École Normale Supérieure de Lyon

Master of Sciences
1st year
Internship 2005-2006
Julien Salort

Georg-August Universität Göttingen

Low frequency oscillation in turbulent Rayleigh-Bénard cubic convection cell

Abstract — In this internship, I worked with PIV (particle image velocimetry) to study a cubic Rayleigh-Bénard cell at high Rayleigh number. The flow has interesting properties such as turbulence, anomalous scalings and large scale structures. The goal was to get some information on the low frequency oscillation.

Keywords — Hydrodynamics, Convection, Turbulence, Particle-Image-Velocimetry

Georg-August Universität Göttingen
Friedrich Hund Platz 1.
37077 Göttingen
Deutschland
`http://www.uni-goettingen.de`
Supervisor : Andreas Tilgner

May 1st — July 22nd 2006

Contents

1 Introduction

1.1 Generalities on Rayleigh-Bénard convection

1.1.1 Experimental apparatus

A fluid (gaz or liquid) is set between two plates kept at two temperatures T_1 and T_2 (see figure 1). The upper plate is cold and the lower plate is warm. If the temperature difference is big enough, a buoyancy-driven convection structure can be observed: if a fluid particle near the bottom plate gets perturbed and moves slightly up, it is then surrounded by colder and thus denser liquid. Therefore buoyancy pushes it even higher. While it moves upwards, the particle gets colder and we get the opposite phenomenon: a slight perturbation downwards will trigger buoyancy forces that will push the particle even lower. This process is summarized on figure 2. The exact physical problem is highly non-linear, which leads to many interesting and unexplained phenomena.

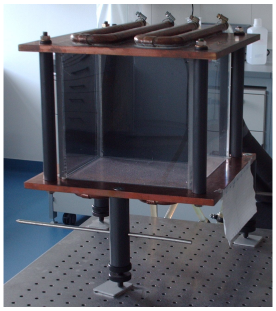


Figure 1: The Rayleigh-Bénard cell used for all the measurements

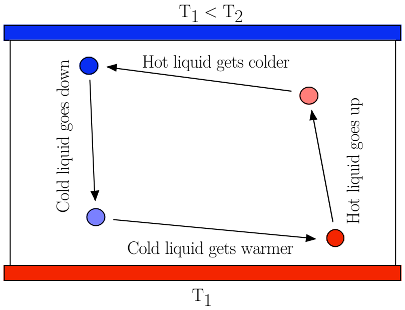


Figure 2: Basic principle of buoyancy-driven thermal convection

1.1.2 Dimensionless numbers

Other forces act against the formation of buoyancy-driven structures. Dissipation slows down fluid movements and thermal conduction reduces the efficiency of convective transport. We can define a dimensionless number to compare buoyancy to dissipation and thermal conduction: the Rayleigh number:

$$\mathrm{Ra} = \frac{\text{Buoyancy}}{\text{Dissipation}}$$

Buoyancy B is given by:

$$B = \Delta\rho \cdot g \cdot V$$

where $\Delta\rho$ is the density gradient, g is the gravitation field and V the volume of a fluid particle. The density gradient is linked to the temperature gradient:

$$\Delta\rho = \rho_{\mathrm{ref}} \cdot \alpha_{\mathrm{ref}} \cdot \Delta T$$

where α is the coefficient of thermal expansion and ΔT the temperature gradient.

The dissipation force W is given by Stokes' law:

$$W \sim \eta d v = \eta d^2 \frac{v}{d} = \eta \frac{d^2}{\Delta t}$$

where $\eta = \rho\nu$ is the dynamical viscosity, d is the displacement of the fluid particle and Δt is the typical duration of the particle displacement. We can derive Δt from thermodynamics:

$$\Delta Q = j d^2 \Delta t \simeq \left(\lambda \frac{\Delta T}{d}\right) d^2 \Delta t$$

and

$$\Delta Q = \rho d^3 C_v \Delta T$$

where λ is thermal conductivity and C_v the heat capacity. Therefore:

$$\Delta t \simeq \frac{\rho d^3 C_v \Delta T}{\lambda \frac{\Delta T}{d} d^2} = \frac{d^2}{\kappa}$$

where $\kappa = \frac{\lambda}{\rho C_v}$ is the thermal diffusivity. We finally get the dissipation force:

$$W = \eta \kappa$$

The Rayleigh number is then given by:

$$\mathrm{Ra} = \frac{B}{W} = \frac{\rho \alpha \Delta T g d^3}{\eta \kappa} = \frac{\alpha g d^3 \Delta T}{\kappa \nu}$$

We can also introduce the Prandtl number to characterize the properties of a given fluid,

$$\mathrm{Pr} = \frac{\nu}{\kappa}$$

it compares viscosity and conductivity.

1.2 Different regimes

1.2.1 Experimental results

Convection starts for Rayleigh numbers greater than a critical number Ra_c. Its value is determined experimentally. It is around 1700. For very low Rayleigh numbers, the fluid remains static and the temperature field is given by conductivity only. For $\mathrm{Ra} > \mathrm{Ra}_c$, convective rolls set in.

A transition is then found to a turbulent coherent state when the Rayleigh number gets larger. The critical value was found experimentally around $\mathrm{Ra}_t = 5 \times 10^7$. Qiu found coherent oscillations in the turbulent regime [Qiu and Tong, 2001]. He worked in particular on a well-defined low frequency oscillation in a cylindrical cell using one-point measurements with Laser-Doppler-Anemometers.

1.2.2 Theoretical models for the low frequency oscillation

A theoretical model was proposed a decade ago [Villermaux, 1995]: the thermal boundary layer instabilities near the top and bottom plates are coupled by the slow motion of the recirculation. We get two coupled oscillators. E. Villermaux solved this system and found that the frequency must be like Ra^γ with $\gamma \approx 0.49$ and the top and bottom boundary layer temperatures should oscillate in opposition of phase.

Those boundary layer instabilities are thought to be triggered by thermal plumes. Those plumes, carried by the large scale flow, hit the boundary layers and excite waves. Those waves further decay to spiraling swirls if inertia dominates over buoyancy or plumes if buoyancy dominates (see [Funfschilling and Ahlers, 2004, Haramina, 2005] for a further description).

1.3 Internship objectives

We want to study the low frequency oscillation in a cubic cell and possibly get several measurements for different Rayleigh numbers using PIV.

One possibility would be to make a Proper-Orthogonal-Decomposition of the velocity field (see [Berkooz et al., 1993, Chatterjee, 2000] for details about POD). This could allow us to simplify greatly the description of the velocity field if only a few modes were enough to describe the flow.

2 Velocity measurements

2.1 Particle Image Velocimetry

The principles of Particle-Image-Velocimetry (PIV) are described in more details in [Raffel et al., 1998]. We seed the cell with polyamid particles ($d = 50\ \mu$m). We assume that the particles don't have any effect on the flow and that they are transported with the fluid. We intend to find the velocity of the seeding particles. We suppose that the measured particle velocity matches the flow velocity. For that purpose, a sheet of light illuminates the particles and images are recorded at t and $t + \Delta t$ (see figure 3). The image is divided into small windows ("interrogation area"), possibly overlapping. The velocity is assumed to be homogeneous in the windows. On each window, we compute the 2D-cross-correlation of the image at t with the image at $t + \Delta t$:

$$C(\xi, \eta) = \sum_{x,y} I(t, x, y) I(t + \Delta t, x + \xi, y + \eta)$$

We expect to find a peak when $(\xi, \eta) = (\xi_0, \eta_0)$, the particle displacement between t and $t + \Delta t$. It is then easy to derive the particle velocity in the window:

$$\begin{cases} v_x = \frac{\xi_0}{\Delta t} \\ v_y = \frac{\eta_0}{\Delta t} \end{cases}$$

An example of velocity field obtained with this method is given on figure 4.

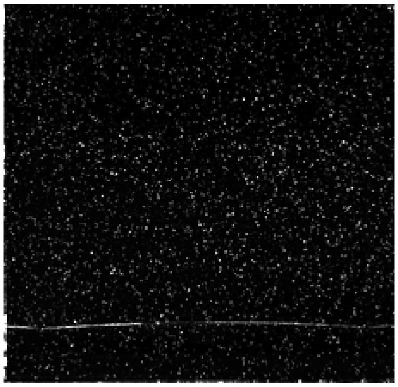


Figure 3: Example of image taken for PIV measurements

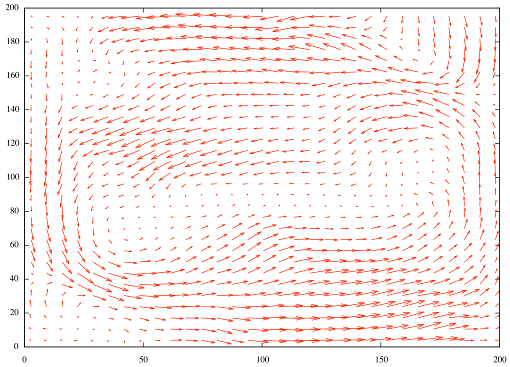


Figure 4: Example of velocity field obtained with PIV

It is also possible to use a finer technique called "Adaptative cross-correlation": a rough velocity field is first computed using the simpler cross-correlation technique described above with non-overlapping big interrogation areas (for example 64x64 pixels interrogation areas). The velocity field can then be calculated with a much higher precision with smaller and possibly overlapping interrogation areas. The windows for the $t + \Delta t$ image are offset so that they follow the particle displacement. The two image cross-correlates much better. It is possible to do that several times with decreasing window size to get a finer and finer velocity field.

2.2 Experimental setup

We use a cubic cell of dimension $L = 20$ cm, Plexiglas walls and copper plates. The temperature is controlled in the plate by a refrigerating/heating circulator (see figure 5). The circulator is heated/colded by two Lauda Ecoline Staredition RE 310 thermostats. The fluctuations of the circulator temperature are around 0.02°C. Copper thermal conductivity is good enough to assume that the plate temperature is homogeneous to better than 0.1°C. Water is used as the working fluid.

