An exploration of moderate *Re* plane Couette turbulence

John R. Elton[*]

School of Physics
Georgia Institute of Technology,
Atlanta, GA 30332-0430
(Dated: April 29, 2007)

The `channelflow` code [1] for determining equilibria and unstable periodic orbits of moderate *Re* plane Couette turbulence is implemented and applied. A look at modification of flow parameters and initial conditions has been made to further investigate phase space properties for plane Couette flow.

Georgia Tech PHYS 4699:
CHAOS, AND WHAT TO DO ABOUT IT
special problems project, spring semester 2007

I. PLANE COUETTE FLOW

Plane Couette flow describes the motion of a fluid constrained by two long plates moving in opposite directions (along the streamwise x axis) at the same constant speed. The distance between the plates in the wall-normal y direction is chosen to be 2, with one plate at 1 and the other at -1. Periodic boundary conditions are imposed on the streamwise x and spanwise z directions which allows for one section of the flow to be analyzed. Given a set of fluid constants and initial conditions the Navier-Stokes equations may be solved numerically to find the velocity field as a function of time (see fluid notes, sect. V). One of the tools for performing this task is the `channelflow` code `couette.cpp` [1] which works as outlined in sect. I A. The velocity field $u(x, y, z)$ can be expanded as a set of orthogonal basis functions

$$u(x, y, z) = \sum a_n \Phi_n(x, y, z) \qquad (1)$$

where the basis set Φ_n spans the same finite-dimensional function space as the chosen $N_x \times N_y \times N_z$ grid. Because of the periodicity in x and z the basis functions are of the form

$$\Phi_{j,k,l} = \phi_l(y) e^{2\pi i (jx/L_x + kz/L_z)} \qquad (2)$$

where j and k are Fourier modes and ϕ_l is a polynomial in y which satisfies the boundary conditions. The coefficients are represented in `channelflow` as `ColumnVectors`. For computations it is often necessary to write the velocity fields as a DNS spectral expansion called a `FlowField` object,

$$u(x, y, z) = \sum_{jkl} \hat{u}_{jkl} T_l(y) e^{2\pi i (jx/L_x + kz/L_z)}, \qquad (3)$$

where the spectral coefficients $\hat{u}_{jkl}$ can be converted back and forth from `FlowFields` to `ColumnVectors` through

[*]Electronic address: `gtg960p@mail.gatech.edu`

the functions `field2vector` and `vector2field`. Use of this is made in the program `couette.cpp` where one can compute the L^2 norm of the expansion coefficients and it is the same as the L^2 norm of the velocity field *from* the laminar state. The L^2 norm $\|f\|$ (familiar Hilbert space from Quantum Mechanics) is defined for a function f on a volume V [2] as

$$\|f\|^2 = \frac{1}{V} \int_V |f|^2 \, dx \, dy \, dz. \qquad (4)$$

A. Program couette.cpp

The program `Couette.cpp` is used to integrate plane Couette flow from an initial field for a specified time. I outline here (with C++ notation) the important parts of this program.
// Define flow parameters
const Real Reynolds = 400.0;
const Real nu = 1.0/Reynolds;
const Real dPdx = 0.0;
const Real Ubulk = 0.0;
Here we have the first important parameter inputs. The Reynolds number (19) determines the viscosity of the flow. We also see the incompressibility condition (18).
// Define DNS parameters
flags.initstepping = CNRK2;
The numerical integration method is a combination of the Crank-Nicholson and Runge-Kutta 2 methods.
// Define size and smoothness of initial disturbance
Real spectralDecay = 0.3;
Real magnitude = 0.1;
int kxmax = 3;
int kzmax = 3;
The spectral decay number determines the rate at which the spectral coefficient in y decays back to the laminar state. The coefficient $\hat{u}_l$ is defined as

$$|\hat{u}_l| = \mathrm{rand}[0, 1] * (\mathrm{spectraldecay})^l \qquad (5)$$

and thus we see that the dependence of $\hat{u}_l$ on spectral decay number is the slope when $\log |\hat{u}_l|$ is plotted against l.
// Construct data fields: 3d velocity and 1d pressure
cout $\ll$ "building velocity and pressure fields..." $\ll$ flush;

FlowField u(Nx,Ny,Nz,3,Lx,Lz,a,b);
FlowField q(Nx,Ny,Nz,1,Lx,Lz,a,b);
cout $\ll$ "done" $\ll$ endl;

Velocity and pressure fields are created for the given box size (N_x, N_y, N_z) as `FlowFields` (3).
// Perturb velocity field
u.addPerturbations(kxmax,kzmax,1.0,spectralDecay);
u *= magnitude/L2Norm(u);

