8-24-2011

Effect of Prostaglandin E2 on Mechanical Stresses Applied by MC3T3-E1 Osteoblast-like Cells on a Soft Hydrogel Substrate

Abhijit Deb Roy
University of Connecticut, abhijit.debroy@gmail.com

Effect of Prostaglandin E2 on Mechanical Stresses Applied by MC3T3-E1 Osteoblast-like Cells on a Soft Hydrogel Substrate

Abhijit Deb Roy

B.Tech., National Institute of Technology, Warangal, 2005

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

at the

University of Connecticut

2011

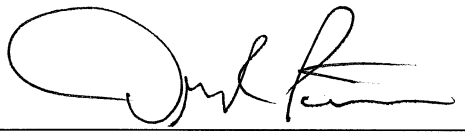
APPROVAL PAGE

Master of Science Thesis

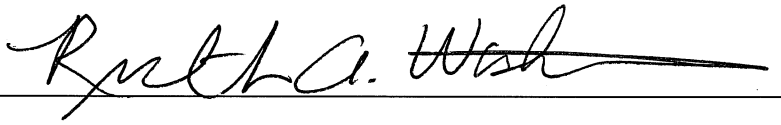
Effect of Prostaglandin E2 on Mechanical Stresses Applied by MC3T3-E1 Osteoblast-like cells on a Soft Hydrogel Substrate

Presented by

Abhijit Deb Roy, B.Tech.

Major Advisor 

Donald R. Peterson

Associate Advisor 

Ruth A. Washington

Associate Advisor 

Carol Pilbeam

University of Connecticut

2011

Acknowledgements

To my major advisor, Dr. Donald Peterson: Thank you for your help, support and encouragement, without which this thesis would not have materialized. Thank you also, for being a great mentor and guide.

To Dr. Ruth Washington: Thank you for your guidance and your help through difficult times, and for giving me the opportunity to work with you. Thank you for having faith in my research abilities and in me.

To Dr, Carol Pilbeam: Thank you for your encouragement and valuable advice. It helped me focus and streamline this thesis.

I would like to thank Dr. Carol Norris, for her help and assistance with the confocal microscope; Dr. Shiva Kotha, under whom I learnt many of the techniques I have used in this study, and Dr. Kristen Govoni, for allowing me to practice cell culture in her laboratory.

I would like to thank my fellow graduate student Robert Knapp for his help with coding, without which the program would not have been possible. Thank you to Yamalia Roberts for your help with the experiments, and for your continual support. Thank you Katelyn Burkhart and Diana Ramirez for patiently making the gel substrates for the experiments.

Finally, Thanks to my family, my friends and my peers, for your support through this journey, which, though challenging, has been immensely rewarding.

Table of Contents

Table of Figures

Table of Tables

Abstract

Osteoblasts are sensitive to mechanical stimuli and release Prostaglandin E2 when exposed to a fluid shear stress. They are also sensitive to sub-micron scale surface patterns and mechanical properties of any substrate they are cultured on. The exact mechanism by which these cells sense mechanical stress and communicate this information is not well established. A study of the mechanical stresses applied by the osteoblasts, under the influence of prostaglandin E2, on a compliant substrate provided information regarding intercellular communication via changes in the substrate surface pattern by changes in magnitudes of the strains on the surface, or changes in the distribution of these strains.

A digital image correlation program was developed using the Levenberg-Marquardt optimization algorithm to analyze images and compare the deformations between pairs of images, in terms of displacements and displacement gradients. Similar image processing programs commonly use the Newton-Raphson algorithm for optimization. The development of the program to analyze changes in the surface pattern of a compliant poly-N-isopropyl acrylamide gel, on which MC3T3-E1 osteoblast-like cells had been plated and exposed to prostaglandin E2, was the objective of this thesis.

Comparisons of the distributions and locations of substrate-embedded fluorescent marker beads before and after the addition of prostaglandin E2 to the

media showed differences in the substrate surface pattern, which was consistent with the morphological changes in the cells observed on the addition of prostaglandin E2. No changes were observed in the magnitudes of the stresses applied by the cells on the substrate surface. These observations also suggest that the MC3T3-E1 cells communicate with one another mechanically by changes in the substrate surface pattern by redistribution of the stresses applied but not by changes in the magnitude of these stresses.

The performance of the Levenberg-Marquardt algorithm was found to be comparable to that of the Newton-Raphson algorithm and the development of this image-analysis program provides an alternative algorithm for future use in digital image analysis.

Further investigation into the effects of different concentrations of prostaglandin E2 on the mechanical behavior of osteoblasts and the behavior of osteoblasts in a three dimensional environment will lead to a better understanding of the means by which these cells communicate. The techniques developed in this study can assist in identifying changes in mechanical behavior of cells and the image-analysis program can be used to compare sub-pixel deformations between sub-micron scale images in cell-based research or in other fields.

1. Introduction

The relationship between substrate mechanical properties and cell behavior has generated interest recently among scientists. It has been shown by Lo et al. (2000) that cells exert contractile stresses on their substrate and that they exert different magnitude of stresses on substrates of different stiffness. Digital image correlation (DIC) is an image-analysis technique that is being increasingly used to analyze similarities and differences between two images in general, and to compare and measure deformations in an image relative to a reference in particular. DIC programs are particularly useful in analysis of microscopy images to displacements, gradients and rotational effects of an agent or phenomenon on a substrate. Osteoblasts are bone cells that are responsible for production of the bone mineral matrix and are involved in the bone remodeling process. Osteoblasts have been observed to be sensitive to mechanical stimuli, including strains, in the substrate on which they are cultured. The mechanism by which these cells sense mechanical stresses and communicate the information to one another is not well established. The stresses applied by the cells on a compliant substrate cause the substrate body to deform. Prostaglandin E2 (PGE2) is one of the first molecules released by osteoblasts in response to mechanical stimuli and they have been known to cause cell shape change in osteoblasts and increase the number of intercellular gap junctions (Ponik et al., 2004; Guignandon et al., 1995; Shen et al., 1986). Inter cellular gap junctions are involved in maintenance of tissue homeostasis and intercellular communication.

The cells are able to sense changes in the strains caused by changes in the stresses they apply and are able to communicate with one another by applying different contractile stresses on the substrate. In this thesis, it is hypothesized that PGE2 causes a change in the stresses applied by MC3T3-E1 osteoblast-like cells on a compliant hydrogel substrate. A digital image correlation (DIC) program was developed in Matlab (Mathworks, Natick, MA, USA) using Levenberg-Marquardt algorithm to compare the deformations (strains) in a poly-N-isopropyl-acrylamide hydrogel caused by MC3T3-E1 cells before and after exposure to 0.05 μM and 0.1 μM PGE2 in the culture media.

1.1 Mechanical Stimulus and Osteoblasts

Mechanical stimuli play a significant role in bone cell activities, especially in bone remodeling and bone cell differentiation behavior. The mechanical stress is manifested as a fluid shear stress in the cell substrate. Figure 1.1 is an illustration of how shear stress affects a substrate, where τ is the shear stress acting on the substrate of height, l, and causing a deformation Δl as shown in the figure. The strain caused by τ is $\Delta l/l$. The stress and the strain are related to the elastic modulus (Y) of the substrate as given in equation (1).

$$Y = \frac{\tau}{\Delta l / l}, \qquad (1)$$

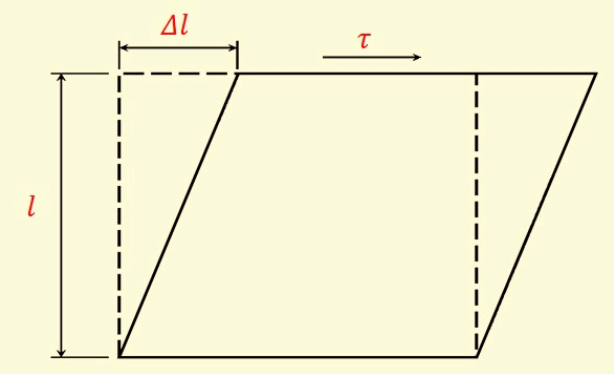


Figure 1.1 Shear Stress Acting on a Substrate

The mechanism by which osteoblasts detect and transduce mechanical stress to activate signaling pathways is not very well understood. Endothelial cells have been observed to communicate with one another by applying mechanical stresses to the substrate (Lo et al., 2000). It has been observed that cells apply stresses on the culture substrate *in vitro* (Igarashi et al., 1994) and that the stresses applied by osteoblast like cells on a soft substrate vary with the stiffness of the substrate. Osteoblasts are primarily responsible to perform the adaptive functions in response to mechanical stress (Skerry, 2008), and have been observed to release Prostaglandin E2 (PGE2) into the extracellular environment in response to mechanical stress (Kumegawa et al., 1984). *In-vitro* studies of osteoblasts have shown that a steady fluid flow induced shear stress causes up-regulation (increased expression) of cyclooxygenase 2 (COX-2) and PGE2 (Bakker et al., 2001). There is a strong direct correlation between mechanical stimulus and an increase in PGE2 release by osteoblasts. COX-2, which is an enzyme involved in the production of PGE2, has been observed to

promote bone formation by increasing proliferation and differentiation of osteo-progenitor cells (Forwood, 1996). PGE2 is known to influence the number of gap junctions (Shen et al., 1986), which are involved in tissue homeostasis as well as intercellular communication. This suggests that PGE2 is involved in the process of osteoblasts adapting to an external mechanical stimulus.

Mechanical stresses applied by cells have been found to vary with the mechanical properties of the substrate they are cultured on *in vitro* (Lo et al., 2000). Substrate mechanical properties have been observed to influence cell proliferation, migration and differentiation (Igarashi et al., 1994; Meazzini et al., 1998) as well as the stresses applied by the cells on the substrate. As seen in equation (1), strain is inversely proportional to the Young's modulus of elasticity and, to observe any changes in the strains in a substrate, it is important to have a compliant (soft) substrate on which the cells are cultured.

1.1.1 Elasticity of the Gel Substrate

The Young's modulus of elasticity of the gel substrates can be measured using a microsphere indentation method, where a small stainless steel microsphere of known diameter is placed on top of the gel substrate containing fluorescent beads. An inverted fluorescent microscope is then used to focus on the set of beads just beneath the ball on the gel surface. The ball is removed using a magnet and the resultant vertical movement of the substrate is measured by re-focusing the microscope on the previous set of fluorescent beads. The

micrometer movement is the distance the gel substrate travelled following ball removal. The distance is used to compute the Young's modulus of elasticity (E) using the Hertz equation (2).

$$E = \frac{3\left(1-\upsilon^2\right)\mathrm{f}}{4r^{1/2}\delta^{3/2}} \ , \tag{2}$$

where r is the marker bead radius, f is the force applied by the stainless steel ball, δ is the indentation of the substrate and υ is the Poisson ratio. The force applied, f, is the gravitational force applied by the stainless steel ball minus its buoyancy. The hydrogel mechanical properties are affected by environment variables, such as temperature and humidity, making it important to perform the measurements under controlled experimental conditions.

1.2 Digital Image Processing

Digital image correlation is a computational technique devised to determine displacements and displacement gradients between digital images. Digital images are compared to evaluate deformations between the two images and these deformations are then analyzed to evaluate the displacement and displacement gradients at the sub-pixel level. Gray scale digital images are discrete arrays of numbers, which indicate the intensity of a pixel at a specific location. The sub-pixel accuracy in displacement measurement is obtained by rendering the discrete numerical arrays as pseudo-continuous by interpolating

between the array points such that differentials can be obtained in the spaces between the original array points. The goal of the digital image correlation is to evaluate the deformation parameters, namely the displacements and the displacement gradients along the axes. This method can be utilized to measure the stresses applied by an osteoblast on a hydrogel substrate by incorporating fluorescent beads in the gel-substrate body and tracking the movement of these beads. Bead displacements are caused by stresses applied by the cells on the substrate and are proportional to the distribution of the applied stresses. The displacement gradients along different axes are the strains along those axes. At a constant modulus of elasticity, it can be seen from equation (1) that the displacement gradients are directly proportional to the magnitude of the cell stresses. Any changes in the displacement and the displacement gradients of the substrate surface provide an estimate of the changes in the magnitude and the distribution of the stresses applied by osteoblasts on the gel substrate.

Digital image correlation uses the cross-correlation function to compare the distribution pattern of the pixel intensities between two images and expresses the deformation between the images as displacements and displacement gradients. In figure 1.2, an example of such a deformation is given, where the pixel located at (x,y) in the original image has an intensity F and, in the deformed image, the pixel has moved to a location (x*,y*) and has an intensity G.

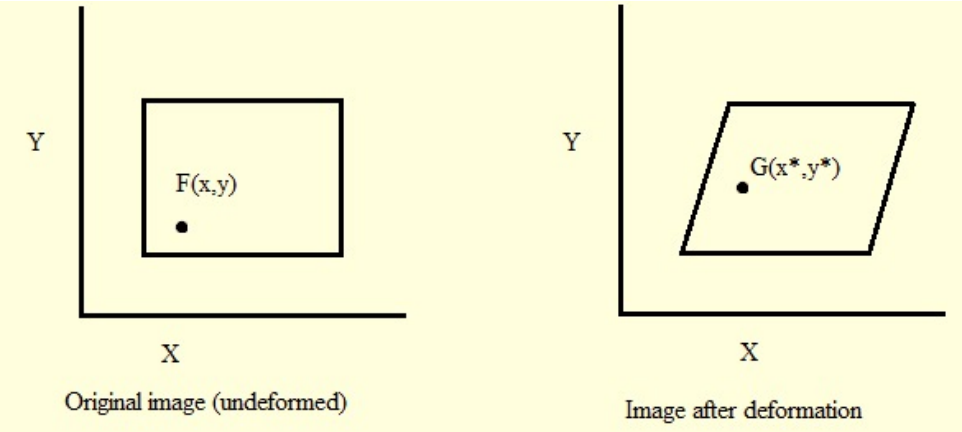


Figure 1.2 Example of Deformation of Images

Mapping the deformed image with the reference (i.e., original) image, the two locations can be related as:

$$x^* = x + u(x,y), \tag{3}$$

$$y^* = y + v(x,y), \tag{4}$$

where u and v are the x and y directional displacement components of the image, respectively. Applying a first order Taylor series expansion on the above relationship, the new location (x^*, y^*) is found to be related to the origin, (x_0, y_0), as given in equations (5) and (6), where $\Delta x = (x - x_0)$, and $\Delta y = (y - y_0)$.

$$x^* = x_0 + u_0 + \frac{\partial u}{\partial x}\Delta x + \frac{\partial u}{\partial y}\Delta y\,, \tag{5}$$

$$y^* = y_0 + v_0 + \frac{\partial v}{\partial x}\Delta x + \frac{\partial v}{\partial y}\Delta y\,, \tag{6}$$

Figure 1.3 shows how different gradients influence a point distribution array.

