Analysis of Michelson Interference Images

Alisa Engsberg
Summer 2008 Internship
University of Connecticut
Storrs, CT 06269
alisa.c.engsberg@gmail.com
August 8, 2008

Abstract

The goal of the Gluonic Excitation Experiment at Jefferson National Laboratory in Newport News, VA, is to measure the excitation spectrum of gluonic degrees of freedom inside mesons. Excited mesons are produced by a beam of polarized gamma rays directed onto a liquid hydrogen target. The gamma rays are produced by sending a high-energy electron beam through a diamond wafer so that coherent bremsstrahlung radiation is induced. In order to accomplish this, the diamond wafer must be thin, flat, and smooth. This paper deals with how the diamond wafers can be evaluated for these characteristics by creating interferograms using a Michelson interferometer and analyzing these interferograms using a powerful computer algorithm called Parallel Simulated Annealing. The aim of the project in question is to obtain the ideal settings for this algorithm to analyze the interferograms.

Background

The Gluonic Excitation Experiment (GlueX), based at Jefferson National Laboratory in Newport News, VA, is aimed at measuring the excitation spectrum of gluonic degrees of freedom inside mesons. In order to do this, a particle accelerator is first used to generate high-energy electrons. A diamond wafer, about twenty microns thick, is placed in the path of the high-energy electrons and as an electron approaches the wafer, its path is changed – i.e., it is accelerated into a path distinct from its original path – by its interaction with nuclei in the plane of the diamond wafer. This reduction of the speed of the electron is also a reduction of its energy, causing the electron to emit this excess energy in the form of a gamma particle. This process – by which a photon is emitted as a charged particle approaches another charged particle and is slowed in its path by it – is known as bremsstrahlung radiation. When this process occurs in a solid of crystalline structure such as a diamond, the effect is amplified as the momentum is taken up by the entire crystalline structure of the diamond, and this is called coherent bremsstrahlung radiation [1].

As the gamma particle and the electron emerge from the far side of the diamond wafer, the path of the electron is bent away by a large magnet but the gamma particle is allowed to continue on its path, traveling away from the electron and towards the detector. En route, the gamma particle is transformed into a pair of quarks by quantum fluctuations. These quarks remain as a

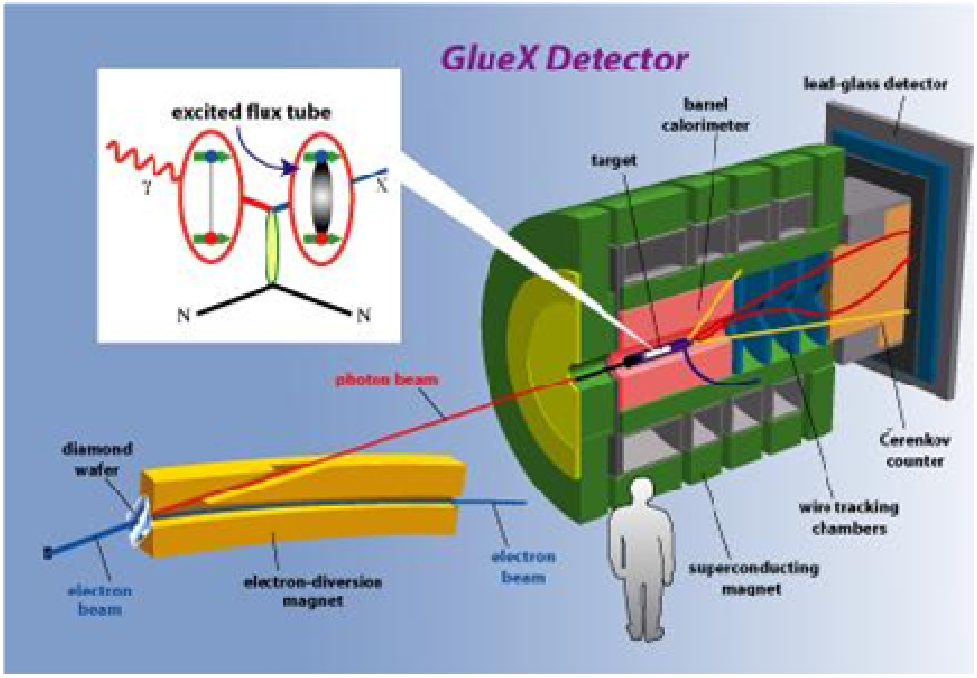

Figure 1. The GlueX Detector

pair because of the strong force existing between them. This force has been modeled as a *gluonic flux tube*, something like an elastic string connecting the two quarks [2].