After the initial `FlowField` objects are set up the perturbations are added and u is updated.
// for (Real t=0; t$\le$ T; t += n*dt)
cout $\ll$ " t == " $\ll$ t $\ll$ endl;
cout $\ll$ " CFL == " $\ll$ dns.CFL() $\ll$ endl;
cout $\ll$ " L2Norm(u) == " $\ll$ L2Norm(u) $\ll$ endl;
cout $\ll$ "divNorm(u) == " $\ll$ divNorm(u) $\ll$ endl;

These are the printed outputs that show up when the program is run. "$t =$ " clearly shows the time step that the program is on. "CFL" is a number related to the flow parameters which needs to be between 0 and 1. For most cases it takes a value around 0.4 - 0.5. It can be approximately given as $CFL \approx \frac{\triangle t u}{\triangle x}$. "L2Norm" (4) gives the distance of the solutions from laminar flow. It is interesting to watch the variations in L2Norm as spectral decay number is varied. It always starts our at the value 0.1 and for large spectral decay rates goes quickly to 0, but grows for decay rates ~ 0.5. "divNorm" is a number which should be very close to zero, $O \sim 10^{-16}$.

II.   WALL LAW AND VERTICAL VORTICITY

Upon reading Gibson's "Upper/lower/laminar branch" blog [3], I found that for plane Couette flow one may linearize the Navier-Stokes equations about a base flow by letting

$$\vec{v} \to U(y)\hat{x} + \vec{v}(x,t) \qquad (6)$$

where the base flow is $U(y)\hat{x}$. In this approximation, how does the base flow respond to a variation in the flow parameters? The theoretical approximation for the base shear flow is the 'Wall Law'. It predicts

$$U(y) \sim y \qquad (7)$$

for $y < 10$ and

$$U(y) \sim 5 + 2.5 * log(y) \qquad (8)$$

for $y > 10$. Here I look at the Wall law program in `channelflow`, which computes the average velocity profile $U(y)$ of a pressure-driven flow and compares it to the theoretical approximation for $U(y)$. This program allows one to investigate the behavior of the base flow and check that it remains fairly stable.

The vertical vorticity, η, is defined by

$$\eta = (\nabla \times \vec{v}) \cdot \hat{y}, \qquad (9)$$

so according to the following simple calculation, a 'wall-law' base flow should contribute nothing to the vertical vorticity. That is, the base flow should be stable with respect to a variation in the flow parameters.

$$\vec{U}(y) = y\hat{x}, \qquad (10)$$

$$\nabla \times \vec{U} = \hat{x}(\frac{\partial U_z}{\partial y} - \frac{\partial U_y}{\partial z}) - \hat{y}(\frac{\partial U_z}{\partial x} - \frac{\partial U_x}{\partial z}) + \hat{z}(\frac{\partial U_y}{\partial x} - \frac{\partial U_x}{\partial y})$$

$$= -\frac{\partial U_x}{\partial y}\hat{z} = -\hat{z},$$

$$(\nabla \times \vec{U}) \cdot \hat{y} = \eta = 0. \qquad (11)$$

No vertical vorticity. This being the case, if one plots the mean velocity profile against y, the theoretical approximation should be more or less returned regardless of the magnitude of the parameters, though the parameters strongly affect the non-laminar part of the flow. This seems somewhat obvious, but it is a good check nevertheless. To check it I have run the `wall-law` program and plotted results for several different cases in which the Reynolds number, the perturbation magnitude, and the spectral decay rate were varied against a base flow with no perturbations. One can see from Figs. 1 and 2 that despite considerable changes in the perturbation parameters and Reynolds number, there is little deviation from the wall law. In fact the perturbation changes are hardly even visible. One can see the slight changes due to the variation in Reynolds number, but to obtain even this small deviation, the Reynolds number was decreased by an order of 100. This confirms the $\eta = 0$ calculation for the mean base flow.

III.   PCF-UTILS METHODS AND RESULTS

I have been able to produce some phase space trajectories for random initial fields, and random initial conditions along eigenvectors from equilibria. First I will document the use of the various `channelflow` programs for the phase space flows, and then give some plots and results.