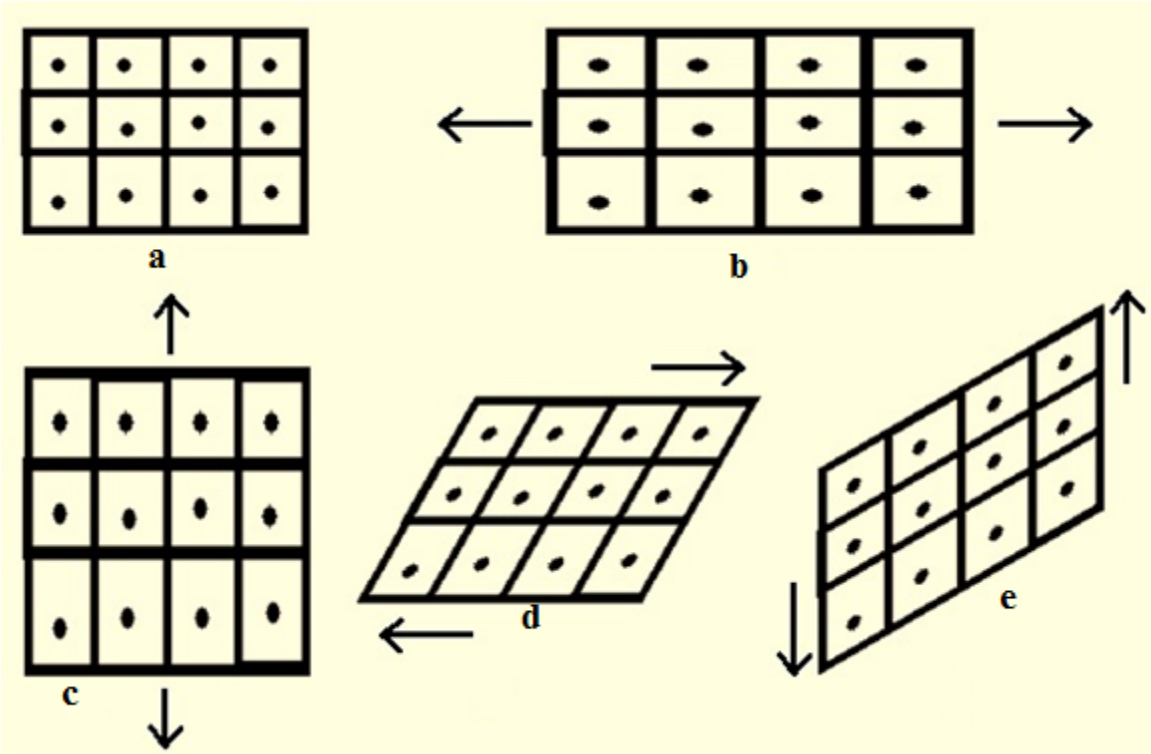


Figure 1.3 Effects of Different Displacement Gradients on a Uniform Point Distribution (a) Original image, (b)$\frac{\partial u}{\partial x} > 0$, (c)$\frac{\partial v}{\partial y} > 0$, (d)$\frac{\partial u}{\partial y} > 0$, (e)$\frac{\partial v}{\partial x} > 0$

For the evaluation of the values of various partial differential functions, it is important that the array distribution is continuous. Digital images are discrete in nature and must be rendered continuous by interpolation. Bicubic interpolation was used on the images for interpolation to ensure that the differential terms are continuous through the region. This enabled measurement of sub-pixel gray scale intensities as well as differential values in between pixels.

The cross-correlation function used in the DIC program is defined in equation (7), where the maximum value of S is 2 when the two images are

inverse of one another and the minimum value of S is 0 when the two images are the same.

$$S(x, y, u, v, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}) = 1 - \frac{\sum[F(x,y) \times G(x*, y*)]}{\sqrt{[\sum(F(x,y)^2) \times \sum(G(x*, y*)^2)]}}, \quad (7)$$

An iterative process is used to evaluate the values of S for different subsets of the pair of images and to find the values of the parameters $u, v, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}$ and $\frac{\partial v}{\partial y}$. Different deforming displacements and gradients are applied on the reference image and it is compared with the deformed image. Since the value of S is minimum when the two images are the same, the problem of finding the parameters evolves into a minimization problem, where the displacements and the displacement gradients applied to the reference image to minimize the value of S gives the values of the parameters. An example to illustrate how DIC works is demonstrated in figure 1.4, where the reference image A is being compared to the distorted image E. The value for S between A and E is 0.88, which is very high.

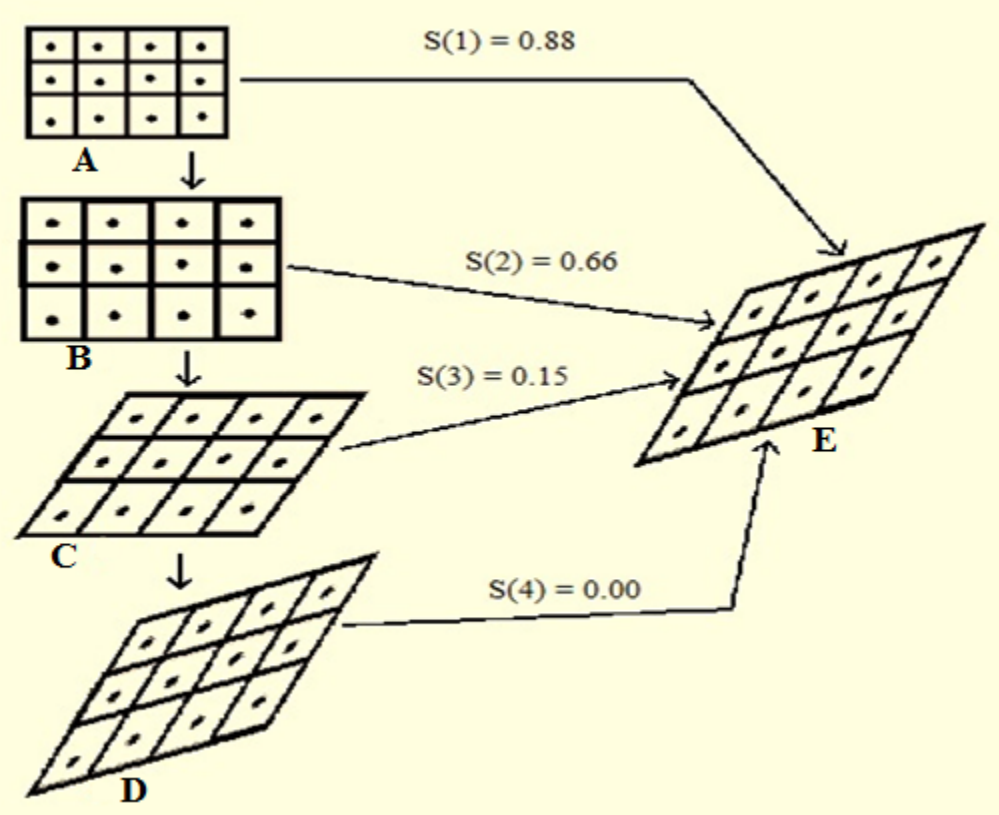


Figure 1.4 Example of Digital Image Correlation

A displacement gradient is first applied to A transforming it to B. The value of S now drops to 0.66 indicating that B is more similar to E than A was. Another displacement gradient is applied to B changing the distribution to C and causing the S to drop further to 0.15. This iterative process continues until the value of S is minimized (in this example, image D with S = 0). The displacements and the displacement gradients applied on the original image are stored and updated through the iterations and the values obtained when the minimum value for S is reached are the optimized values for the parameters. The displacements and gradients to be applied to the original image at each step is determined by the Levenberg-Marquardt algorithm.

1.2.1 Optimization Algorithms

Digital image correlation is a computationally intensive numerical iterative process and must be optimized for good performance. There are many optimization algorithms available, but the Levenberg-Marquardt (LM) appears to be the most suitable for this application. The Levenberg-Marquardt algorithm can be represented as a blend of the simple steepest-descent method and the Gauss-Newton method, which are methods to solve non-linear least squares minimization problems. These methods start with initial estimates, or guesses, for the parameters, which are iteratively changed until the desired degree of accuracy is obtained. Some of the minimization algorithms relevant to this application are explained briefly in the following section using a non-linear function $f(x)$.

1.2.1.1 Steepest-Descent Algorithm

Let $f(x)$ be a non-linear function of x where x is a real entity. It is assumed that there exists a real value of x that minimizes $f(x)$. The simplest iterative method to reach the solution to the minimization problem is the steepest-descent method where the iterations start with an initial estimate for the solution and the parameter values are changed iteratively according to equation (8),

$$x_{i+1} = x_i - \lambda \left(\partial f(x_i)\right)/\partial x, \tag{8}$$

where λ is an arbitrarily chosen step size. This method suffers from convergence problems if the step size is too large. If the step size is too small, the convergence is extremely slow. An example of the steepest-descent method is illustrated in figure 1.5,

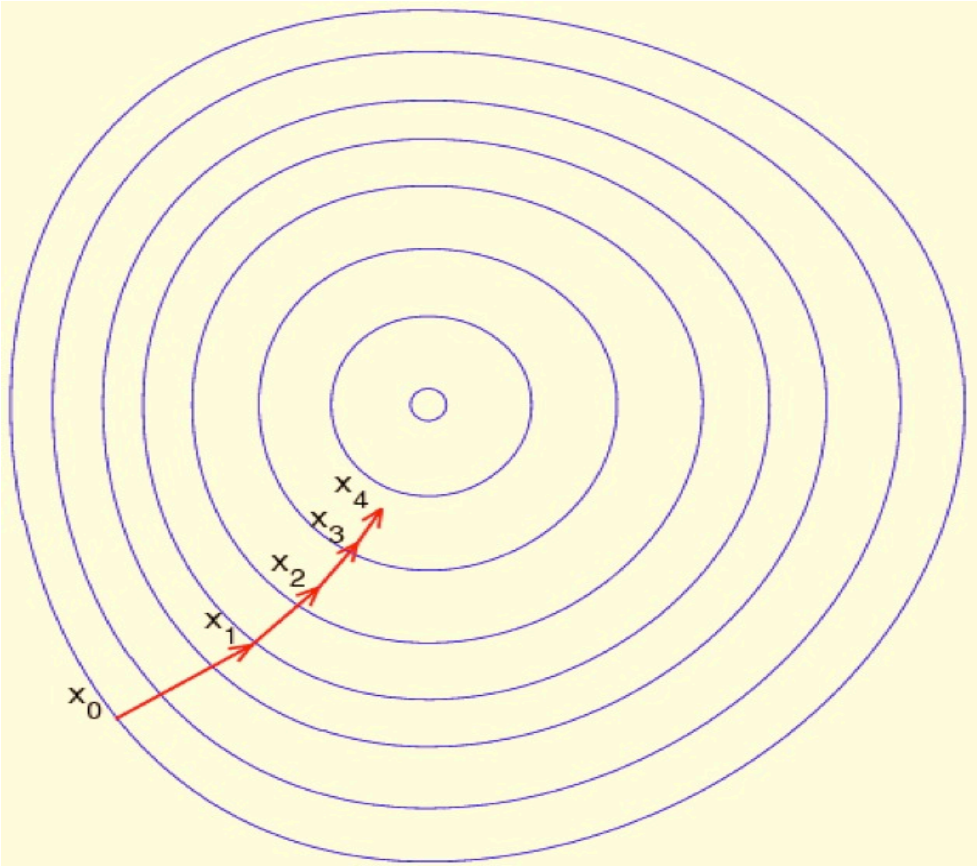


Figure 1.5 Example of Steepest-Descent Algorithm

where the blue circles indicate the surface distribution of a function f(x). The initial guess for the minima is x_o, and x_1 is computed using equation (8). Subsequently x_2 and x_3 and the other iterative values for the parameter are computed until the required accuracy is reached.

1.2.1.2 Gauss-Newton Algorithm and Newton-Raphson Algorithm

Gauss-Newton method, illustrated in figure 1.6, uses a Taylor series expansion of $\frac{\partial f}{\partial x}$ and ignores the higher order terms (i.e., order > 2) assuming $f(x)$ to be quadratic around x_{min}. The step size in this method is updated according to equation (9).

$$x_{i+1} = x_i - \left(\frac{\partial^2}{\partial x^2}(f(x_i))^{-1} \frac{\partial f(x_i)}{\partial x}\right), \tag{9}$$

This method ensures rapid convergence provided that the initial guess is close to the real solution but, if the initial guess is far from the solution, convergence may not be reached.

