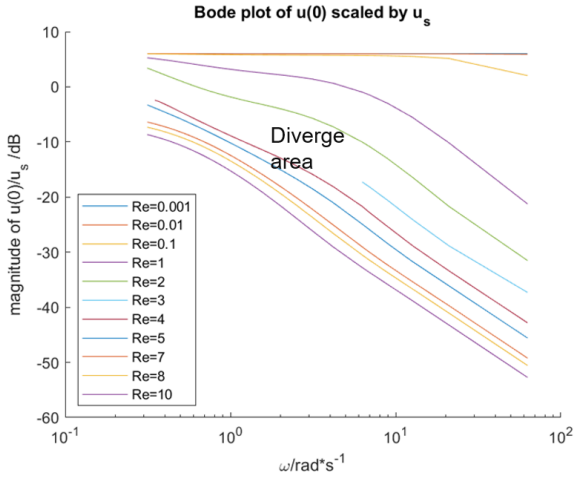
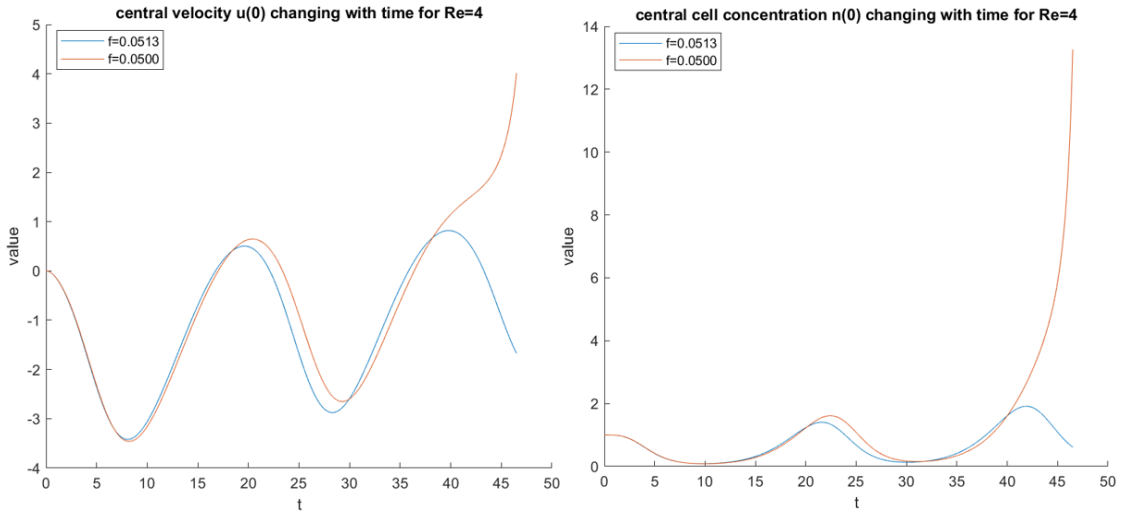


Figure 4.1: Bode diagram for the amplitude of central line velocity $u(0)$ scaled by steady state central line velocity u_s for different Reynolds number. u_s are calculated by setting the pressure gradient to be stable at 1 and extracting the central velocity from the converged result. The amplitude is calculated by calculated the difference between the maximum and minimum value of the data after a long period such that the system is vibrating stably.

In order to investigate the divergence mechanism, the oscillation of central velocity and cell concentrations with time on the divergence boundary are plotted in Figure4.2a and 4.2b for Reynolds number being equal to 4.



(a) Central velocity change

(b) Central concentration change

Figure 4.2: Oscillation of concentration velocity and cell concentration at $Re = 4$

Can be seen after two periods of the oscillation, the red line representing the condition at a frequency of $f = 0.0500$ blows up. The central-line velocity difference with the value of blue line at $f = 0.0513$ becomes increasingly bigger with time. After the third peak, it seems that the backward pressure gradient fails to darg the system back from the divergence. Although due to the indirect influence of the backward pressure gradient experienced by the system, a small resistance is experienced near $t = 41s$, the cell concentration still rapidly reaches an extremely high level.

The snapshots of the velocity and concentration distribution are also shown from Figure4.3 to 4.7.

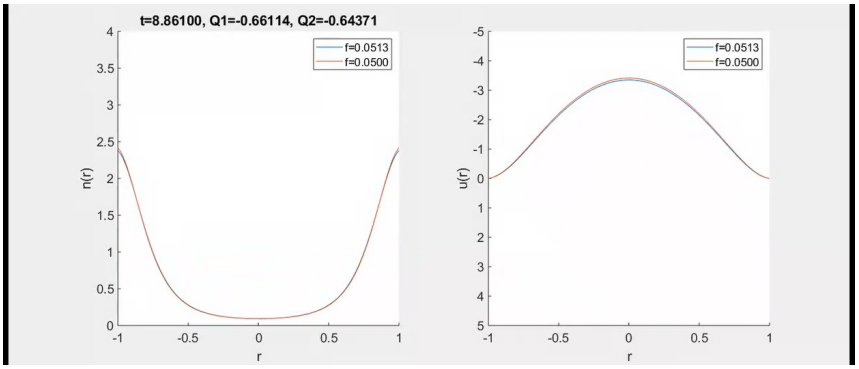


Figure 4.3: Snapshot1: $t = \frac{1}{4}T_{f=0.05}$, the cell moves to the wall and velocity towards up. Little difference on the peak value occurs because of the minor frequency difference as well as the resultant time for development. (Period is not exact because of a minor initial velocity profile are imposed at first to stabilise the system)

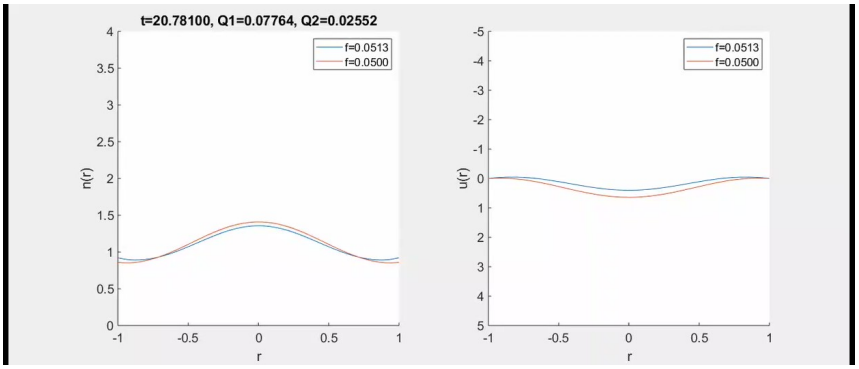


Figure 4.4: Snapshot2: $T = \frac{3}{4}T_{f=0.05}$, the cell moves to the centre smoothly to the peak due to the negative velocity profile. The difference on the peak value becomes bigger

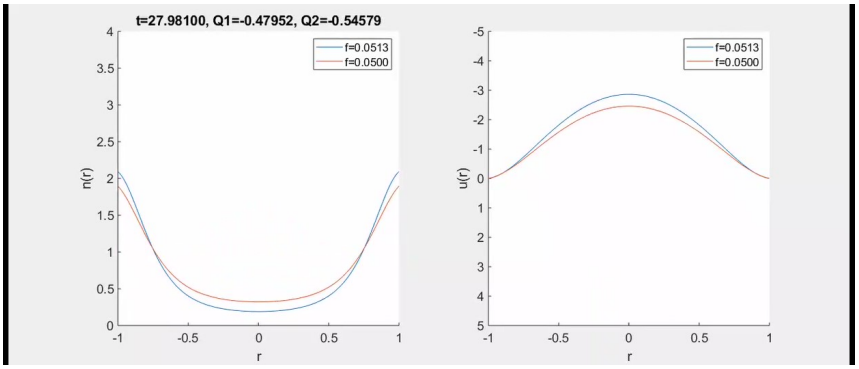


Figure 4.5: Snapshot3: $T = \frac{5}{4}T_{f=0.05}$, the peak value of each state is smaller than former period value, indicating that both systems are still not stable.