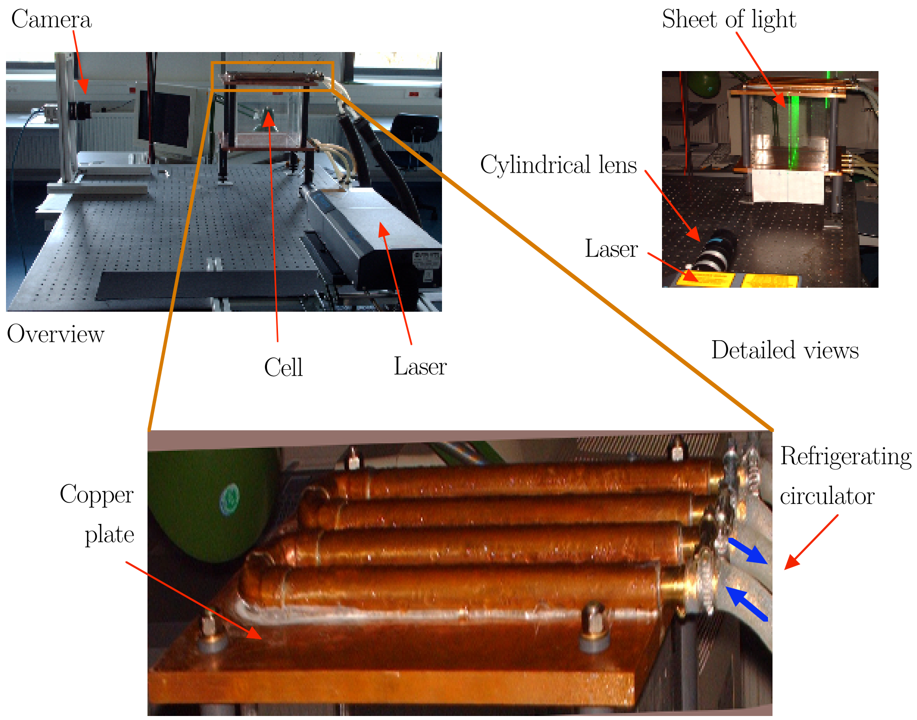


Figure 5: Experimental setup

The PIV system consists in a Class-IV YAG laser that delivers two pulses, a PCO PixelFly camera with double shuttering mode and a trigger controlled by a computer using a software from ILA[1], PIVSync. The camera can take two 1392x1024-pixel images very rapidly. The delay between the two images can vary between 10 ms and 90 ms. The frequency of the laser pulses can vary from 1 to 15 Hz. However, it is not possible experimentally to record more than 4 double-images per second in the computer. The computation is finally done with another software from ILA, VidPIV. It calculates the cross-correlation and the adaptative cross-correlation, comes with several subpixel estimators, velocity filters and can interpolate missing vectors.

2.3 Experimental parameters

2.3.1 Pulse separation

To make measurements, one has to choose a value for the pulse separation Δt. It is possible to use any value from 10 to 75 ms. When the pulse separation gets longer, the particle displacement is bigger and the precision on the velocity should be better. However, the particles may have gone out of the window and the non-linearity of their trajectories becomes more apparent, which brings confusion for tracking them. Therefore, one need to find a compromise.

To do that, I decided to make measurements for different values of Δt. For each of these measurements, I measured the mean signal-to-noise ratio (ie. amplitude of the peak/noise in the correlogram) to quantify the confusion in the particle displacement and I calculated the maximum velocity precision:

[1] Intelligent Laser Application

if one assumes that the subpixel estimator can locate the peak to approximately 0.1 pixel, then the best velocity precision for a given Δt is $\Delta v_{\text{pix}} = 0.1/\Delta t$. One can then derive the relative velocity error: $\Delta v_{\text{pix}}/v_{\text{mean}}$.

I've done 14 measurements for different pulse separations from 15 to 75 ms. Each of them lasted 20 seconds with a sampling rate of 1 Hz. I used 32x32-pixel interrogation areas. The results are plotted on figure 6.

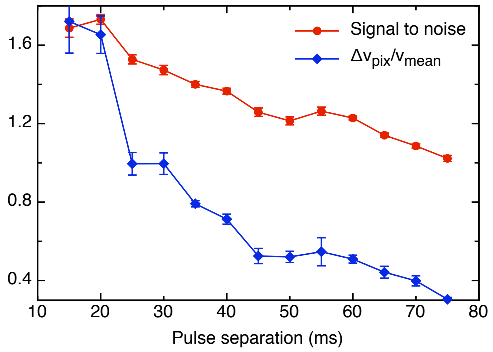
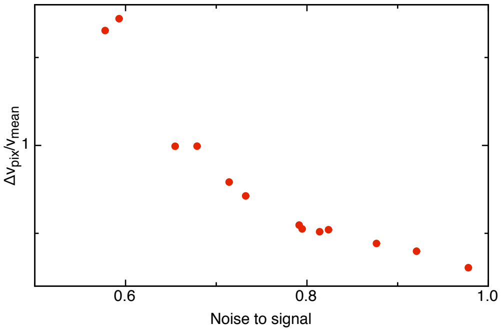


Figure 6: Measured signal-to-noise and relative velocity error for various pulse separation

To find the best parameter, we plot each measurement in a noise-to-signal/relative velocity error space. The measurement which is the closest from the origin is the best one. The origin would be the ideal measurement with no noise and no error (ie. infinite precision). To find the closest point, I plotted the distance to the origin $\sqrt{(\text{noise-to-signal})^2 + (\text{velocity error})^2}$ versus the pulse separation (see figure 7 and 8).



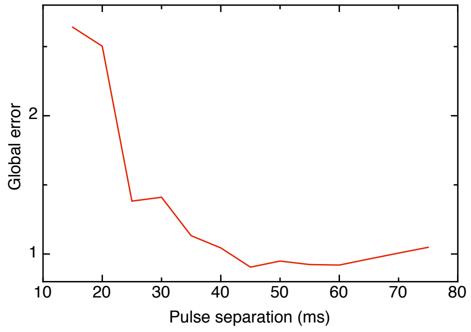


Figure 7: Plot of the measurements in the error space

Figure 8: Distance to the origin versus pulse separation

As one can see, the best pulse separation appears to be around 45 ms. However, the global error doesn't change much from 40 ms onwards. The most important factor of error is the precision on the peak coordinates. The noise brought by the confusion when the pulse separation is too big is not very important.

2.3.2 Temporal and spatial sampling rates

It is possible to choose a temporal sampling rate from 1 Hz[2] to 4 Hz (the laser can do up to 15 Hz but the computer turns out to be unable to record images so rapidly). To choose the appropriate sampling rate, one has to remember that we're interested in low frequency oscillation around 0.01 Hz. Therefore, a sampling rate of 0.02 Hz should be enough according to Shannon's theorem.

Additionally, the main limitation is disk space. Each double-image recorded by the camera takes 5.4 MB on disk. If we want to make statistics on the low frequency oscillation, we want to record for a long enough time. For example, to record 50 oscillations, we need to record for 50×100 s = 5000 s. With a sampling rate of 1 Hz, we get already 5000×5.4 MB = 27000 MB. Then, we need to store the VidPIV file which contains the calculated vector arrays for each step of the calculation (cross-correlation, velocity filters, adaptative cross-correlation, etc.). That's why I decided to use only 1 Hz sampling rate. It is possible to record with higher sampling rates for shorter times if one wants to study higher frequency oscillations.

It is also possible to choose a spatial sampling rate, ie space resolution, when we compute the velocity field from the raw images using techniques described above. Our limitation is now computer time. A practical problem is also that VidPIV is rather buggy. It tends to crash each time the user asks for a long calculation. The only solution is then to divide a long calculation into small ones and launch each of them one after the other by hand. It takes already more than two days to complete the calculation to get a vector every 32 pixels. If we want to increase the resolution to 16 pixels or to 8 pixels, the number of vectors to calculate increases exponentially and so does the computer time (approximately linear with the number of vectors to calculate).

That's why most of the final measurements I did use 64x64-pixel-wide interrogation areas every 32 pixels (each interrogation area overlaps with eight nearby interrogation areas) using 1 Hz sampling rate.

2.4 Experimental problems

2.4.1 Bubbles and tracing particles

The first experimental problem one has to deal with when using the cell is bubbles: there are two kinds of bubbles, big bubbles at the top the cell that may appear over time or when filling the cell and very small waterborne bubbles.

The former may change the boundary conditions and hence always have to be removed. To suppress them, one has to tilt the cell so that all the big bubbles move to the corner of the cell where a hole was made to allow us to fill the cell and add tracing particles. It is then easy to add some water and remove those bubbles.

The latter can actually reflect light and act as tracing particles provided they are not too big. The problem is that their size is quite random and their density is out of control. That's why it's better to suppress them too. To do that, I had to degas water with a pump before using it. It takes a couple of days for the water to be degassed completly.

Some experimental problems around the tracing particles also arise over time: at the beginning, the particles tend to deposit on the bottom plate. This is solved simply by adding more particles everyday. The phenomenon is slow enough to allow us to make measurements over several hours without problem. However, after about ten days, one can observe some threads of length around 1 cm. They are very likely to be agglomerated polyamid particles. When those threads pass through the sheet of light, a very big spot is recorded on the image, which prevents us from computing velocity vectors at some random points every now and then. The solution is then to empty and clean the cell and fill it back with some new degassed water.

[2] I noticed at the very end of the internship that it was actually possible to trigger the camera with fractional sampling rates, 0.25, 0.5 or 0.75 Hz

2.4.2 Peak locking

Introduction

The first statistical analysis one can do with the data is to calculate velocity histograms at several points of the cell. This has been done several times already (see [Qiu et al., 2003] for an example) therefore we expect to find one gaussian peak. However, as shown on figure 9, what we get doesn't look like a gaussian peak. It seems to consist in a superposition of several peaks, which additionally don't seem to be gaussian.

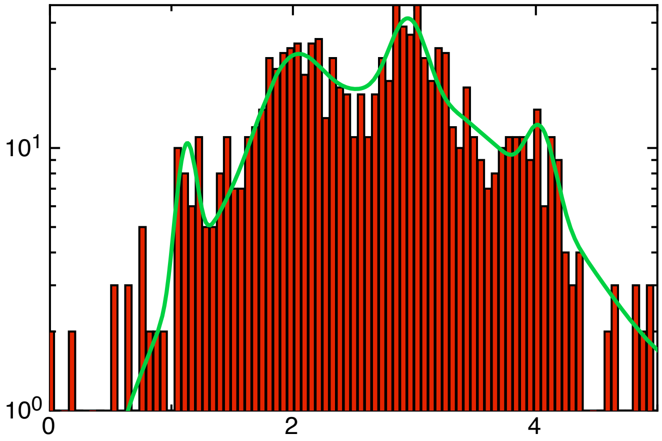


Figure 9: Velocity histograms showing peak locking

One can notice that those peaks are evenly spaced. They actually coincide with whole pixel values. This is due to a biais in the subpixel estimator usually referred to as "Peak locking": the value of the particle displacement (in pixels) tend to be rounded preferentially to integer values. Qiu didn't have this problem because he used LDA methods which naturally don't need to use this kind of subpixel estimators.

However, those results are consistent with what other people using PIV found (see [Christensen, 2004] for an example). Christensen's extensive study showed that the only statistical operation that is insensitive to peak locking is mean velocity profile. It is then of high importance to try to reduce peak locking errors as much as we can.

The impact of pulse separation

Christensen found that the effects of peak locking are stronger when fluctuating displacements are small and asserts that when the particle images are under resolved ($d_\tau / d_{\text{pix}} < 2$, where d_τ is the particle displacement and d_{pix} the pixel size), peak locking is a significant error independently of the choice of sub-pixel estimator.

That's why, the first thing I tried was to increase the pulse separation. I found in 2.3.1 that the ideal pulse separation was around 45 ms with respect to signal-to-noise and image sensibility. Additionally, while precision dramatically dropped down when the pulse separation was shorter, it was not too much affected when the pulse separation was longer. The idea is then to increase d_τ to prevent the images from being under-resolved and to lessen the effects of peak locking on statistical analysis.

