XPCS Analysis Procedure (Mathematically and Illustratively)

The below document is an attempt to document some of the critical steps involved in the analysis of any XPCS time series to yield correlation functions. While this document is not written or meant to be self-explanatory, it is meant to be a guide with which a beamline staff or a user with prior experience in XPCS can guide a new user to make some of the abstract concepts clearer or at least less abstract.

This document is written using Matlab's Live Editor (Version 2016 and later) so that a user can play with the code or tweak things to get a better feel for the analysis and not be restricted to a black box type approach.

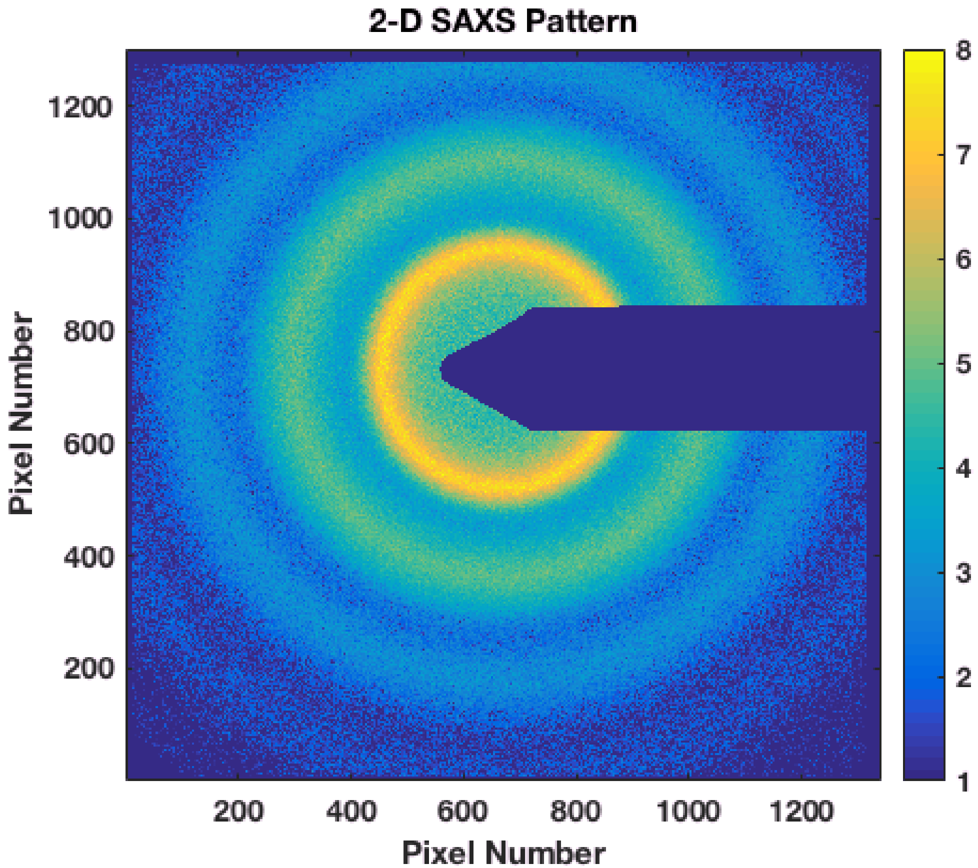
%v1: January 2018

SAXS 2-D pattern:

```
% take a few time series images, use the shear from silica 100 nm
viewresult('A016_Silica50vpnew_1_100nm_ME20_Gap0p5_ZeroShear_Sq0_002_0001-0522.hdf');
```

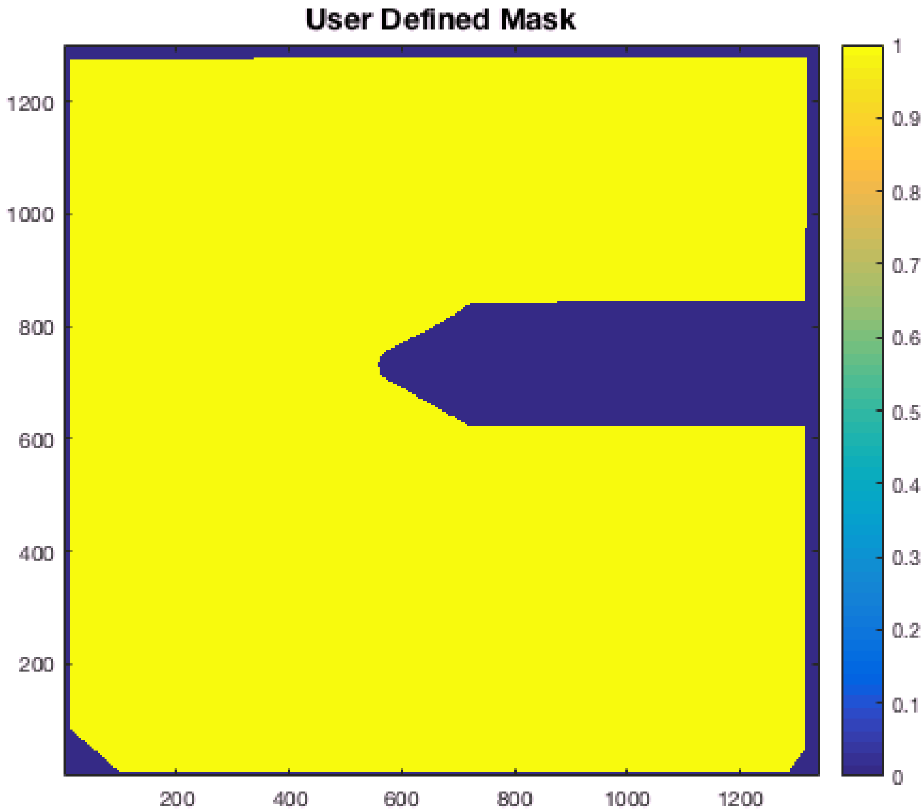
```
Loading from the HDF5 file: RESULTS Group: /exchange, XPCS Group: /xpcs
g2 fits already exist in the HDF5 file. Bringing up the viewresult window...
-----------------------------------------------
```

```
viewresult_debug;
img=udata.result.aIt{1};
figure(11);imagesc(log(img),[1,8]);axis xy;axis image;colorbar;
xlabel('Pixel Number','fontsize',16,'fontweight','bold');
ylabel('Pixel Number','fontsize',16,'fontweight','bold');
title('2-D SAXS Pattern','fontsize',14,'fontweight','bold');
set(gca,'fontsize',14,'fontweight','bold');
```