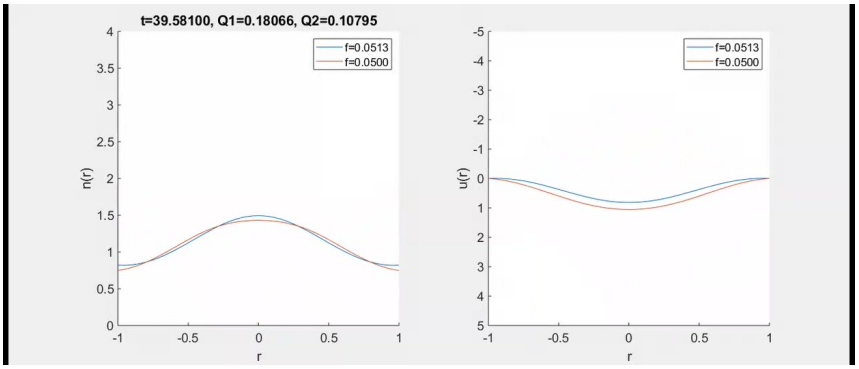


Figure 4.6: Snapshot4: $T = \frac{7}{4}T_{f=0.05}$

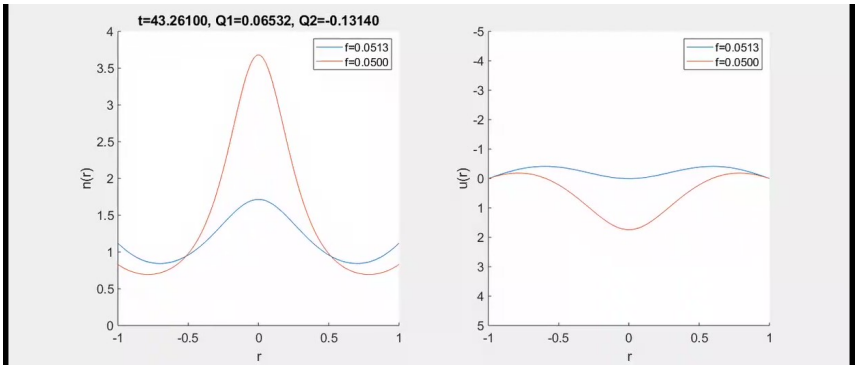


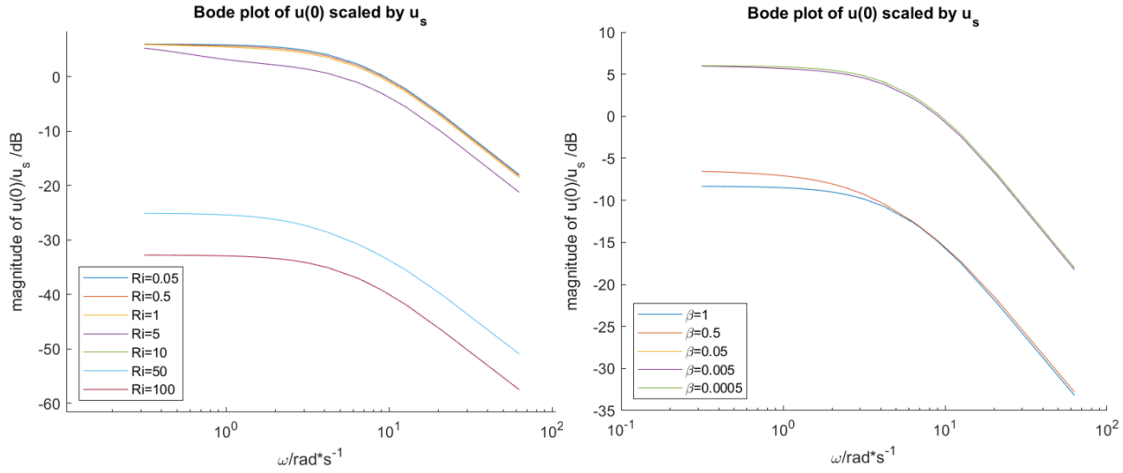
Figure 4.7: Snapshot4: $T = \frac{7}{4}T_{f=0.05} + 3.68s$, with a negative pressure gradient, comparing with Figure4.6 the velocity profile for the blue line is pushed upward but that for the red line is still increase downward and finally to divergence.

The reason for this performance is very similar to that in last chapter. At certain stage the pressure gradient can not maintain the velocity distribution stable because the cell distribution strength goes too much that a positive pressure gradient will not push the flow profile back to the profile like the case 2 shown in the schematic diagram Figure3.14.

Furthermore, a bigger Reynolds number represents a stronger velocity profile, the system will less prone to be influenced by cell distribution, as a result the critical frequency is smaller enough such that the cell profile is enough extreme to influence the flow to be stable.

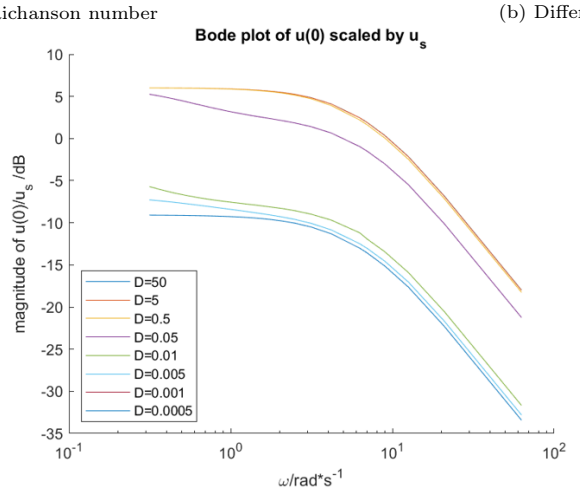
Other Parameters' Effects

The effects of other parameters are also documented in Figure4.8. Can be seen the regular patterns are similar to that of Reynolds number in Figure4.1. The distance between different conditions decrease from either side of the diverge area. Furthermore, the bigger Ri, smaller β and smaller D has a smaller relative magnitude. Further investigation is needed to explore more of the performance.



(a) Different Richanson number

(b) Different β



(c) Different D

Figure 4.8: Bode diagram for the amplitude of central line velocity $u(0)$ scaled by steady state central line velocity us for different Ri, β and D

Chapter 5

Conclusion and Evaluation

5.1 Modelling

In this project, an unsteady solver with constant flow rate and pressure gradient based on Kessler's 1986 model [Kes86b] are developed, semi-implicit first order disicritisation scheme are taken for cell conservation law. And pure implicit first order discritisation scheme are taken for momentum equation. The code is validate with a typical parameters

The first important lesson is that the conservation form of disicritisation, and the order of the boundary condition makes a great difference on the conservation of the cell number, or total flow rate. The second lesson is that the compensation coefficient K_p should scale with different time step dt, and a negative flow rate requires a negative K_p . Besides, non-equal-spaced mesh are tried to apply but it needs more mathematical manipulation of the inhomogeneous.

The model can be optimised by utilising a scheme of discritization, or interpolation of the middle point with higher order and more accuracy. Besides, different way of modelling the random swimming behaviour of the microorganisms can also be applied instead of using a simple diffusion coefficient.