A.   How to implement

The documentation and descriptions here are similar to those provided in the `PCF-utils` repository, but with specifics and details to running on my machine.
Method I : A random initial condition
1.) Create a random initial field.
This is done by the program `randomfield` which takes the arguments of box size, magnitude, and smoothness of field. To run:
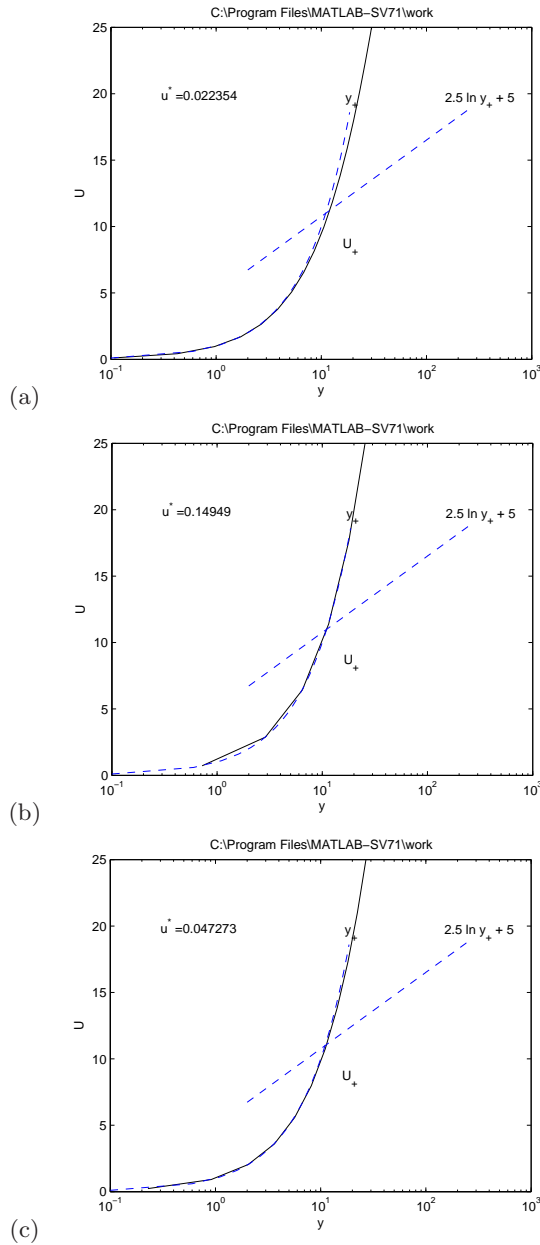./randomfield.x -Nx 48 -Ny 35 -Nz 48 -a 1.14 -g 2.5 -s 0.5 -m 0.3 urand.48.35.48

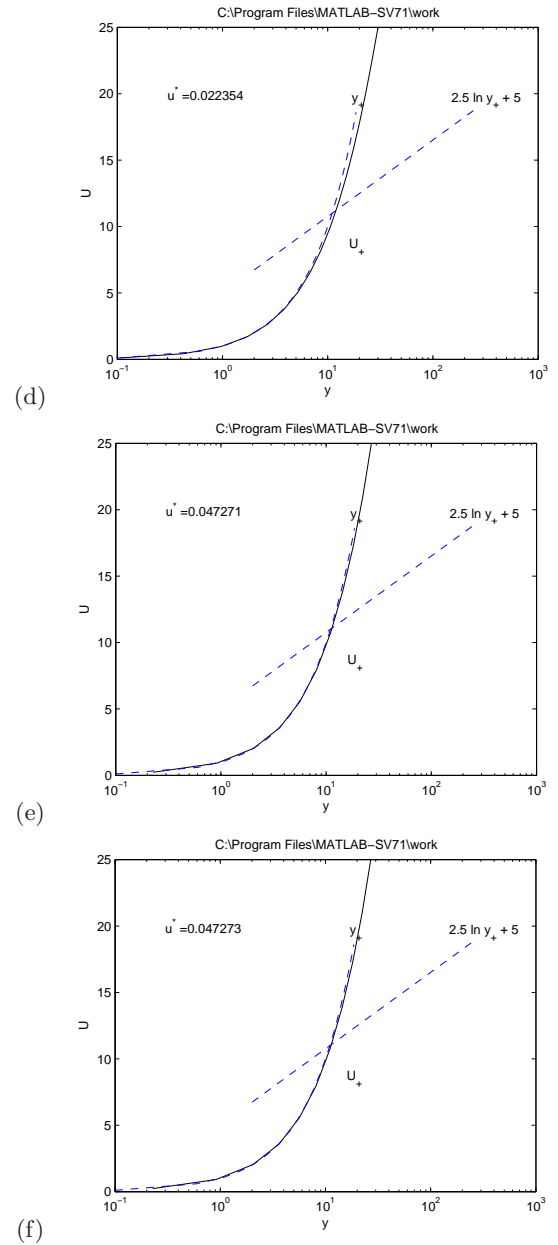FIG. 1: Variation in Re with perturbation magnitude = 0.01, decay rate = 0.01. (a) Base flow (b) Re = 4,000 (c) Re = 40

FIG. 2: Perturbation variations at Re = 40. (a) Base flow (b) Magnitude = 1.0, decay rate = 0.01 (c) Magnitude = 0.01, decay rate = 0.1

where urand.48.35.48 is the output file name.
Random field satisfies the boundary and zero-divergence conditions.

