

Advanced Measurement Techniques in Fluid Mechanics and Heat Transfer

Prof. Saptarshi Basu

Department of Mechanical Engineering

Indian Institute of Science, Bengaluru

Week – 04

Lecture - 17

Introduction to Experimental Data Processing

Hi, everyone. So in this portion of the course, we'll be looking at tools and techniques that are used to extract information from experimental data. So, the tools and techniques that we discuss here are not specific to any experimental technique and can be applied across all techniques that we study in the current course. For processing experimental image data, we'll be using an open-source toolbox called Imaging. And for processing experimental time series signals, we'll be using functions in MATLAB. Now, let's take a look at the content of this portion of the course.

Introduction to Experimental Data Processing

 **Prof. Saptarshi Basu**
Pratt and Whitney Chair Professor,
Mechanical Engineering
Indian Institute of Science, Bangalore-560012

 
BASU LAB

Lecture delivered by
Akhil Aravind
(Teaching Assistant)
Mechanical Engineering
Indian Institute of Science, Bangalore-560012



So, we'll first look into the kinds of data that we get from the experiments. So, depending on the kind of experimental technique that is used, we can either have visual-based data, which is basically an image, or we can have data from a sensor, which basically gives a time series variation of a particular parameter of interest. For example, if we use a pressure sensor, we will get a time series variation of pressure at a particular point in the flow field. So, once we have made this distinction clear, we will go into tools that we can use to extract, say, parameters of our interest from the images.

For example, this image on the right basically shows the response of a flame when it

interacts with the shockwave. So, during this interaction process, we might be interested in parameters such as heat release rate. So, we learn about the tools that we can use to extract this information from an image. And once we have processed this parameter over a sequence of images, we again essentially have a time-series variation of this parameter. So, we always end up with a time series variation in most of our experiments.

So, then after that, we look into how this time series data can be represented using a Fourier series approximation. Here again, we look into different techniques. We look into discrete Fourier transforms. We look into fast Fourier transforms, which are basically an easier way to implement the discrete Fourier transforms. Then we look at spectrograms.

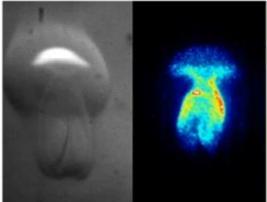
After that, we look into data filtering techniques to filter out the noise from the data. So, experimental data are usually very noisy in nature, and we need to filter it out. So here we are going to look at two commonly used techniques. One way is by using FFT itself. So once we have the FFT, we can remove the noise from it using the data from the FFT.

Or we can use filters such as the Savitzky-Golay filter, moving average filters, or median filters. So we'll have an overview of these techniques. So here I have shown typical experimental images. So, these are from different techniques: shadowgraphy, Schiller, and chemiluminescence. All of these techniques are covered in the current course.

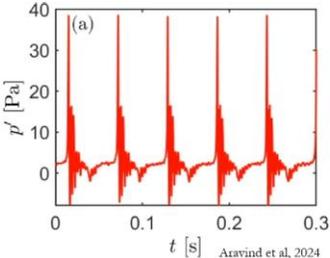
Contents

- Data Acquisition Methods in Experiments
- Image data extraction
- Fourier Analysis of time Series data
 - Fourier Series Representation
 - Discrete Fourier Transforms (DFT)
 - Fast Fourier Transform (FFT)
 - Spectrograms
- Data Filtering
 - FFT for noise filtering
 - Moving average, median and Savitsky Golay Filter





Aravind et al, 2024



Aravind et al, 2024

So, here we can see a shadowgraphic image of a droplet that is atomizing in high-speed flow. We can see that the droplet deforms and continuously changes its shape during the deformation process. And to understand this dynamics, we need to quantify the rate at which this deformation happens. And this is information that we need to extract from the image sequence. Similarly, if we look into the interaction of a jet flame with a blast wave,

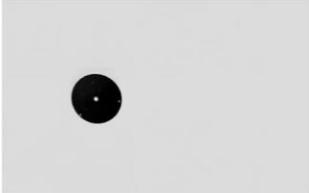
we can see that the flame lifts off and its area continuously changes during this interaction process.