5.2 Bifurcations Under Different Fixed Flow Rate Conditions

In this section, first the converged cell concentration and velocity distributions for different Ri are plotted together in upflow and downflow separately. One found is that the locations of the peak coincide with the that of the flow velocity, and does not change with the Ri. Another is that a divergence will be expected with high Ri form the fact that the peak values increase with higher Ri.

As a result, by setting a sequence of Ri, the bifurcation point at $Ri = 8$ is found consistent with Kessler's analytical result.

Furthermore, in order to investigate the conditions near the 0 point, another bifurcation point is also documented at $Ri = 25$ matching Fung's result qualitatively.

5.3 Bifurcations Under Different Fixed Pressure Gradient Conditions

In this section, different pressure gradient and initial flow rate are applied and one interesting bifurcation is found, that is a divergence then convergence performance manifests for $p_z < 0 Q_0 < 0$. The divergence gap widens with a bigger magnitude of p_z. An explanation is given by viewing the snapshots of the velocity and concentration distributions. The cell concentration profile and the pressure gradient influence the velocity profile in opposite directions. As a result, bigger magnitude of p_z and smaller magnitude of Q_0 requires bigger Ri to converge the system.

The attempt of finding the edge state are partially failed. The zero flow rate is proved not to be the edge state condition by adding the resolution of the Q_0 and find the edge state flow rate at different Ri and p_z conditions. However, the pattern of the critical Q_0 in this type of initial condition is clear to be logarithmic, and the magnitude of critical Q_0 is smaller with bigger Ri.

Further researches can be done to investigate more on the edge state by adding the resolution of the Q_0 to find more accurate edge states for different Ri, or changing different type of initial condition, (sine-like or parabolic) to find an same type of edge state.

5.4 Pulsatile Flow Under Sinusoidal Wave Pressure Gradient

In this part, the linearied state space system is derived and different parameters' effect are investigated. The diverge area is found for Reynolds number condition and the explanation is given qualitatively by look into the central velocity and concentration, and the profiles changing with time.

More blank is left in this section, more patches of conditions for Reynolds number, β and D in the order of 1, and wider range of frequencies should be taken to investigate the divergence performance. Besides from the fact that the steady station converges, i.e. the system converges when $\omega = 0$.

Appendix A

Derivation of cell conservation law and the analytical results

A.1 Cell conservation law

The conservation of cell number ($N = nv$) is:

$$\frac{\partial n}{\partial t}dv + \sum_{out} N - \sum_{in} N = 0 \tag{A.1}$$

In x direction:

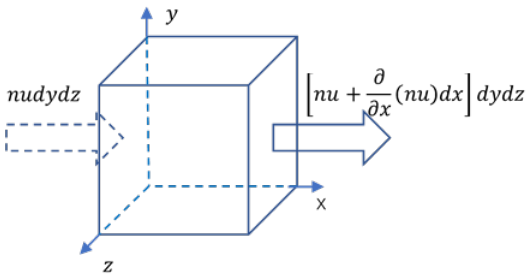


Figure A.1: Cell conservation law on horizontal direction in a flow cell

combined the 3 directions

$$\sum_{in} N - \sum_{out} N = \left[\frac{\partial}{\partial x}\ (nu)\, dx\right] dydz + \left[\frac{\partial}{\partial y}\ (nv)\, dy\right] dxdz + [\frac{\partial}{\partial z}\ (nw)dz]dxdy \tag{A.2}$$

As a result:

$$\frac{\partial n}{\partial t}dxdydz = \left[\frac{\partial}{\partial x}\ (nu) + \frac{\partial}{\partial y}\ (nv) + \frac{\partial}{\partial z}\ (nw)\right] dxdydz \tag{A.3}$$

$$or, \quad \frac{\partial n}{\partial t} + \nabla\,(nu) = 0$$

Add flux:

$$\frac{\partial n}{\partial t} + \nabla\,(nu) = \nabla j \tag{A.4}$$

With assumptions:

$$\nabla\,(nu) = u\nabla n \tag{A.5}$$

$$\nabla j = \frac{1}{r}\frac{\partial}{\partial r}(rj) \tag{A.6}$$

As a result

$$\frac{\partial n}{\partial t} + u\nabla n = \frac{1}{r}\frac{\partial}{\partial r}(rj) \tag{A.7}$$

Add α to Navier-Stokes equation

$$\frac{\partial \mathbf{u}}{\partial t} = -P_z + \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left(\mathbf{r}\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right)\right] + \alpha\mathbf{n} \tag{A.8}$$

Analytical steady equations

$$u\nabla n = \frac{1}{r}\frac{\partial}{\partial r}(rj) \tag{A.9}$$

$$0 = -P_z + \frac{1}{Re}\left[\frac{1}{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}\left(\mathbf{r}\frac{\partial \mathbf{u}}{\partial \mathbf{r}}\right)\right] + \alpha\mathbf{n} \tag{A.10}$$

where:

$$j = -n\beta\frac{du}{dr} - D\frac{dn}{dr} \tag{A.11}$$

A.2 Solve n(r) Analytically

Cell conservation law with no flux:

$$j = -n\beta\frac{du}{dr} - D\frac{dn}{dr} = 0 \tag{A.12}$$

$$\Leftrightarrow n = n(0)e^{-(\beta/D)(u-u(0))} \tag{A.13}$$

Where $n(0)$ and $u(0)$ are cell concentration and velocity at the centre. NS equation

$$0 = -P_z + \frac{1}{Re}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)\right] + \alpha n \tag{A.14}$$

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) = -Re\ \alpha n + Re\ P_z \tag{A.15}$$

Combine Eq.A.15 and Eq.A.13.

$$\log n = -\frac{\beta}{D}(u - u(0))\log n(0) \tag{A.16}$$

$$-\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \log n}{\partial r}\right) = -\frac{Re\ \alpha\beta}{D}n + \frac{Re\ \beta}{D}\ P_z \tag{A.17}$$

With $P_z = 0$:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \log n}{\partial r}\right) = \frac{Re\ \alpha\beta}{D}n \tag{A.18}$$

$$n\left(r\right) = \frac{n\left(0\right)}{\left(1 + \gamma n\left(0\right)r^2\right)^2} \tag{A.19}$$

where

$$\gamma = \frac{Re\ Ri\beta}{8D} \tag{A.20}$$

Solve $n(0)$ Analytically

Set an average concentration n_0 such that

$$n_0 = \frac{\oiint_s n dA}{A} \tag{A.21}$$

Replace n according to Eq.A.19

$$\int_0^R n\left(r\right) r dr = \frac{1}{2} n_0 R^2 \tag{A.22}$$

or

$$\frac{n\left(0\right) R^2}{1 + \gamma n\left(0\right) R^2} = n_0 R^2$$

As a result, when $\gamma n\left(0\right) R^2 \to 1$, the code blows up.