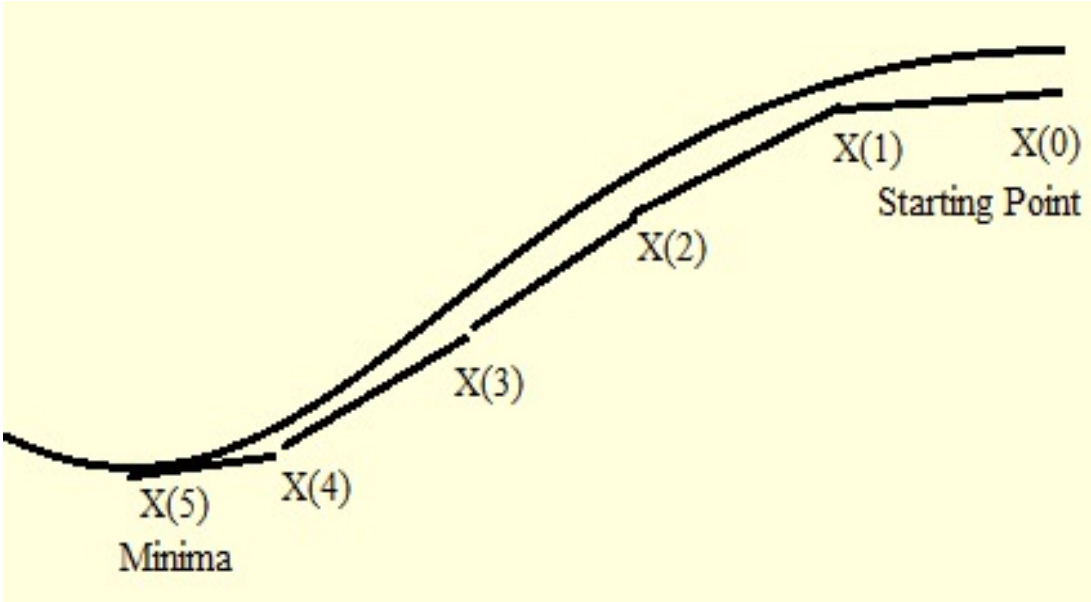


Figure 1.6 Example of Gauss-Newton Algorithm

The curve is the graphical representation of a function f(x) and x_0 is the initial guess. The iterations are performed using equation (9) until a minima is obtained. The quadratic nature of the Gauss-Newton algorithm ensures that the successive steps in the minimization process closely follows the curve; however, this causes the optimization to be inefficient in areas of low curvature.

Currently most DIC programs use the Newton-Raphson algorithm for optimization. This method represents a modification of the Gauss-Newton method, where the iterative step is calculated as given in equation (10).

$$x_{i+1} = x_i - f(x_i) / \left\{ \frac{\partial f(x_i)}{\partial x} \right\}, \quad (10)$$

The Newton-Raphson method is computationally faster than the Gauss-Newton method, as it does not compute the second-degree differential array. It suffers from convergence issues like the Gauss-Newton method.

1.2.1.3 Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm uses a blend of the steepest-descent method and the Gauss-Newton method methods incorporating a variable step-size that is dependent on the value of the error. The iterative step for the Levenberg-Marquardt method is given as equation (11).

$$x_{i+1} = x_i - \left[\frac{\partial^2}{\partial x^2} \left(f(x_i) \right) + \lambda \times diag\{ \left(\frac{\partial^2}{\partial x^2} \left(f(x_i) \right) \right) \} \right]^{-1} \frac{\partial f(x_i)}{\partial x}, \quad (11)$$

where λ is an arbitrary number and $diag\{(\frac{\partial^2}{\partial x^2}\big(f(x_i)\big)\}$ is the diagonal matrix of $\frac{\partial^2}{\partial x^2}\big(f(x_i)\big)$. This method applies a large step in the direction of low curvature and a small step in the direction of high curvature.

1.3 Previous Research

Prostaglandin E2 and its effects on osteoblasts have been studied extensively because of its intimate association with the osteoblasts' response to mechanical stimuli. Kumegawa et al. (1984) and Hakeda et al. (1985) studied the effects of PGE2 on alkaline phosphatase in MC3T3-E1 cells. Shen et al. (1986) and Yang et al. (1998) showed that prostaglandins cause a change in osteoblast cell morphology as well as an increase in intercellular communication. Guignandon et al. (1995) and Hughes-Fulfor et al. (1996) studied the effects of gravitational changes on osteoblasts and found that PGE2 is involved in the adaptive mechanism of osteoblasts and that PGE2 caused a change in the osteoblasts cell morphology. The involvement of PGE2 in alkaline phosphatase activity and intercellular communication suggests that PGE2 plays a significant role in the adaptive mechanism of osteoblasts exposed to fluid shear stress. Bakker et al. (2001) showed that the production of PGE2 by osteoblasts is shear stress dependent. Ponik et al. (2004) found that formation of focal adhesion induces PGE2 release by osteoblasts suggesting that PGE2 plays a role in intercellular communication. Meazzini et al. (1998) had shown that there is osteoblast cytoskeletal modulation when the cells are exposed to mechanical

stimulus. Malone et al. (2007) demonstrated that PGE2 has an effect on the actin cytoskeleton in MC3T3-E1 cells. Zhao et al. (2006) have shown that MC3T3-E1 cells respond to submicron scale surface pattern. Dembo et al. (1999) demonstrated a way to compare and measure the stresses exerted by fibroblasts on a compliant gel substrate, which showed a way of examining intercellular communication by mechanical means. Lo et al. (2000) showed that cells exert different stresses on substrates of different stiffness and also that the movement of MC3T3-E1 cells can be directed by stretching the substrate on which they are plated. Reinhart-King et al. (2008) found that endothelial cells communicate with one another via strains exerted on a compliant substrate. The findings of Lo et al. (2000) and Reinhart-King et al. (2008) suggest that osteoblasts may communicate to one another via modulation of mechanical strains on the culture substrate, and that this modulation may be dependent on the mechanical properties of the culture substrate.

Most DIC softwares developed in the past have used the Newton-Raphson algorithm for optimization. The algorithm used in the Levenberg-Marquardt DIC program developed was based on the structure used by Bruck et al. (1989). They had used Newton-Raphson algorithm to create a program for the estimation of displacement gradients along with the displacements by using first order partial differential terms. Lu et al. (2000) demonstrated an improvement in the accuracy of the displacement and displacement gradients by using second order partial differential terms for the analysis. Kuendong (2000) also developed

a similar program and illustrated a method to account for out of plane displacements. Schreier et al. (2002) conducted extensive work on the errors in correlation method due to the mismatch in the shape functions of the regions of interest being analyzed using the DIC program. Pan et al. (2008) analyzed the choice of the size for the region of interest and showed that, for optimal convergence, the region of interest used must contain enough information to be identified as a unique region in the total image. This helped in the determination of the amount of fluorescent beads to be added to the gel matrix.

2. Methods

MC3T3-E1 cells were cultured in a compliant p-NIPA gel substrate containing fluorescent marker beads and exposed to 0.05 and 0.10 μM PGE2. Fluorescent confocal images were taken of the substrate surface before and after the addition of the PGE2 and compared using a DIC program developed in Matlab using Levenberg-Marquardt algorithm.

2.1 MC3T3-E1 Cell Culture

MC3T3-E1 osteoblast-like cells were cultured in α-minimum essential media (α-MEM) containing 10% volume/volume (v/v) fetal bovine serum (FBS), with 1% penicillin and streptomycin antibiotics and 0.1% fungizone antifungal agent, under standard conditions (37ºC, 5% CO_2 and 75% humidity). Well-dispersed cells were seeded to the fibronectin coated pNIPA gel substrate at ~5000 cells per glass bottom dish. A low number of cells were seeded so that the average distance between individual cells was at least equivalent to the size of a cell. This helped determine the deformations caused on the gel substrate body by a cell independent of the influence of neighboring cells. To reduce cell division during the imaging process, the media was replaced with α-MEM containing the antibiotics and the antifungal agents without the FBS before the experiment.

The cells were stained with 3,3'-dilinoleyloxacarbocyanine perchlorate (DiO, Invitrogen # D3898) before being placed on the hydrogel surface. The

stock solution was prepared by dissolving DiO in dimethyl sulfoxide. DiO is not readily soluble in water and must be treated to minimize precipitation when added to the cell culture media. The dye was dissolved in chloroform at 50mg/ml concentration and the solution was mixed with an equal volume of octadecylamine and heated up to 45°C. DiO was precipitated by adding ice-cold methanol. The mixture was centrifuged for 10 minutes at 3000 rpm and the supernatant was discarded. The remaining solvent was allowed to evaporate and the pellet was dissolved in dimethyl-sulfoxide at a concentration of 1.5 mg/ml. This stock solution was stored at -20°C for future use. Before staining, the stock solution was heated up to 37°C in a water-bath and centrifuged for five minutes at 2000 rpm. Fifty µl of the supernatant was added to the cell culture containing 8 ml media and the culture dish was incubated at 37°C for two hours. The media was aspirated and the culture was washed three times with PBS to remove debris. The stained cells were used to seed the hydrogel substrate.

2.2 Poly-N-isopropylacrylamide Hydrogel Substrate

Poly-N-isopropylacrylamide hydrogels were prepared on 35mm glass bottom dishes from Mattek Corp. (Agawam, MA, USA). Forty percent weight/volume (w/v) acrylamide and two percent w/v N,N'-methylene bis-acrylamide stock solutions were sterilized by filtering through 0.22 μm filter. The filtration also removed debris from the stock solutions improving the quality of the confocal microscope images.

Glass bottom dishes were made hydrophilic to improve adhesion of the gel to the dish by activation with 3-Aminopropylmethoxysilane (APS). The glass surfaces on the dishes were passed through the inner flame of a Bunsen burner, smeared with 0.1 molar (M) sodium hydroxide and air-dried. The dried surfaces were smeared with APS and incubated at room temperature for about 10 minutes. The surfaces were washed with de-ionized water until clear, incubated with 0.5% glutaraldehyde for an hour at room temperature, washed with de-ionized water until clear, and then air-dried. These activated glass-bottom dishes were used within two weeks of activation. Number 1 glass cover slips with 18 mm and 12 mm diameter were siliconized to deactivate the surface, which ensured minimal adhesion of the gel surface to the cover slip. The cover slips were washed with de-ionized water and the kept in a 2% v/v solution of dimethyldichlorosilane in chloroform for 10 minutes. They were removed from the solution, air dried in a chemical hood, and washed with de-ionized water. Deactivation of the surface was confirmed visually by checking the shape of water droplets on the cover slips.

For good confocal image quality, it is important that the gel substrate is horizontal. The confocal microscope imaging focal plane is horizontal and any incline in the gel surface would severely impair the quality of the image. The effect of a non-horizontal substrate surface is shown in figure 2.1. For the gel surface strain analysis, the confocal images should be taken as close to the gel surface within the gel body as possible. Due to the objective used in the imaging

process, the gel thickness could not exceed 1.9mm. The imaging was carried out at standard cell culture conditions i.e., 37ºC, 5% CO_2 and 75% humidity. The high temperature and humidity causes expansion of the gel body and, to account for this expansion, the initial gel thickness was kept below 1.0 mm. The volume of the monomer solution to make a gel substrate this thin is very small making the polymerization highly susceptible to inhibition by atmospheric oxygen.

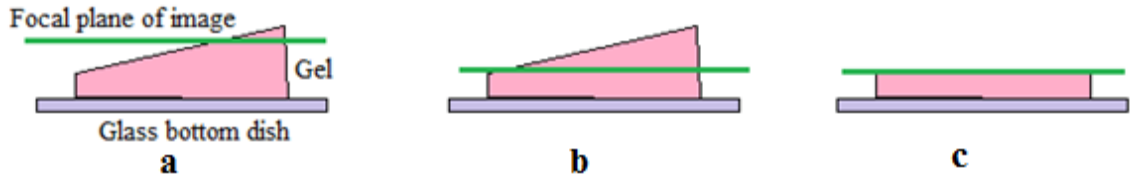


Figure 2.1 Effect of a Non-horizontal Surface on Confocal Imaging (a) Incorrect Imaging Due to Insufficient Information, (b) Incorrect Imaging Due to Varying Depth, (c) Correct Imaging

A horizontal polymerization chamber was designed to ensure that the gels were horizontal. A desiccating chamber was fixed on a stable horizontal surface using adhesives and a standard 6-well plate was fixed inside the desiccator and the plate was kept horizontal. The 35mm glass bottom dishes fit snugly in the wells and this ensured that the dishes were horizontal during polymerization. The plate being fixed to the desiccator ensured that the dishes remain stable and do not move from any disturbances during the substrate preparation. The chamber set-up is shown in figure 2.2. To minimize the presence of oxygen, CO_2 was initially used to flush the chamber; however, this caused severe slowing of the polymerization due to a drop in the pH. Alternatively, nitrogen gas was used to

eliminate the effects of pH and to reduce the polymerization inhibition by oxygen of the monomer solution making the polymerization process more efficient.

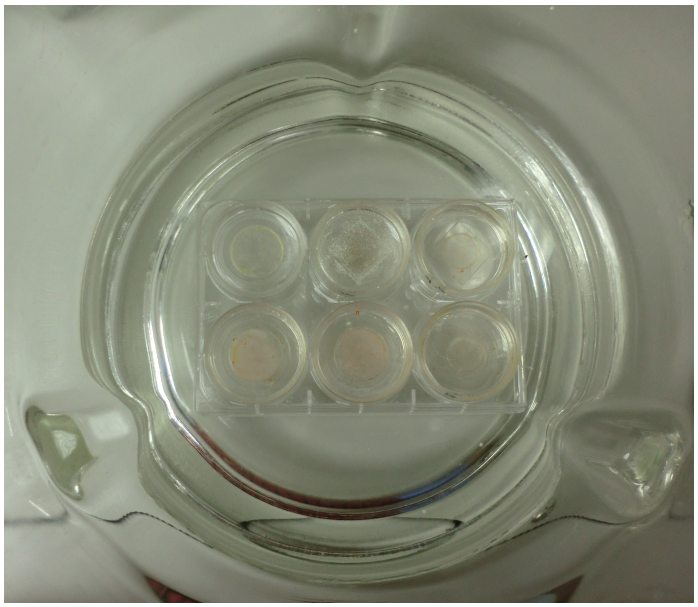


Figure 2.2 Horizontal Polymerization Chamber

The gel substrates were made in two layers to control the thickness and help focus the microscope on the gel surface during microscopy. For the first layer, the monomer solution for p-NIPA was prepared with 7.5% w/v acrylamide and 0.1% w/v N,N'-methylene bis-acrylamide in 0.01 M Hepes at pH of 8.5. The pH was adjusted using 1M sodium hydroxide and 1M hydrochloric acid. The monomer solution was degassed under partial vacuum for 60 minutes to remove dissolved oxygen. The polymerization promoter TEMED and the initiator, freshly prepared 10% ammonium persulfate solution, were added to the monomer solution to initiate polymerization. Immediately after addition of the initiator, 50μl of the monomer solution was pipetted on to the activated glass bottom dish and a 12mm deactivated cover slip was gently placed on the droplet to form a thin flat

layer of the polymerizing solution in between the glass bottom dish and the cover slip. After polymerization, which took about 45 minutes, the gel was flooded with 50mM Hepes at pH of 8.5. The top cover slip was then removed using forceps and the gel was washed thoroughly using 50mM Hepes on a shaker. The gel formed in this step had a thickness of 45μm and the process is shown in figure 2.3.