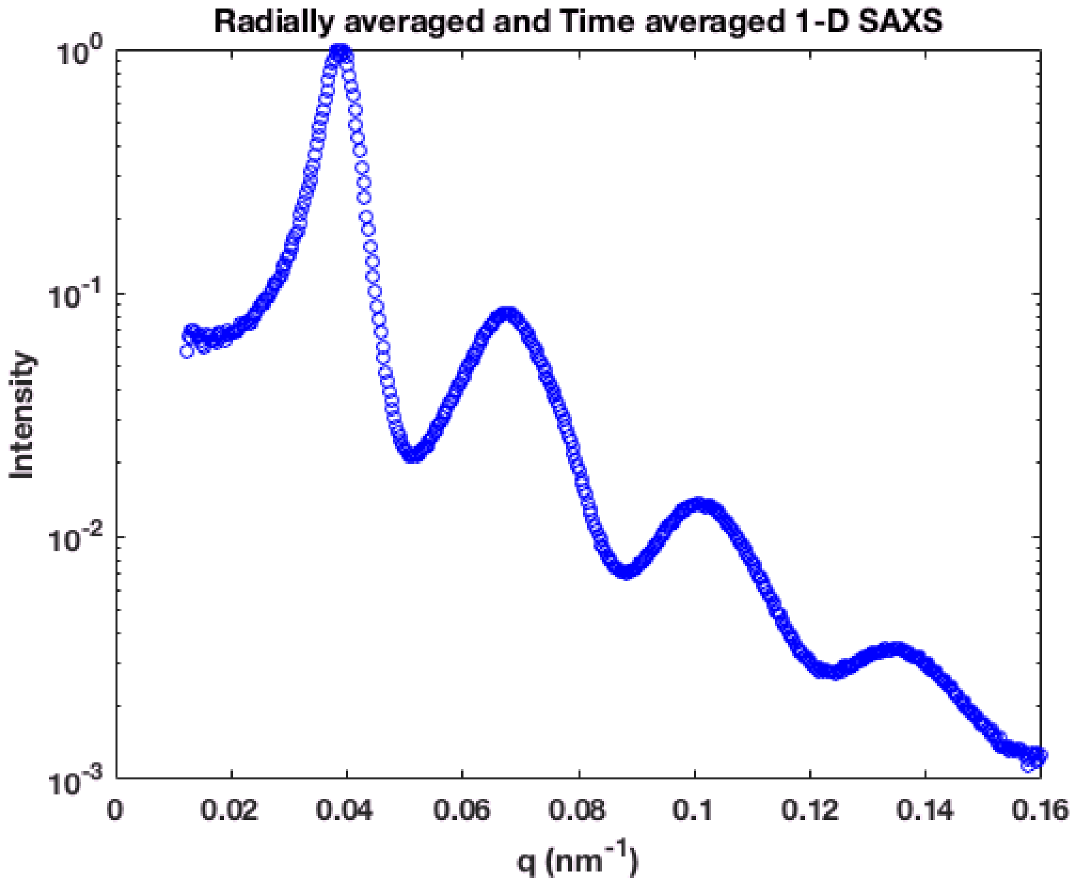
User defined Geometrical Mask (setting boundaries):

```
mask=h5read('A016_Silica50vpnew_1_100nm_ME20_Gap0p5_ZeroShear_Sq0_002_0001-0522.hdf','/xpcs/ma
figure(23);
imagesc(mask');axis xy;axis image;colorbar;
title('User Defined Mask','fontsize',14,'fontweight','bold');
```



Radially Averaged I(q):

```
q=udata.result.staticQs{1};
Iq=udata.result.Iqphi{1};
figure(12);
semilogy(q*10,Iq/max(Iq),'bo');
xlim([0 0.16]);
title('Radially averaged and Time averaged 1-D SAXS','fontsize',16,'fontweight','bold');
xlabel('q (nm^{-1})','fontsize',16,'fontweight','bold');
ylabel('Intensity','fontsize',16,'fontweight','bold');
set(gca,'fontsize',14,'fontweight','bold');
```



Correlation Function g2:

The numerator would be the correlation component and the denominator will be the normalization so that g2 decays to unity at long delay times. Normalization is a topic in itself and so is only briefly discussed below (Phase-2).

$$g_2(q, dt) = \frac{\langle\ I(q,t)\, I(q,t+dt)\ \rangle}{\langle\ I(q,t)\ \rangle^2}$$

Phase-1: During correlation analysis, individual pixels are correlated with itself as a function of time. Let us call this as G2 and is of the size of the image array.

Phase-2: The qmaps come into play for the normalization. Basically, there are 2 Maps, naively one is for Static (typically finer) and the other is for Dynamics (typically coarser).

For transmission, (q and phi) are the maps used.

For reflection, (qr and qz) or (qy and qz) are typically used.

It is an enforced requirement to select the ratio of (# in (static) Map 1 / (dynamic) # in Map 2) to be an integer for both Static and Dynamics.

Typical thumb rule is to pick this integer to be some where between 5 and 10 **OR** the other way to look at is to pick the static map to be such that each bin is no bigger than a single pixel width.

Phase-2a: During the **first normalization step**, G2 from the pixels that belong to the Static partition (Map 1) are averaged together and normalized by the time averaged intensities over the group of pixels in that static partition (Map 1). **Reiterating**, each bin in Map 2 comprises of integral number of bins in Map 1.