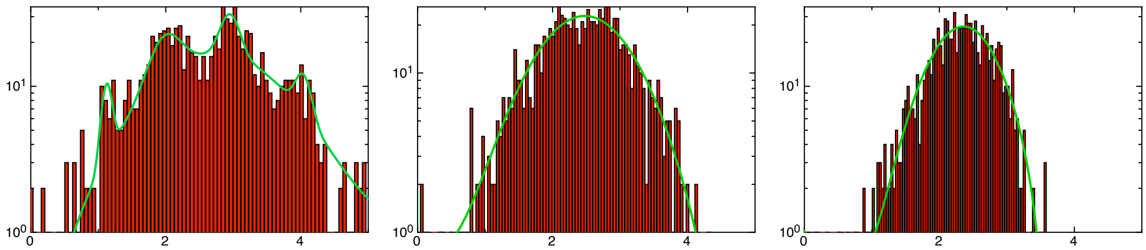
Unfortunately, this didn't really change the velocity histogram. Peak locking was still clearly visible on the histograms even for $\Delta t = 80$ ms (the largest value we can use with our PIV system).

Subpixel estimators

Several subpixel estimators are available in VidPIV: several gaussian estimators and a Whittaker estimator. The default is Whittaker and changing it to gaussian doesn't make the histograms better.

Adaptative cross-correlation and Velocity smoothing

However, two things do make histograms better: Adaptative cross-correlation (ACC) and most of all “Velocity smoothing” (VS) instead of bare cross-correlation (CC). VidPIV documentation states that their “Adaptative Cross-Correlation” algorithm does reduce peak-locking. The “Velocity smoothing” node works using a weighted mean that is rather similar to the interpolation method. This improve the overall accuracy of individual measurements. Smoothing both prior the application of adaptative cross-correlation and after dramatically improve the velocity histograms (see figure 10).



(a) CC only

(b) CC + VS

(c) CC + VS + ACC + VS

Figure 10: One point velocity histograms obtained with various methods

An important issue at this point is computer time. Even if the latest method (CC+VS+ACC+VS) seems to be the best, it is also the most time consuming. No doubt that a CC+VS+ACC+VS+ACC+VS method would be even better. However, considering that the internship only lasts 3 months, it was clear that the best methods were unaffordable from the computer time point of view. That’s why all the measurements were done using the CC+VS method.

Even though peaks are still visible, the “Velocity smoothing” node considerably reduces the effect of peak locking. This is hopefully enough to make other statistical analysis.

2.5 Simple properties of the velocity field

After having determined how to use the setup and what were the best parameters, I made four real measurements for several temperature differences (see table 1). Here is the experimental protocol for these final measurements:

- Degas distilled water with a pump for a couple of days
- Clean and fill the cell, tilt it to get rid of bubbles
- Make sure the cell is precisely perpendicular to the sheet of light to avoid reflexions on the back wall
- Make sure the sheet of light hit the cell exactly in the middle (where side-effects are less important)
- Set the thermostats temperature
- Set the laser frequency to 8 Hz with the synchroniser and the camera to take an image once every eight laser pulses (the camera sampling rate is therefore 1 Hz)
- Start the laser
- Wait about an hour to make sure the thermostats, the plate and the laser are stable
- Make sure the camera is focused on the sheet of light

- Acquire 5100 images with the “CamWare” software and save them to the external hard drive
- Create 17 VidPIV files. Import 300 CameWare images in each (VidPIV crashes if one tries to import the 5100 images in the same project file) and configure the appropriate linear mapping (this allows to get values in mm/s instead of pixels/s)
- On each file, compute Cross-Correlation, Velocity filters and velocity smoothing
- Export the vector arrays to comma-separated files for later data analysis

Before trying to study the “low frequency oscillation”, I computed velocity profiles and velocity average. They can be compared to what other scientists found earlier. This allowed me to make sure the measurements were accurate.

N	ΔT	Ra
1	10 K	1.1285×10^9
2	20 K	2.2571×10^9
3	30 K	3.3856×10^9
4	35 K	3.9499×10^9

Table 1: Final measurements: 1 Hz sampling rate, 5100 seconds

2.5.1 Velocity profile

Velocity profiles have already been studied in a Rayleigh-Bénard cell. In particular, Qiu found that the flow field could be divided into three regions (see [Qiu et al., 2000]): the thin viscous boundary layer, for $y/L \leq 0.04$, a central region for $0.25 \leq y/L \leq 0.75$ and an intermediate region for $0.04 \leq y/L \leq 0.25$.

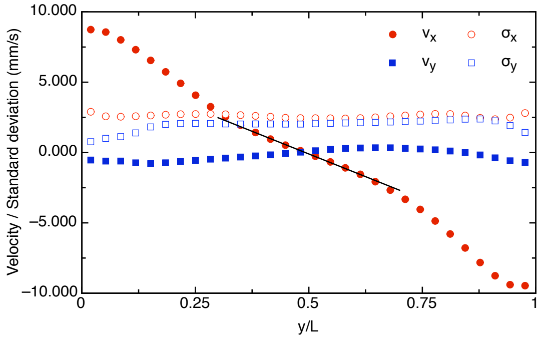


Figure 11: Mean velocity and standard deviation (vertical) profile for $x = \frac{L}{2}$

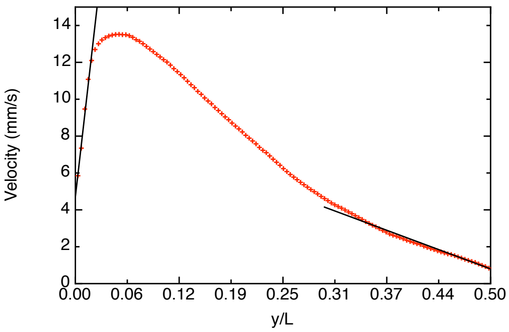


Figure 12: High resolution mean v_x (vertical) profile for $x = \frac{L}{2}$

I computed the velocity vertical profiles and the standard deviation vertical profiles at the middle of the cell (see figure 11). However, the viscous boundary layer was not to be seen. Therefore, I computed the velocity field with the highest resolution possible: the data obtained this way is about 4 GB, it is thus very difficult to make too many data analysis on it: it takes a lot of memory and a lot of time. Therefore, since velocity profiles are not our primary goal, I only computed v_x vertical profile with these high precision data (see figure 12).

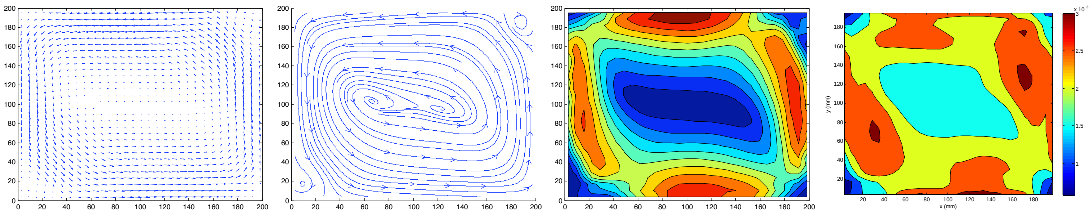
One can check the flow properties in these three regions:

1. **Viscous boundary layer** ($y/L \leq 0.04$)
   Qiu writes that v_x increases linearly and reach a maximum at $y/L = 0.04$. This is indeed what we get on figure 12. The precise value seems to be greater than 0.04 in our case. This may come from geometrical parameters. Qiu worked in a cylindrical cell, whereas we worked in a cubic cell.
2. **Central region** ($0.25 \leq y/L \leq 0.75$)
   According to Qiu, σ_x becomes larger than v_x in this region, the velocity fluctuations are approximately homogeneous, v_x is well fitted by a linear function and v_y remains rather small. This is also what we get on figure 11.
3. **Intermediate region** ($0.04 \leq y/L \leq 0.25$)
   In Qiu's experiments, the velocity fluctuations in this regions remained approximately the same as those in the central region, σ_x was small compared to v_x but was greater than v_y. This is also what we get. However, again, the figures do not match exactly. Qiu found $\sigma_y \simeq 4$ mm/s.

The velocity profiles we get are quite satisfactory. They match rather well with Qiu's velocity profiles. This tends to show that the measurements are accurate.

2.5.2 Velocity average

Another simple analysis one can do is to calculate the velocity average in the lighted plan (see figure 13). One can see the fundamental structure of the flow which consists in one big roll and two small recirculating vortexes on opposite edges.



(a) Vector representation (b) Stream representation (c) Scalar representation (d) Velocity fluctuations

Figure 13: Velocity average

One can notice additionally that the mean velocity field shows a symmetry with respect to the diagonal.

3 Low frequency oscillation

3.1 First observations

It is not actually possible to see this "low frequency oscillation" with bare eyes. Therefore, the first thing is to make sure we can measure it and to check that we get the same results as were observed earlier (by Qiu for example, see [Qiu et al., 2003]). So I wanted to look at one-point time-series and fourier spectra. I computed power spectra at various points in the cell. Some of them, mostly near the bottom and top plates, show indeed a clear low frequency peak (see figures 14 and 15). Some of them, particularly at the center of the cell, do not show any peak.

Although the setup is not really efficient to determine Ra-dependency (it is difficult to make a lot of measurements at various temperature, there is not thermistors to accurately determine the plate temperature, etc.), I decided to check if the low frequency oscillation I observed on figure 15 complied with the experimental law obtained by Qiu et al.,

$$\frac{f_0 L^2}{\kappa} = 0.167 \times \mathrm{Ra}^{0.47}$$

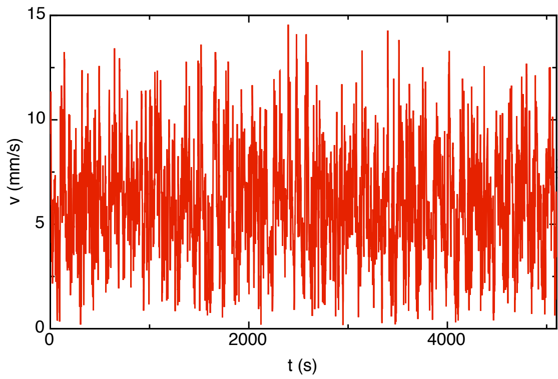


Figure 14: Velocity time-series at x=126 mm and y=36 mm

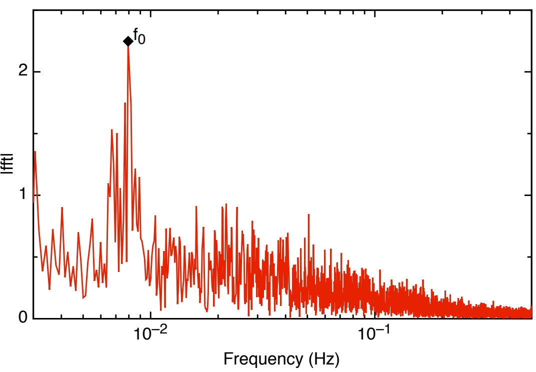


Figure 15: Velocity power spectrum at x=126 mm and y=36 mm

Therefore, I measured f_0 for each of the four measurements shown on table 1. Of course, a fit with only four data points does not prove anything. However, it is interesting to fit the data anyway to compare the numerical values of the coefficients, and therefore make sure the frequency values are the same as those observed by Qiu. The fit function shown on figure 16 is

$$\frac{f_0 L^2}{\kappa} = 0.23 \times \mathrm{Ra}^{0.42}$$

Considering how imprecise the fit was supposed to be, this result seems satisfactory. Additionally, geometrical parameters may have an impact on the coefficients.