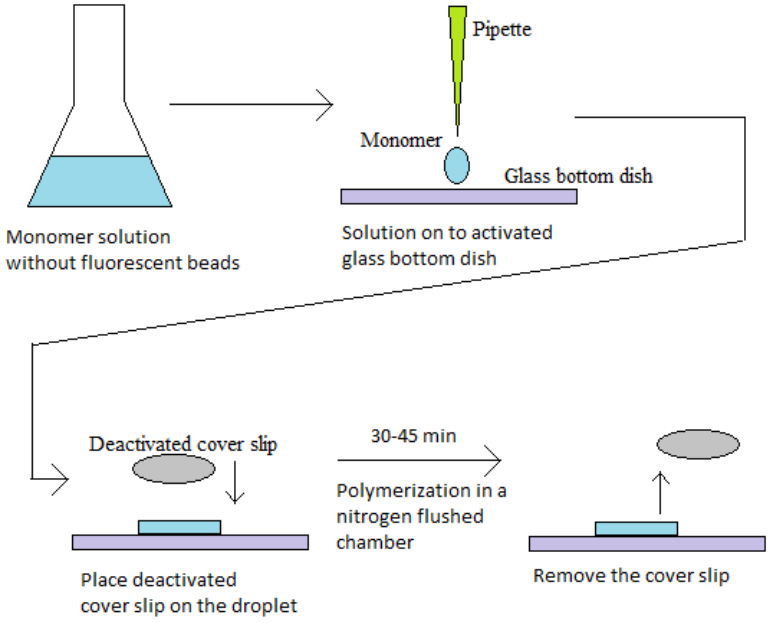


Figure 2.3 Steps to Prepare the First Layer in the Gel Substrate

For the second layer of the gel substrate, the monomer solution was prepared as described in the first step. Before degassing, 0.84μm diameter Nile Red fluorescent beads (Spherotech Inc. # FP-0856) were added to the monomer mixture at 0.02mg/ml concentration. After degassing for 60 minutes, the promoter and the initiator were added to start the polymerization. For the second gel layer, 100μl of the degassed solution containing the fluorescent beads was added on

top of the first gel layer and an 18mm diameter deactivated cover slip was used to make the gel and this step is illustrated in figure 2.4. The total thickness of the gel formed was approximately 60μm and the thickness of the gel layer containing the fluorescent beads on top of the transparent gel layer was about 15μm.

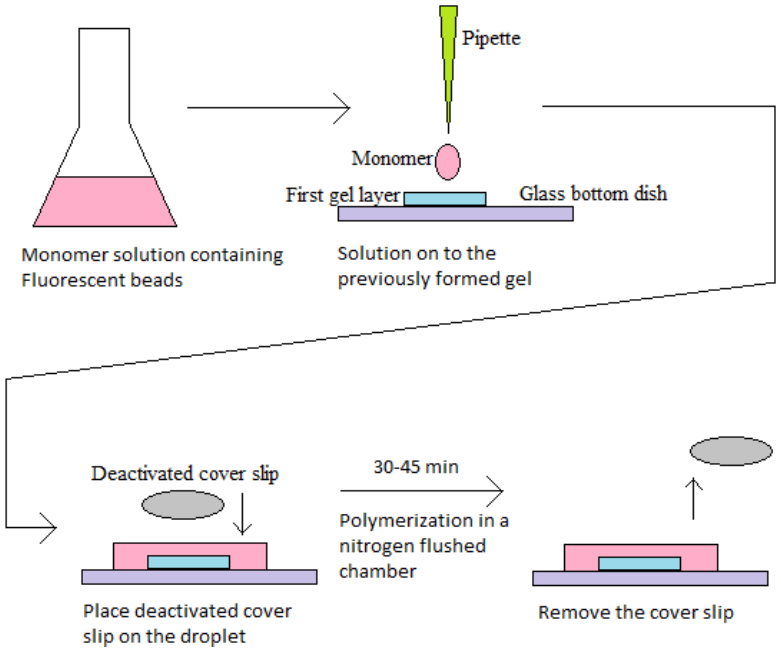


Figure 2.4 Steps to Prepare the Second Layer in the Gel Substrate

p-NIPA gels are very non-reactive and this property is important in using it as an ideal inert growth substrate for cells minimizing any chemical interactions with the cells plated on it. Due to its inert nature, p-NIPA substrates do not encourage cell adhesion on its surface. By attaching a fibronectin layer good focal adhesions of the cells on the gel substrate were ensured. The gel substrate was covered with a freshly prepared working solution of 1mM sulfo-SANPAH and 0.5% DMSO in 50mM Hepes at pH of 8.5. The gel was exposed to UV light (320-

350 nm wavelength) for 8 minutes, which activated the gel surface and this was repeated to ensure activation of the entire surface. The activated gels were washed thoroughly in 50mM Hepes, at pH of 8.5, three times to remove excess reagents. Bovine fibronectin in PBS at 15μg/ml was added to the gel surface and left on a shaker at room temperature overnight to allow the proteins to bind to the activated surface. The fibronectin-coated gels were washed with 70% ethanol to sterilize and then with sterile PBS three times. The MC3T3-E1 cells were seeded on the substrate at 5000 cells per culture dish. These steps are illustrated in figure 2.4. The cells were incubated overnight under standard cell culture conditions and prior to the experiment the media was replaced with media without FBS.

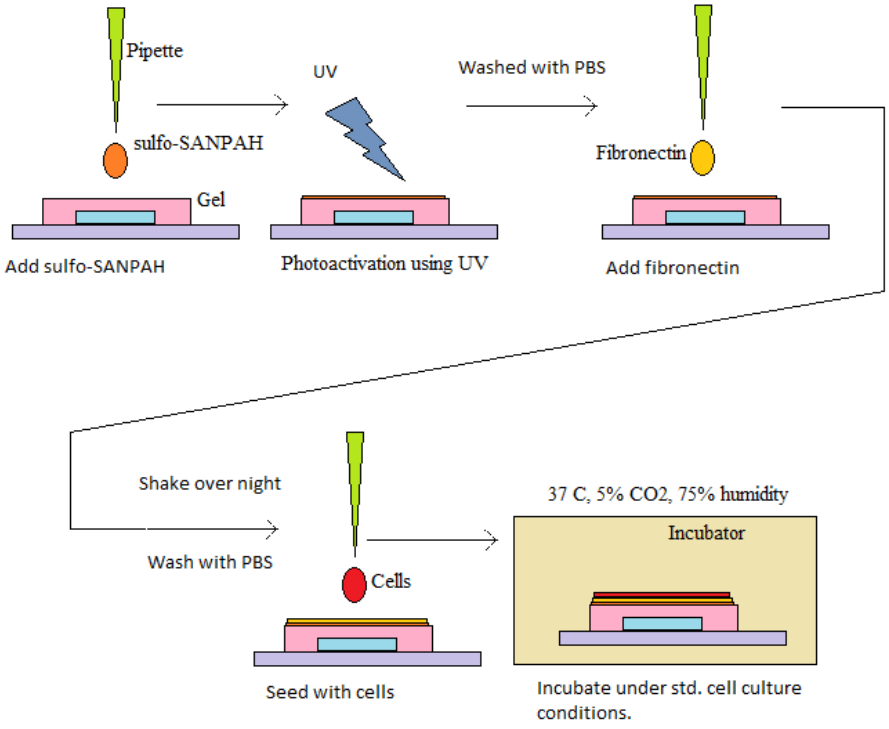


Figure 2.5 Preparation of the Gel Substrate for Cell Culture and Seeding the Substrate with Cells.

2.2.1 Young's Modulus of Elasticity of the Hydrogel Substrate

The Young's modulus of elasticity of the gel substrates were measured using the microsphere indentation method using a stainless steel microsphere of 0.64 mm diameter and the Nile Red fluorescent beads used earlier. The indentation procedure was performed five times at random locations on the gel substrate. The Poisson ratio of acrylamide gels has been experimentally estimated by Li *et al.* (1993) to be 0.3, which was used in these calculations. Measurements were taken under standard cell culture conditions with the substrate submerged in the cell culture media. The density of the cell culture media was assumed to be the same as that of water. The top and side views of the steel microsphere on the p-NIPA gel used in the indentation process is shown in figure 2.6.

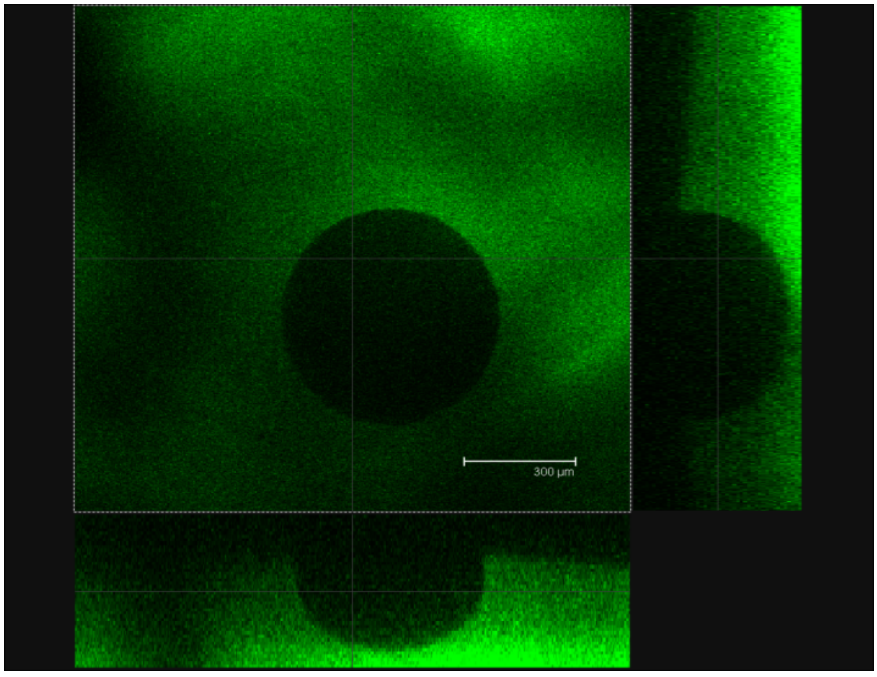


Figure 2.6 Microsphere Indentation Method

The parameters for the experimental setup are given in table 2.1. The Young's moduli for the gel calculated are given in table 2.2.

Table 2.1 Parameters Used in Measurement of Young's Modulus

Parameter	Value
Radius of steel ball	0.32 mm
Poissons Ratio	0.3
Density of Steel Ball	7800 kg/m^3
Weight of Steel Ball	10.492 μN
Buyoant Force	1.345 μN
Net Force	9.147 μN

Table 2.2 Observed Young's Moduli

Observation #	Indentation (μm)	Young's Modulus (kPa)
1	4.8	33.23
2	5.2	29.84
3	4.6	35.21
4	5.1	30.44
5	4.9	32.25
Average Young's Modulus		32.194

2.3 Experimental Procedure

Three sets of cell-gel systems were observed with Nikon A1R Confocal microscope. Each glass-bottom dish with the cells had 1.5 ml of FBS free α-MEM. The confocal microscope had a live cell chamber that allowed the experiment to be carried out under standard cell culture conditions. The glass

bottom dish was fixed to the live cell chamber stage using vacuum grease to minimize chances of the dish shifting during the addition of PGE2. A cell that was located away from neighboring cells by a distance at least equivalent to twice the average size of a cell was used for imaging. The gel surface was approximately located using phase contrast microscopy. Fluorescent images were taken using a 40x objective with a numerical aperture of 0.6 and an adjustable working distance of 2.8 to 3.6 mm. The Perfect Focus System, which is a feature of the microscope, was utilized to neutralize effects of thermal expansion of the stage. After capturing the first image, PGE2 was added to the existing media in the dish so that the media contained 0.05 μM or 0.1 μM PGE2, based on the experiment performed. Previous experiments had shown that the morphological changes in the cells on exposure to PGE2 was stabilized after about 90 minutes, as shown in figure 2.7. After 90 minutes of adding the media containing PGE2, another image was taken and the images of the beads before and after addition of PGE2 were analyzed using digital image correlation.

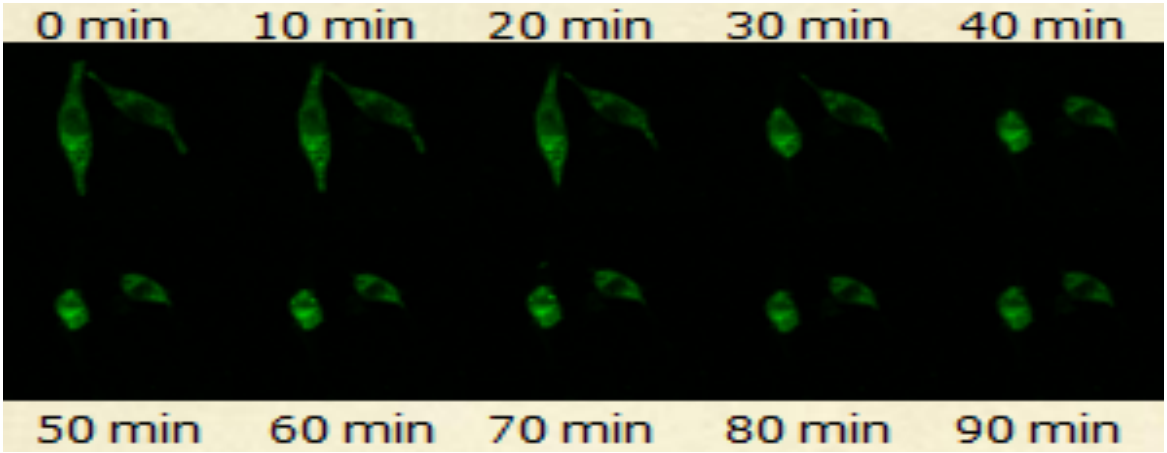


Figure 2.7 Change in MC3T3-E1 Cell Morphology on Exposure to 0.1 µM PGE2

Figure 2.8 is a representative image set of the cell and the beads before the addition of PGE2 (0.05 μM in this case) and figure 2.9 is a representative set after 90 minutes of adding 0.05 μM PGE2 to the system. The images were averaged eight times to improve the signal to noise ratio in the images. The fluorescent laser beams were also shone alternately to ensure that there was no overlap of signals between the red and the green fluorescence.