Appendix B

Matlab codes

B.1 Fixing the flow rate

```
% ————————————————————————————————————————————————————————————————————————%
%             Solve a trainsent 1D laminar downflow pipe flow                 %
% ————————————————————————————————————————————————————————————————————————%
% PDE non—dimentional):
%       du/dt= —2/Re 1/R [du/dr]_(r=R)—2Ri/R^2 N+1/Re [1/r d/dr(rdu/dr)]+Ri n
%       dn/dt=—1/r d/dr[beta rn du/dr+Dr dn/dr]+Kp(N0—N)n
% Method:
%       semi—implicit finite difference,forword time,central space discritization
% Boundary condition:
%       r = 0 —— dn/dr = 0              symmetric boundary
%       r = 0 —— du/dr = 0              symmetric boundary
%       r = R —— dn/dt = —n beta /D 2R  wall boundary
%       r = R —— u = 0                  wall boundary
% Initial condition:
%       n = 1, u = (Q0 pi^2)/(2pi—4)
% ————————————————————————————————————————————————————————————————————————%
%                           Author: Songrui LI                                %
%                             Date: June 23rd                                  %
% ————————————————————————————————————————————————————————————————————————%

%house keeping
clear,clc;
close all;

%% Preparation
%set mesh dimension, time step and initailize u
R = 1;                                % Radius of pipe
n=200;                               % Non_Boundary number of nodes
dt = 1e—3;                            % Time step
r= [linspace(0,R,n)];                 % Mesh (equal space discretisation)
dr = r(2:end)—r(1:end—1);             % space step
Kp= 100 ;                             % compensate term

%for saving snap shot
%snap = [1:200/dt];
snap = [0];
sn = 1;
x_snap = zeros(length(snap),2*n);

%set max iteration number
iter_number=5e6;
%set max and min residual
Res_max = 1e3;
Res_min = 1e—10;

%set flow parameter
pz = —4;                              % pressure
Q0 = —1.004;                           % flow rate
titl = '1—4—50—1_004';
D = 5e—2;                             % Diffusivity of cells
```

```
beta = 5e-2;                          % Gyrotastic length
Re = 1;                              % Reynolds number

% start different parameters
alpha_loop = [50]';         % bouyant rate

u_loop = zeros(length(alpha_loop),n);            % results
N_loop = zeros(length(alpha_loop),n);            % results
Q_loop = zeros(length(alpha_loop),1);            % results

for Loop = 1:length(alpha_loop)
alpha = alpha_loop( Loop);

%set initial condition and save
N = ones(1,n);
u = Q0*pi*pi/(2*pi-4)*cos(pi*r/2);
N0 = N;
u0 = u;
x = [u,N];
x0 = [u0,N0];

%% set iteration
%% matrix K3 midpoint interpolation
%first order
K3 = sparse(diag([1*ones(1,n-1),2])+diag(1*ones(1,n-1),1));
K3 = 1/2*K3;

% %second order Lagrange Midpoint Interpolation
% ...
    %https://d3cw3dd2w32x2b.cloudfront.net/wp-content/uploads/2011/06/dyadic_interpolation.pdf
% K3 = ...
    sparse(diag(9/16*ones(1,n))+diag(9/16*ones(1,n-1),1)+diag(-1/16*ones(1,n-2),2)+diag(-1/16*ones(1,n-
% K3(1,:)=[1/2,1/2,zeros(1,n-2)];
% K3(end-1,:)=[zeros(1,n-2),1/2,1/2];
% K3(end,:)=[zeros(1,n-1),1];

%% parameters
t = 0;                                % Current time
x_1 = x0;                             % Save result of one timestep afterwards
RES = [0];                            % Save all residual data
Num = [Num_int(dr,K3,N0,r)];          % Save all cell number data
Q = [Q_int(dr,K3,u0,r)];              % Save all cell number data

%% set K,K1 and K2:
%matrix K second order
K = sparse(zeros(n));
K(:,end)=3/2;
K(:,end-1)=-2;
K(:,end-2)=1/2;
K = 1/dr(end)*K;
%matrix K4
K4 = sparse([zeros(1,n-3),1/2,-2,3/2]);
K4 = 1/dr(end)*K4;
%matrix K1
K1 = sparse(diag([-1,1*ones(1,n-1)])+diag(-1*ones(1,n-1),-1));
K1(1,2) = 1;
K1 = 1./[dr(1),dr]'.*K1;
%matrix K2
K2 = sparse(diag([-1*ones(1,n-1),1])+diag(ones(1,n-1),1));
K2(end,end-1) = -1;
K2 = 1./[dr,dr(end)]'.*K2;

% %second order
% %matrix K1
% K1 = sparse(diag(3/2*ones(1,n))+diag(-2*ones(1,n-1),-1)+diag(1/2*ones(1,n-2),-2));
% K1(1,:)=[-3/2,2,-1/2,zeros(1,n-3)];
% K1(2,:)=[-1/2,0,1/2,zeros(1,n-3)];
% K1 = 1./[dr(1),dr].*K1;
% %matrix K2
% K2 = sparse(diag(-3/2*ones(1,n))+diag(2*ones(1,n-1),1)+diag(-1/2*ones(1,n-2),2));
% K2(end,:)=[zeros(1,n-3),1/2,-2,3/2];
% K2(end-1,:)=[zeros(1,n-3),-1/2,0,1/2];
% K2 = 1./[dr,dr(end)].*K2;
```

```
122
123 %% set G
124 %matrix G
125 G = sparse(eye(n)-dt/Re*diag(1./r')*K1*(diag(K3*r')*K2));
126 G(1,:)=[-3/2,2,-1/2,zeros(1,n-3)];
127 G(n,:)=[zeros(1,n-1),1];
128 % matrix H
129 H = sparse(eye(n)-dt*diag(1./r)*K1*(D*diag(K3*r')*K2));
130 H(1,:)=[-137/60,5,-5,10/3,-5/4,1/5,zeros(1,n-6)];
131 H(end,:)=[zeros(1,n-6),-1/5,5/4,-10/3,5,-5,0];
132 %% start iteration
133 for iter=1:iter_number
134
135     % next time step N_1
136     % calculate K_comp and K_H
137     K_comp = sparse(dt*Kp*(Num(1)-Num(end))*eye(n));
138
139     %step 1
140     u_1=u'-dt*pz+dt*alpha*N'-2*dt*alpha/R/R*Num(1);
141     u_1(1)=0;
142     u_1(end)=0;
143     u_1=(G\u_1)';
144     % step 2
145     H(end,end)=137/60-beta/D*dr(end)*K4*u_1';
146     N_1=N'-dt*diag(1./r)*K1*beta*diag(K3*r')*((K3*N').*(K2*u'))+K_comp*N';
147     N_1(1)=0;
148     N_1(end)=0;
149     N_1=(H\N_1)';
150
151     % combine x_1
152     x_1 = [u_1 N_1];
153
154     % claculate residual and save
155     Res = norm(x-x_1);
156     RES = [RES, Res];
157
158     % update x
159     x = x_1;
160     u = x(1:n);
161     N = x(n+1:end);
162
163     %calculate and save cell number
164     Num = [Num,Num_int(dr,K3,N,r)];
165     %calculate and save cell number
166     Q = [Q,Q_int(dr,K3,u,r)];
167
168     % update time and display with Res
169     t = t+dt;
170 %     disp('current time t and Residual Res: ')
171 %     disp([t, Res])
172
173     % save snapshot
174     if length(snap) ≥ sn && iter == snap(sn)
175     x_snap(sn,:) = x;
176     sn = sn+1;
177     end
178
179     %realtime plot
180     %pic(r,R,n_t,x,x0,dt,RES,sn,x_snap,Num,Q,Kp)
181
182 %     %stop at a time
183 %     if t>1
184 %         break;
185 %     end
186
187     %define convergance
188     if conv(Res, Res_max, Res_min,t)
189         disp('t, Res,alpha')
190         disp([t, Res,alpha])
191         break;
192     end
193 end
194
```

```
%% plot final result
pic(r,R,n,x,x0,dt,RES,sn,x_snap,Num,Q,Kp,alpha);

%% end loop

u_loop(Loop,:) = u;
N_loop(Loop,:) = N;
Q_loop(Loop,:) = Q(end);
end

%save(titl)
%% fuctions
%% calculate Q
function Q= Q_int(dr,K3,u,r)
    u_Q=K3*u';
    r_Q=K3*r';
    Q = [(dr.*u_Q(1:end-1)')*r_Q(1:end-1)];                    % Save all Q ...
        data, Q(1)=Q(0)
end

%% calculate Num
function Num= Num_int(dr,K3,N,r)
    N_Q=K3*N';
    r_Q=K3*r';
    Num = [(dr.*N_Q(1:end-1)')*r_Q(1:end-1)];                    % Save all Q ...
        data, Q(1)=Q(0)
end

%% define convergance
function conv = conv(Res, Res_max, Res_min,t)
    if (Res>Res_max || 0)                %check the residual
        disp('what a pity U DIverged');
        conv = 1;
    elseif Res<Res_min
        disp('OMG Congrats finally U CONverged');
%         disp('Converged time:')
%         disp(t)
        conv = 1;
    else
        conv = 0;
    end
end

%% plot
function pic = pic(r,R,n_t,x,x0,dt,RES,sn,x_snap,Num,Q,Kp,alpha)
figure(1)                                                  % plot result
    subplot(2,1,1);
%   plot(linspace(-R,R,2*n-1),[fliplr(x0(n+2:end)),x0(n+1:end)])
%     hold on
%     if sn > 1
%         for sn1 = 1:sn-1
%            ...
    plot(linspace(-R,R,2*n-1),[fliplr(x_snap(sn1,n+2:end)),x_snap(sn1,n+1:end)])
%         end
%     end
    plot([-1*fliplr(r(2:end)),r],[fliplr(x(n_t+2:end)),x(n_t+1:end)])
    tit = append("n(r), dt = ",string(dt),"s, ","Kp= ",string(Kp), " Ri = ...
        ",string(alpha));
    title(tit)
    xlabel('r')
    ylabel('n')
%     legend({'t_0','t_1','t_2','t_3' 't_4','conv.'},'Location','southwest')
    hold on

    subplot(2,1,2);
%   plot(linspace(-R,R,2*n-1),[fliplr(x0(2:n)),x0(1:n)])
%     hold on
%     if sn > 1
%         for sn1 = 1:sn-1
%            plot(linspace(-R,R,2*n-1),[fliplr(x_snap(sn1,2:n)),x_snap(sn1,1:n)])
%         end
%     end
    plot([-1*fliplr(r(2:end)),r],[fliplr(x(2:n_t)),x(1:n_t)])
```

```
264     set(gca,'Ydir','reverse');      % Reverse the axis for better looking
265     tit = append("u(r), dt = ",string(dt),"s, Ri = ",string(alpha));
266     title(tit)
267     xlabel('r')
268     ylabel('u')
269 %     legend({'t_0','t_1','t_2','t_3' 't_4','conv.'},'Location','southwest')
270 %     hold off
271     hold on
272     set(gcf, 'Position',  [750, 50, 600, 700])
273     drawnow
274
275 figure(2)                                                % plot Num and Q
276     %plot flow rate
277     subplot(2,1,1);
278     plot(dt:dt:(length(Q)-1)*dt,Q(2:end)-Q(1))
279     tit = append("flow rate difference with Q0(t), dt = ",string(dt),"s");
280     title(tit)
281     xlabel('t')
282     ylabel('Q-Q0')
283     set(gcf, 'Position',  [150, 50, 600, 700])
284     drawnow
285     hold on
286
287     %plot cell number
288     subplot(2,1,2);
289     plot(dt:dt:(length(Num)-1)*dt,Num(2:end)-Num(1))
290     tit = append("cell number difference with N0(t), dt = ",string(dt),"s, ","Kp= ...
            ",string(Kp));
291     title(tit)
292     xlabel('t')
293     ylabel('Num-Num0')
294     hold on
295     drawnow
296
297 % figure(3)                                                % plot Residual
298 %     plot(dt:dt:(length(RES)-1)*dt,RES(2:end))
299 %     title('Residual(t)')
300 %     xlabel('t')
301 %     ylabel('Res')
302 end
```