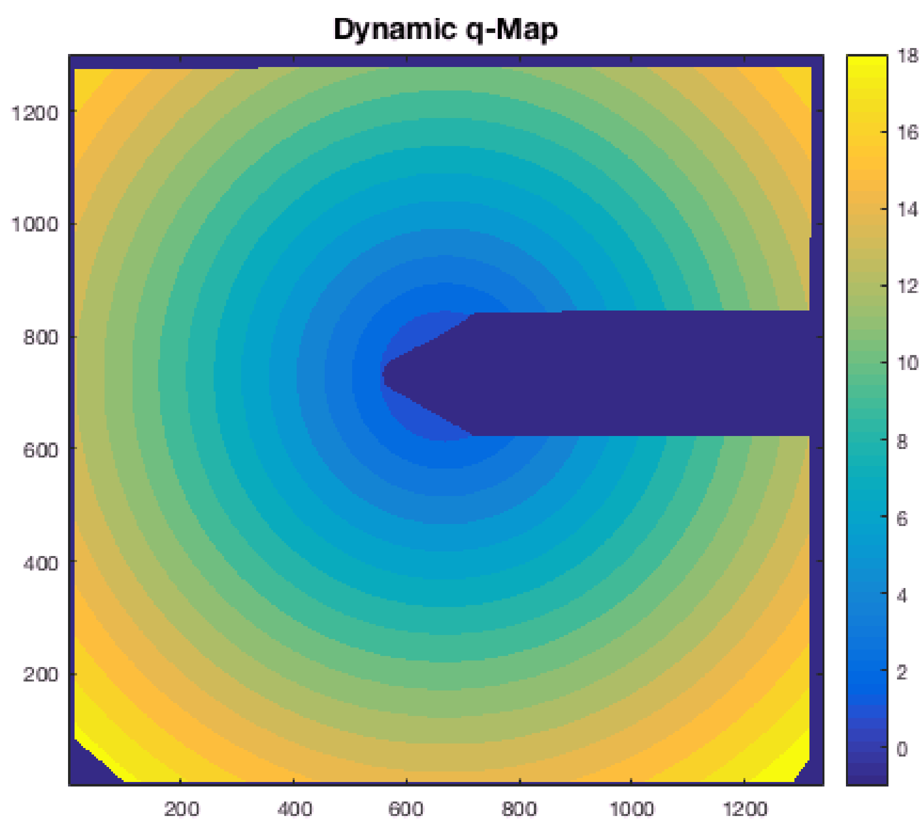
Phase-2b: During the **second and final normalization step**, the sub-averages from the first step are further averaged to give you the final g2 as specified in Map 2.

Note: If Phase-2 is not split into two as described above and if the intensity varies over the dynamic partition which is usually the case as it is typically coarser, then the intensity variation within a bin (single q or phi) will contribute to contrast variation and the baseline might not reach unity, even for a completely dynamic and ergodic system.

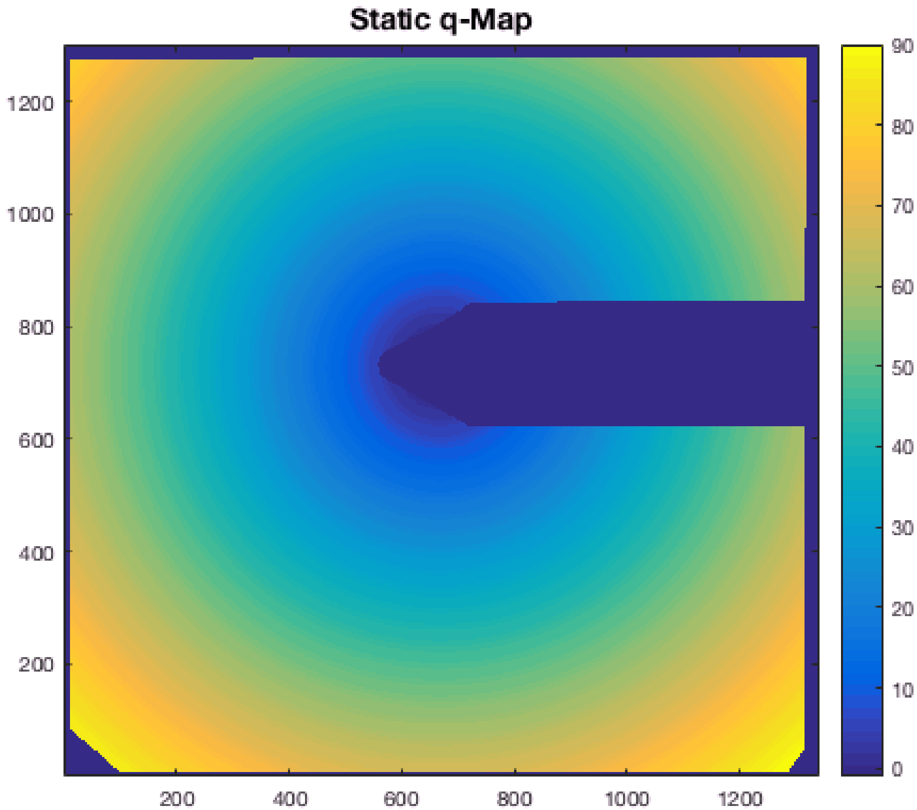
Qmap or Wave vector map:

take a qmap.h5 file (qmap_example.h5) and plot dqmap (Map 2) and sqmap (Map 1)

```
dqmap=transpose(h5read('qmap_example.h5','/data/dynamicMap'));
sqmap=transpose(h5read('qmap_example.h5','/data/staticMap'));
figure(13);
imagesc(dqmap);axis xy;axis image;colorbar;
title('Dynamic q-Map','fontsize',14,'fontweight','bold');
```