2.) From the initial field, integrate the Navier-Stokes equations. Program `couette` does this for a specified time interval and saves the flowfield in data/u1 as filename u1. The path data/u1 needs to be created before running. To run:
./couette.x -T0 0 -T1 400 -o data/u1 -l u1 urand.48.35.48
Program couette takes a rather long time to run.

3.) From a linearly independent set of fields on the trajectory, create an orthonormal basis. Program `makebasis` outputs this set to a specified directory. To run:
./makebasis.x -o basis1 data/u1/u10 data/u1/u110 data/u1/u120 data/u1/u130
where {u10, u110, u120, u130} is the basis set.

4.) Project fields onto the basis set for a specified interval of time. Program `projectseries` inputs the basis and outputs to the file u1.asc. To run:
./projectseries.x -T0 0 -T1 400 -bd basis1/ -Nb 4 -d data/u1 -ul u1 -o u1.asc

5.) With the file u1.asc at hand, all that is left is to load into matlab and graph. In order to do this matlab

needs to have `PCF-utils` in its path. In the matlab window type:

path(path,'C:/cygwin/home/OWNER/workspace/channelflow/PCF-utils')

Next load the file by entering:

load u1.asc

Finally to plot the trajectory enter:

plot3(u1(:,1), u1(:,2), u1(:,3));

Method II : A random perturbation from an equilibrium

1.) Suppose we already have an equilibrium point uUB, and an associated eigenfunction, UBef1, such as for the upper branch. Adding a perturbation along the eigenfunction simply involves taking a linear combination of the two (in which the perturbation coefficient is small) using the program `addfields`. To run:

./addfields.x 1 uUB 0.01 UB_ef1 uUB_0.01_ef1

where 1 and .01 are the coefficients. uUB_0.01_ef1 is the output field.

2.) Now take this field uUB_0.01_ef1 close to the equilibrium and go back to step 2.) of method 1 to perform the integration, make the basis set, project, and plot.

B. Method 1 results: 3/22/07

Using method 1 I have added trajectories for several different random initial fields, see Fig. 3. The parameters I use in `randomfield` are smoothness and magnitude of the initial field. For Fig. 3(a), values of $s = 0.5$, $m = 0.8$ are used. For Fig. 3(b) I decreased magnitude to $m = 0.3$ and smoothness to $s = 0.1$.

The difference in the plots for these varied values is quite interesting. There is a similar dense area at about $y = 0.15$. More could be extracted from many more comparisons of nearby values and longer integration times. There may be a particular value of either of these parameters for which the structure of the state space flow changes quickly.

C. Method 1 results: 4/29/07

Upon longer integration times of Fig. 3 more can be seen about the suspicious region in Fig. 4. I have attempted to bound this region by projection onto the xy and yz planes to give the rough values $0.1 \leq x \leq 0.15, 0.15 \leq y \leq 0.2, 0 \leq z \leq 0.05$. As can be see from Fig. 4, upon running the time out to $T = 400$ the trajectories do eventually leave this region, but not for quite some time. It is curious that each random trajectory finds this region and stays trapped for some time. This seems to suggest that the region must be fairly strongly attracting. In Fig. 5 I have added Fig. 4(b) along with a phase space plot of the upper branch in order to get a better feel for where this region lies. In comparison with the upper branch plot my figure is in an entirely different
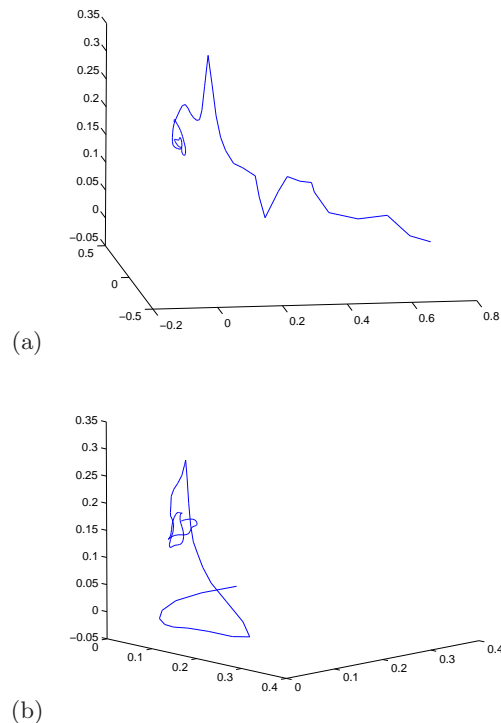

(a)

(b)