Within the detector there is a target, and as the pair of quarks enters the target, they scatter off an incoming proton and are jolted into a quark-antiquark bound state. Sometimes when this happens, the gluonic flux tube described above is excited into vibration. Depending on the energy of the gluonic flux tube and its other quantum signatures, a given pattern of final state particles is created, and this is measured by the detector. Thus, one can tell by the pattern of the final state particles what sort of gluonic excitations occurred inside the detector during the experiment [2]. All of this is shown in Fig. 1 (courtesy of Reference [3]).

However, all of this begins with bremsstrahlung radiation due to a diamond wafer, and bremsstrahlung radiation occurring in a way that is useful for the experiment requires a very flat, smooth diamond about 20 microns thick. This is a very thin diamond wafer, and besides being very difficult to manufacture, it would be impossible to tell just by looking at it whether or not it was flat and smooth enough. So instead, a Michelson Interferometer is used to look at the surface of the diamond.

Methods

In order to determine the features of the surfaces of a diamond wafer, Michelson interference images were created and also simulated, and both the real images and the simulations

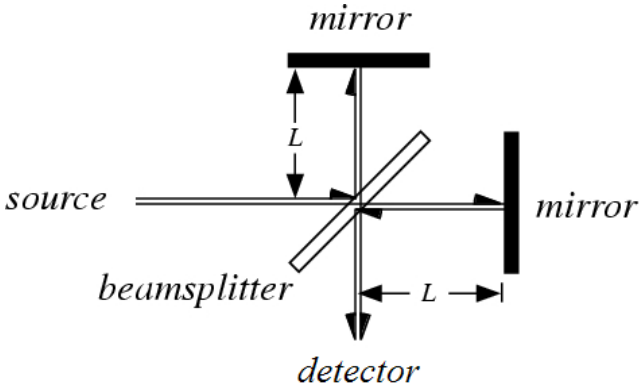

Figure 2. Michelson Interferometer.

were analyzed using a computer algorithm called Parallel Simulated Annealing. Through this process detailed information was obtained about the optimal settings for the algorithm to analyze the images.

A Michelson Interferometer is a bench-top instrument that begins with a laser. A beamsplitter is placed in the path of the laser beam and the beam is split, approximately half of the beam continuing straight on its prior path, and half of it redirected on a path ninety degrees from the original path, as shown in Fig. 2 (courtesy of Ref. [4]). This portion of the beam then reflects off a mirror and returns to the beamsplitter, so that half of it is reflected back toward the laser and lost for the purposes of the Interferometer, and half of the beam continues on a straight path from the mirror. At almost the same time, the portion of the beam that went straight in the original division is also reflected off a mirror placed in its path and returns to the beamsplitter, so that half of this beam continues straight toward the laser and is lost and half of it is reflected by ninety degrees, joining up with the beam that was original reflected ninety degrees.

Now there are two beams traveling together, and depending on the relative length of the

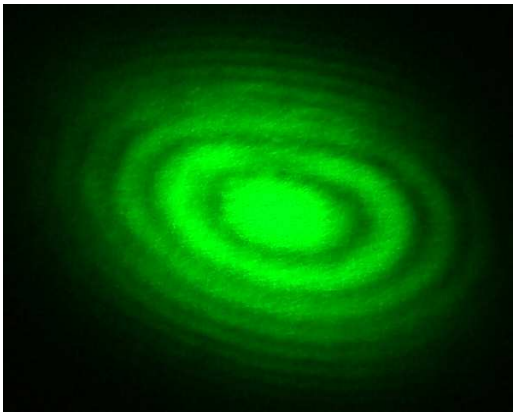

Figure 3. The interference produced by a Michelson Interferometer.

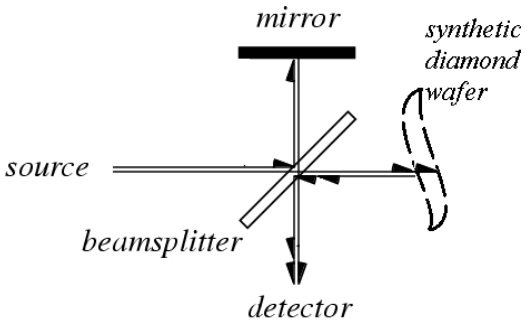

Figure 4. Michelson Interferometer with the diamond wafer in place.

two arms of the Michelson Interferometer, they could have any phase difference between them. Any two beams of light traveling together in the same direction at the same time interfere with each other, and the phase difference between the two determines the appearance of the image of the interference.