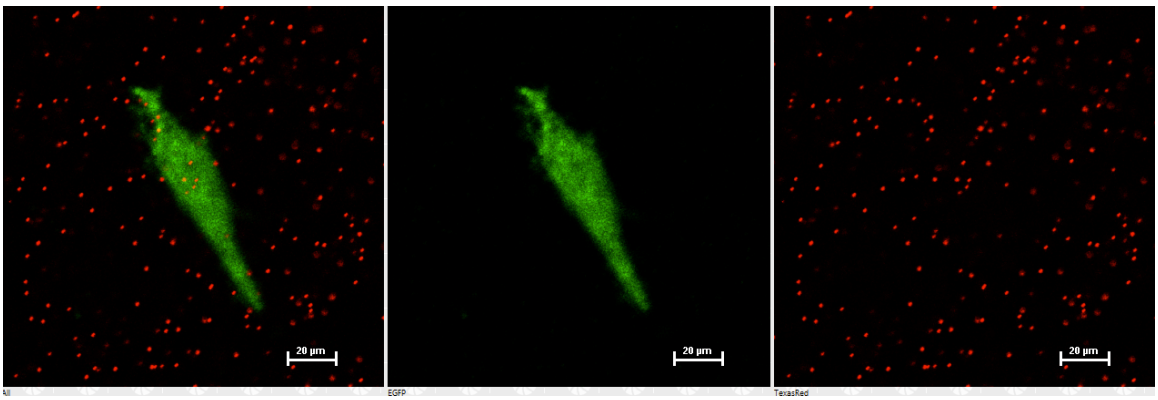


Figure 2.8 Confocal Image of MC3T3 Cell Stained with DiO on p-NIPA Gel Substrate Containing Nile Red Fluorescent Beads.

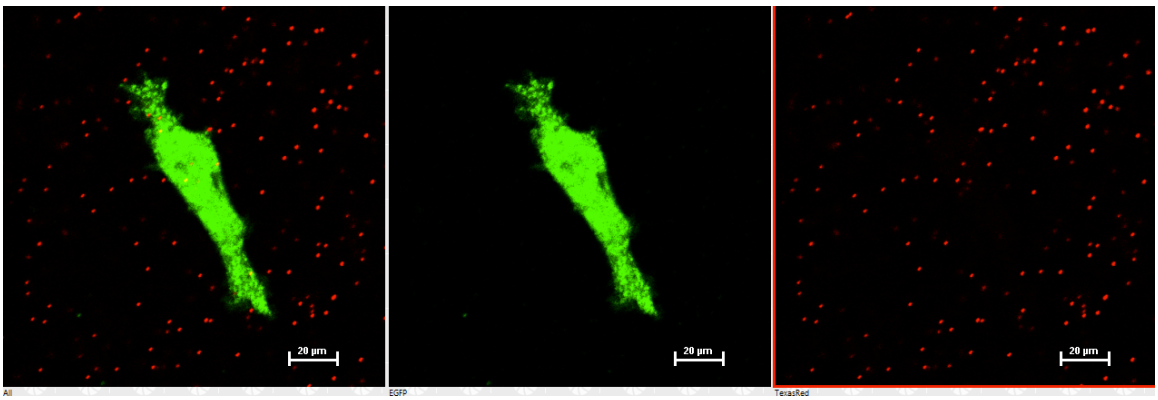


Figure 2.9 Confocal Image of the Cell in figure 2.7 After Exposure to 0.05 μM PGE2 for 90 Minutes.

The image sets were saved in 8-bit nd2 image format, which is a Nikon default image format. ImageJ software (NIH, Bethesda, MD, USA) with the LOCI plug-in was used to open and view the images. For the DIC analysis, the images of the fluorescent beads were converted to 8-bit gray scale tiff (tagged image file format) images. The image of the cell was used to identify the region of interest in the image set. To eliminate any disparity in gray scale intensities, because of lighting changes between the two images, the dynamic range of the gray scale was rendered equal in both the image sets using ImageJ.

2.4 Digital Image Correlation Program

The DIC algorithm for the DIC program is shown in figure 2.10. The Levenberg-Marquardt algorithm was used to iteratively change the values in the parameter vector. The divergence limit is used to check if two successive values of the correlation factor are very close to one another, and, if they are close enough, the optimal result is assumed to have been reached. The oscillation counter is used to ensure that the program does not run endlessly in one region if it does not achieve convergence. At the end of the iterations in a region, if the value of the correlation factor is not below a specified value, it is assumed that the minima was not reached in that region. In these regions, the parameter values were estimated by averaging the parameter values of the nearest neighbors. The values for the divergence limit, oscillation limit, and correlation coefficient were 0.001, 1000, and 0.001 respectively.

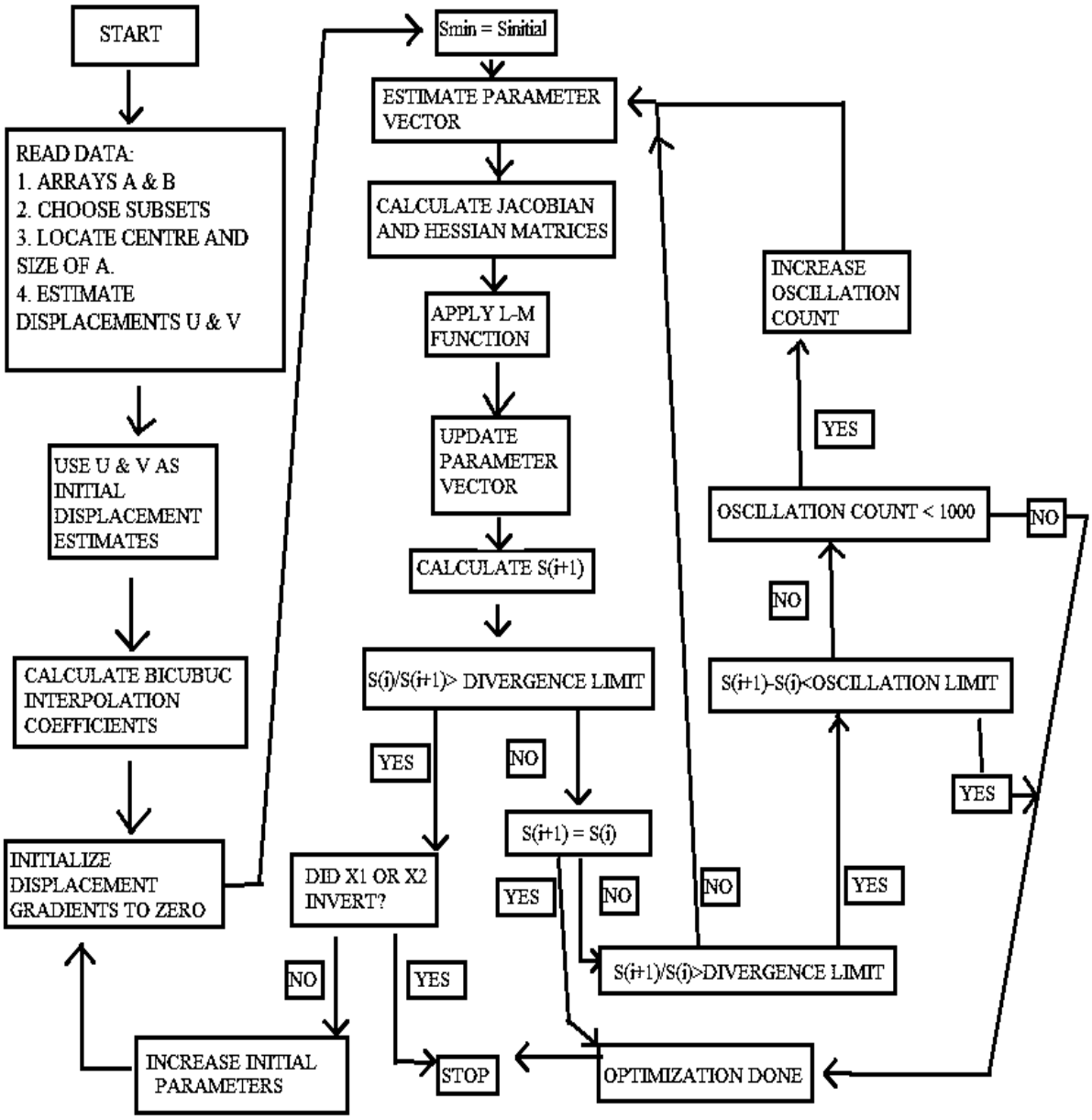


Figure 2.10 Algorithm for DIC Program

The DIC program was validated using a random 500 X 500 array was generated with integer values between 0 and 255 (i.e., the range for 8-bit images). Known displacement and gradient deformations ($u, v, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}$) were applied on a reference image one at a time to create a new image. These images

were compared using the program and the results were analyzed using two-sided Student's t-test with a 99% confidence level to ensure that the analysis of the experimental data was at least applicable at a 95% confidence level. A unit displacement was used to test the program for the parameter u and the results are shown in figure 2.11, where the mean value from the results was 0.999943 and the standard deviation was 0.001080.

Another DIC program was made using the Newton-Raphson algorithm to compare to the performance of the Levenberg-Marquardt algorithm. The DIC program with the Newton-Raphson algorithm was validated using the same array sets as above. The results for the value of u using the Newton-Raphson algorithm are given in figure 2.12. The mean value from the results was 0.999959 and the standard deviation was 0.001147. The run-times for both DIC programs were comparable. The parameter values for the regions, which did not converge, were estimated by an average of the values of their nearest neighbors. Student's t-tests were used to compare the means of the parameters estimated using the two programs at a confidence level of 99% to test if the parameter values obtained were different. The p-values from the tests indicate that the values were not significantly different at 99% confidence level.

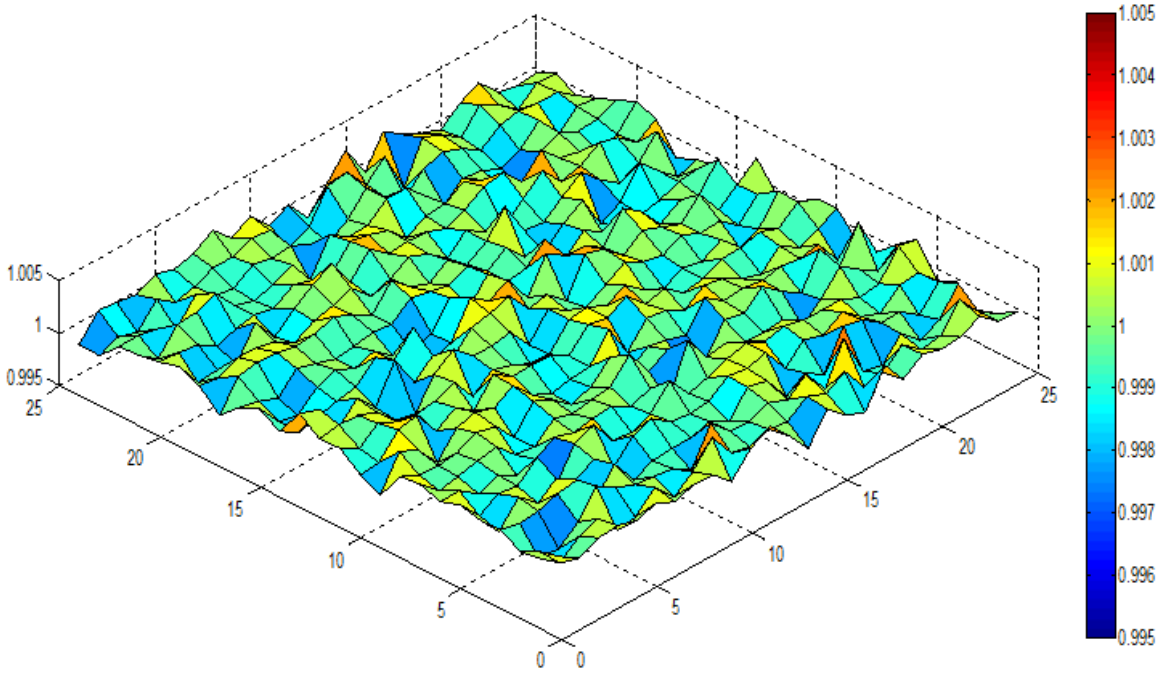


Figure 2.11 Results for Validation of DIC Program Using Levenberg-Marquardt Algorithm with a u Displacement of 1.0

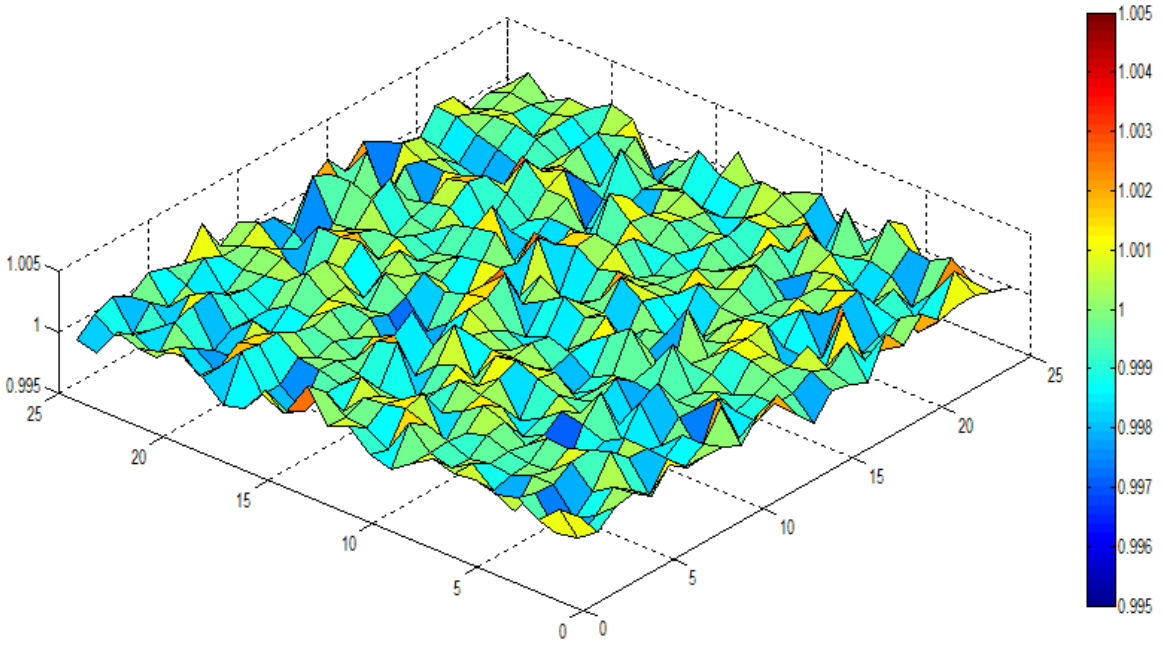


Figure 2.12 Results for Validation of DIC Program Using Newton-Raphson Algorithm with a u Displacement of 1.0

3. Results

The displacements of the fluorescent beads in the gel body is directly proportional to the change in the stresses applied by the cells on the compliant gel substrate before and after the addition of PGE2. Representative samples of the parameters, in terms of pixels, are plotted for a sample each of 0.05 μM and 0.1 μM PGE2 in figures 3.1 through 3.12. Figures 3.13 and 3.14 show the scatter diagrams for the u displacements, and figures 3.15 and 3.16 show the scatter diagrams for the v displacements at different PGE2 concentrations. Figures 3.17 and 3.18 show the histograms for u and v displacements at different PGE2 concentrations. For all analyses, only the magnitudes of the displacement and the gradients were considered. Parameters for regions where the algorithm did not converge were substituted with the arithmetic mean of the parameters of the nearest neighbors. The accuracy of the DIC program is 0.1% of the pixel size. For the statistical analysis, the parameter values were rounded up to three significant decimal figures.