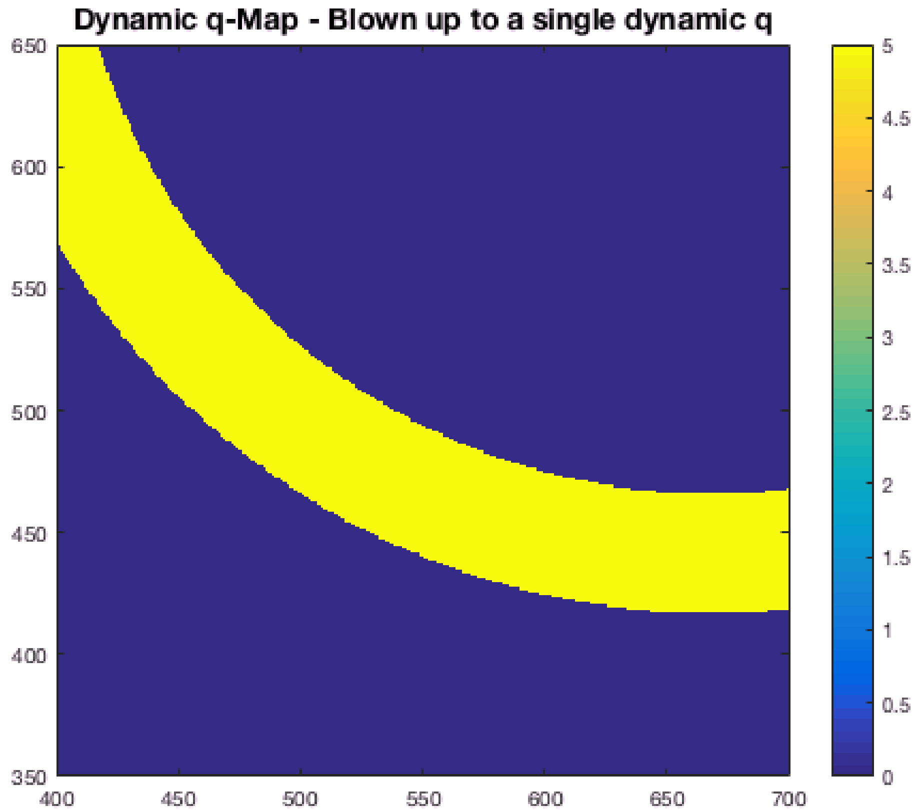

```
figure(14);
imagesc(sqmap);axis xy;axis image;colorbar;
title('Static q-Map','fontsize',14,'fontweight','bold');
```



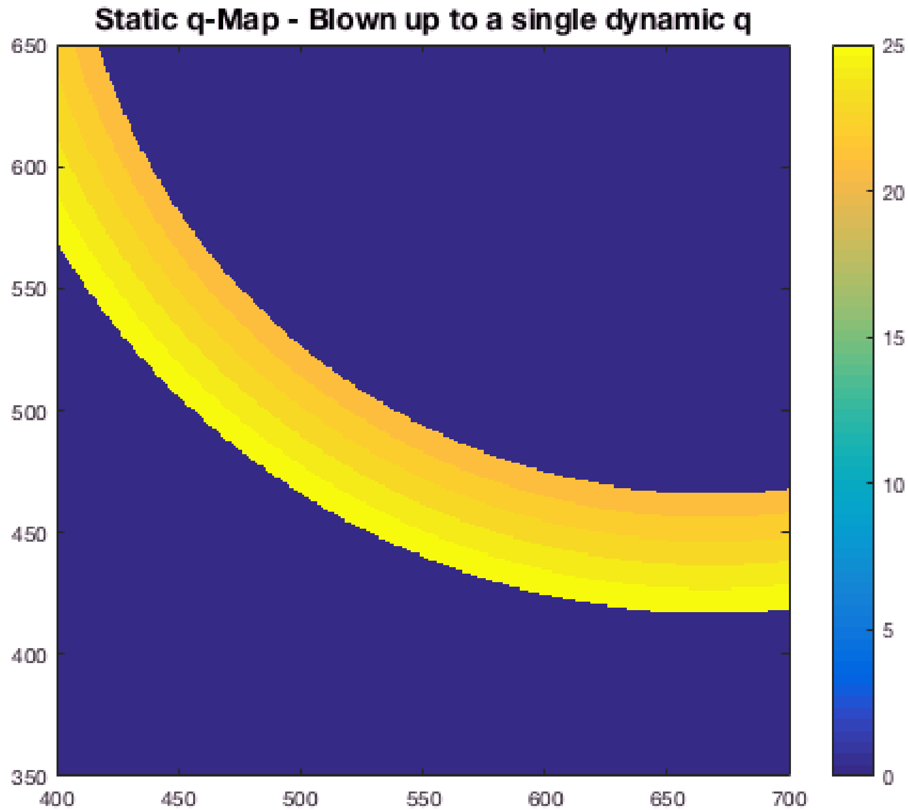
Zoom of Qmap or Wave vector map:

show a blow up of one dqmap value and the corr. sqmap values in it

```
dqmap_q=(dqmap==5)*5;
sqmap_q=zeros(size(sqmap));

for jj=21:25
    sqmap_q = sqmap_q + (sqmap==jj)*jj;
end

figure(15);
imagesc(dqmap_q);axis xy;axis image;colorbar;
xlim([400 700]);ylim([350 650]);
title('Dynamic q-Map - Blown up to a single dynamic q','fontsize',14,'fontweight','bold');
```




```
figure(16);
imagesc(sqmap_q);axis xy;axis image;colorbar;
xlim([400 700]);ylim([350 650]);
title('Static q-Map - Blown up to a single dynamic q','fontsize',14,'fontweight','bold');
```



MULTITAU Algorithm:

In this algorithm, the sampling time is increased while computing longer delay times. The reason is, for longer delay times, there are fewer combinations of frames that contribute to the correlations for a finite length of the time series of the data. This would result in poor statistics for the correlations as the delay times are increased resulting in a poorly defined baseline thereby making the correlation function less useful for quantitative analysis. Thus, in a pioneering paper by Schatzel in 1990, this algorithm was introduced with the above concept of analysis. However, since the time series is typically collected at a constant sampling time, the sampling time is increased during data analysis. **This is done by summing or averaging frames at shorter time intervals so as to increase the statistics for longer time delays. While this will smear the data because of the so called smoothing process inherent with the above averaging, for systems that are in equilibrium or stationary such that the correlation is only a function of the difference in the observation time and not dependent on the absolute time.**

In this logarithm, the averaging is done in multiple levels resulting in the name "multi-tau". It typically starts with 8 values of dt in the so called Zeroth level where there is no averaging done and hence the data is pristine (as acquired). As a default, dt = 1:8 are computed in this manner.

For the First level, two successive frames are averaged together to yield a new time series that is half the length of the original length of the time series. It has to be noted here that in this reduced time series, the time interval between two successive frames is 2x that of the original time series. Likewise, another 8 dt's are computed as a default which gives values of dt=2 to 16 in steps of 2. It is to be noted that the first half of the dt values in this level are already computed in the level prior to this and hence discarded (the lower level is always more precise because of less averaging of the frames).