Light is an electromagnetic wave, that is, it is a propagating electromagnetic field that varies in strength as a sine wave. Thus if the electromagnetic wave begins by pointing towards the sky at peak strength, it will initially decrease in strength until it is down to nothing and then increase in strength as an electric field pointing towards the ground. For two beams of light to have a phase difference means that as they travel together in the same direction at the same time, the peak strengths of the two waves occur at different times. The two waves are "off" from each other. If they had the same phase, then as they propagated, the peak strengths would add to each other, making the peaks twice as high and the valleys twice as low and thus making the intensity of the light four times as great, since intensity is the square of the electric field. However, when the two waves are off, their electric fields will perhaps at some places be pointing in the same direction but

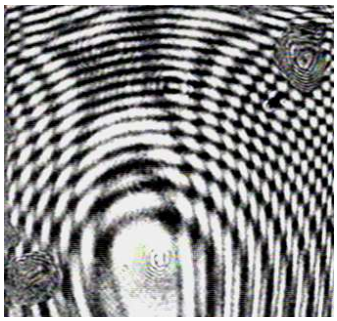

Figure 5. An Interferogram made with a Michelson Interferometer with a diamond wafer in place of one of its mirrors.

with different strengths and at some places pointing in different directions. Thus the overall field will be sometimes stronger than the field of either beam and sometimes weaker. Sometimes, the electric fields will cancel each other out entirely, so that there is no electric field at that point. This interaction, if caught in one place on a surface or by a camera, creates a pattern of interference fringes like the ones in Fig. 3 (courtesy of Ref. [6]).

In an ordinary Michelson interferometer, two nearly-planar mirrors are used at the end of the two arms of the interferometer, but for the experiment in question, one of those mirrors is replaced by a diamond wafer of the sort described above, as is shown in Fig. 4 (courtesy of Ref. [5]), and the assumption in doing this is that the surfaces of the diamond wafer are not planar. This means that the interference fringes will not be linear bands (as they would be for the mirrors), but rather elaborate swirls as shown in Fig. 5 (courtesy of Ref. [5]), as the pattern is highly complicated by the fact that the light reflected by the diamond wafer has not a single phase, but many phases, depending on the relative height of the surface of the diamond wafer at the point at which the light reflected from it. If, for example, there was a divot in the diamond wafer within the area which the laser beam struck, the portion of the beam reflected from the bottom of the divot would have a phase difference relative to the portion of the beam reflected from the edge of the divot.

A second added complication is that there are now actually three beams of light interfering with each other instead of just two, since the diamond wafer, instead of reflecting all the light in one fell swoop with perfect reflectivity, with very limited reflectivity reflects a tiny portion of the beam off of its front surface back toward the beamsplitter and a second tiny portion off of its back surface towards the beam splitter. This makes for three beams – two relatively weak ones from the diamond and one relatively strong one from the mirror. The weakness of the diamond beams, however, is not a matter for concern, since the beam from the mirror is strong enough to make strong interference patterns.

Thus, by placing the diamond wafer in the position of the second mirror of the Michelson Interferometer, a very complex interference image, or interferogram, is generated, with detailed information about the surfaces of the diamond wafer embedded in it. The next step is to analyze these images in order to extract from them the information needed about the surfaces of the diamond wafer.

The analysis of the interferograms is done by a cluster of forty-one servers using a powerful algorithm called Parallel Simulated Annealing, or ParSA, which is designed to mimic the process of annealing by which crystals are formed. In the process of forming crystals, annealing is a process in which the substance is heated up until it enters a disordered state, such as a liquid, and then gradually cooled to allow it to enter the highly ordered state of a crystal. As the substance is cooled, its molecules spontaneously arrange themselves into a crystalline structure.

Similarly, Simulated Annealing begins by assigning a preliminary solution to its problem and then the algorithm scans through solution space more slowly in order to find a solution that is closer to the best solution than the preliminary solution. As the algorithm scans, it assigns to each solution it comes upon a *cost function,* a value which indicates whether the solution is a good one or a poor one. A good solution is assigned a low cost function, and a poor solution is assigned a high cost function. In order to find the solution with the lowest cost function, the algorithm begins with a "heating up" stage, in which the "temperature" is increased. In the algorithm, the "temperature" corresponds to the probability that a poor solution (i.e., a solution have a high cost function) will be accepted. This "heating up" stage randomizes the starting point of the work, allowing the program to begin the "cooling down" stage from an arbitrary point in the solution space. Then the "cooling down" stage decreases the "temperature" so that only better solutions are accepted, and increasingly better solutions as time goes on. This process of "heating up" and then "cooling down" mimics the annealing process of growing crystals, but in the case of the algorithm its purpose is to ensure that the search does not end in some local minimum before it finds any solution near the true solution.