The data shows that the displacement parameters had values above 0.001and the gradients had a values below 0.001. This indicates that there was a change in the location of the beads on addition of PGE2 but no displacement gradients were created. The change in location of the beads is consistent with the change in the cell morphology, which causes a distortion in the substrate and moves the beads. A lack of gradients indicate that there was no change in the

magnitude of the strains in the substrate, which shows that there was no change in the stresses applied by the cells on the substrate on addition of PGE2.

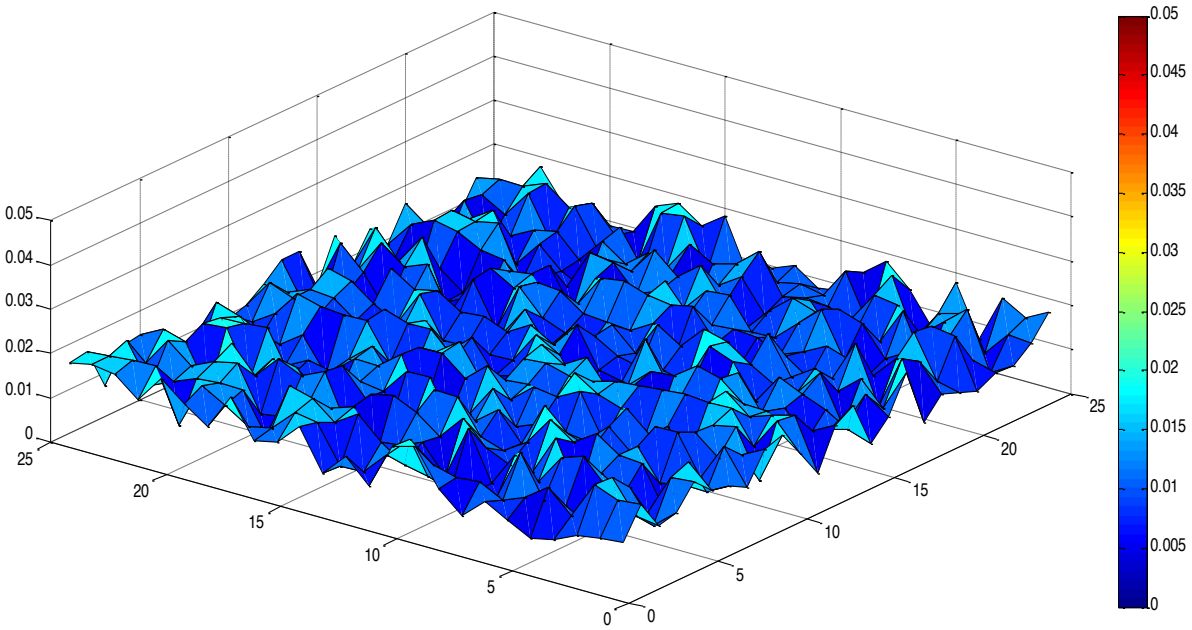


Figure 3.1. Distribution of u for a Sample with 0.05 µM PGE2

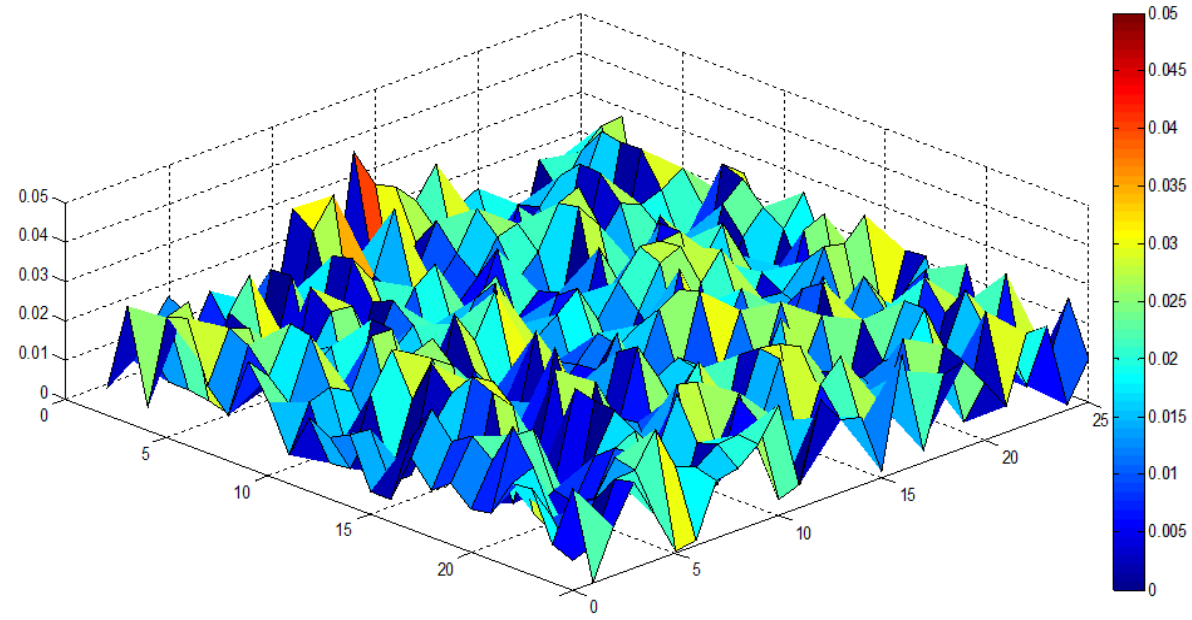


Figure 3.2 Distribution of u for a Sample with 0.10 µM PGE2

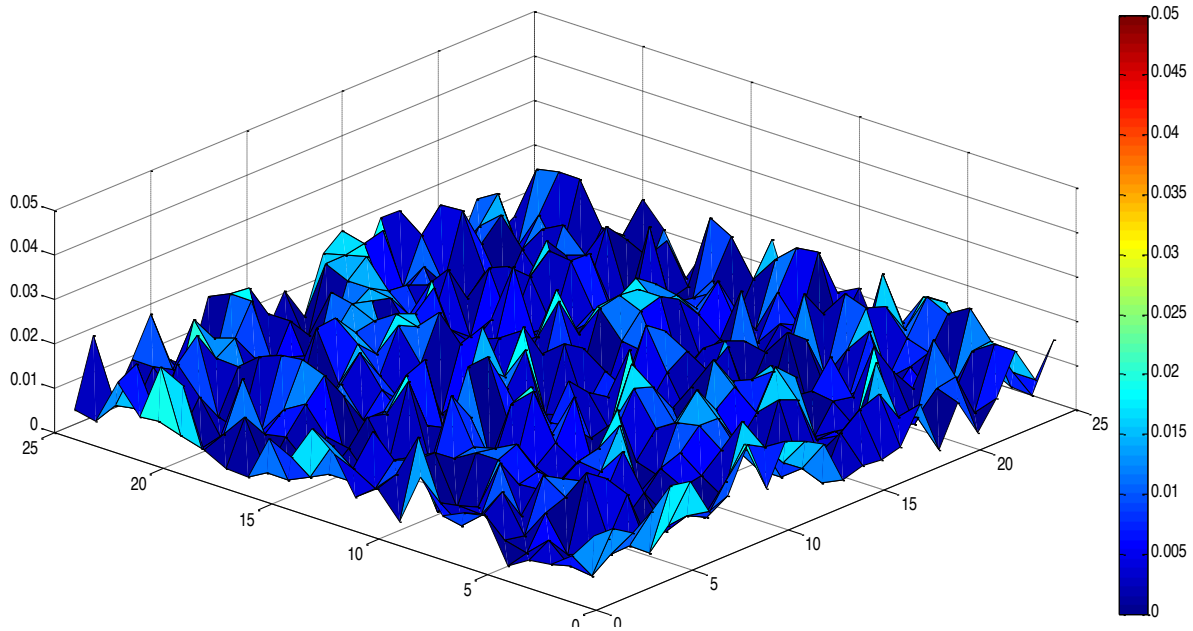


Figure 3.3 Distribution of v for a Sample with 0.05 µM PGE2

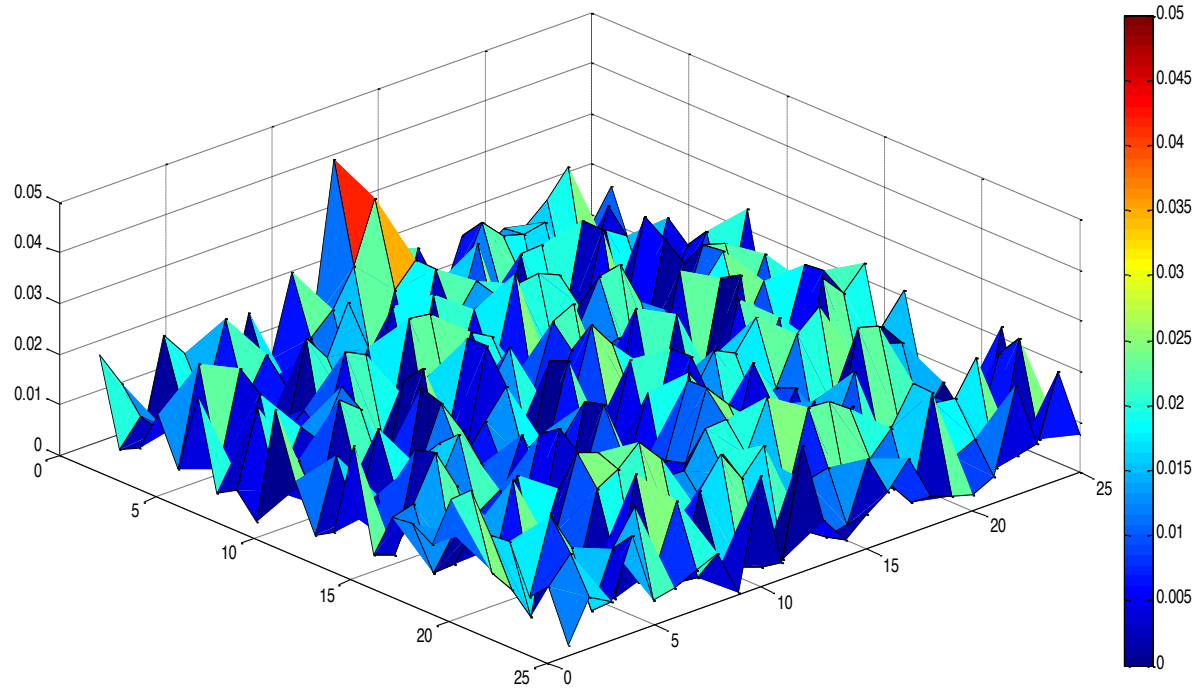


Figure 3.4 Distribution of v for a Sample with 0.10 µM PGE2

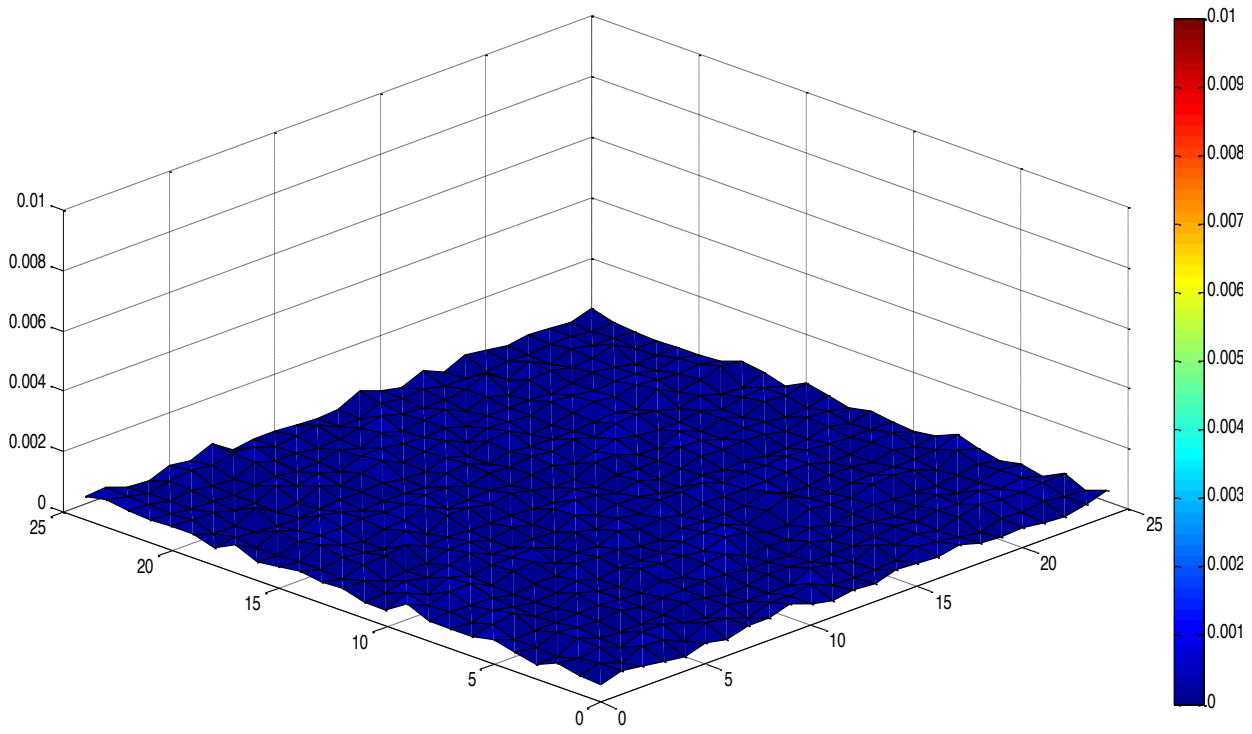


Figure 3.5 Distribution of $\frac{\partial u}{\partial x}$ for a Sample with 0.05 µM PGE2