Multi tau is shown below in a mathematical and illustrative form:

```
Num_Levels=8;
Tau_per_levels=8; %called as delays per level or dpl
level=NaN(Num_Levels,Tau_per_levels); %1st dim is number of levels and 2nd dim is taus/level

%show example of how a few multi tau levels look like
level(1,1:Tau_per_levels)=(1:1:8);
level(2,1:Tau_per_levels)=(2:2:16);
level(3,1:Tau_per_levels)=(4:4:32);
level(4,1:Tau_per_levels)=(8:8:64);
level(5,1:Tau_per_levels)=(16:16:16*8);
level(6,1:Tau_per_levels)=(32:32:32*8);
level(7,1:Tau_per_levels)=(64:64:64*8);
level(8,1:Tau_per_levels)=(128:128:128*8); %%2^(8-1):2^(8-1):2^(8-1)*2^8
% disp(level);

% %%write a general expression for multi tau
for tau=1:Num_Levels
    level(tau,1:Tau_per_levels)=(2^(tau-1):2^(tau-1):2^(tau-1)*Tau_per_levels);
end
% disp(level);

tableval=array2table(level);

disp(tableval);
```

~~level1~~	~~level2~~	~~level3~~	~~level4~~	~~level5~~	~~level6~~	~~level7~~	~~level8~~
1	2	3	4	5	6	7	8
~~2~~	~~4~~	~~6~~	~~8~~	10	12	14	16
~~4~~	~~8~~	~~12~~	~~16~~	20	24	28	32
~~8~~	~~16~~	~~24~~	~~32~~	40	48	56	64
~~16~~	~~32~~	~~48~~	~~64~~	80	96	112	128
~~32~~	~~64~~	~~96~~	~~128~~	160	192	224	256
~~64~~	~~128~~	~~192~~	~~256~~	320	384	448	512
~~128~~	~~256~~	~~384~~	~~512~~	640	768	896	1024

Crude G2 Calculation Example:

```
%take some point detector data in time
clear I G2 G2mean
I=(1:16);

dt=1;
G2{dt}=(I(1:end-dt).*I(dt+1:end));

dt=2;
G2{dt}=(I(1:end-dt).*I(dt+1:end));

dt=3;
G2{dt}=(I(1:end-dt).*I(dt+1:end));

for dt=1:8
    G2{dt}=(I(1:end-dt).*I(dt+1:end));
end

G2mean=cellfun(@mean,G2);
```

Two-Time Algorithm:

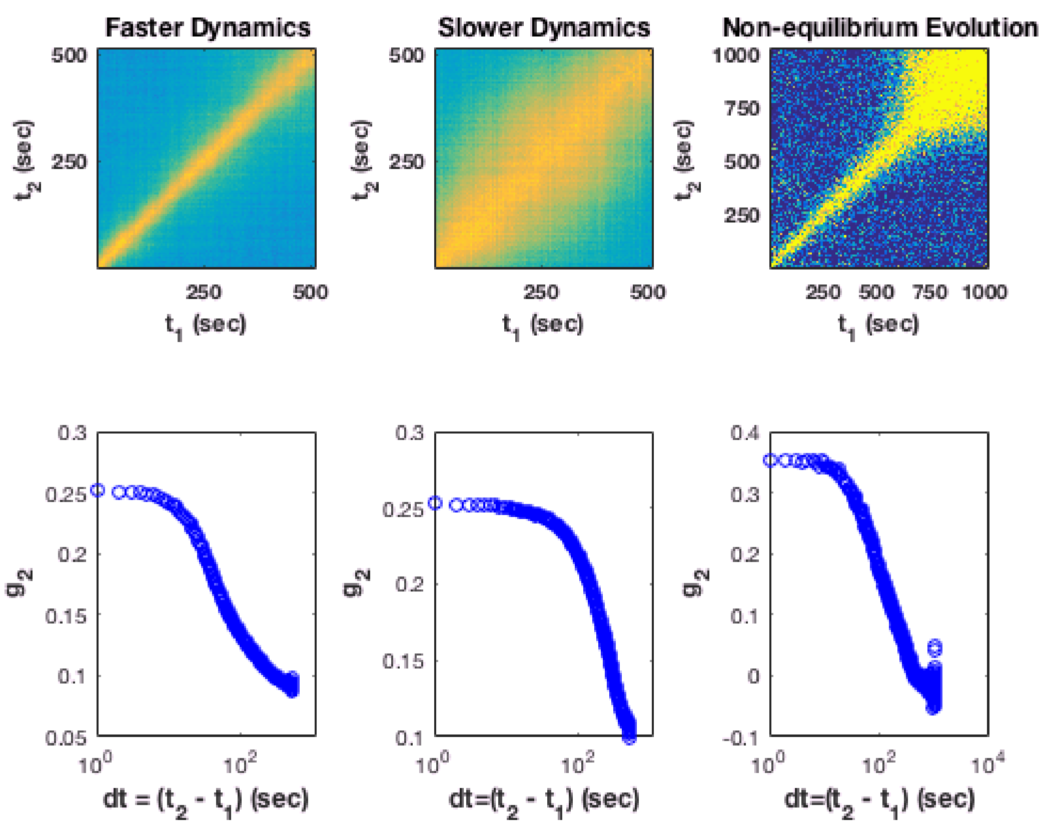
$$C(t_1, t_2, q) = \frac{\langle I(t_1, q) I(t_2, q) \rangle}{\langle I(t_1, q) \rangle \langle I(t_2, q) \rangle},$$

If the system is not in equilibrium or in steady state such as a system that is quenched or sheared, under the influence of an external field or any driven system, the measured dynamics or fluctuations are **not only a function of the difference in the observation time but is critically dependent on a time origin which could vary from one system to another. So under such circumstances, a two-time algorithm is defined where the correlation function is calculated as a function of t1 and t2, where t1 can be considered as the frames in time and t2 likewise. This algorithm is more intuitive to understand than multi-tau. For a given frame, say frame 1 or t1=1, correlations are computed with every other possible frame and this gives a row of correlations in a so called two-time matrix. This procedure is repeated for frame=2 or t1=2 with every other frame. The procedure will yield a N x N square matrix as the two-time correlation function where N is the total number of frames or time steps. From the above procedure, one can readily see that the matrix is symmetric whereby the upper and lower triangular parts are identical by definition.**