FIG. 3: Trajectories for a random initial condition for smoothness and magnitude values of (a) $s = 0.5$, $m = 0.8$, $T = 100$ (b) $s = 0.1$, $m = 0.3$, $T = 100$
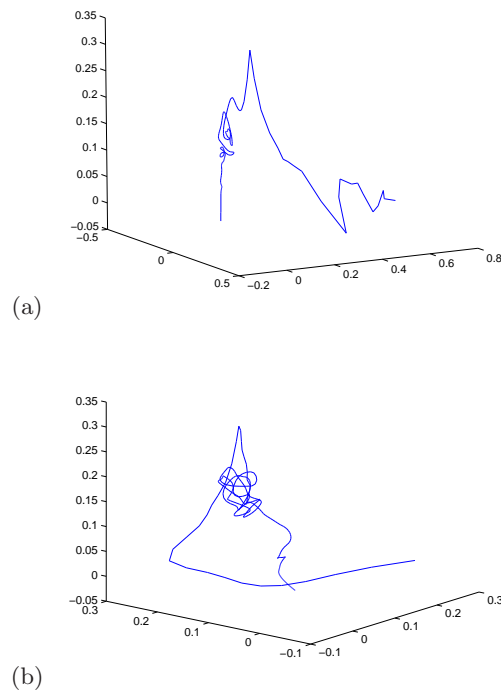

(a)

(b)

FIG. 4: Trajectories for a random initial condition for smoothness and magnitude values of (a) $s = 0.5$, $m = 0.8$, $T = 400$ - file `u1.ff` (b) $s = 0.1$, $m = 0.3$, $T = 400$ - file `u2.ff`
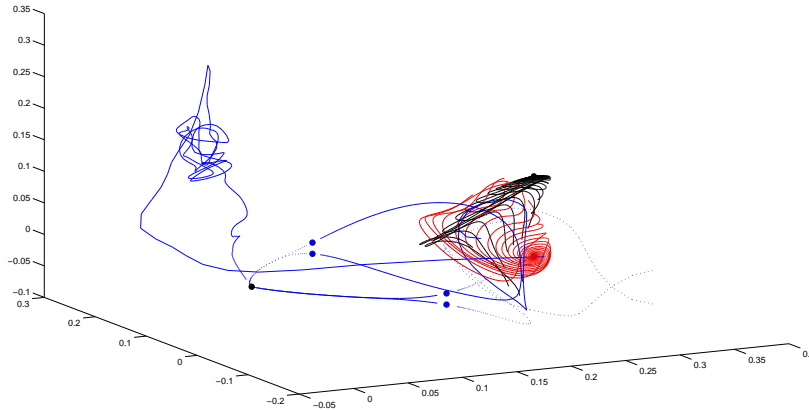
FIG. 5: UB and u2

region of the space. There may be an interesting behavior connecting the regions or possibly another unstable manifold in the area near `flowfield` u2.

D. Method 2 results 4/29/2007

Starting from Waleffe's upper branch, as recomputed at higher accuracy by Gibson [4], I have added plots for perturbations along each of the 7 eigenfunctions in Figs. 6 and 7. The difficulty is determining and computing an equilibrium (I have used a precomputed Waleffe's upper branch) and deciding where to start. This having been done already, I have the following in Figs. 6 and 7. The region connecting the UB and the NB would be an interesting place to look as it is a little less well understood. For this it would be necessary to know which eigenfunction points in that direction. It is necessary to compare these plots with previously computed data to make sure they are sensible and to get more of a feel for what the behavior means. The spiral behavior at the onset of (e) and (f) would suggest a comparatively large imaginary part of the eigenvalues for these two. Sure enough, from Table 1 of Gibson's blog "State space portraits of the LB, NB, UB" [5], eigenfunctions (e) and (f) of Fig. 7) have eigenvalues $\lambda = 0.01539 + i(0.28418)$ which have much larger imaginary part than any of the others.

In Figs. 8 and 9 perturbations from the same eigenfunctions have been plotted only this time using the *same* basis for each. The chosen basis is one which sustains typical upper branch behavior for the given flow parameters. This allows for a comparison between these plots and previously computed upper branch data in, for example, Gibson [5].