The Simulated Annealing algorithm is a good process for finding a good solution, but in order to improve both the quality of the solutions it produces and its time-efficiency, some refinements are necessary. The first improvement is to divide the search into multiple independent runs (MIR). This modification means that instead of each chain heating up and cooling down in one go, each chain undergoes many heating and cooling cycles containing shortened heating and cooling stages. So the algorithm heats up, accepting poor solutions, and then cools down, accepting good solutions, in many cases ending up in some local minimum. Then it begins again from another arbitrary point and yields another solution. Thus each chain yields many solutions – one best solution for each run. All of these solutions are recorded in the output of the program. In

this way, the global minimum can be selected as the best solution, and the interferogram and corresponding surface of the diamond wafer can be reproduced from that solution. The second improvement is that the algorithm is parallelized, and in this form it is called Parallel Simulated Annealing, or ParSA. This means that several chains are initiated at the same time, and multiple units, corporately called a cluster (but not to be confused with the complete, forty-one-server cluster), work cooperatively on finding a solution in each chain. The nodes cooperate in this by starting at the same point and moving separately from there, in order that they might rejoin at the best solution found by the cluster after a heating and cooling cycle in order to begin another heating and cooling cycle from that point [5].

The final product of all of this analysis will indicate a solution for the two surfaces of the diamond wafer. Thus, as adjustments are made by the manufacturer in how they manufacture the diamonds, the adjustments can be tested by this process in order to determine whether they are helpful adjustments or not in accomplishing the goal of smooth flat diamond surfaces.

At the outset of the portion of this project in question, the algorithm had been tested on a simulated interferogram made from two fifty by fifty pixel surfaces, each with only a slight curvature. This was a very simple test problem, and the results of that test are described in Reference [4]. The real diamond surfaces will be much more difficult problems – the images will be 400 pixels by 400 pixels, and the diamond wafer will have all sorts of complex little bumps and imperfections in its surface instead of being smooth and nearly planar.

For this reason, Legendre polynomials are a useful tool for describing these surfaces. Legendre polynomials are advantageous for these purposes because unlike Taylor expansions, which lose accuracy in some spatial location when truncated, Legendre polynomials lose a degree of accuracy indiscriminately across the entire surface when the higher degrees are cut off. This allows a degree of Legendre polynomials specified for a simulation to be related to a chosen minimum feature size to be resolved in the surfaces.

However, Legendre polynomials are ordinarily polynomials of only one variable and two variables are needed to create a surface. This obstacle is easily overcome since the product of two Legendre polynomials is also a Legendre polynomial. Thus, a Legendre polynomial in x is multiplied by a Legendre polynomial in y to create a single Legendre polynomial of two variables.

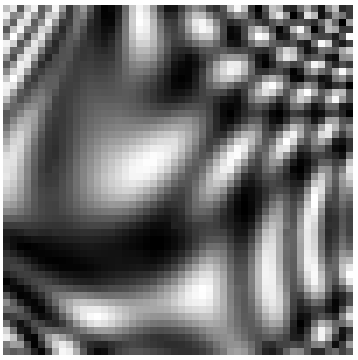

Figure 6. Simulated Interferogram

There is no need for the polynomial in x and the polynomial in y to be polynomials of the same degree, and thus there are endless possible combinations of Legendre polynomial degrees.

In order to store these polynomials in a form that is useful, an n by n matrix is created, where n is the desired number of degrees in the surface being generated. The indices of the matrix elements are always one greater than the degrees of the polynomials, since the indices begin at one and the degrees begin at zero. To create a random surface generator, we create a corresponding matrix of coefficients to go with the Legendre polynomials and have the elements of the coefficient matrix be randomly generated. In this way, each of the Legendre polynomials is randomly weighted, and when the polynomials are summed, the result is a surface of detail to the degree specified by n and generated randomly. Eq. 1 is used to describe their interference and create an image of that interference. Thus, two random surfaces and their interferogram have been created. Fig. 6 shows the final interferogram.

At the outset of this project, very little was known about how much time the cluster would require to analyze the real interferogram images obtained from the Michelson interferometer. In order to get this information, it was necessary to start small and simple, specifically with simulated surfaces fifty pixels by fifty pixels and third degree polynomials and work up to the size and complexity corresponding to an actual interferogram image. For any given image and solution dimension, the rate of convergence using the MIR algorithm is characterized by two parameters, a and k [5]. The method for determining these parameters is illustrated by the following procedure.