In the below sections, the above concept is shown mathematically and illustratively.

```
a=dir('A01*Shear*.mat');

load(a(1).name);
figure(113);
subplot(2,3,1);
imagesc(TwoTimeInfo.C{1},[0 .3]);axis xy;axis image;%colorbar;
xlabel('t_{1} (sec)','fontsize',12,'fontweight','bold');
ylabel('t_{2} (sec)','fontsize',12,'fontweight','bold');
set(gca,'xtick',[0 250 500],'fontweight','bold','fontsize',12);
set(gca,'xticklabel',{'0','250','500'},'fontweight','bold','fontsize',10);
set(gca,'ytick',[0 250 500],'fontweight','bold','fontsize',12);
set(gca,'yticklabel',{'0','250','500'},'fontweight','bold','fontsize',10);
title('Faster Dynamics','fontsize',12,'fontweight','bold');
g2{1}=TwoTimeInfo.g2{1};
deltaframes=TwoTimeInfo.deltaframes;
subplot(2,3,4);
semilogx(deltaframes,g2{1},'bo','markersize',6);
xlim([1,1100]);
% title('Faster Dynamics case','fontsize',12,'fontweight','bold');
xlabel('dt = (t_{2} - t_{1}) (sec)','fontsize',12,'fontweight','bold');
ylabel('g_{2}','fontsize',12,'fontweight','bold');

load(a(2).name);
subplot(2,3,2);
imagesc(TwoTimeInfo.C{1},[0 .3]);axis xy;axis image;%colorbar;
xlabel('t_{1} (sec)','fontsize',12,'fontweight','bold');
ylabel('t_{2} (sec)','fontsize',12,'fontweight','bold');
set(gca,'xtick',[0 250 500],'fontweight','bold','fontsize',12);
set(gca,'xticklabel',{'0','250','500'},'fontweight','bold','fontsize',10);
set(gca,'ytick',[0 250 500],'fontweight','bold','fontsize',12);
set(gca,'yticklabel',{'0','250','500'},'fontweight','bold','fontsize',10);
title('Slower Dynamics','fontsize',12,'fontweight','bold');
g2{2}=TwoTimeInfo.g2{1};
deltaframes=TwoTimeInfo.deltaframes;
subplot(2,3,5);
semilogx(deltaframes,g2{2},'bo','markersize',6);
xlim([1,1100]);
% title('Slower Dynamics','fontsize',12,'fontweight','bold');
```

```
xlabel('dt=(t_{2} - t_{1}) (sec)','fontsize',12,'fontweight','bold');
% ylabel('AUTOCORRELATION g_{2}','fontsize',12,'fontweight','bold');
ylabel('g_{2}','fontsize',12,'fontweight','bold');


b=dir('Si1*.mat');
load(b.name);
subplot(2,3,3);
imagesc(TwoTimeInfo.C{1},[0 .3]);axis xy;axis image;%colorbar;
xlabel('t_{1} (sec)','fontsize',12,'fontweight','bold');
ylabel('t_{2} (sec)','fontsize',12,'fontweight','bold');
set(gca,'xtick',[0 250 500 750 1000],'fontweight','bold','fontsize',12);
set(gca,'xticklabel',{'0','250','500','750','1000'},'fontweight','bold','fontsize',10);
set(gca,'ytick',[0 250 500 750 1000],'fontweight','bold','fontsize',12);
set(gca,'yticklabel',{'0','250','500','750','1000'},'fontweight','bold','fontsize',10);
title('Non-equilibrium Evolution','fontsize',12,'fontweight','bold');
g2{3}=TwoTimeInfo.g2{1};
deltaframes=TwoTimeInfo.deltaframes;
subplot(2,3,6);
semilogx(deltaframes,g2{3},'bo','markersize',6);
% xlim([1,1100]);
% title('Non-equilibrium Evolution','fontsize',12,'fontweight','bold');
xlabel('dt=(t_{2} - t_{1}) (sec)','fontsize',12,'fontweight','bold');
ylabel('g_{2}','fontsize',12,'fontweight','bold');
```