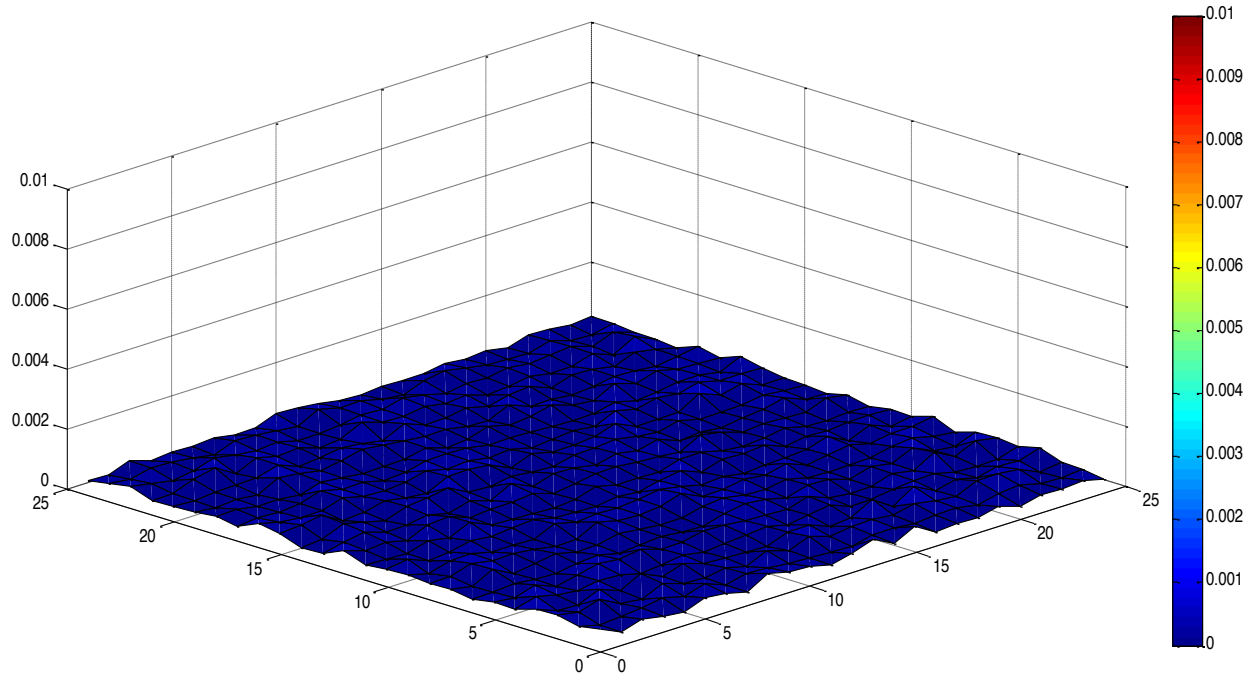


Figure 3.6 Distribution of $\frac{\partial u}{\partial x}$ for a Sample with 0.10 µM PGE2

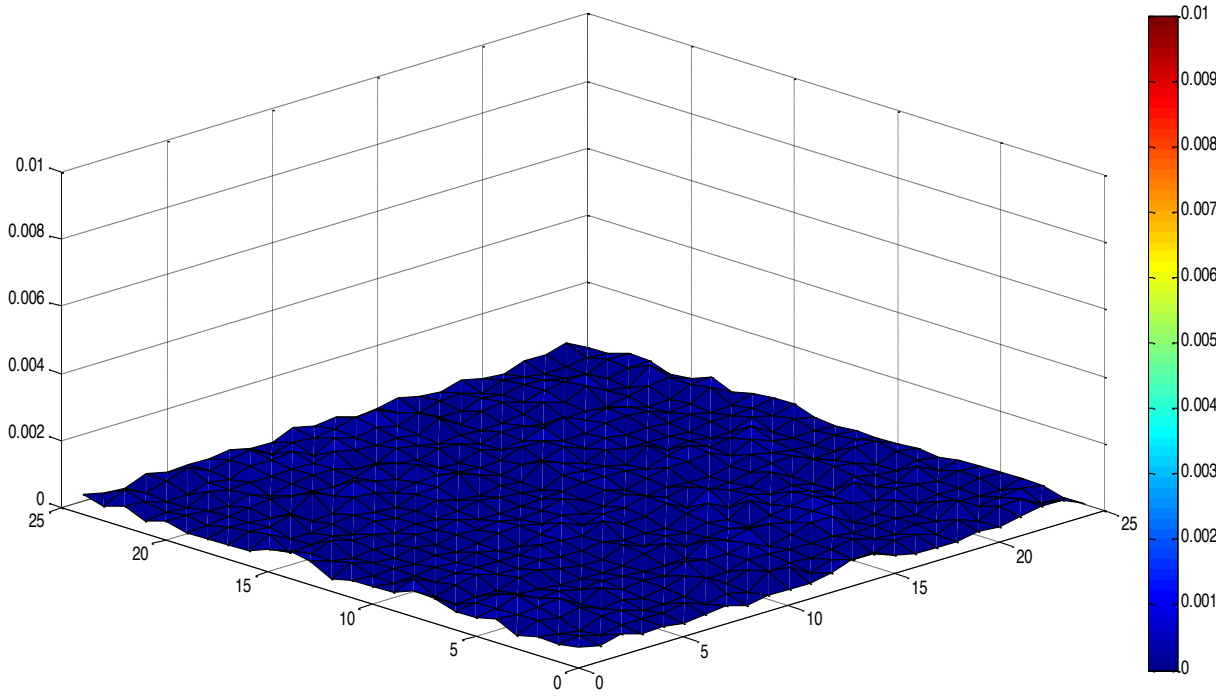


Figure 3.7 Distribution of $\frac{\partial v}{\partial x}$ for a Sample with 0.05 µM PGE2

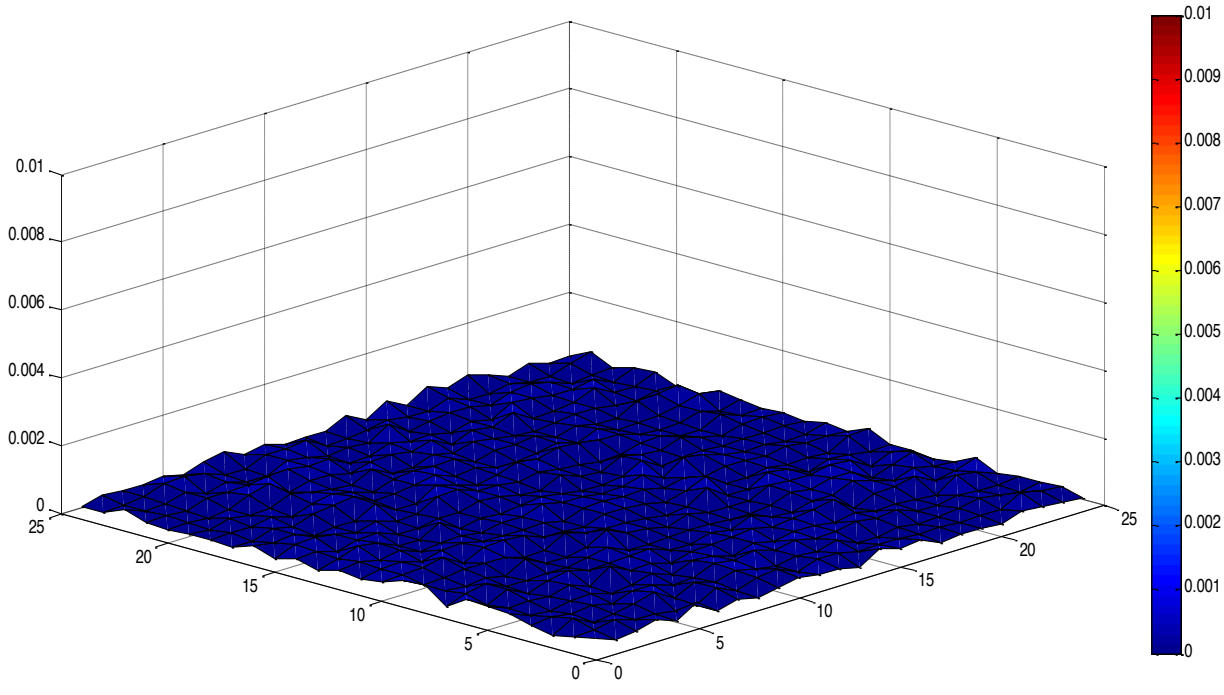


Figure 3.8 Distribution of $\frac{\partial v}{\partial x}$ for a Sample with 0.10 µM PGE2

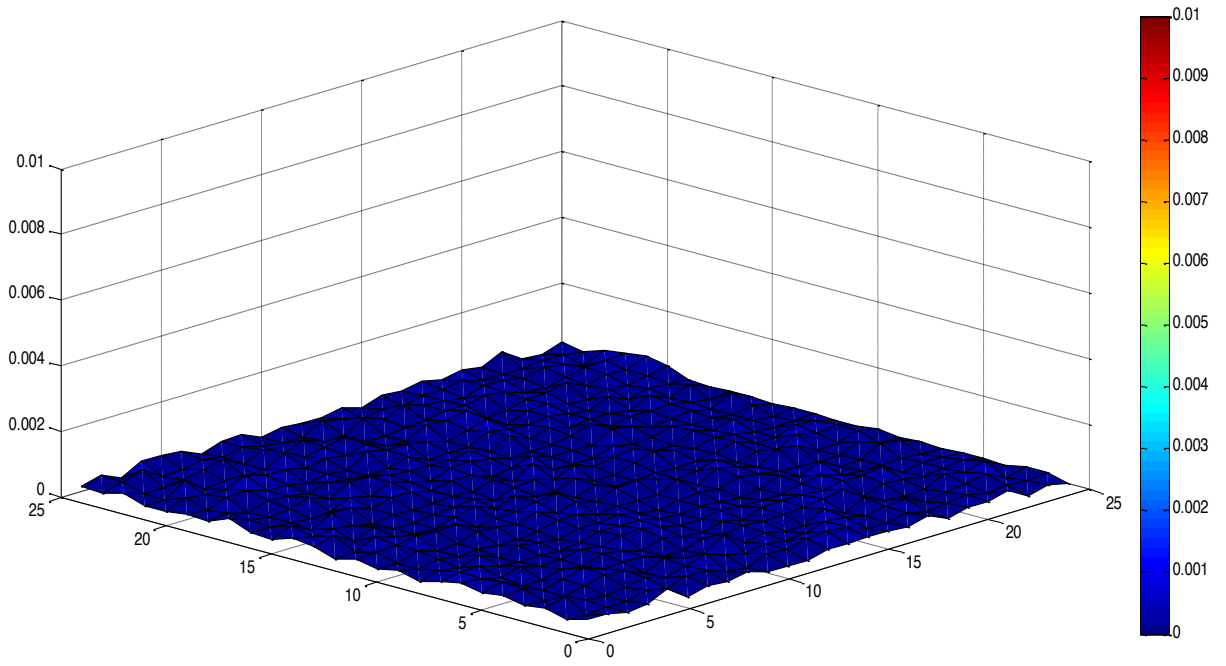


Figure 3.9 Distribution of $\frac{\partial u}{\partial y}$ for a Sample with 0.05 µM PGE2

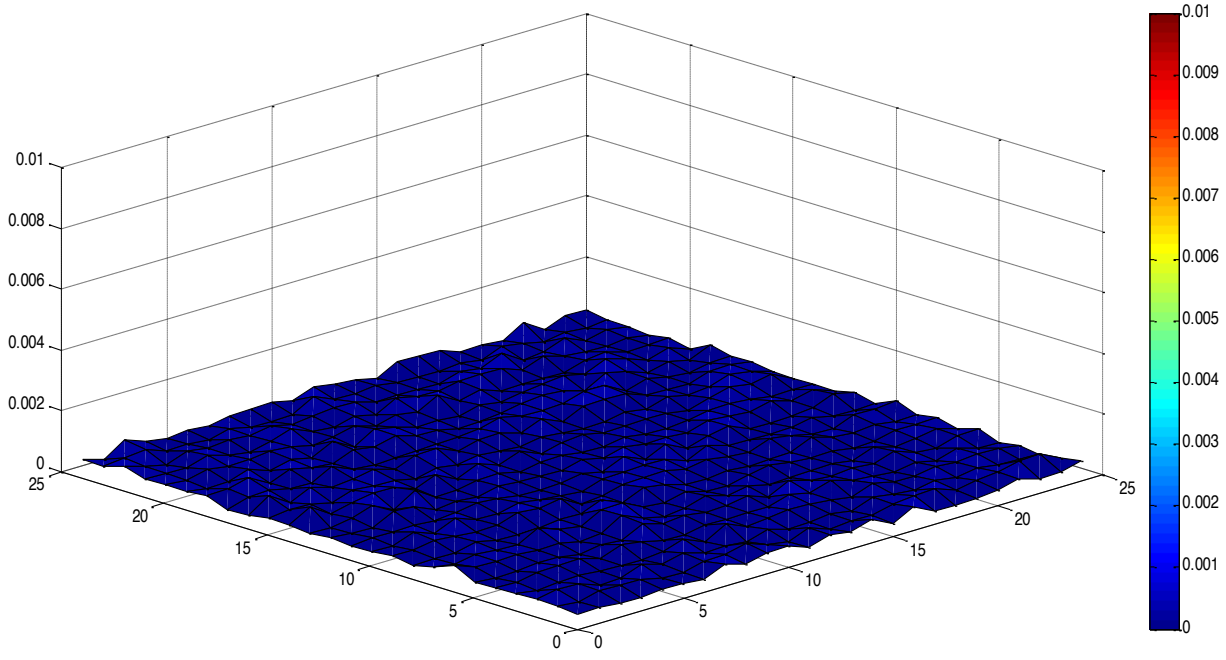


Figure 3.10 Distribution of $\frac{\partial u}{\partial y}$ for a Sample with 0.10 µM PGE2