Again, we need to quantify the rate at which these changes are happening in order to study this process. Again, this is information that we need to extract from the image sequence. Similarly, I have another example shown here of a droplet that is impacting a surface. So, as a droplet impacts a surface, it starts to spread radially outward. And to study the dynamics, we need to understand the rate at which this spreading happens.

And this is again information that we need to extract from the images. The technique that we use to extract such information will be discussed later on in this lecture. So this is the other class of data that we have, which is basically data from sensors. These are time series signals. Here we have sensors that are sensitive to specific parameters in the flow field.

Data Acquisition Methods in Experiments

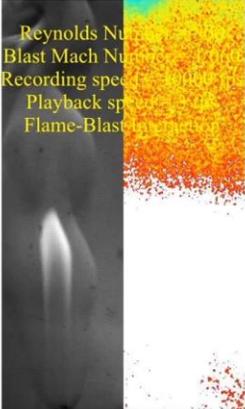
- Visual Sensing
 - Using high-speed cameras to capture the visual data of the phenomena (Output: Images)
 - Examples
 - Shadowgraphy
 - Schlieren Visualization
 - Particle Image Velocimetry
 - Chemiluminescence Imaging
 - IR Thermography



Sharma et al, 2023

	6 ≤ We ≤ 132 Deposition				
GLASS					
	$t = 0$	$t = 3.5\text{ms}$	$t = 4.3\text{ms}$	$t = 6.2\text{ms}$	$t = 12.3\text{ms}$
PDMS	We ≤ 6		6 < We ≤ 132		
	$t = 4\text{ms}$	$t = 4.9\text{ms}$	$t = 1.7\text{ms}$	$t = 3.4\text{ms}$	$t = 6.9\text{ms}$

Aravind et al, 2024



Reynolds Number
Blast Mach Number
Recording speed
Playback speed
Flame-Blast interaction

Vadlamudi et al, 2023

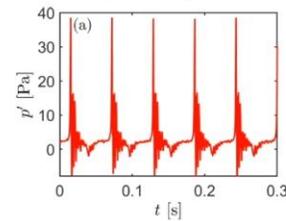
So, if you want to measure the velocity variations, you can use a hot wire anemometer; or if you want to measure the pressure variations, you can use a piezoelectric sensor; or if you want to measure the temperature field at a specific point, you can use a thermocouple. So, irrespective of what sensor we use, we always end up getting a time series signal showing the variation of this parameter at a specific location in the flow field. Usually, the output of any of these sensors is an analog signal, either a current or a voltage signal, and then we acquire the signal using a data acquisition unit. We will come to that in a later slide. A very important distinction to note here is that, in an image sequence, we actually had both spatial and temporal data.

Whereas from a sensor, we only get the temporal variation of a parameter at a specific location in the flow field. If we want to know how this parameter varies in another location in the flow field, we need to place the sensor at that location and perform the experiment again. Whereas from an image, we can obtain both the spatial and temporal information simultaneously. This is a very important distinction to note. So, as we saw before, an image sequence has data on both space and time.

Data Acquisition Methods in Experiments



- Sensor Based Measurements
 - Employing specialized sensors to directly measure the physical properties such as pressure, temperature , etc. (Output: Typical a Current/Voltage Signal)
- Examples
 - Piezoelectric sensor
 - Thermocouples
 - Photomultiplier Tubes (PMT)
 - Hot wire anemometers



<https://www.pcb.com>



<https://www.electronicwings.com>



<https://www.dantecdynamics.com/>

Whereas data from a sensor only has variation in the time domain. However, if we look into an image sequence, We cannot resolve all the link scales and time scales associated with events. To understand that limit, let us look at this example here. So here before that. We need to understand that an image essentially consists of an array of very small units that are called pixels.

And each pixel has an intensity value and a color value associated with it. So, these two images represent satellite images of the same geographical location. However, the pixels in both of them represent two different physical metrics. For the image on the left, each pixel represents 30 meters, whereas for the image on the right, each pixel represents 1 meter. So, comparing the two images, we can see that there are patterns in the second image that are not resolved in the first image.