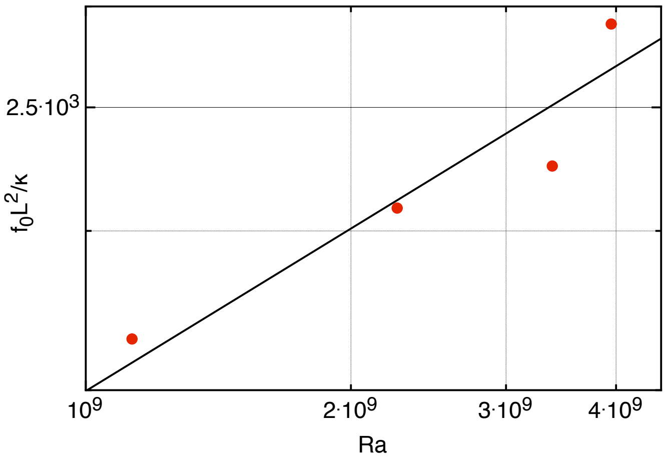
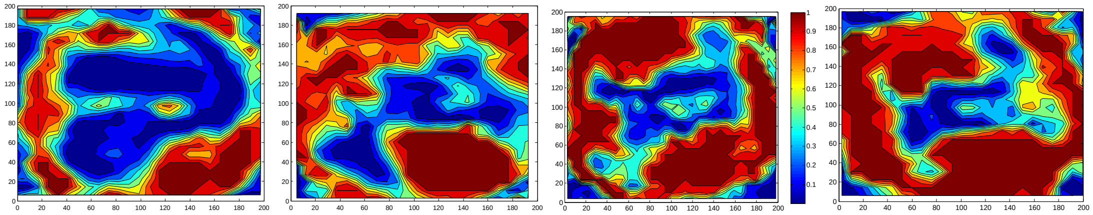


Figure 16: Oscillation frequency f_0 evolution with Rayleigh number (power-law fit)

3.2 Fourier analysis

3.2.1 Location of the peak

As mentioned in section 3.1, the frequency analysis does not reveal a low frequency peak everywhere in the cell. The peak seems to be easier to find near to bottom and top plates. Therefore, I tried to figure out exactly where the peak was. To do that, I plotted the amplitude of the fourier transform, $\hat{v}(f_0, x, y)$ in space (see figure 17).



(a) $\Delta T = 10$ K
(b) $\Delta T = 20$ K
(c) $\Delta T = 30$ K
(d) $\Delta T = 35$ K

Figure 17: Amplitude of the fourier transform $\hat{v}(f_0, x, y)$ for the four measurements

The edge between the areas with a peak (in red) and the areas without peak (in blue) shown in figure 17 is not very smooth, probably because 5100 images are not enough to have a precise statistics. Hence, it is difficult to comment on the differences between the four measurements. One can only give general qualitative observations that are true for all of them.

There is a low frequency oscillation in the intermediate region (see 2.5.1 for a definition). No peak can be observed in the central region. One can also compare the peak location (figure 17) with the velocity average and fluctuations (figure 13). In other words, the oscillation seems to be really strong inside the main roll.

3.2.2 Coherence function

An interesting function to find correlations and phase relations is the coherence function. The coherence function of two timeseries $f_1(t)$ and $f_2(t)$ is defined as,

$$\mathrm{Coh}_{f_1 f_2}(\nu) = \frac{\hat{f}_1(\nu) \cdot \hat{f}_2(\nu)^*}{\sqrt{|\hat{f}_1(\nu)|^2 |\hat{f}_2(\nu)|^2}}$$

where the hat represents the fourier transformation and the star the complex conjugate.

For example, one could calculate the coherence function between the top and the bottom intermediate regions to see how both oscillations are correlated. As one can see on figure 18, the amplitude of the coherence function has a clear peak for $f = f_0 \simeq 0.01$, which means that the oscillation near the top and the oscillation near the bottom are coherent. It is also possible to read the phase of the coherence function to know the phase difference. Apparently, the phase difference is around zero, which suggests that the top and bottom layers oscillate in phase.

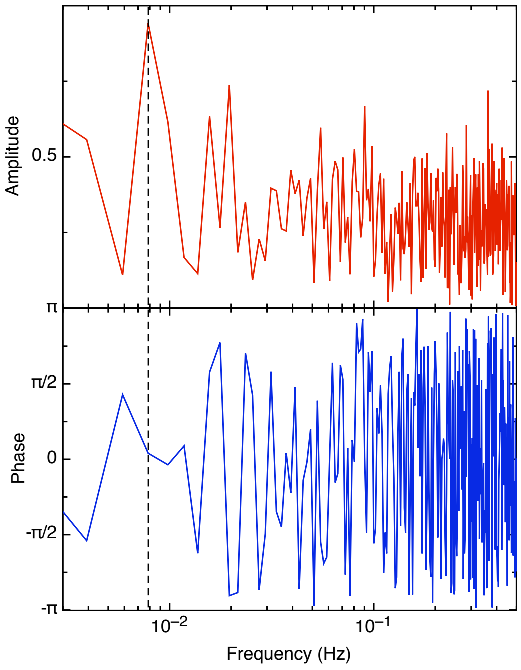


Figure 18: Coherence fonction between the velocity $\|\vec{v}\|$ at y = 182.06 mm and the velocity y = 17.113 mm

However, this is not what we expected since other people found previously that the top and bottom layers oscillated in opposition of phase:

1. a researcher of Andreas Tilger's group, Tomi Haramina, has been working on temperature measurements in this very Rayleigh-Bénard cell. For his PhD thesis [Haramina, 2005], he measured the temperature on a vertical line at the center of the cell and computed the coherence function. He found that the top and bottom layers oscillated with a phase difference around π.

2. other people, like G. Ahlers, measured the angle $\Theta(t)$ of the horizontal component of plume velocity [Funfschilling and Ahlers, 2004]. He found that $\Theta(t)$ near the top and bottom plates had the same frequency but were anticorrelated.

3. a decade ago, E. Villermaux proposed a simple theoretical model (see section 1.2.2). His model was based on experimental data where the top and bottom layers were found to oscillate in opposition of phase. His sources are [Castaing et al., 1989] and [Wu, 1991].

Yet, all those results can be consistent because all those people measured the temperature and not the velocity. If one assumes that the low frequency oscillation has something to do with emissions of plumes then one has to keep in mind that one gets emission of hot plumes near the bottom layer and cold plumes near the top layer.

It is unclear whether plume emissions speed up or slow down local velocity. We assume in the following discussion that it speeds it up. If it had slowed it down, it wouldn't have changed the result. When hot plumes are emitted, the temperature will first increase then decrease when the plume gets through, so will the velocity. Therefore, we expect the velocity and the temperature to be in phase. When cold plumes are emitted, the temperature will first decrease then increase when the plume gets through, whereas velocity will still first increase and then decrease. Therefore, we expect the velocity and the temperature to be in opposition of phase.

As a result, the velocity and the temperature should oscillate in phase near the bottom layer but in opposition of phase near the top layer. Since the top and bottom temperatures oscillate in opposition of phase, the top and bottom velocities should then oscillate in phase, which is indeed what we get on figure 18. Direct measurements of the phase difference between temperature and velocity were done by [Qiu et al., 2004].

Then, it would be interesting to calculate the coherence function of the horizontal component of the top and bottom velocities. Since the v_x axis has the same direction as the roll flow near the bottom boundary layer, it will increase then decrease exactly like $\|\vec{v}\|$ when a plume gets through. However, since the v_x axis has the opposite direction as the roll flow near the top boundary layer, it will decrease then increase. Therefore, v_x should behave like the temperature and top and bottom horizontal velocities should be found to oscillate in opposition of phase. We find indeed on figure 19 that the phase difference is around $-\pi$.

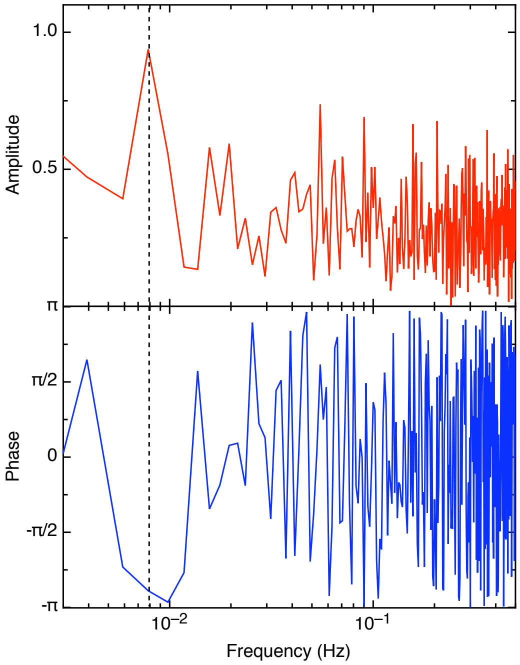


Figure 19: Coherence fonction between the horizontal velocity v_x at y = 182.06 mm and the velocity y = 17.113 mm

3.2.3 Phase differences

Introduction

To get a better idea of how this oscillation happens in space, it is interesting to calculate the phase everywhere in the cell and to plot the isophase lines. This could allow us to locate different structures in the cell and perhaps to have a better idea of how they are correlated.

First of all, I wanted to make sure the coherence function was a good tool to find phase differences. To do that, I tried to plot the isophase line of a known vector field. I defined a fake vector field as follow:

$$\begin{cases} v_x &= v \cdot \cos\alpha \\ v_y &= v \cdot \sin\alpha \\ v &= v_0 + \sin(2\pi f_0 t + \varphi(x,y)) + \\ & \quad \sin(2\pi f_1 t) + \\ & \quad \sin(2\pi f_2 t + \text{random noise}) + \\ & \quad \text{random noise} \end{cases}$$

where f_1 and f_2 are two other frequencies and $\varphi(x,y)$ a phase equal to zero everywhere except in three regions (see figure 20). The random noise is gaussian. The velocity field obtained this way contains several frequencies and some random noise. We are only interested in the phase of the f_0 oscillation.