B.2 Shaking the pressure gradient

```
1  % ------------------------------------------------------------------------%
2  %             Solve a trainsent 1D laminar downflow pipe flow             %
3  % ------------------------------------------------------------------------%
4  % PDE non-dimentional):
5  %       du/dt= pz -2/Re 1/R [du/dr]_(r=R)+1/Re [1/r d/dr(rdu/dr)]+Ri n
6  %       dn/dt=-1/r d/dr[beta rn du/dr+Dr dn/dr]+Kp(N0-N)n
7  % Method:
8  %       semi-implicit finite difference,forword time,central space discritization
9  % Boundary condition:
10 %       r = 0 -- dn/dr = 0              symmetric boundary
11 %       r = 0 -- du/dr = 0              symmetric boundary
12 %       r = R -- dn/dt = -n beta /D 2R  wall boundary
13 %       r = R -- u = 0                  wall boundary
14 % Initial condition:
15 %       n = 1, u = (Q0 pi^2)/(2pi-4)
16 % ------------------------------------------------------------------------%
17 %                          Author: Songrui LI                             %
18 %                            Date: June 23rd                               %
19 % ------------------------------------------------------------------------%
20
21 %house keeping
22 clear,clc;
23 close all;
24
25 %% Preparation
26 %set mesh dimension, time step and initailize u
```

```
R = 1;                                  % Radius of pipe
n=200;                                 % Non_Boundary number of nodes
dt = 1e-3;                              % Time step
r= [linspace(0,R,n)];                   % Mesh (equal space discretisation)
dr = r(2:end)-r(1:end-1);              % space step
Kp= 100 ;                               % compensate term

% %for save snap shot
% snap = [0];
% sn = 1;
% x_snap = zeros(length(snap),2*n);

%set max iteration number
iter_number=5e6;
%set max and min residual
Res_max = 1e3;
Res_min = 1e-6;

%set flow parameter
A = 1;                          %amplitude and period
alpha = 5;                         %period
titl='D_0_005';
Q0 = 1e-4;                           % flow rate
D = 5e-3;                            % Diffusivity of cells
beta = 5e-2;                         % Gyrotastic length
Re = 1;                              % Reynolds number

% start different parameters
%T_loop = [1:.2:8]';          % bouyant rate
T_loop = [1e-1:2e-1:9e-1 1:.5:20]';    % bouyant rate
%T_loop = [110:10:160]';          % bouyant rate

time_loop=[300*ones(1,5),300*ones(1,20),300*ones(1,17),300*ones(1,2)];

u_loop = zeros(length(T_loop),n);             % results
N_loop = zeros(length(T_loop),n);             % results
u0_loop = zeros(length(T_loop),200/dt);             % results save first 200s
N0_loop = zeros(length(T_loop),200/dt);             % results save first 200s
Q_loop = zeros(length(T_loop),1);             % results

for Loop = 1:length(T_loop)
T = T_loop( Loop);
pz = append(string(A),"sin(","2pi/",string(T),"t)");

%set initial condition and save
N = ones(1,n);
%u = Q0/2*ones(1,n);
u = Q0*pi*pi/(2*pi-4)*cos(pi*r/2);
%u = -12*Q0*r.*(r-1);
N0 = N;
u0 = u;
x = [u,N];
x0 = [u0,N0];

%% set iteration
%% matrix K3 midpoint interpolation
%first order
K3 = sparse(diag([1*ones(1,n-1),2])+diag(1*ones(1,n-1),1));
K3 = 1/2*K3;

% %second order Lagrange Midpoint Interpolation
% ...
    %https://d3cw3dd2w32x2b.cloudfront.net/wp-content/uploads/2011/06/dyadic_interpolation.pdf
% K3 = ...
    sparse(diag(9/16*ones(1,n))+diag(9/16*ones(1,n-1),1)+diag(-1/16*ones(1,n-2),2)+diag(-1/16*ones(1,n-
% K3(1,:)=[1/2,1/2,zeros(1,n-2)];
% K3(end-1,:)=[zeros(1,n-2),1/2,1/2];
% K3(end,:)=[zeros(1,n-1),1];

%% parameters
t = 0;                                  % Current time
x_1 = x0;                               % Save result of one timestep afterwards
RES = [0];                              % Save all residual data
```

```
Num = [Num_int(dr,K3,N0,r)];        % Save all cell number data
Q = [Q_int(dr,K3,u0,r)];            % Save all cell number data

%% set K,K1 and K2:
%matrix K second order
K = sparse(zeros(n));
K(:,end)=3/2;
K(:,end-1)=-2;
K(:,end-2)=1/2;
K = 1/dr(end)*K;
%matrix K4
K4 = sparse([zeros(1,n-3),1/2,-2,3/2]);
K4 = 1/dr(end)*K4;
%matrix K1
K1 = sparse(diag([-1,1*ones(1,n-1)])+diag(-1*ones(1,n-1),-1));
K1(1,2) = 1;
K1 = 1./[dr(1),dr]'.*K1;
%matrix K2
K2 = sparse(diag([-1*ones(1,n-1),1])+diag(ones(1,n-1),1));
K2(end,end-1) = -1;
K2 = 1./[dr,dr(end)]'.*K2;

% %second order
% %matrix K1
% K1 = sparse(diag(3/2*ones(1,n))+diag(-2*ones(1,n-1),-1)+diag(1/2*ones(1,n-2),-2));
% K1(1,:)=[-3/2,2,-1/2,zeros(1,n-3)];
% K1(2,:)=[-1/2,0,1/2,zeros(1,n-3)];
% K1 = 1./[dr(1),dr].*K1;
% %matrix K2
% K2 = sparse(diag(-3/2*ones(1,n))+diag(2*ones(1,n-1),1)+diag(-1/2*ones(1,n-2),2));
% K2(end,:)=[zeros(1,n-3),1/2,-2,3/2];
% K2(end-1,:)=[zeros(1,n-3),-1/2,0,1/2];
% K2 = 1./[dr,dr(end)].*K2;

%% set G
%matrix G
G = sparse(eye(n)-dt/Re*diag(1./r')*K1*(diag(K3*r')*K2));
G(1,:)=[-3/2,2,-1/2,zeros(1,n-3)];
G(n,:)=[zeros(1,n-1),1];
% matrix H
H = sparse(eye(n)-dt*diag(1./r)*K1*(D*diag(K3*r')*K2));
H(1,:)=[-137/60,5,-5,10/3,-5/4,1/5,zeros(1,n-6)];
H(end,:)=[zeros(1,n-6),-1/5,5/4,-10/3,5,-5,0];
%% start iteration
for iter=1:iter_number

    % next time step N_1
    % calculate K_comp and K_H
    K_comp = sparse(dt*Kp*(Num(1)-Num(end))*eye(n));
    K_H = sparse(dt*Kp*(Q(1)-Q(end))*eye(n));

    %step 1
    u_1=u'-dt*(A*sin(2*pi/T*t))+dt*alpha*N'-2*dt*alpha/R/R*Num(1);
    u_1(1)=0;
    u_1(end)=0;
    u_1=(G\u_1)';
    % step 2
    H(end,end)=137/60-beta/D*dr(end)*K4*u_1';
    N_1=N'-dt*diag(1./r)*K1*beta*diag(K3*r')*((K3*N').*(K2*u'))+K_comp*N';
    N_1(1)=0;
    N_1(end)=0;
    N_1=(H\N_1)';

    % combine x_1
    x_1 = [u_1 N_1];

    % claculate residual and save
    Res = norm(x-x_1);
    RES = [RES, Res];

    % update x
    x = x_1;
    u = x(1:n);
```

```
171      N = x(n+1:end);
172
173      %calculate and save cell number
174      Num = [Num,Num_int(dr,K3,N,r)];
175      %calculate and save cell number
176      Q = [Q,Q_int(dr,K3,u,r)];
177
178      % update time and display with Res
179      t = t+dt;
180  %      disp('current time t and Residual Res: ')
181  %      disp([t, Res])
182
183  %      % save snapshot
184  %      if length(snap) ≥ sn && iter == snap(sn)
185  %      x_snap(sn,:) = x;
186  %      sn = sn+1;
187  %      end
188
189      % save n0 and u0
190      N0_loop(Loop,iter)=N(1);
191      u0_loop(Loop,iter)=u(1);
192      %realtime plot
193      %plot_one_condition_ani(r,R,n,x,x0,dt,RES,sn,x_snap,Num,Q,Kp,alpha,pz,Q0)
194
195      %stop at a time
196      if t>time_loop(Loop)+1e-8
197          break;
198      end
199
200  %      %define convergance
201  %      if Conv(Res, Res_max, Res_min,t)
202  %          disp('t, Res,alpha')
203  %          disp([t, Res,alpha])
204  %          break;
205  %      end
206  end
207
208  %% plot final result
209  %pic(r,R,n,x,x0,dt,RES,0,0,Num,Q,Kp,alpha);
210
211  %% end loop
212  T
213  u_loop(Loop,:) = u;
214  N_loop(Loop,:) = N;
215  Q_loop(Loop,:) = Q(end);
216  end
217
218  save (titl);
219
220  %% fuctions
221  %% calculate Q
222  function Q= Q_int(dr,K3,u,r)
223      u_Q=K3*u';
224      r_Q=K3*r';
225      Q = [(dr.*u_Q(1:end-1)')*r_Q(1:end-1)];                    % Save all Q ...
             data, Q(1)=Q(0)
226  end
227
228  %% calculate Num
229  function Num= Num_int(dr,K3,N,r)
230      N_Q=K3*N';
231      r_Q=K3*r';
232      Num = [(dr.*N_Q(1:end-1)')*r_Q(1:end-1)];                    % Save all Q ...
             data, Q(1)=Q(0)
233  end
234
235  %% define convergance
236  function conv = Conv(Res, Res_max, Res_min,t)
237      if (Res>Res_max || 0)                  %check the residual
238          disp('what a pity U DIverged');
239          conv = 1;
240      elseif Res<Res_min
241          disp('OMG Congrats finally U CONverged');
```

```
%         disp('Converged time:')
%         disp(t)
        conv = 1;
    else
        conv = 0;
    end
end

%% plot
function pic = pic(r,R,n_t,x,x0,dt,RES,sn,x_snap,Num,Q,Kp,alpha)
figure(1)                                                   % plot result
    subplot(2,1,1);
%   plot(linspace(-R,R,2*n-1),[fliplr(x0(n+2:end)),x0(n+1:end)])
%     hold on
%     if sn > 1
%         for sn1 = 1:sn-1
%             ...
    plot(linspace(-R,R,2*n-1),[fliplr(x_snap(sn1,n+2:end)),x_snap(sn1,n+1:end)])
%         end
%     end
    plot([-1*fliplr(r(2:end)),r],[fliplr(x(n_t+2:end)),x(n_t+1:end)])
    tit = append("n(r), dt = ",string(dt),"s, ","Kp= ",string(Kp), " Ri = ...
        ",string(alpha));
    title(tit)
    xlabel('r')
    ylabel('n')
%     legend({'t_0','t_1','t_2','t_3' 't_4','conv.'},'Location','southwest')
    hold on

    subplot(2,1,2);
%   plot(linspace(-R,R,2*n-1),[fliplr(x0(2:n)),x0(1:n)])
%     hold on
%     if sn > 1
%         for sn1 = 1:sn-1
%            plot(linspace(-R,R,2*n-1),[fliplr(x_snap(sn1,2:n)),x_snap(sn1,1:n)])
%         end
%     end
    plot([-1*fliplr(r(2:end)),r],[fliplr(x(2:n_t)),x(1:n_t)])
    set(gca,'Ydir','reverse');      % Reverse the axis for better looking
    tit = append("u(r), dt = ",string(dt),"s, Ri = ",string(alpha));
    title(tit)
    xlabel('r')
    ylabel('u')
%     legend({'t_0','t_1','t_2','t_3' 't_4','conv.'},'Location','southwest')
%     hold off
    hold on
    set(gcf, 'Position',  [750, 50, 600, 700])
    drawnow

figure(2)                                                   % plot Num and Q
    %plot flow rate
    subplot(2,1,1);
    plot(dt:dt:(length(Q)-1)*dt,Q(2:end)-Q(1))
    tit = append("flow rate difference with Q0(t), dt = ",string(dt),"s");
    title(tit)
    xlabel('t')
    ylabel('Q-Q0')
    set(gcf, 'Position',  [150, 50, 600, 700])
    drawnow
    hold on

    %plot cell number
    subplot(2,1,2);
    plot(dt:dt:(length(Num)-1)*dt,Num(2:end)-Num(1))
    tit = append("cell number difference with N0(t), dt = ",string(dt),"s, ","Kp= ...
        ",string(Kp));
    title(tit)
    xlabel('t')
    ylabel('Num-Num0')
    hold on
    drawnow

% figure(3)                                                   % plot Residual
```

```
%     plot(dt:dt:(length(RES)-1)*dt,RES(2:end))
%     title('Residual(t)')
%     xlabel('t')
%     ylabel('Res')
end
```