IV. DISCUSSION

The preceding results summarize my efforts at discovering and attaining information about the dynamics of plane Couette turbulence. From the outset my approach has been to try something, see what happens, and possibly adjust. The nature of the problem forces this somewhat "hit or miss" technique. The most important results produced in this way are the random initial condition trajectories from Fig. 4 and the trajectories from equilibrium in figures Figs. 8 and 9. The 'knotted' region in Fig. 4 would probably be the most interesting property to look at in future investigations. I would like to have been able to draw more comparisons and conclusions between my data and previously computed data from other sources, and to give more results, but ultimately time and complexity of the problem did not allow this. The methods set up would however provide a nice starting point for a future investigation project by myself or another. The method relies heavily on the use of all of the `channelflow` programs aforementioned as well as PCF data which has already been compiled and analyzed. Numerical computation power is therefore very important in investigating this problem.
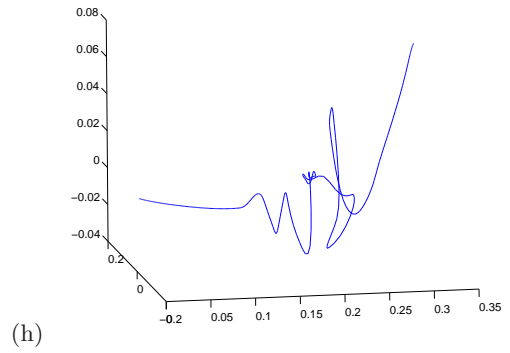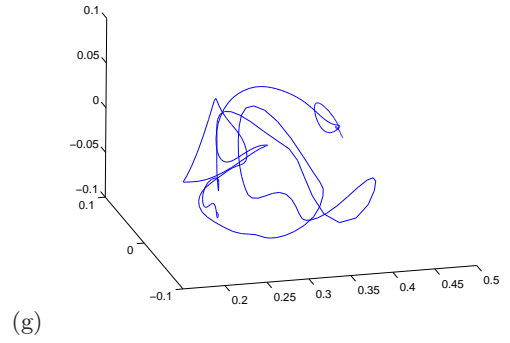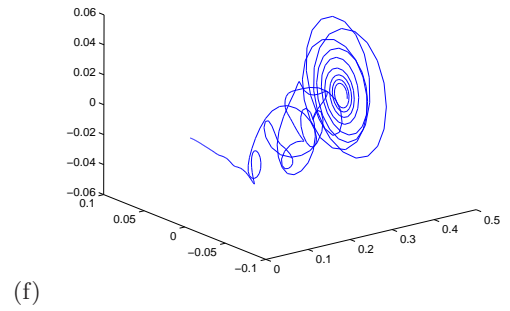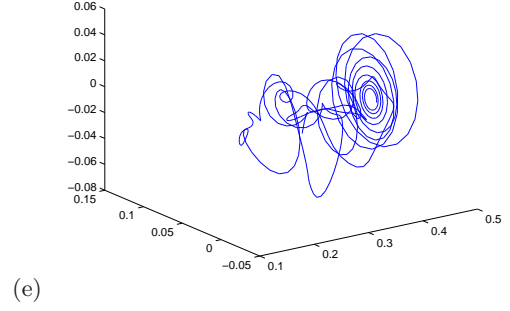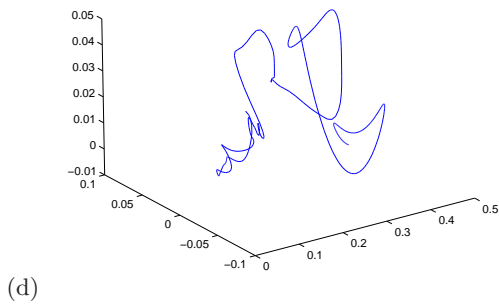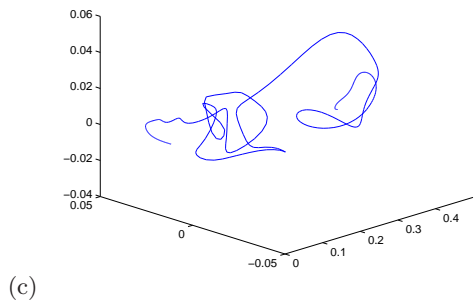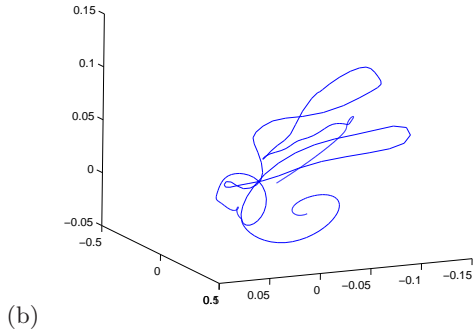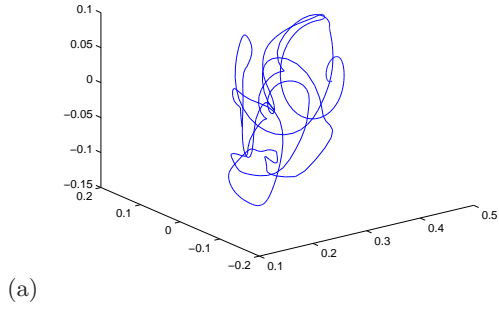
(a)

(b)

(c)

(d)

FIG. 6: Trajectories along eigenfunctions from upper branch equilibrium (a) eigenfunction 1 file uUB1.ff need to remake (b) eigenfunction 2 file uUB2.ff need to remake (c) eigenfunction 3 file uUB3.ff need to remake (d) eigenfunction 4 file uUB4.ff

(e)

(f)