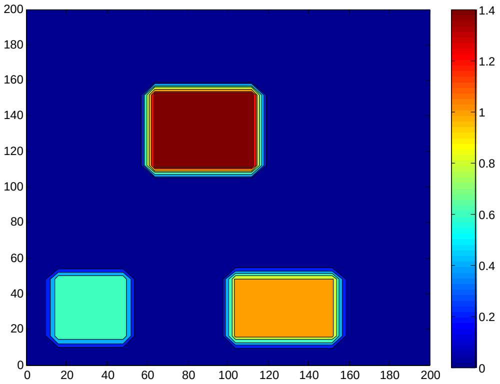


Figure 20: $\varphi(x,y)$ function used to generate a fake vector field

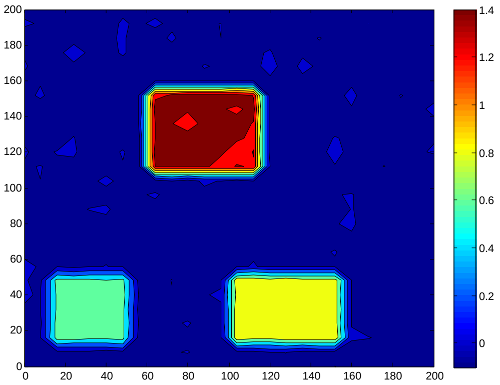


Figure 21: Isophase line calculated from the fake vector field using the coherence function

The isophase line found for this fake vector field by the coherence function method is given on figure 21. They are quite similar to the actual function we used. Therefore, one can assume that this method is good to determine the isophase lines for the real velocity field.

Isophase lines

The isophase lines are shown on figure 22. Like the peak localisation figures (section 3.2.1), it would have probably been better to have more than 5100 data points. However, one can draw some general conclusions:

- The top and bottom layers oscillate in phase. This oscillation in phase appear at the top left and bottom right corners. At the top right and bottom left, the small recirculating vortexes perhaps perturb the oscillation,
- The left oscillation seems to be in phase with the right oscillation, they have a phase around $\pi/2$ relative to the top and bottom oscillations.

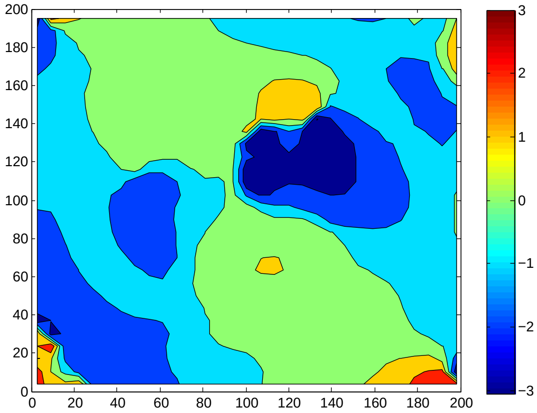


Figure 22: Isophase line of the low frequency oscillation of $\|\vec{v}\|$

3.2.4 Low-pass filtered movies

It is possible to make a movie with the data using Matlab. It is also possible to filter each velocity timeseries and to make a movie with the filtered data (see appendix C.1). The idea is to focus on the low frequency oscillation and to see what actually oscillates and how. It is not really possible to use a band-pass filter to keep only the low frequency oscillation because there are only a couple of values around f_0 in the fourier space, which would probably destroy the data too much. That's why, I made low-pass filtered movies.

One can use various representation for the velocity field: vector representation, streamline representation or scalar representation. They were shown previously on figure 13 for the velocity average. Thus, I made movies with all those representations. However, only the streamline representation brought easy qualitative observations. Some images extracted from this movie are shown on figure 23. In the other representations, one could see that the roll oscillate somehow but it was difficult to figure out any phase relations or any structure with bare eyes.

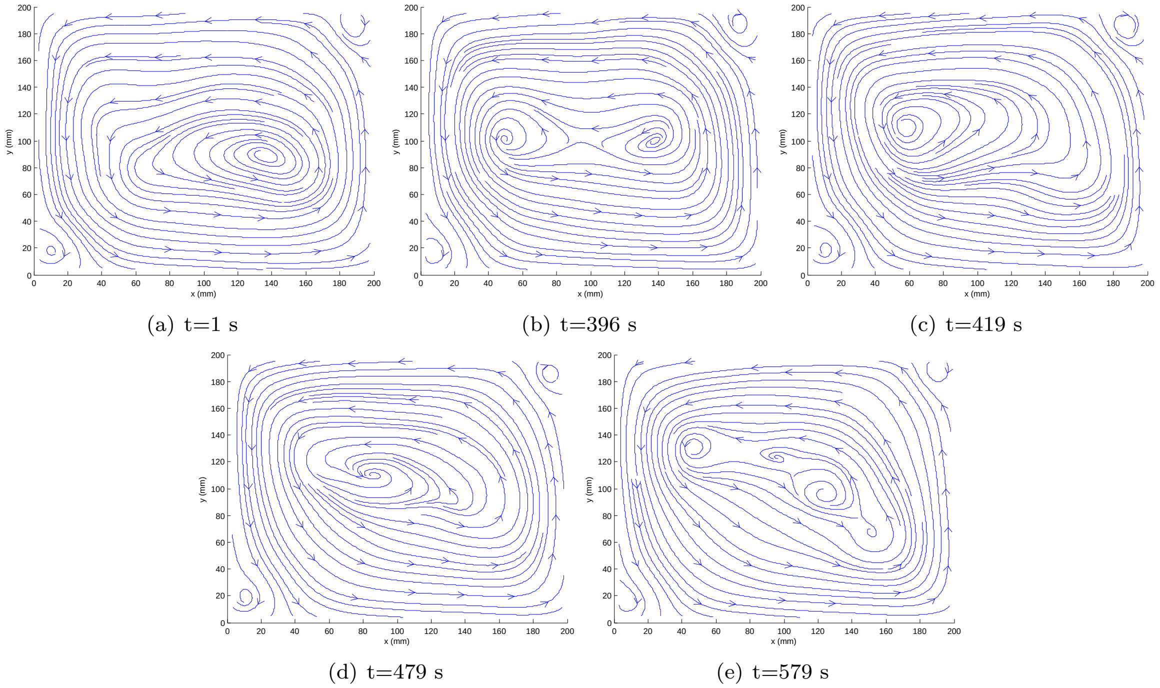
On the streamline representation, one can see the structure of the central region. This structure changes over time. One has to keep in mind that this is the central structure *averaged over approximately 100 seconds*. There might be other structures in the central region but they oscillate too fast and have been filtered. Sometimes, there is only one vortex, possibly on the right side as shown on subfigure 23(a), on the left side as shown on subfigure 23(c) or at the center as shown of subfigure 23(d). Sometimes there are several vorteces: two vorteces can be shown on subfigure 23(b) and up to four vorteces can be shown on subfigure 23(e).

Time	Structure
1 s	One vortex (right)
71 s	Two vorteces
88 s	One vortex (center)
112 s	Two vorteces
152 s	One vortex (right)
186 s	Two vorteces
214 s	One vortex (left)
234 s	Two vorteces
280 s	One vortex (right)
312 s	Two vorteces
405 s	One vortex (left)
435 s	Two vorteces
461 s	One vortex (right)

Time	Structure
498 s	Two vorteces
579 s	Three to four vorteces
609 s	One vortex (right)
627 s	Two vorteces
646 s	One vortex (right)
678 s	Two vorteces
714 s	One vortex (right)
736 s	Two vorteces
843 s	One vortex (right)
877 s	Two vorteces
971 s	One vortex (right)
1009 s	Two to three vorteces
1040 s	Two vorteces

Time	Structure
1095 s	One vortex (left)
1119 s	Two vorteces
1153 s	Three vorteces
1161 s	Two vorteces
1214 s	One vortex (left)
1266 s	Two vorteces
1298 s	One vortex (left)
1334 s	Two vorteces
1425 s	One vortex (left)
1440 s	Two vorteces
1519 s	One vortex (right)
1527 s	Two vorteces
1584 s	One vortex (left)

Table 2: Changes of structure in the central region during the first 1600 seconds



(a) t=1 s (b) t=396 s (c) t=419 s

(d) t=479 s (e) t=579 s

Figure 23: Some images extracted from the low-pass filtered streamline movie. The center region shows different kinds of structures

It is difficult to find a precise periodicity in those structure changes without a quantitative criterium that would allow a computer to decide how many structures are present in the central region. I tried to note by hand the moment at which a change of structure occurs for the first 2000 seconds. The raw result for the first 1600 seconds is shown on table 2. A graphical representation is shown on figure 24.

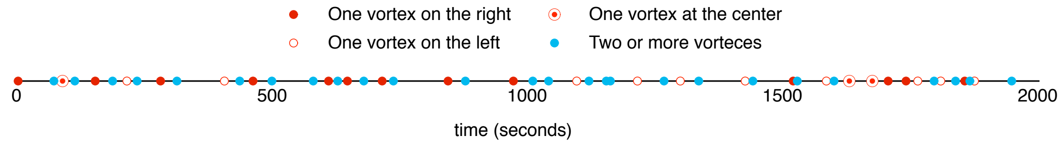


Figure 24: Graphical representation of the changes of structure in the central region for the first 2000 seconds

One can wonder if there is a connection between the low frequency oscillation and those changes of structure. Since the determination of the time at which the changes of structure occur is highly imprecise, it is difficult to know if their possible periodicity matches the periodicity of the large scale oscillation. However, it is possible to calcule the mean time interval between each one-vortex structure using the data from table 2. I found $\Delta t_{\text{mean}} = 94.93$ s. The associated frequency is 0.0105 which is close to f_0. Of course, if the time interval were more precise, we would know whether the large-scale oscillation frequency and the changes of structure in the central region frequency match or not. To do that, one should find a way to accurately determine the number of structures on each image.

3.3 Proper Orthogonal Decomposition

3.3.1 Introduction

Basic principles of POD

A more complete discussion of POD can be found in [Berkooz et al., 1993, Chatterjee, 2000]. Consider a fonction $a(x,t)$ that depends on both space and time. We want to rewrite it with separated variables, ie.

$$a(x,t) = \sum_{k=1}^{m} q_k(t) v_k(x) \tag{1}$$

where the v_k functions are modes, the q_k functions are time-dependent coefficients and m is the number of modes.

Additionally, we want both $(v_k)_k$ and $(q_k)_k$ to be orthogonal basis, ie

$$\int_{\mathbf{R}} v_{k_1}(x) v_{k_2}(x) \mathrm{d}x = \delta_{k_1 k_2}$$

It is possible to formulate the problem in terms of matrices. This is interesting because we're using Matlab to do the computation. Consider we have N sets of m simultaneous measurements. We define A, a $N \times m$ matrix such that A_{ij} is the i^{th} measurement of the j^{th} probe. The POD decomposition then consists in diagonalising the matrix A, ie.

$$A = U \Sigma V^T \tag{2}$$

such that U is an $N \times N$ orthogonal matrix, V is an $m \times m$ orthogonal matrix and Σ is a $N \times m$ diagonal matrix.

If one defines the matrix $Q = U\Sigma$, one can note that

$$A = QV^T \Leftrightarrow A_{ij} = \sum_{k=1}^{m} q_{k,i} v_{k,j}$$

This is the discrete version of the equation 1 with $q_{k,i}$ corresponding to $q_k(t)$ and $v_{k,j}$ corresponding to $v_k(x)$. Thus, the modes are given by the V matrix. The values in the diagonal matrix Σ are called singular values. Those values are non negative.