Using a fifty pixel by fifty pixel image with three degrees of Legendre polynomials, several jobs were run with a variety of settings for the cooling rate and run length. When these jobs were finished, the data were analyzed to find a and k. From these two values it is possible to find the optimal cooling rate and run duration for this image size and solution resolution. These optimized

parameters provide a starting point for a job involving a slightly larger or more complex image, and this allows the jobs to expediently increase in size and complexity up to the desired size and complexity – that of an actual diamond wafer. When the simulations of size and complexity similar to that of an actual diamond wafer have been analyzed, then a real diamond wafer can be analyzed in the same way, and the lab at the University of Connecticut will be enabled to provide feedback to the manufacturers on their manufacturing processes and adjustments.

To explain in more detail, there are six parameters that govern the cooling rate and duration of the annealing simulation within the MIR strategy. These are named *alpha*, *beta*, *end_temperature*, *Beta_Runtime*, *Run_Factor*, and *Samples*. *Alpha* (distinct from a) determines the rate at which the temperature decreases during cooling. The temperature T_n at step n in the cooling sequence is computed from that in the previous step using the relation $T_n = alpha * T_{n-1}$. So if *alpha* is set equal to 0.5, the cooling will be much more aggressive than if *alpha* is set equal to 0.9 and much less aggressive than if *alpha* is set equal to 0.2. There are advantages and disadvantages to each, since a run that cools too quickly is more likely to get stuck at a local minimum, whereas a run that does not cool quickly enough may not progress downward enough to end up at a good solution at all. *End_temperature* is the temperature at which the algorithm will end its cooling phase. *Beta_Runtime* is the factor by which the duration of a run is increased. Each run, that is, each heating and cooling cycle, consists of N steps. A sequence of runs with increasing N is called a sample. The first run in a sample always has N = 100,000, the second has N = 100,000 * *Beta_Runtime*, the third has N = 100,000 * $Beta\_Runtime^2$, and so on.

These runs continue until N exceeds the product 100,000 * *Run_factor*, at which point the MIR sample terminates. At that point, the sample is ended and a new sample begins, until the number of *Samples* specified have been completed.

When values have been assigned to each of these six parameters, the jobs can be run. Once the jobs have been run, several values can be found from the simulation results. These values are needed in order to find a and k. The first of these values are r(N) and u(N), where r is the number of runs of length N that were completed, and u(N) is the number of those runs that failed to produce acceptable solutions. A solution is acceptable if its cost function is below the *Cost_accepted*, that is, the maximum cost function that it can have and still be a visually indistinguishable match to the original image. With these two values, we can determine the value

of p, the probability of non-convergence, which is given by $p(N) = (u(N) + 1)/(r(N) + 2)$. The expected behavior of p is $p(N) = (a/N)^k$, for the same constant a, k that depend on the size of the problem and the chosen cooling parameters. At this point, if p is plotted against N as a log-log plot, the linear fit will give a as the slope and k as the y-intercept.

The statistical uncertainty of the data points in the log-log plot can be found by first finding the Variance of p, which is given by $V(p) = ((n_f + 1)(N - n_f + 1))/((N + 2)^2(N + 3))$, and then finding the standard deviation by $s = \sqrt{V(p)}$.

Results

Conclusion

In this experiment, simulated interference images were created and analyzed using the algorithm parSA. A large number of runs were completed on this image and the data collected were analyzed, showing that the ideal run length for this image was … In the next stages of this experiment, it will be important to create larger and more complex simulations, more and more closely resembling the real data, in order to determine the optimal settings to be used to analyze the real data.

Acknowledgements

I would like to thank Dr. Jones for giving me this amazing opportunity to work in his lab, and Jefferson Lab for providing the funding for it. I'd also like to thank Carl Nettleton, Igor Senderovich, and Woody Underwood for all of their help throughout the project.

Bibliography

[1] J. D. Kellie, et al., The Selection and Performance of Diamond Radiators used in Coherent Bremsstrahlung Experiments. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 545(1-2): 164-180, 2005.

[2] R. Clark, et al., Photoproduction of Gluonic Excitations and Unusual Mesons. *http://dustbunny.physics.indiana.edu/~dzierba/newhd/CDR_MASTER.ps.gz*

[3] GlueX Detector, 2004. *http://www.gluex.org*

[4] L. Motta. Michelson interferometer. Wolfram Science World.

[5] M. Demas, Analysis of Synthetic Diamond Wafer Interferograms Using a Parallel Simulated Annealing Algorithm. *http://zeus.phys.uconn.edu/halld/diamonds/MattDemasThesis-5-2008.pdf*, 2008.

[6] Falcorian. Michelson interferometer green laser interference. *http://commons.wikimedia.org/wiki/Image:Michelson_Interferometer_Green_Laser_Interference.jpg*.