(g)

(h)

FIG. 7: Trajectories along eigenfunctions from upper branch equilibrium (e) eigenfunction 5 file `uUB5.ff` (f) eigenfunction 6 file `uUB6.ff` (g) eigenfunction 7 file `uUB7.ff` (h) eigenfunction 0 file `uUB0.ff`
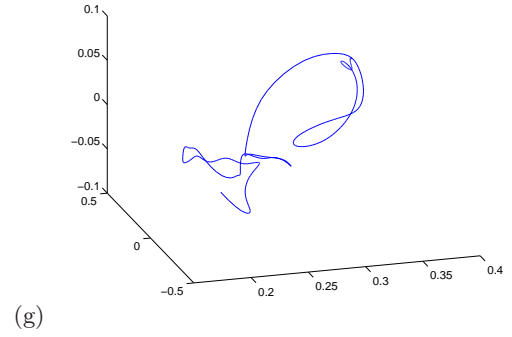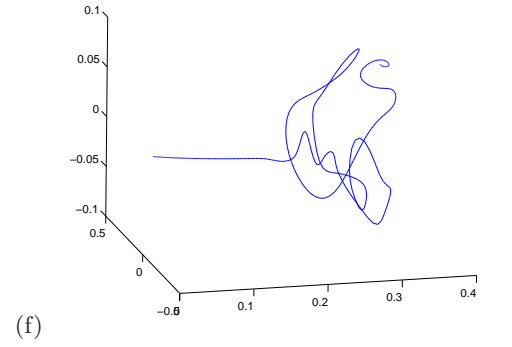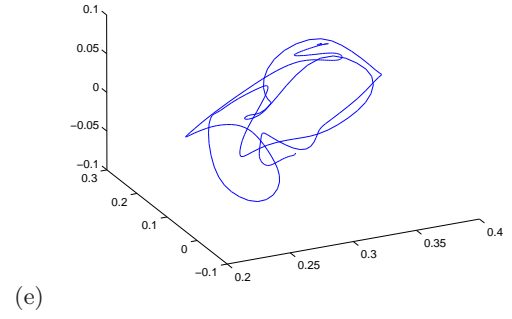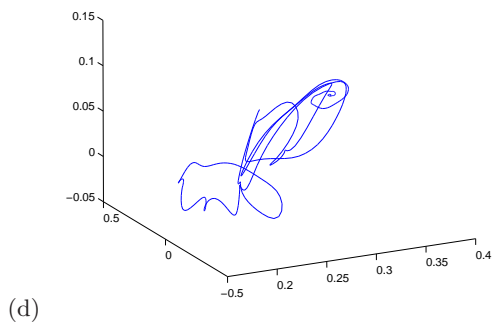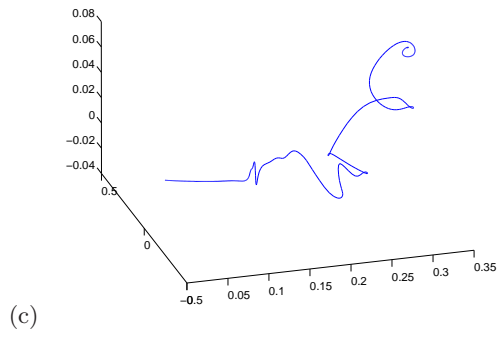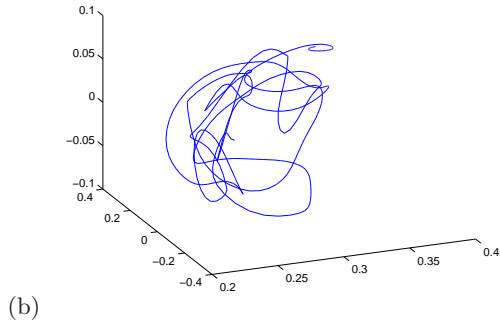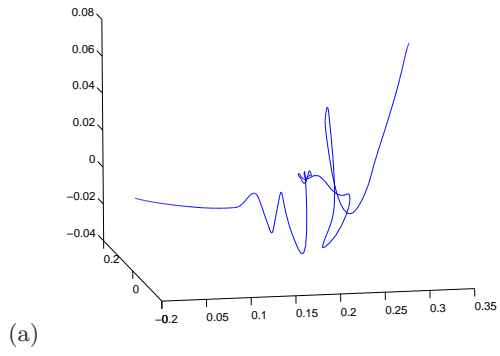
(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)
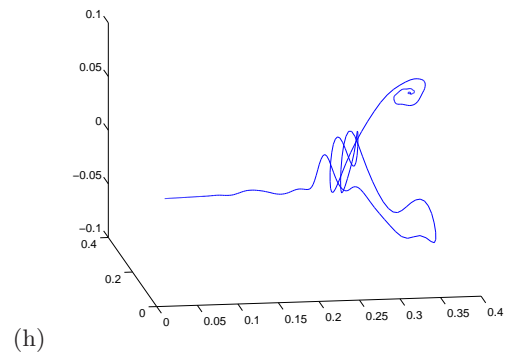
FIG. 8: Trajectories along eigenfunctions from upper branch equilibrium (a) eigenfunction 0 file `uUB0.asc` (b) eigenfunction 1 file `uUB1.asc` (c) eigenfunction 3 file `uUB3.asc` (d) eigenfunction 5 file `uUB5.asc`