That is because the length scale associated with these small patterns is lower than 30 meters, which is the pixel resolution or spatial resolution of that image. It was visible in the second image because the length scale associated with these patterns is greater than the spatial resolution, which is one meter per pixel. The same concept applies even in the time domain. So, here we have data that is sampled on a yearly basis, and here we have data that is sampled on a daily basis. So, we can see that there are fluctuations on a daily basis, but

these fluctuations won't be reflected when an annual sample data is plotted.

Again, the reason is that the time scale associated with this fluctuation is very small when compared to the spatial resolution at which the yearly data is sampled. So, in an image sequence, we have a limit on both this length scale and time scale that can be resolved. That is specified in terms of spatial resolution, which is representative of the number of pixels for one physical unit, which is basically a meter or millimeter. It is generally specified in pixels per mm or pixels per meter. And then we have temporal resolution, which is basically nothing but the rate of the number of image sequences that we have captured of the physical event in one second.

It is usually specified in frames per second. Say, if we have a frame rate of 50 fps, that means that we have captured 50 images in one second; that means that the time between two consecutive images is one divided by 50 seconds. The same concept applies even to sensor data, as we already discussed; sensors do not have spatial information; they only have data in the time domain. Although data from a sensor is essentially analog in nature, it is usually a current signal or a voltage signal that is analog in nature. We usually use this data acquisition unit to capture the data from the sensors.

And these data acquisition units sample the data at specific time intervals. So there is a sampling rate. If you have a sampling rate of 10 kilohertz, it takes 10,000 samples from the sensor in one second. So the time between two consecutively sampled data points is 1 divided by 10,000. So any time scale that is lower than this particular threshold cannot be resolved from that data anymore.

Characteristics Experimental Data



○ Resolution

○ Image Data

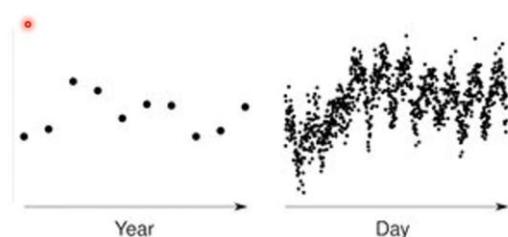
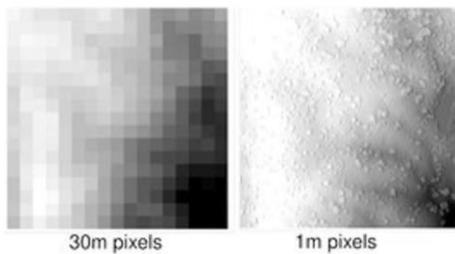
- Spatial Resolution (Specified in px/m)
- Temporal Resolution (Specified by the frame rate of the camera, fps)

○ Sensor Data

- Temporal resolution (Specified by the Sampling rate of the acquisition unit such as DAQ)



<https://www.vvdatools.com/>



Mahood et al, 2023

Now that we have seen that there is a limit on the time scale and length scale that can be

resolved from our experimental data, let us look at what this limit is. To understand this, we'll start off with an example. Say I have pressure fluctuations in my flow field, and these pressure fluctuations are represented by solid lines and solid curves. Let's say these pressure fluctuations have a frequency of f , which is 10 hertz now. Now suppose I use a pressure sensor to measure these pressure fluctuations, and the sampling rate that I set in my data acquisition unit is, say, f_s , and say this f_s by f is, say, 20 hertz, is, say, 2, which basically means that f_s is 20 hertz.