B.3 Plot combined picture Figure3.7 and 3.6

```
% ------------------------------------------------------------------------%
%               Plot resuls with different pz Ri and Q0                   %
% ------------------------------------------------------------------------%
%                          Author: Songrui LI                             %
%                            Date: July 20rd                              %
% ------------------------------------------------------------------------%
close all
clear
clc

load('data_pz_Q0.mat')
rng(0,'twister');

% linestyle
lt=["--",':','-.','-','--o',':*','-.*','-*'];

%linecolor
lc1=[207 57 89
    209 55 165
    173 54 210
    98 53 211
    52 79 212
    51 163 213
    50 214 183
    49 215 100
    80 216 48
    205 217 47
    218 169 49
    220 107 44
    222 42 42
    223 41 123
    ]/255;

lc2=[243 77 77
    243 239 53
    242 190 68
    123 214 78
    42 250 151
    98 194 194
    234 63 247
    203 107 148
    59 251 91
    ]/255;


% plot different color
lcolor=[-6 -4 -2 -1 0 .5 .75 1 2 3 4];
linest=[-4 -3 -2 -1 0 1 2 4];
lindic=[8 9 10 11 12 13 14 15];
figure(1)
for i=1:35
    name=append('p_z=',string(u(1,i)),' Q_0=',string(u(1,i)));

    u0=u(3:end,i);
    mc= lc2(u(2,i)== linest,:);  %marker color
    alpha0=alpha(u0~=0);
    u1=u0(u0~=0);
    lc=lc1(u(1,i)== lcolor,:);

    lin_indic=[0:0.5*lindic(u(2,i)== linest):0.5*201];  % all marker indice