An interesting property of the POD decomposition is that the first n modes provide the best n terms approximation. Consequently, the fundamental mode is nothing but the time average. A lower-rank approximation of A is then given by:

$$A_k = U \Sigma_k V^T \text{ with } \begin{cases} (\Sigma_k)_{i,i} = 0 \text{ for } i > k \\ (\Sigma_k)_{i,i} = \Sigma_{i,i} \text{ for } i \leq k \\ (\Sigma_k)_{i,j} = 0 \text{ for } i \neq j \end{cases} \tag{3}$$

To calculate the U, Σ and V matrices, one can note the following properties:

- Premultiplying equation 2 with its transpose gives $A^T A = V \Sigma^2 V^T$. Therefore V is the matrix of the eigenvectors of $A^T A$
- Similarly, $AA^T = U \Sigma^2 U^T$. Therefore U is the matrix of the eigenvectors of AA^T
- The singular values, $\sigma_i = \Sigma_{i,i}$, are the squares of the eigenvalues of $A^T A$

Therefore, the POD decomposition can be derived from the regular eigenvalue decomposition of $A^T A$ and AA^T.

However, it is not necessary to do that by hand because Matlab provides a function for the POD analysis (also known as Singular Values Decomposition), `[U,S,V] = SVD(X)`.

POD for vector arrays

What we want to do is a POD analysis of the velocity field. We may find that only a few modes are enough to describe to flow behaviour. If it is the case, this may simplify greatly the study of the flow and particularly the low frequency large-scale oscillation we're interested in. To do that, we need to decompose a vector array $\vec{a}(x, y, t)$. In other words, the modes have to be vectors and equation 1 becomes:

$$\vec{a}(x, y, t) = \sum_{k=1}^{m} q_k(t)\vec{v}_k(x, y) \tag{4}$$

Equation 4 can be rewritten as two scalar equations:

$$\begin{cases} a_x(x, y, t) = \sum_{k=1}^{m} q_k(t)(v_k)_x(x, y) \\ a_y(x, y, t) = \sum_{k=1}^{m} q_k(t)(v_k)_y(x, y) \end{cases} \tag{5}$$

In other words, we want a proper orthogonal decomposition of two functions $a_x(x, y, t)$ and $a_y(x, y, t)$ and we want the time-dependant coefficients to be the same for those two functions. The discrete matrix form of this problem is:

$$\begin{cases} A_x = U\Sigma V_x^T \\ A_y = U\Sigma V_y^T \end{cases}$$

where A_x is the matrix of the measurements of the horizontal velocity component and A_y is the matrix of the measurements of the vertical velocity component, such that $(A_x)_{i,j}$ is the i^{th} measurement of the j^{th} probe. As one can notice, there is now only one indice for space. In equation 4, there were two variables x and y. This is not problem because space is discretised. Therefore, we can give an number to each position where velocity is calculated: $\vec{p_j} = (x_j, y_j)$.

Then, from a pratical point of view, one can derive this two dimensional POD from the scalar `SVD` function mentionned before using the following Matlab function:

```
function [U S Vx Vy] = mySvd(Ax, Ay)
    [n_t n_p]=size(Ax);
    A=[Ax Ay];
    [U S V] = svd(A);
    Vx=V(1:n_p,:);
    Vy=V(n_p+1:2*n_p,:);
```

One can easily check that the resulting matrices U, S, V_x and V_y have the desired properties.

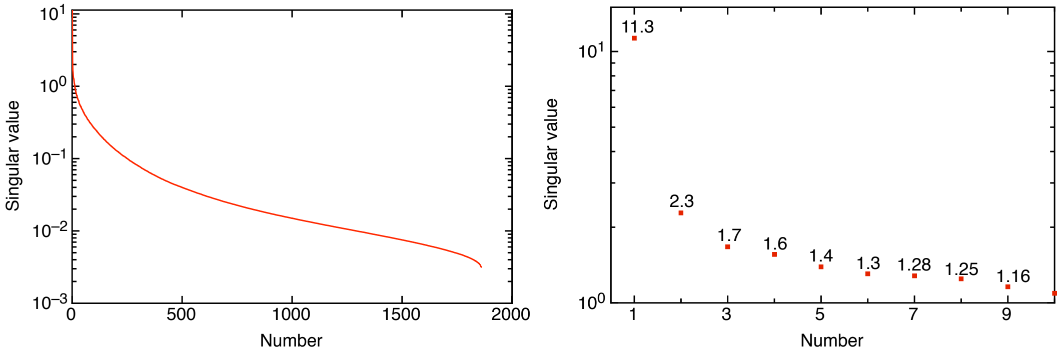
3.3.2 POD analysis of the velocity field

The results shown below were done using the Matlab script of appendix C.2 with the measurement number 3 (see table 1). The results obtained with the other measurements are similar.

Singular values

Singular values are positive numbers that give the weight of each mode. They give information on which modes are required to get a good approximation of the flow. Our interest would be to have the smallest number of modes possible. The singular values calculated from the POD analysis of our velocity field is shown on figure 25.

Unfortunately, restricting the data to a couple of modes is obviously impossible. There is less than a factor of 5 between the fundamental (ie. time average) $k = 1$ mode and the $k = 2$ mode. Then the singular values decrease very slowly. As one can see on subfigure 25(b), the first ten singular values are around 1 and 2. Then, we have $\sigma_{249} = 10^{-1}$ and $\sigma_{1291} = 10^{-2}$. In other words, if one wants a reasonable approximation, one needs 250 modes, if not 1300 modes (we could neglect modes only if their singular value is negligible).



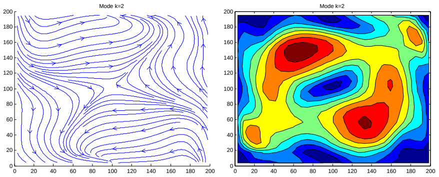
(a) Global view

(b) Detailed view

Figure 25: Singular values calculated from the POD analysis of the velocity field

Modes

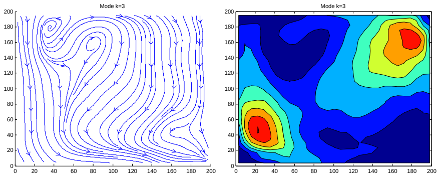
The first two non-fundamental modes are shown on figures 26 and 27. The fundamental mode is only the time-averaged velocity field. It was discussed previously on figure 13.



(a) Stream

(b) Module

Figure 26: $k = 2$ mode



(a) Stream

(b) Module

Figure 27: $k = 3$ mode

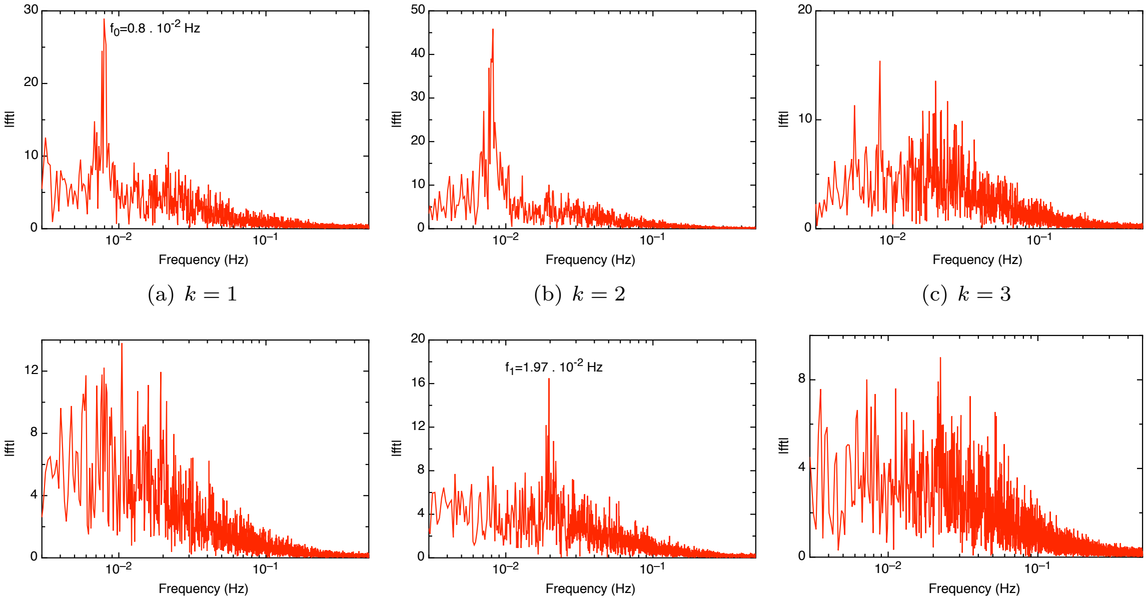
There are a couple of interesting points one can notice on those modes:

1. There is a stagnation point on the $k = 2$ mode. There were none on the fundamental one. Yet, one could notice on the low-pass filtered movies discussed in section 3.2.4 that a stagnation point appears and disappears over time. None can be seen on subfigure 23(a) or 23(c) but one can be seen on subfigure 23(b). Perhaps this comes from the contribution of this $k = 2$ mode whose weight might oscillate in time.

2. Both $k = 2$ and $k = 3$ modes are approximately symmetrical with respect to the diagonal. The time-average also were. This is something we expected therefore it gives confidence in the accuracy of the measurements.

3.3.3 Frequency analysis of the POD coefficients

It is interesting to make a spectral analysis of the time-dependant coefficients of the main POD modes, particularly to see which modes have the low-frequency f_0 peak. As one can see on figure 28, the fundamental $k = 1$ mode and the $k = 2$ mode have a clear peak at f_0, the third mode has a weak peak. All the following modes do not show any peak near f_0. However, the fifth mode show a clear peak for $f_1 = 1.97 \cdot 10^{-2}$ Hz. Although this doesn't seem to be a clear multiple of $f_0 \approx 0.8 \cdot 10^{-2}$ Hz, it may still be an harmonic of the large-scale oscillation. One should not forget that the value of f_0 is not measured that precisely (see figure 16), a rough numerical approximation is $f_0 \approx 10^{-2}$ Hz and $f_1 = 2f_0 \approx 2 \cdot 10^{-2}$ Hz.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

(d) $k = 4$ (e) $k = 5$ (f) $k = 6$

Figure 28: Spectral analysis of the coefficients of the first six POD modes

Since no clear peak is to be seen for $k > 5$, one might think that the first five modes hold all the dynamics of the large-scale oscillation. A simpler approximation would be to consider only the first two modes because they are the only modes with a very clear peak for f_0. The changes of structure of the central region discussed in section 3.2.4 might originate from the oscillation of the weight of this second mode (which shows a stagnation point).

First of all, one may want to know the phase difference between the coefficients of those two modes. To do that, I calculated the coherence function of the two coefficients:

$$\text{Coh} = \frac{\hat{q}_1 \hat{q}_2^*}{\sqrt{|\hat{q}_1| \cdot |\hat{q}_2|}}$$

As one can see on figure 29, the phase difference between q_1 and q_2 is around $-\pi/2$. Therefore, the f_0 component of $|q_1|$ and the f_0 component of $|q_2|$ are similar to $|\cos\varphi|$ and $|\sin\varphi|$. The former is maximal when the later is minimal (ie. for $\varphi \equiv 0\ [\pi]$ for cosines). In other words, we found indeed that the first and the second mode dominate alternatively. Of course, the first mode remains preponderant since its singular value is five times larger than the one of the second mode.