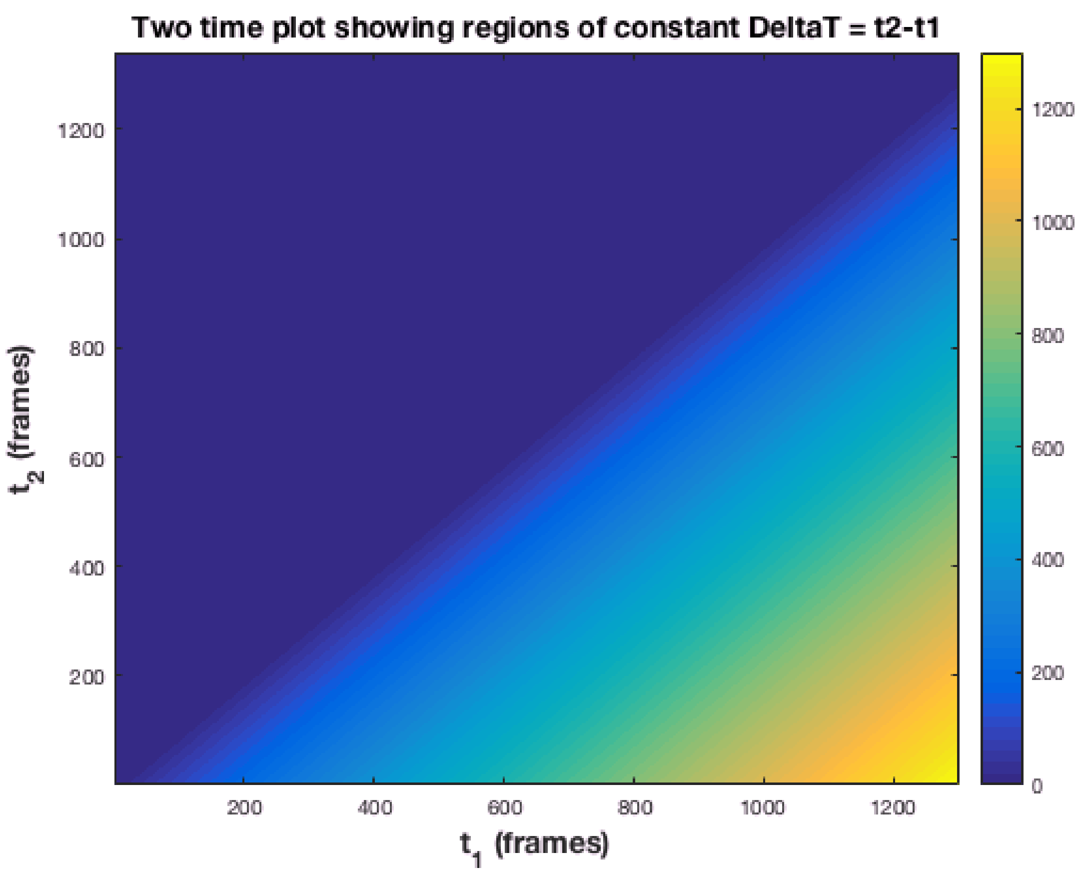


Concept of Twotime to One time:

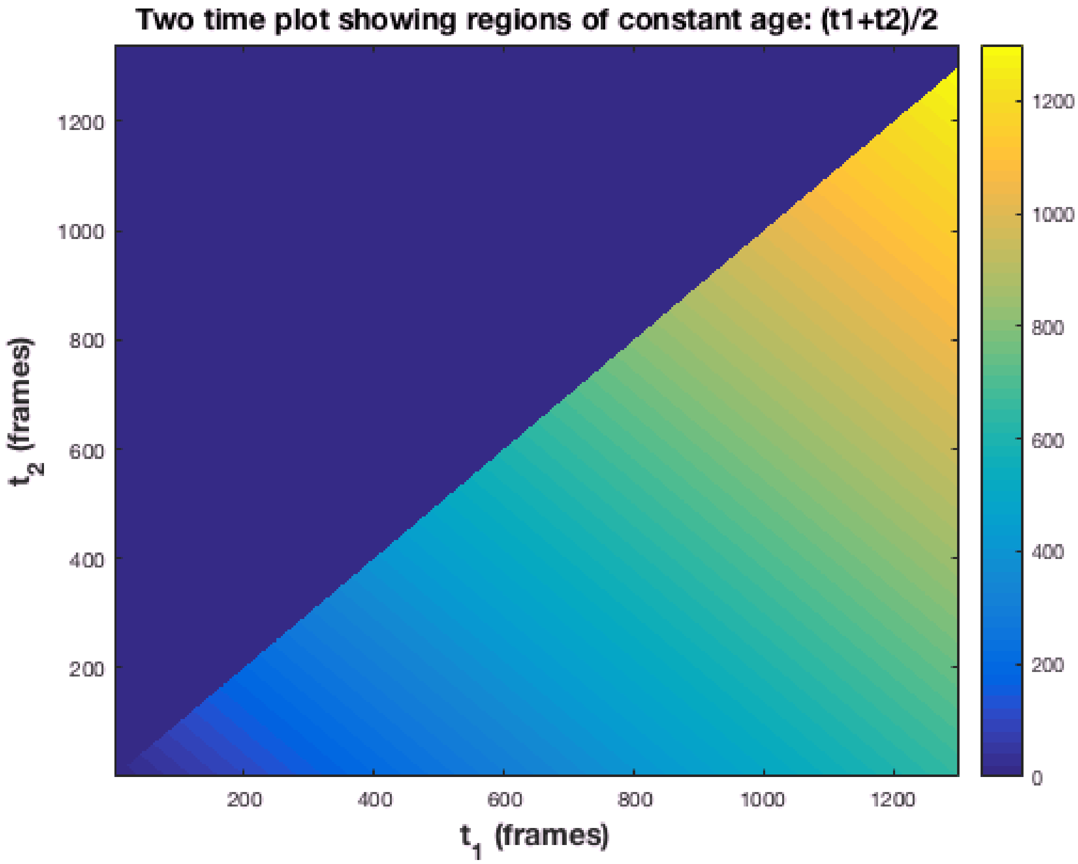
As explained above and as seen in the illustrations, each diagonal in the two time matrix corresponds to a single **dt** value with the main diagonal corresponding to dt=0 (which is not computed as that is selfcorrelation and will be abnormally high and will overwhelm everything else. At APS, we set dt=0 to be the same as dt=1 so as to give a continuous feeling, some people like to set the main diagonal to Zero.

If the correlation is the same along the diagonal, then the system is in equilibrium and obviously the system is **not in equilibrium** if the correlation varies along the diagonal. This is a test to decide whether **multitau or two time algorithm** is the most appropriate analysis for the data set in question.