FIG. 9: Trajectories along eigenfunctions from upper branch equilibrium (e) eigenfunction 7 file `uUB7.asc` (f) eigenfunction 2 file `uUB2.asc` (g) eigenfunction 4 file `uUB4.asc` (h) eigenfunction 6 file `uUB6.asc`

V. APPENDIX: SOME NOTES ON FLUIDS

I have compiled some reference notes on the material which I have been reading out of Lautrup [6] that can be referred back to as a guide or quick reminder.

A. Conservation of Mass

The rate at which mass is gained in a volume V is equal to the rate of loss through its surface S.

$$\frac{d}{dt}\int \rho dV + \oint \rho(\vec{v}\cdot\hat{n})dS = 0 \qquad (12)$$

Using Gauss' Theorem, this can be reformulated as the continuity equation,

$$\frac{\partial \rho}{\partial t} + (\vec{v}\cdot\nabla)\rho = -\rho\,\nabla\cdot\vec{v} \qquad (13)$$

For an incompressible velocity field the divergence of v is 0 and a slightly simplified continuity equation can be used. Defining the 'material time derivative' as $\frac{D}{Dt} = \frac{\partial}{\partial t} + \vec{v}\cdot\nabla$, the comoving acceleration is given as

$$\frac{D\vec{v}}{Dt} = \frac{\partial \vec{v}}{\partial t} + (\vec{v}\cdot\nabla)\vec{v} \qquad (14)$$

The nonlinear term $(\vec{v}\cdot\nabla)\vec{v}$ is known as the 'convective' or 'inertia' term. It is the acceleration that is due to the material transporting the particle. In cartesian coordinates it can be written as,

$$(\vec{v}\cdot\nabla)\vec{v} = (v_x\frac{\partial v_x}{\partial x})\hat{i} + (v_y\frac{\partial v_y}{\partial y})\hat{j} + (v_z\frac{\partial v_z}{\partial z})\hat{k} \qquad (15)$$

For a steady flow, $\frac{\partial \vec{v}}{\partial t} = 0$ and the inertia term is the only source of acceleration. Finally, from Newton's second law, we have Cauchy's equation of motion for a force density f:

$$f = \rho\left(\frac{\partial \vec{v}}{\partial t} + (\vec{v}\cdot\nabla)\vec{v}\right) \qquad (16)$$

The Cauchy equation is the governing equation for the motion of all fluids.

B. Viscosity, Reynolds Number Re, and Navier-Stokes

A fluid which flows along the x axis with velocity $v(y)$ which is independent of the coordinate x is said to be laminar. The shear stress between layers due to friction is a measure of the viscosity of the fluid. The force density in Cauchy's equation is made up of a pressure gradient and of the stress tensor, which incorporates a viscosity term $\nu\nabla^2 v$. ν is known as the kinematic viscosity of the fluid. Combining all of these terms and assuming the density of the fluid to be constant we arrive at the incompressible Navier-Stokes equations:

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v}\cdot\nabla)\vec{v} = -\frac{1}{\rho}\nabla p + \nu\nabla^2 v \qquad (17)$$
$$\nabla\cdot\vec{v} = 0. \qquad (18)$$

The relation between the inertia of a fluid and its viscosity gives rise to the Reynolds Number

$$Re = \frac{\rho(\vec{v}\cdot\nabla)\vec{v}}{\eta\nabla^2 v} \qquad (19)$$

This ratio implies that large Re fluids flow freely while small Re fluids are highly viscous. For a flow between 2 plates separated by a distance d (problem at hand) with average flow velocity U, the Reynolds number can be given as

$$Re = \frac{Ud}{\nu}. \qquad (20)$$

[1] J. F. Gibson, Tech. Rep., Georgia Inst. of Technology (2008), Channelflow.org.
[2] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry* (Cambridge Univ. Press, Cambridge, 2012), 2nd ed.
[3] J. Gibson (2007), upper/lower/laminar branch research blog.
[4] J. Gibson (2007), plane Couette flow data.
[5] J. Gibson (2007), research blog for UB, LB, NB state-space portraits.
[6] B. Lautrup, *Physics of Continuous Matter* (Institute of Physics, Bristol, 2006).