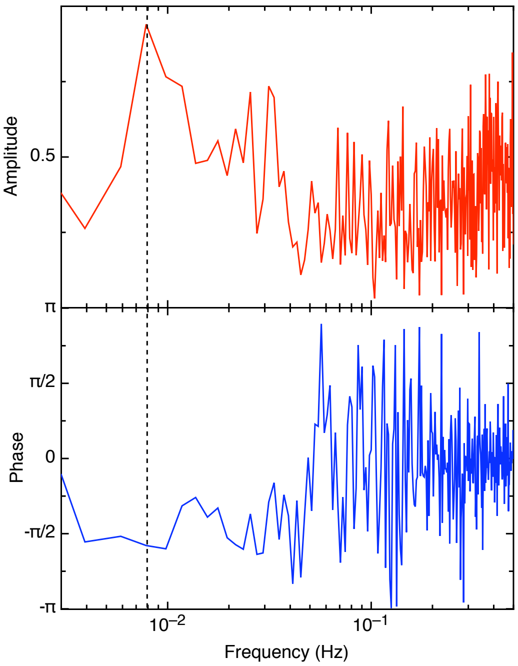
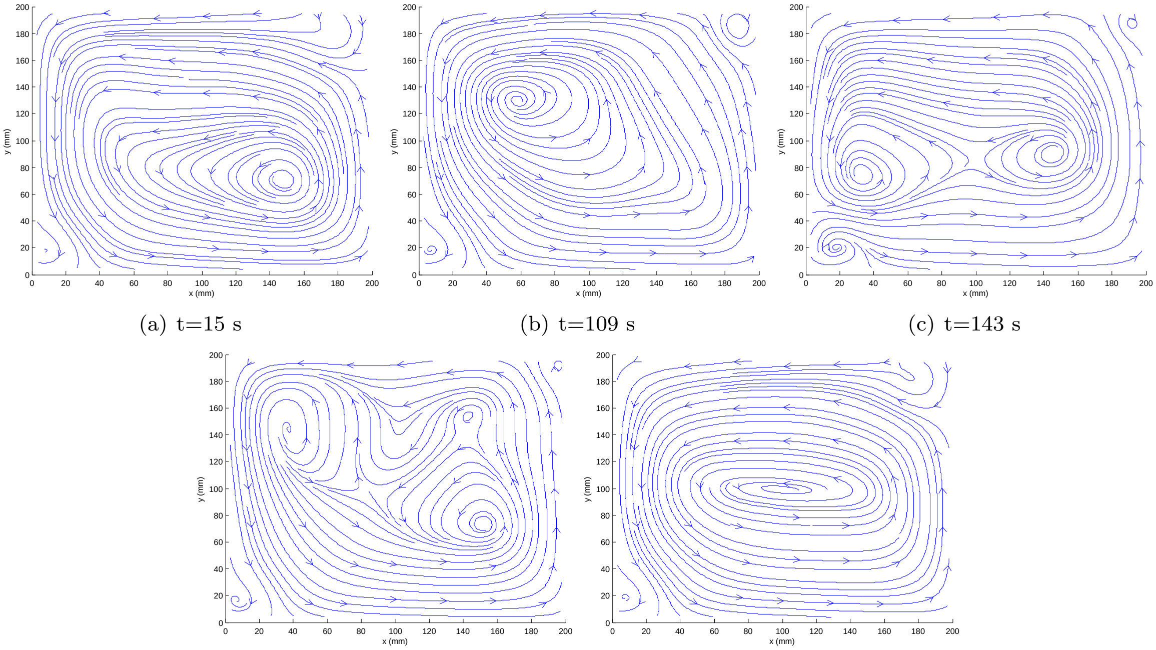


Figure 29: Coherence fonction between q_1 and q_2

3.3.4 POD-filtered movies

Since it appeared in section 3.3.3 that the first two modes held an approximated description of the low-frequency large-scale oscillation in our cell, it made sense to generate a movie using a 2$^{\mathrm{nd}}$ order lower-rank approximation of A (see equation 3 page 18 for details). However, throughout this movie, there are *always* two structures in the central region and a stagnation point. It seems that the first two modes are not enough to get the changes in the central structure.

Therefore, I tried to generate a movie with the first six modes. As shown on figure 30, one can see the same kinds of structure changes in the central region as were observed on figure 23. However, a major difference in this movie is that the changes occur a lot more rapidly (see table 3).



(a) t=15 s

(b) t=109 s

(c) t=143 s

(d) t=185 s

(e) t=250 s

Figure 30: Some images extracted from the 6$^{\mathrm{th}}$ order lower-rank approximation of the velocity field streamline movie. The center region shows different kinds of structures

Apparently those changes of structure happen approximately once every ten seconds, although it is unclear whether there is a periodicity or not. The fact that we saw only a change every 100 seconds in the low-pass filtered movie in section 3.2.4 came from the fact that the faster changes were filtered out. In reality, they are much more rapid and there is no obvious connexion to the low-frequency oscillation mentionned before. Two modes were obviously not enough to describe this phenomenon.

Time (s)	Structure
16 s	One vortex (right)
35 s	Two vorteces
37 s	One vortex (right)
39 s	Two vorteces
71 s	One vortex (right)

Time (s)	Structure
75 s	Two vorteces
96 s	One vortex (right)
100 s	One vortex (center)
110 s	One vortex (left)
125 s	Two vorteces

Time (s)	Structure
130 s	One vortex (right)
140 s	Two vorteces
153 s	One vortex (right)
155 s	Two vorteces
158 s	One vortex (right)

Table 3: Changes of structure in the central region seen in the 6$^{\mathrm{th}}$ order lower-rank approximation of the velocity field movie

However, it is still possible to comment on the connexion between the low-frequency oscillation and the first two modes. The first two modes clearly oscillate at the frequency f_0. This means that there is a superposition of several kinds of oscillating modes. The most important are

- the first mode, ie. the roll oscillating as a whole (the first mode consists in one big roll as can be seen on figure 13)
- the second mode, which is a lot more difficult to interpret. It has a clear stagnation point at the center of the cell (see figure 26) but the central structure cannot be described with only two modes (as discussed above). Therefore it is unclear whether there is a connexion between this stagnation point and the f_0 oscillation.

4 Conclusion

4.1 Generalities

In this internship, I not only learnt a lot about Rayleigh-Bénard convection, I also discovered that getting a lot of data on a specific phenomenon does not automatically give any answer. In our case, we have a lot of data. The PIV system provides us with gigabytes of images and vector arrays. Yet, it remains difficult to understand how the oscillation is sustained or even to describe it precisely.

4.2 Rayleigh-Bénard convection

I spent quite some time during this internship to figure out the best parameters to use the setup optimally (position of the camera and the laser, pulse separation, peak-locking, sampling rate, minimal number data points for the POD to make sense, space discretization, optimal use of VidPIV). Then I made sure the velocity field was accurate. To do that, I computed some known quantities and compared them with the expected results (velocity average, velocity profiles, relation between f_0 and Ra). Finally, I analysed the low-frequency large-scale oscillation using Fourier analysis (coherence functions, low-pass filters) and Proper Orthogonal Decomposition.

There is no doubt that the accuracy of the measurements would improve if one:

1. Gets data for a longer time, ie. get more than 5100 images. This will require to dedicate one entire 300 GB harddrive to the raw images
2. Uses a finer adaptative cross-correlation algorithm.

I couldn't have done this during the internship because I would have run out of time. The computation with a finer algorithm on more than 5100 images would take much more computer time but should allow one to get a better insight of the flow properties.

Appendix

Aknowledgements
Matlab scripts
References

A Aknowledgements

Above all, I thank Andreas Tilgner who welcomed me in his group. The discussions we had were highly interesting for me and allowed me to get past most problems. Most of the time, I could just pop in his office and he would be available for a talk.

I also thank the other members of the group, particularly Tomi Haramina who helped me to get started, the system administrator who was always available when I had a probem (Matlab not starting, need of an external hard drive and some other things), the technical people, particularly Manfred Erden who were available for technical problems (nozzle missing in the thermostat, ...) and finally the nearby researchers, Ole Hanekop, Ellen Hage, Stefan Bernhard, Simon Schmitz, Dennis Rippe and the others for their warm reception.

B References

Particle Image Velocimetry

[Christensen, 2004] Christensen, K. (2004). The influence of peak-locking errors on turbulence statistics computed from piv ensembles. *Experiments in Fluid*, 36(3):484–497.

[Raffel et al., 1998] Raffel, M., Willert, C., and Kompenhans, J. (1998). *Particle Image Velocimetry, A practical Guide*. Springer.

Proper Orthogonal Decomposition

[Berkooz et al., 1993] Berkooz, G., Holmes, P., and Lumley, J. L. (1993). The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid. Mech.*, 25:539–75.

[Chatterjee, 2000] Chatterjee, A. (2000). An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817.

Rayleigh-Bénard Convection

[Brown et al., 2005] Brown, E., Nikolaenko, A., and Ahlers, G. (2005). Reorientation of the large-scale circulation in turbulent rayleigh-bénard convection. *Phys. Rev. Lett.*, 95:084503.

[Castaing et al., 1989] Castaing, B., Gunaratne, G., Heslot, F., Kadanoff, L., Libchaber, A., Thomae, S., Wu, X., Zaleski, S., and Zanetti, G. (1989). *J. Fluid Mech.*, 204(1).

[Funfschilling and Ahlers, 2004] Funfschilling, D. and Ahlers, G. (2004). Plume motion and large-scale circulation in a cylindrical rayleigh-bénard cell. *Phys. Rev. Lett.*, 92(19):194502.

[Haramina, 2005] Haramina, T. (2005). *Coherent structures in turbulent Rayleigh-Bénard convection*. PhD thesis, Georg-August Universität, Göttingen.

[Haramina and Tilgner, 2006] Haramina, T. and Tilgner, A. (2006). Spatial correlation of temperature in turbulent rayleigh-bénard convection. *Not published yet.*

[Qiu et al., 2003] Qiu, X.-L., Shang, X.-D., Tong, P., and Xia, X.-A. (2003). Velocity oscillations in turbulent rayleigh-bénard convection.

[Qiu et al., 2004] Qiu, X.-L., Shang, X.-D., Tong, P., and Xia, X.-Q. (2004). Velocity oscillations in turbulent rayleigh-bénard convection. *Physics of Fluids*, 16(2):412.

[Qiu and Tong, 2001] Qiu, X.-L. and Tong, P. (2001). Onset of coherent oscillations in turbulent rayleigh-bénard convection. *Phys. Rev. Lett.*, 87(9):094501.

[Qiu et al., 2000] Qiu, X.-L., Yao, S. H., and Tong, P. (2000). Large-scale coherent rotation and oscillation in turbulent thermal convection. *Physical Review E*, 61(6).

[Villermaux, 1995] Villermaux, E. (1995). Memory-induced low frequency oscillations in closed convection boxes. *Phys. Rev. Lett.*, 75(25):4618.