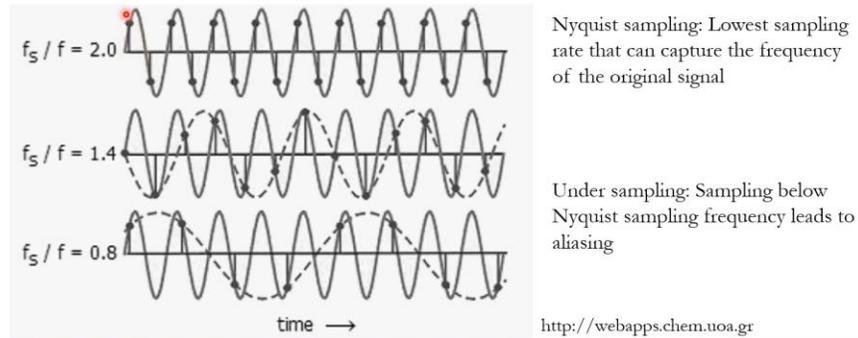
Then the data points that we capture from the pressure sensor are these that I have shown in dots here, and if we connect these dots by using a curve, we will be able to replicate the same periodicity as the pressure fluctuations that are happening in the flow field. Although we will not be able to reconstruct the entire pressure fluctuations properly since we are missing many data points in between, the periodicity at least is replicated. Now suppose we reduce the sampling rate further. Say now f_s by f is 1.4, which basically means that I am sampling it at 14 Hz instead of 20 Hz.

Now, the data points I get can be traced out by a different curve. Here, a dotted line shows a completely different frequency compared to the original pressure fluctuations, which was F . So, whatever frequency I report from my pressure sensor will be completely wrong when compared to the actual event that is happening. So this actually shows what the limit is at which I can sample my data so that I can capture the physical phenomenon. Say that if the physical phenomena have a frequency of f , my sampling frequency should be at least two times that.

Now, this can be said in terms of temporal resolution, as well. So, the temporal resolution, which is basically nothing but 1 divided by the sampling rate or 1 divided by the frames per second, should be lower than half of the lowest time scale that is associated with the problem. Again, the lowest time scale is essentially one divided by the highest frequency. Here, there is only one frequency, so this is simply $\frac{1}{f}$. Now, the same analogy applies even in the spatial domain.

To capture, say if we want to capture a length scale of L , then our spatial resolution should be less than $L/2$. So, this essentially is the Shannon-Nyquist criteria, and this is very important when setting the recording rates for our sensors and cameras. So, we should have a general idea about what the expected length scale and time scale are, as well as the lowest length scale and time scale of the fluid flow that we want to capture. And then accordingly set the temporal and spatial resolutions, or basically the frame rates or the pixel resolution.

- Shannon-Nyquist Criteria
 - Temporal Resolution $<$ (smallest timescale that we intend to capture/2)
 - Spatial Resolution $<$ (smallest lengthscale that we intend to capture/2)
- Aliasing occurs when a signal is sampled at a rate lower than twice its highest frequency, causing high-frequency components to appear as lower frequencies in the sampled data.



Now that we have distinguished the different data types, let us look into how to process each of these data types individually.

We'll start off by processing image data. Before that, let us look at what an image is. So, as I already mentioned before, an image is essentially an array of a large number of subunits that are called pixels. And the number of pixels in an image is usually specified in this format. So here, if I say that I have an image that is 1920 x 1080 pixels.

It basically means that I have 1,920 pixels along the horizontal direction and 1,080 pixels along the vertical direction. And then these units, these numbers of pixels along each dimension, essentially depend upon the camera hardware, the sensor hardware, and the images usually are of two types; they can either be a grayscale image or an RGB image. In a grayscale image, each pixel has information only on the intensity. Whereas an RGB image contains information on both the intensity and the color. We look at each of these image types individually now.

Experimental Data



- Image is an array of fundamental units called pixels that contain information on intensity and color
- An image is usually specified in terms of pixel dimensions (Eg: 1920 x 1080 px)
- The number of pixels captured depends on the sensor hardware of the camera used
- Two types of images
 - Greyscale Image
 - RGB Image

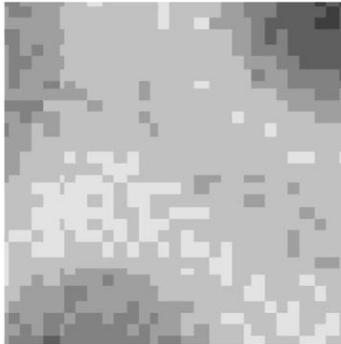


Jindal et al, 2016

So here I have a typical grayscale image. If we keep zooming in on this grayscale image, we can see the pixels in it. And underlying each pixel is essentially a numerical value that quantifies the intensity at that pixel. Now, the intensity is specified in different ways depending on the bit depth of the image. For example, if we have an 8-bit image, the lowest intensity is assigned a value of 0, which corresponds to the color black. Whereas the highest intensity is assigned a value of 255, which is $2^8 - 1$, and that is assigned a color of white.