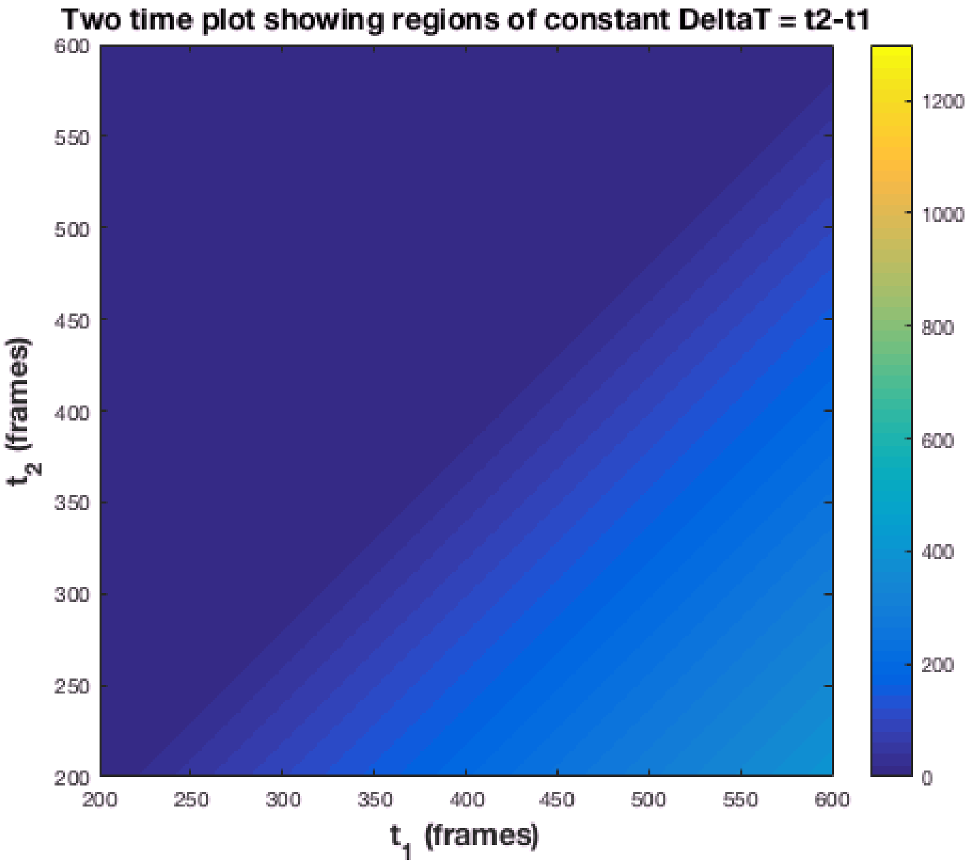
```
%explain two time to one time g2 concept
[z1,z2]=meshgrid(1:size(dqmap,1),1:size(dqmap,2));

dt=z1-z2;
dt=triu(dt); %upper and lower triangular are symmetrical, so get rid of one

t1plust2=(z1+z2)/2;
t1plust2=triu(t1plust2); %upper and lower triangular are symmetrical, so get rid of one

figure(31);imagesc(dt);axis xy;title('Two time plot showing regions of constant DeltaT = t2-t1
xlabel('t_1 (frames)','fontweight','bold','fontsize',14);
ylabel('t_2 (frames)','fontweight','bold','fontsize',14);
```



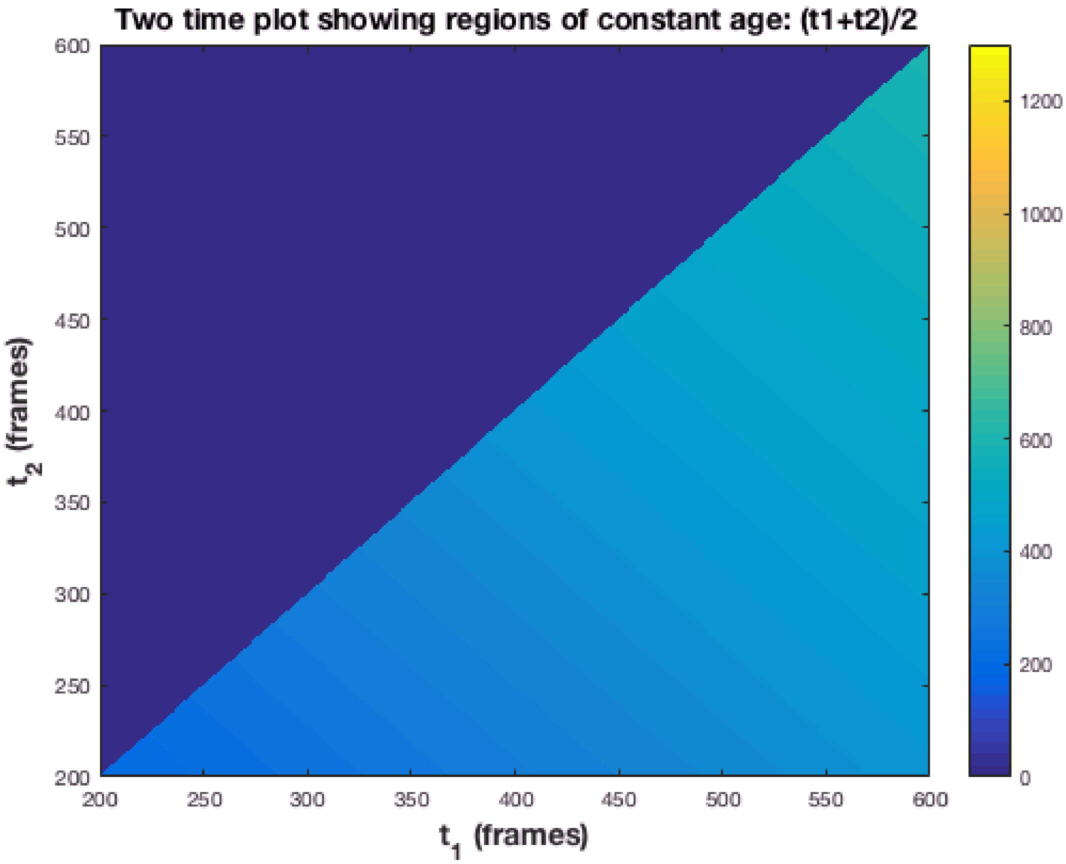

```
figure(32);imagesc(t1plust2);axis xy;title('Two time plot showing regions of constant age: (t1
xlabel('t_1 (frames)','fontweight','bold','fontsize',14);
ylabel('t_2 (frames)','fontweight','bold','fontsize',14);
```




```
figure(33);imagesc(dt);axis xy;title('Two time plot showing regions of constant DeltaT = t2-t1
axis image;
xlim([200,600]);ylim([200,600]);
xlabel('t_1 (frames)','fontweight','bold','fontsize',14);
ylabel('t_2 (frames)','fontweight','bold','fontsize',14);
```




```
figure(34);imagesc(t1plust2);axis xy;title('Two time plot showing regions of constant age: (t1
xlim([200,600]);ylim([200,600]);
xlabel('t_1 (frames)','fontweight','bold','fontsize',14);
ylabel('t_2 (frames)','fontweight','bold','fontsize',14);
```



The description ends here for now, will try to add more as time goes on.