```

```
    ax1=alpha0(alpha0==[0:0.5:0.5*(length(alpha0)-1)]');
    ay1=u1(alpha0==[0:0.5:0.5*(length(alpha0)-1)]');

    mi1l=ax1==lin_indic;
    mi1ll=false(length(ax1),1);
    for j=1:length(lin_indic)
        mi1ll=mi1ll|mi1l(:,j);
    end
    mi1=1:length(ax1);
    mi1=mi1(mi1ll);

    ax2=alpha0(alpha0≠[0:0.5:0.5*(length(alpha0)-1)]');
    ay2=u1(alpha0≠[0:0.5:0.5*(length(alpha0)-1)]');

    mi2l=ax2==lin_indic;
    mi2ll=false(length(ax2),1);
    for j=1:length(lin_indic)
        mi2ll=mi2ll|mi2l(:,j);
    end
    mi2=1:length(ax2);
    mi2=mi2(mi2ll);

    plot(ax1,ay1,'-o','MarkerFaceColor',mc,'MarkerEdgeColor',mc,'MarkerIndices', ...
        mi1,'MarkerSize',3,'Color', lc,'DisplayName',name);
    hold on
    plot(ax1(end),ay1(end),'r*','MarkerEdgeColor',mc);
    %legend('-DynamicLegend','Location','southwest');
    plot(ax2,ay2,'-o','MarkerFaceColor',mc,'MarkerEdgeColor',mc,'MarkerIndices', ...
        mi2,'MarkerSize',3,'Color',lc);
    if(isempty(ax2))
    else
        plot(ax2(1),ay2(1),'r*','MarkerEdgeColor',mc);
    end
    drawnow
end
xlabel('Ri');
ylabel('u(0)')
title('central velocity changing with Ri and pz and Q_0')

figure(2)
for i=1:29
    name=append('p_z=',string(N(1,i)),' Q_0=',string(N(1,i)));

    u0=N(3:end,i);
    mc= lc2(N(2,i)== linest,:);  %marker color
    alpha0=alpha(u0≠0);
    u1=u0(u0≠0);
    lc=lc1(N(1,i)== lcolor,:);

    lin_indic=[0:0.5*lindic(N(2,i)== linest):0.5*201];  % all marker indice

    ax1=alpha0(alpha0==[0:0.5:0.5*(length(alpha0)-1)]');
    ay1=u1(alpha0==[0:0.5:0.5*(length(alpha0)-1)]');

    mi1l=ax1==lin_indic;
    mi1ll=false(length(ax1),1);
    for j=1:length(lin_indic)
        mi1ll=mi1ll|mi1l(:,j);
    end
    mi1=1:length(ax1);
    mi1=mi1(mi1ll);

    ax2=alpha0(alpha0≠[0:0.5:0.5*(length(alpha0)-1)]');
    ay2=u1(alpha0≠[0:0.5:0.5*(length(alpha0)-1)]');

    mi2l=ax2==lin_indic;
    mi2ll=false(length(ax2),1);
    for j=1:length(lin_indic)
        mi2ll=mi2ll|mi2l(:,j);
    end
    mi2=1:length(ax2);
    mi2=mi2(mi2ll);

```

```
    plot(ax1,ay1,'-o','MarkerFaceColor',mc,'MarkerEdgeColor',mc,'MarkerIndices', ...
        mi1,'MarkerSize',3,'Color', lc,'DisplayName',name);
    %legend('-DynamicLegend','Location','southwest');
    hold on
    plot(ax1(end),ay1(end),'r*','MarkerEdgeColor',mc);
    plot(ax2,ay2,'-o','MarkerFaceColor',mc,'MarkerEdgeColor',mc,'MarkerIndices', ...
        mi2,'MarkerSize',3,'Color',lc);
    if(isempty(ax2))
    else
        plot(ax2(1),ay2(1),'r*','MarkerEdgeColor',mc);
    end
    set(gca, 'YScale', 'log')
    drawnow
end

xlabel('Ri');
ylabel('N(0)')
title('central concentration changing with Ri and pz and Q_0')
```