So, if the intensity lies between the highest and the lowest, it takes on a value between 0 and 255, and each number that you see here is representative of that intensity level. Similarly, if we have a 16-bit image, the lowest and highest ranges are 0 and $2^{16} - 1$. And again, any intensity between the highest and lowest is assigned a value between these two ranges. So, as I had already mentioned, each pixel in a grayscale image only has information on the intensity; there is no information on the color.

- Grayscale Images
 - Contains only shades of grey with no color information
 - Each pixel represents a single intensity value (lowest for black, highest for white).



26	24	31	38	39	43	40	39	41	39	45	44	43	40	35	26	23	21	16	13	12	15	10	9
24	27	31	33	39	36	41	38	39	40	42	45	43	44	34	29	27	22	17	14	13	11	8	10
25	25	31	37	41	41	41	41	37	36	36	45	43	40	39	33	27	23	15	16	15	13	12	13
26	25	24	36	39	38	41	37	39	39	38	39	43	37	39	35	30	24	20	15	12	14	13	11
24	25	28	36	39	39	37	38	37	38	42	41	42	42	38	37	27	26	23	15	17	13	15	15
27	24	33	38	37	42	37	39	41	41	35	41	38	46	42	40	36	31	28	24	20	20	17	15
26	25	31	32	32	38	40	37	38	35	46	39	42	46	40	39	38	41	35	28	27	24	19	20
27	22	32	35	38	34	38	40	37	39	40	41	37	37	38	43	41	38	38	31	30	27	25	25
24	30	30	40	40	40	39	36	41	34	39	40	40	40	41	42	42	44	42	42	37	36	33	37
28	34	37	41	41	40	43	38	41	42	40	37	35	35	40	37	39	41	42	41	42	42	40	40
31	35	41	40	42	44	45	46	43	37	42	39	36	39	39	38	41	39	40	38	45	42	42	41
34	38	40	42	41	44	44	48	47	41	43	38	39	39	37	39	39	39	39	37	42	44	40	39
38	41	41	47	46	49	46	44	47	42	39	41	36	39	36	38	36	35	39	37	37	39	42	41
42	48	47	45	50	49	46	46	40	50	48	47	46	35	36	40	37	39	40	40	37	36	37	41
44	45	54	51	54	48	53	45	47	44	47	43	53	41	41	36	38	40	40	39	37	37	37	37
44	47	47	55	49	50	48	51	46	44	44	45	47	45	40	43	36	40	36	36	32	37	39	39
43	46	50	51	54	52	50	48	47	47	51	46	42	41	42	41	41	39	38	38	39	38	40	40
45	50	50	47	43	48	44	45	50	45	45	44	48	44	43	43	43	40	41	34	35	38	40	38
46	45	43	42	34	38	36	41	42	43	44	43	42	44	51	48	43	41	41	38	36	39	35	39
41	39	39	35	35	29	29	31	33	36	39	37	44	43	46	43	43	43	39	40	39	39	41	39
38	30	34	29	28	29	27	26	28	32	33	38	42	42	44	44	45	48	39	43	40	45	39	43
29	30	29	24	24	23	27	29	25	28	30	38	33	40	43	43	46	44	46	46	42	42	46	42
26	24	25	23	22	21	26	25	23	28	29	35	41	39	42	41	43	48	48	44	45	46	46	40
25	24	22	22	20	22	17	19	23	23	25	27	32	36	40	41	49	48	40	48	47	43	48	44