[Wu, 1991] Wu, X. (1991). PhD thesis, University of Chicago.

C Matlab scripts

C.1 makeMovie.m

This Matlab script generates several movies without filters and with a low-pass filter.

```
clear
nstart=0;
nfiles=5100;
nend=nstart+nfiles-1;
% Reading files in a matrix A
for i=nstart:nend
    filename = strcat('VF/Export',num2str(i,'%04d'),'.dat')
    data=load(filename);
    if(i==0)
        xs=data(:,1);
        ys=data(:,2);
    end;
    Ax(i+1-nstart,:)=data(:,3);
    Ay(i+1-nstart,:)=data(:,4);
end;

% Make a movie with vectors without filter
figure(9);
clf;
for t=1:2000
    quiver(xs,ys,Ax(t,:)',Ay(t,:)');
    xlim([0 200]);
    ylim([0 200]);
    xlabel('x␣(mm)');
    ylabel('y␣(mm)');
    title(strcat('Pas␣de␣filtrage␣--␣t␣=␣',num2str(t),'␣s'));
    mov(t)=getframe(gcf);
end;
movie2avi(mov,'Raw.avi');
clear mov;
close

% Make a movie with energy without filter
figure(11);
clf;
EccMax=0;
EccMin=1e10;
rho=1000; % kg/m**3
for t=1:2000
    Ecc = 0.5 * rho * (Ax(t,:).*Ax(t,:)+Ay(t,:).*Ay(t,:));
    if(max(Ecc) > EccMax)
        EccMax=max(Ecc);
    end;
    if(min(Ecc) < EccMin)
        EccMin=min(Ecc);
    end;
end;
for t=1:2000
   Ecc = 0.5 * rho * (Ax(t,:).*Ax(t,:)+Ay(t,:).*Ay(t,:));
   [xi yi Ecci]=myGridData(xs,ys,Ecc);
   clf;
   contourf(xi,yi,Ecci,10);
   xlim([0 200]);
   ylim([0 200]);
   caxis([EccMin EccMax]);
   colorbar;
   xlabel('x␣(mm)');
   ylabel('y␣(mm)');
   title(strcat('t␣=␣',num2str(t),'␣s'));
   mov(t)=getframe(gcf);
end;
movie2avi(mov,'Energie.avi');
clear mov;
close

% Make a movie with vectors with low-pass filter
f=(1:nfiles)/nfiles;
h=zeros(1,nfiles);
%h(1,(0.01*nfiles-30):(0.01*nfiles+30))=1; %passe bande
freq_cut=0.02;
h(1:(freq_cut*nfiles))=1;
figure(11);
clf;
[l c] = size(xs);
for i=1:l
    Axf(:,i)=real(ifft(h.*fft(Ax(:,i)')));
    Ayf(:,i)=real(ifft(h.*fft(Ay(:,i)')));
end;

for t=1:2000
    quiver(xs,ys,1000*Axf(t,:)',1000*Ayf(t,:)',0);
    xlim([0 200]);
    ylim([0 200]);
    xlabel('x␣(mm)');
    ylabel('y␣(mm)');
    title(strcat('Cutoff␣frequency:␣',num2str(freq_cut),'␣Hz␣--␣t␣=␣',...
                                 num2str(t),'␣s'));
    mov(t)=getframe(gcf);
end;
movie2avi(mov,'LPF.avi');
clear mov;
close;

% Make a movie with stream line with low-pass filter
figure(11);
clf;
for t=1:2000
    [xi yi U] = myGridData(xs,ys,Axf(t,:)');
    [xi yi V] = myGridData(xs,ys,Ayf(t,:)');
    clf;
    streamslice(xi,yi,U,V);
    xlim([0 200]);
    ylim([0 200]);
    xlabel('x␣(mm)');
    ylabel('y␣(mm)');
    title(strcat('Cutoff␣frequency:␣',num2str(freq_cut),'␣Hz␣--␣t␣=␣',...
                                 num2str(t),'␣s'));
    mov(t)=getframe(gcf);
end;
movie2avi(mov,'streamLPF.avi');
clear mov;
close

% Make a movie with energy with a low-pass filter
EccMax=0;
EccMin=1e10;
rho=1000; % kg/m**3
for t=1:2000
    Ecc = 0.5 * rho * (Axf(t,:).*Axf(t,:)+Ayf(t,:).*Ayf(t,:));
    if(max(Ecc) > EccMax)
        EccMax=max(Ecc);
    end;
    if(min(Ecc) < EccMin)
        EccMin=min(Ecc);
    end;
end;
for t=1:2000
    Ecc = 0.5 * rho * (Axf(t,:).*Axf(t,:)+Ayf(t,:).*Ayf(t,:));
    [xi yi Ecci]=myGridData(xs,ys,Ecc');
    [xi yi U]=myGridData(xs,ys,Axf(t,:)');
     [xi yi V]=myGridData(xs,ys,Ayf(t,:)');
    clf;
    contourf(xi,yi,Ecci,10);
    xlim([0 200]);
    ylim([0 200]);
    caxis([EccMin EccMax]);
    colorbar;
    xlabel('x␣(mm)');
    ylabel('y␣(mm)');
    title(strcat('Cutoff␣frequency:␣',num2str(freq_cut),'␣Hz␣--␣t␣=␣',...
                             num2str(t),'␣s'));
    hold on
    h = streamslice(xi,yi,U,V);
    set(h,'Color','w');
    hold off
    mov(t)=getframe(gcf);
end;
movie2avi(mov,'EnergyStream_LPF.avi');
clear mov;
close
```

C.2 pod.m

This Matlab script computes the POD analysis of the measured velocity field, displays singular values and modes with various representations and generates POD-filtered movies.

```
nfiles=5100;
% Reading measurements in matrix A
for i=0:(nfiles-1)
    filename = strcat('VF/Export',num2str(i,'%04d'),'.dat')
    data=load(filename);
    if(i==0)
        xs=data(:,1);
        ys=data(:,2);
    end;
    Ax(i+1,:)=data(:,3);
    Ay(i+1,:)=data(:,4);
end;
% POD Analysis
[U S Vx Vy]=mySvd(Ax,Ay);
Q=U*S;

% Displaying the module of the modes
for k=1:3
    figure(k);
    V=sqrt(mean(Q(:,k))^2*Vx(:,k).^2+mean(Q(:,k))^2*Vy(:,k).^2);
    [xi yi Vi]=myGridData(xs,ys,V);
    contourf(xi,yi,Vi);
    xlim([0 200]);
    ylim([0 200]);
    title(strcat('Mode␣k=',num2str(k)));
end;

% Displaying the modes with vectors
for k=1:3
    figure(k+3);
    quiver(xs,ys,mean(Q(:,k))*Vx(:,k),mean(Q(:,k))*Vy(:,k));
    xlim([0 200]);
    ylim([0 200]);
    title(strcat('Mode␣k=',num2str(k)));
end;

% Displaying the modes with streamlines
for k=1:3
    figure(k+6);
    [xi yi ui]=myGridData(xs,ys,Vx(:,k));
    [xi yi vi]=myGridData(xs,ys,Vy(:,k));
    streamslice(xi,yi,ui,vi);
    xlim([0 200]);
    ylim([0 200]);
    title(strcat('Mode␣k=',num2str(k)));
end;

% Singular values
figure(7);
clf;
semilogy(diag(S));
title('Singular␣values');
xlabel('Number');
ylabel('Singular␣value');

% Spectral analysis of the time-dependant coefficients
for k=1:5
        figure(k);
        subplot(2,1,1);
        Qf=Q(:,k)-mean(Q(:,k));
        plot(((-nfiles+1):(nfiles-1))/60,xcorr(Qf,Qf));
        title(strcat('Autocorrelation␣of␣q␣-␣<q>␣(k=',num2str(k),')'));
        xlabel('\delta␣t␣(minutes)');
        xlim([-20 20]);
        %ylim([-2e6 6e6]);

        subplot(2,1,2);
        semilogx((0:(nfiles-1))/nfiles,abs(fft(Qf)));
        title(strcat('Spectral␣analysis␣de␣q␣-␣<q>␣(k=',num2str(k),')'));
        xlabel('Frequency␣(Hz)');
        ylabel('Spectral␣density');
        xlim([3e-3 0.5]);
        pause;
end;

% Movie of the POD-filtered velocity field
h=figure('BackingStore','off','Position',[10 10 900 900],...
                  'Resize','off','MenuBar','none');
clf;
Qf=zeros(size(Q));
Qf(:,1:2)=Q(:,1:2);
Axf=Qf*Vx';
Ayf=Qf*Vy';
EccMax=0;
EccMin=1e10;
rho=1000; % kg/m**3
for t=1:2000
    Ecc = 0.5 * rho * (Axf(t,:).*Axf(t,:)+Ayf(t,:).*Ayf(t,:));
    if(max(Ecc) > EccMax)
        EccMax=max(Ecc);
    end;
    if(min(Ecc) < EccMin)
        EccMin=min(Ecc);
    end;
end;
for t=1:607
    subplot(2,2,1);
    quiver(xs,ys,1000*Q(t,1)*Vx(:,1),1000*Q(t,1)*Vy(:,1),0);
    xlim([0 200]);
    ylim([0 200]);
    xlabel('x␣(mm)');
    ylabel('y␣(mm)');
    title('Mode␣k=1');

    subplot(2,2,2);
    quiver(xs,ys,1000*Q(t,2)*Vx(:,2),1000*Q(t,2)*Vy(:,2),0);
    xlim([0 200]);
    ylim([0 200]);
    xlabel('x␣(mm)');
    ylabel('y␣(mm)');
    title('Mode␣k=2');

    subplot(2,2,3);
    quiver(xs,ys,1000*(Q(t,1)*Vx(:,1)+Q(t,2)*Vx(:,2)),...
                   1000*(Q(t,1)*Vy(:,1)+Q(t,2)*Vy(:,2)),0);
    xlim([0 200]);
    ylim([0 200]);
    xlabel('x␣(mm)');
    ylabel('y␣(mm)');
    title(strcat('q_1(t)v_1+q_2(t)v_2␣t␣=␣',num2str(t),'␣s'));
    subplot(2,2,4);
    Ecc = 0.5 * rho * (Axf(t,:).*Axf(t,:)+Ayf(t,:).*Ayf(t,:));
    [xi yi Ecci]=myGridData(xs,ys,Ecc');
    [xi yi U]=myGridData(xs,ys,Axf(t,:)');
    [xi yi V]=myGridData(xs,ys,Ayf(t,:)');
    contourf(xi,yi,Ecci,10);
    xlim([0 200]);
    ylim([0 200]);
    caxis([EccMin EccMax]);
    colorbar;
    xlabel('x␣(mm)');
    ylabel('y␣(mm)');
    title(strcat('q_1(t)v_1+q_2(t)v_2␣t␣=␣',num2str(t),'␣s'));
    hold on
    h = streamslice(xi,yi,U,V);
    set(h,'Color','w');
    hold off
    mov(t)=getframe(gcf);
end;
movie2avi(mov,'POD1+2.avi');
```