B.4 Calculate the amplitude

```
% ------------------------------------------------------------------------%
%             calculate amplitude after a set of calculation              %
% ------------------------------------------------------------------------%
% ------------------------------------------------------------------------%
%                          Author: Songrui LI                             %
%                            Date: Aug 30th                               %
% ------------------------------------------------------------------------%
close all
NPmax=zeros(length(T_loop),1);
Nf1=zeros(length(T_loop),1);
NP2max=zeros(length(T_loop),1);
Nf2=zeros(length(T_loop),1);
Pmax=zeros(length(T_loop),1);
f1=zeros(length(T_loop),1);
P2max=zeros(length(T_loop),1);
f2=zeros(length(T_loop),1);

for loop=1:5

[NP2max(loop),NPmax(loop)]=calculate_biggest_and_mid(10,1,N0_loop(loop,:),dt,298,298,300);
[P2max(loop),Pmax(loop)]=calculate_biggest_and_mid(10,2,u0_loop(loop,:),dt,298,298,300);

end

for loop=6:25

[NP2max(loop),NPmax(loop)]=calculate_biggest_and_mid(10,1,N0_loop(loop,:),dt,280,280,300);
[P2max(loop),Pmax(loop)]=calculate_biggest_and_mid(10,2,u0_loop(loop,:),dt,280,280,300);

end

for loop=26:42

[NP2max(loop),NPmax(loop)]=calculate_biggest_and_mid(10,1,N0_loop(loop,:),dt,250,250,300);
[P2max(loop),Pmax(loop)]=calculate_biggest_and_mid(10,2,u0_loop(loop,:),dt,250,250,300);

end

for loop=43:44

[NP2max(loop),NPmax(loop)]=calculate_biggest_and_mid(10,1,N0_loop(loop,:),dt,220,220,300);
[P2max(loop),Pmax(loop)]=calculate_biggest_and_mid(10,2,u0_loop(loop,:),dt,220,220,300);

end

NPmax=NPmax+1;
NP2max=NP2max+1;

```

```
clearvars -except titl T_loop NPmax Nf1 NP2max Nf2 Pmax f1 P2max f2 A D alpha Re ...
    Ri beta
save(append("f",titl));

function [Bigs, Midd]=calculate_biggest_and_mid(limi,figg,S,dt,t0,t_plot,t)
figure(figg)
hold on
t_sano=t_plot:dt:t;
plot(t_sano,S(t_plot/dt:t/dt));
title('Original signal S(t)')
xlabel('t')
ylabel('value')
drawnow
set(gcf, 'Position',  [750, 50, 600, 500])

SS=S(t0/dt:t/dt);
Bigg=max(SS);
Smal=min(SS);
Midd=Bigg/2+Smal/2;
Bigs=Bigg-Smal;
if(Bigs>limi)
    Bigs=0;
    Midd=0;
end
end
```

B.5 Generate Bode diagram

```
% ------------------------------------------------------------------------%
%                          Plot the bode diagram                          %
% ------------------------------------------------------------------------%
% ------------------------------------------------------------------------%
%                           Author: Songrui LI                            %
%                             Date: Aug 30th                              %
% ------------------------------------------------------------------------%
%%
%close all

name='Re=10';
PPlot(2*pi./T_loop,20*log10(NPmax-2),20*log10(NP2max),20*log10(Pmax),20*log10(-P2max),name,beta)

figure(3)
set(gca, 'XScale', 'log')

%%
figure(3)
ylabel('magnitude of u(0)/u_s /dB')
title('Bode plot of u(0) scaled by u_s')
%%
function PPlot=PPlot(alpha_loop,NPmax,NP2max,Pmax,P2max,name,beta)

figure(3)
hold on
plot(alpha_loop,P2max,'DisplayName',name)
% plot3(alpha_loop,beta*ones(44,1),P2max,'DisplayName',name)
legend('-DynamicLegend','Location','southwest');
    ylabel('magnitude of u(0)/dB')
    xlabel('\omega/rad*s^{-1}')
    title('Bode plot of u(0)')
drawnow
end
```

Bibliography

[Bee20] Martin A Bees. Advances in bioconvection. *Annual Review of Fluid Mechanics*, 52:449–476, 2020.

[CLS75] S Childress, M Levandowsky, and EA Spiegel. Pattern formation in a suspension of swimming microorganisms: equations and stability theory. *Journal of Fluid Mechanics*, 69(3):591–613, 1975.

[FBH20] Lloyd Fung, Rachel N Bearon, and Yongyun Hwang. Bifurcation and stability of downflowing gyrotactic micro-organism suspensions in a vertical pipe. *arXiv preprint arXiv:2001.08072*, 2020.

[GLS+10] H Christopher Greenwell, LML Laurens, RJ Shields, RW Lovitt, and KJ Flynn. Placing microalgae on the biofuels priority list: a review of the technological challenges. *Journal of the royal society interface*, 7(46):703–726, 2010.

[HB02] NA Hill and MA Bees. Taylor dispersion of gyrotactic swimming micro-organisms in a linear flow. *Physics of Fluids*, 14(8):2598–2605, 2002.

[HP14] Yongyun Hwang and TJ Pedley. Stability of downflowing gyrotactic microorganism suspensions in a two-dimensional vertical channel. *Journal of fluid mechanics*, 749:750–777, 2014.

[Hun98] J.C.R. Hunt. Lewis fry richardson and his contributions to mathematics, meteorology, and models of conflict. *Annual Review of Fluid Mechanics*, 30(1):xiii–xxxvi, 1998.

[HWZ+18] Shuhao Huo, Zhongming Wang, Shunni Zhu, Qing Shu, Liandong Zhu, Lei Qin, Weizheng Zhou, Pingzhong Feng, Feifei Zhu, Wei Qi, and Renjie Dong. Biomass accumulation of chlorella zofingiensis g1 cultures grown outdoors in photobioreactors. *Frontiers in Energy Research*, 6, 06 2018.

[Kes84] John O Kessler. Gyrotactic buoyant convection and spontaneous pattern formation in algal cell cultures. In *Nonequilibrium cooperative phenomena in physics and related fields*, pages 241–248. Springer, 1984.

[Kes85a] John O Kessler. Co-operative and concentrative phenomena of swimming micro-organisms. *Contemporary Physics*, 26(2):147–166, 1985.

[Kes85b] John O Kessler. Hydrodynamic focusing of motile algal cells. *Nature*, 313(5999):218–220, 1985.

[Kes86a] John O Kessler. The external dynamics of swimming micro-organisms. *Progress in phycological research*, 4:258–307, 1986.

[Kes86b] John O Kessler. Individual and collective fluid dynamics of swimming cells. *Journal of Fluid Mechanics*, 173:191–205, 1986.

[MF03] A Manela and I Frankel. Generalized taylor dispersion in suspensions of gyrotactic swimming micro-organisms. *Journal of Fluid Mechanics*, 490:99–127, 2003.

[MTB+09] Giancarlo Marafioti, Sihem Tebbani, Dominique Beauvois, Giuliana Becerra, Arsene Isambert, and Morten Hovd. Unscented kalman filter state and parameter estimation in a photobioreactor for microalgae production. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 7, 01 2009.

[PK90] TJ Pedley and John O Kessler. A new continuum model for suspensions of gyrotactic micro-organisms. *Journal of fluid mechanics*, 212:155–182, 1990.