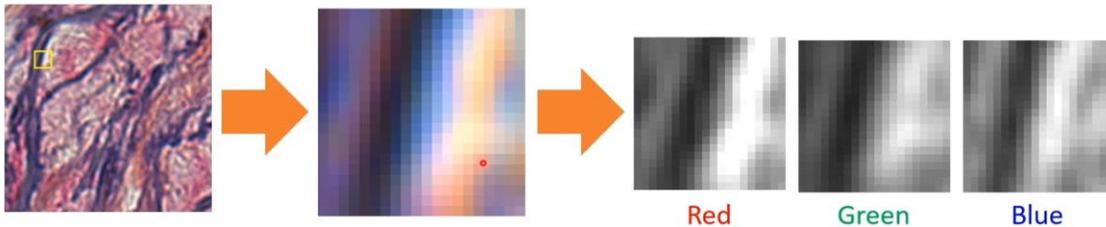
Type	Levels
8 bit	$2^8 = 256$
12 bit	$2^{12} = 4096$
16 bit	$2^{16} = 65536$

Dr. Dominic Alibhai & Dr. Stephen Cross, Lecture Notes, 2021

Now, let us look at RGB images. So, unlike in the grayscale images where the intensity level was specified only in one channel, which is a grayscale channel. In an RGB image, the intensity levels are specified in three channels: red, green, and blue. So if it is an 8-bit image, the intensity levels in each of these channels can be assigned a value between 0 and 255. So basically, each pixel is associated with three values here, unlike in the grayscale image where it is only assigned one value corresponding to the grayscale channel. Here, it has three values corresponding to three channels.

The reason we have chosen red, green, and blue as the channels to specify a color image is that these are the primary colors, and a combination of these colors in different proportions can help you replicate any other color. So, because each pixel is associated with three values now, the memory occupied by an RGB image is much greater when compared to a grayscale image. And most of the time, in most image processing that we do for experimental data, most images are grayscale in nature. And for further processing, whatever we do in the rest of this lecture, we will be focusing on grayscale images.

- RGB Images
 - Contains three channels: Red, Green, and Blue, and can produce a wide range of colors
 - Each pixel has three intensity values (one for each channel).



Dr. Dominic Alibhai & Dr. Stephen Cross, Lecture Notes, 2021

- Typically, RGB images are converted to greyscale images (using a channel weighted average formulation for intensity) for further processing

Okay, so now we look into how we can clean up image data so that we can extract features of our interest from that data.

So here I will be doing it by using an example. So here I'm showing a raw image sequence of a flame that is actually propagating through a channel. The curved object that you find moving inside this rectangular box is essentially a flame that is propagating inside a channel. It ignites at a specific location. Yeah, it ignited there, then it propagated upstream, and after that, it extinguished. So here I'll demonstrate our image cleaning and processing techniques using this raw image sequence.

So here we might be interested in parameters such as the rate at which the flame is propagating and the intensity profile associated with the flame during this periodic ignition extension cycle. So I'll demonstrate techniques to extract these features. But you can always find synonymous features to extract from any problem. So whatever the techniques that are being demonstrated, they are not limited to this particular problem.

They can be applied to any problem in general. And these are the processes that are typically involved. Since our image data is usually very noisy in nature, we need to perform background subtraction to clear out the background noise. Even then, we might have to do filtering so that we can increase the signal-to-noise ratio again. And then we perform a process of thresholding where we basically convert this grayscale image.

All of these are grayscale images. So this grayscale image is converted into a binarized image where we only have 0 and 1. We don't have levels from 0 to 255 anymore. We only have 0 and 1. So once we have a binarized image, it helps us actually identify our object of interest. Here it's basically the curved interface that you see, the curved interface that

represents

the

flame.

So, once we have our object identified, we can trace how that object is moving in different image sequences, and then we can extract information such as the rate at which that object is moving, the area associated with it, or the dimensions of the object. So, once we have the temporal variation of these features, we will be able to understand and characterize the dynamics of this problem. So for this, I'll be using an open-source toolbox called Img, which we'll see in the next slide.