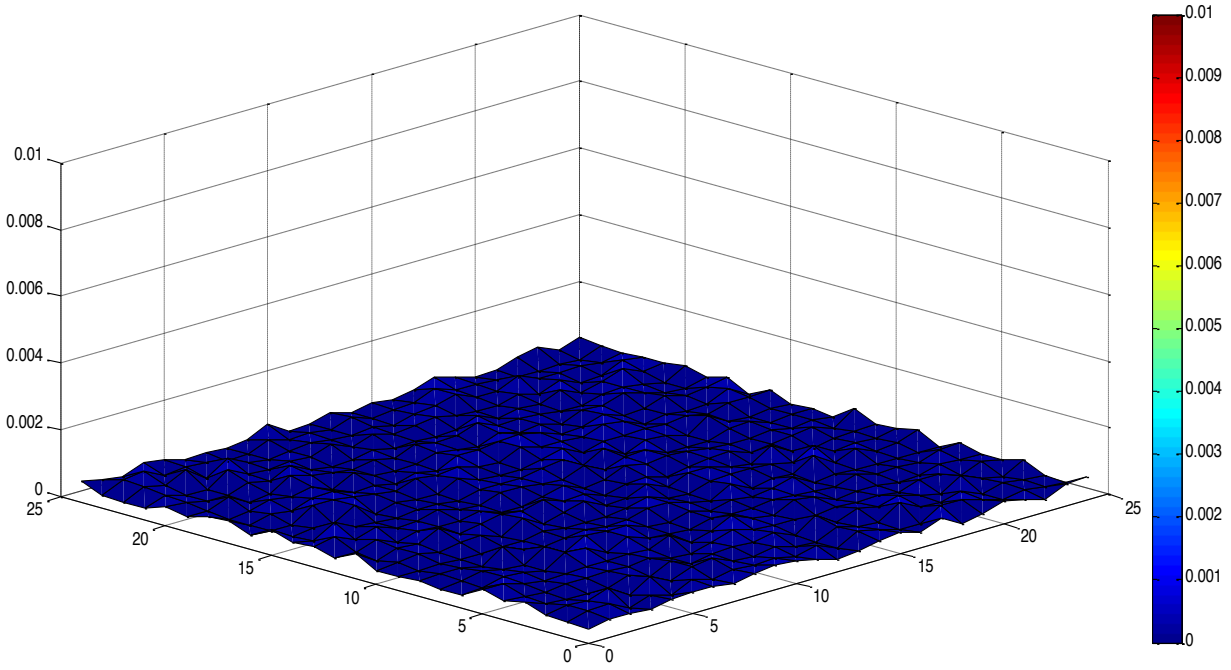


Figure 3.11 Distribution of $\frac{\partial v}{\partial y}$ for a Sample with 0.05 μM PGE2

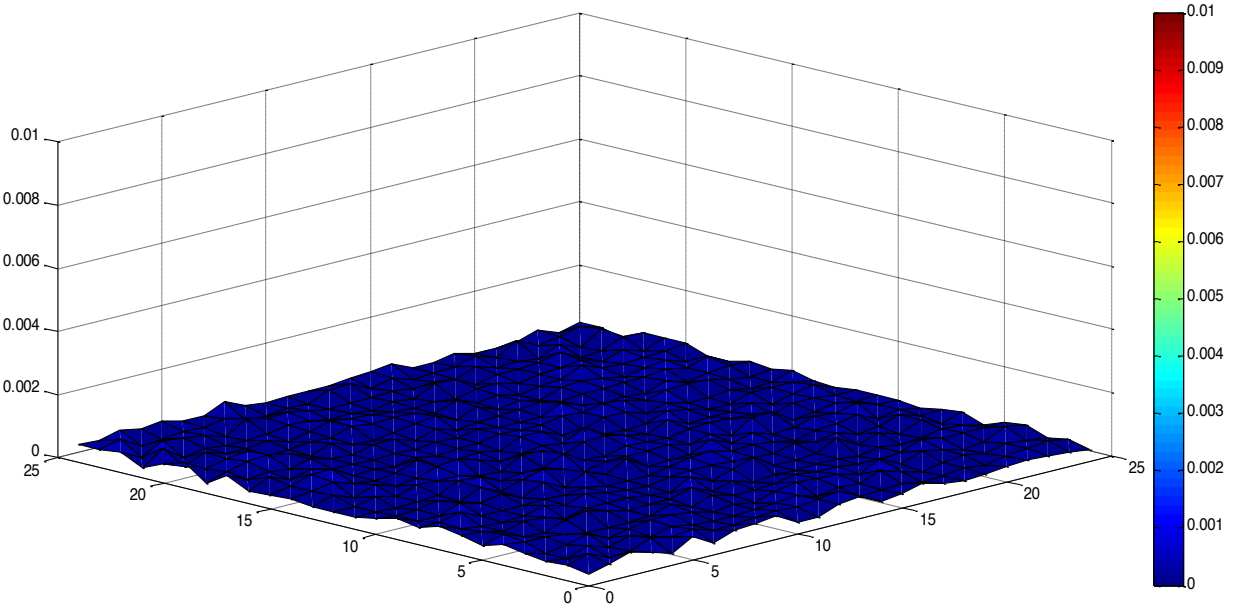


Figure 3.12 Distribution of $\frac{\partial v}{\partial y}$ for a Sample with 0.10 μM PGE2

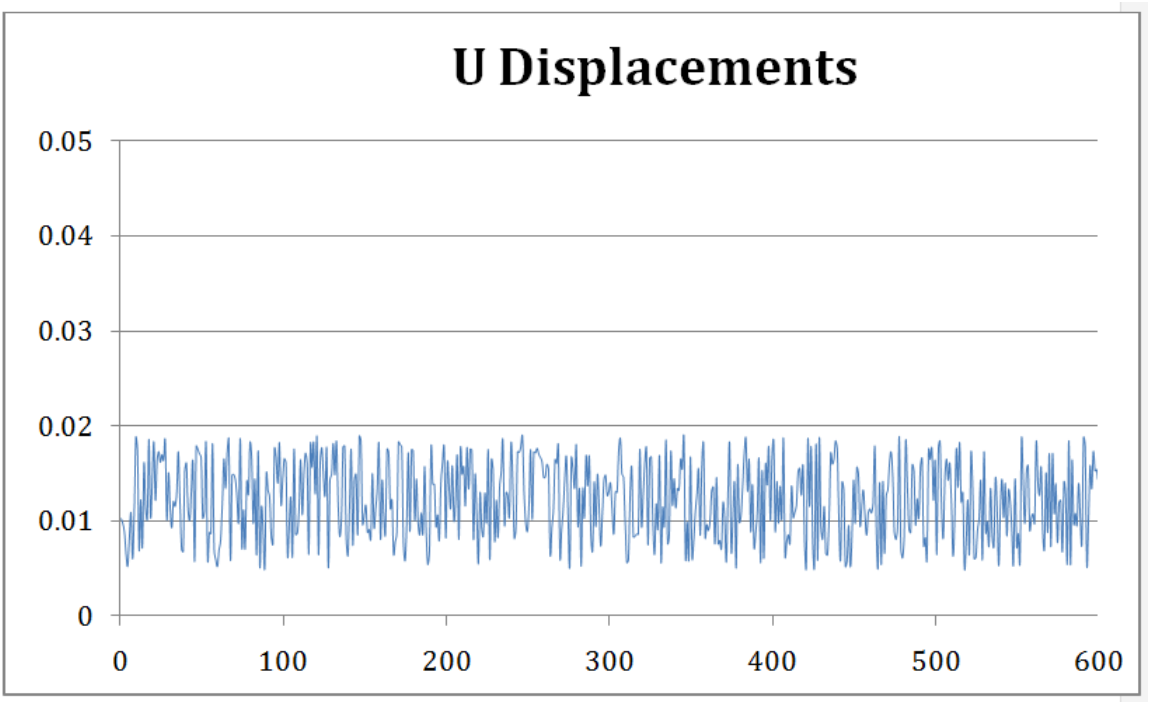


Figure 3.13 Scatter Diagram of u for a Sample with 0.05 μM PGE2

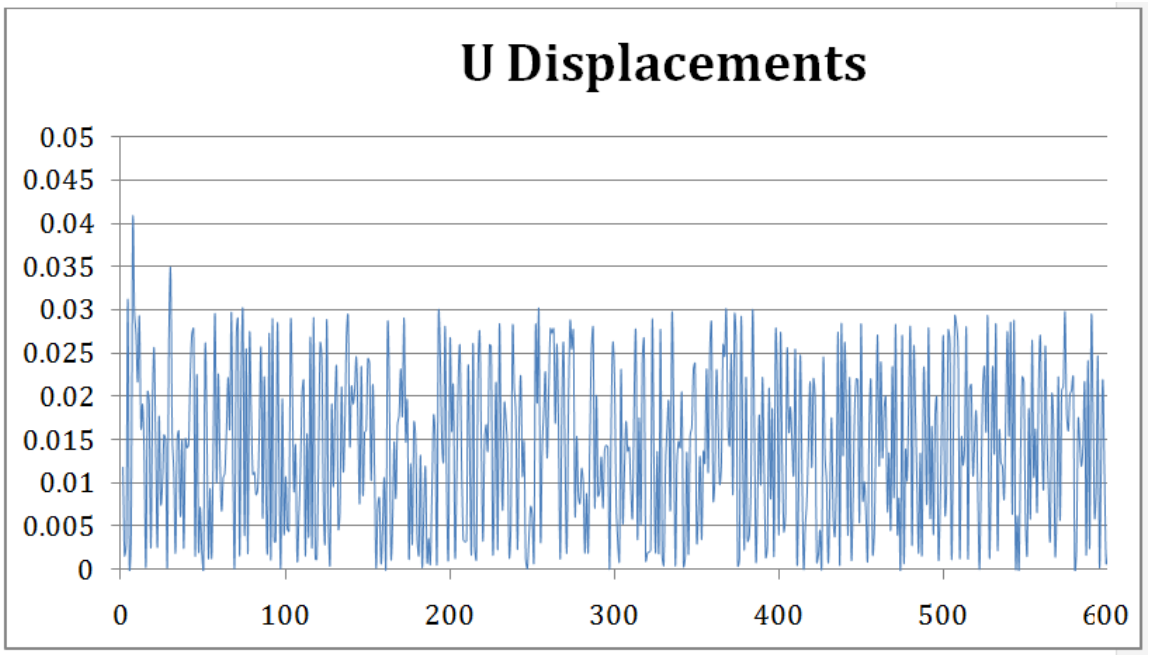


Figure 3.14 Scatter Diagram of u for a Sample with 0.10 μM PGE2

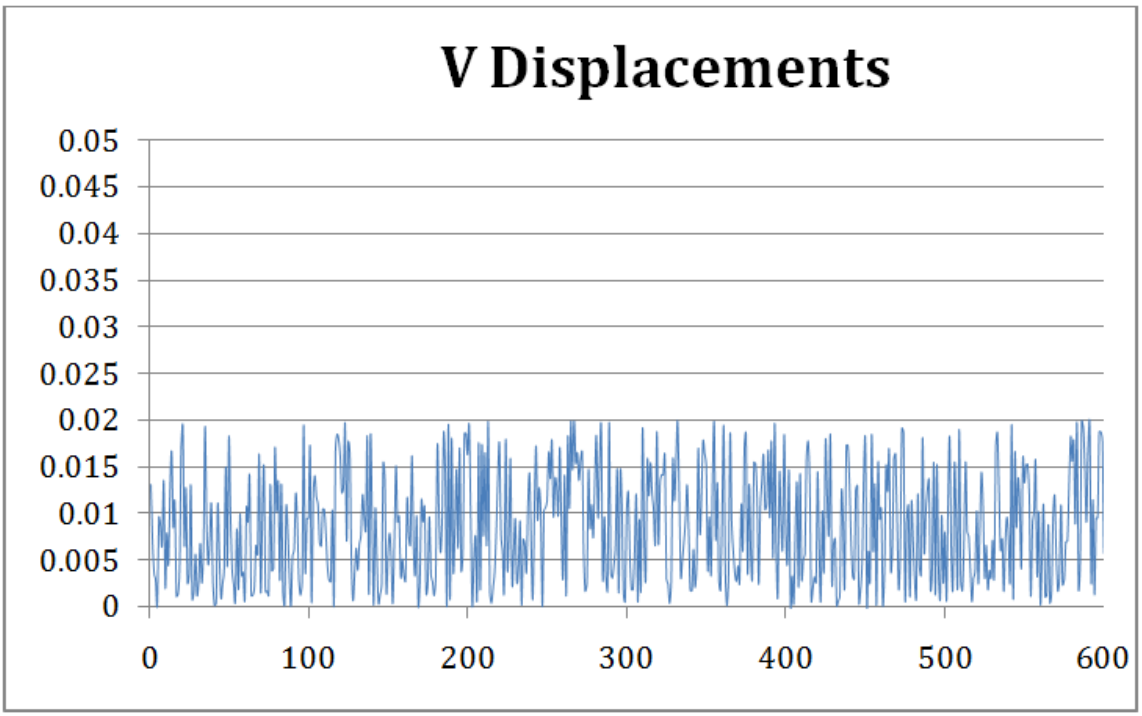


Figure 3.15 Scatter Diagram of v for a Sample with 0.05 µM PGE2

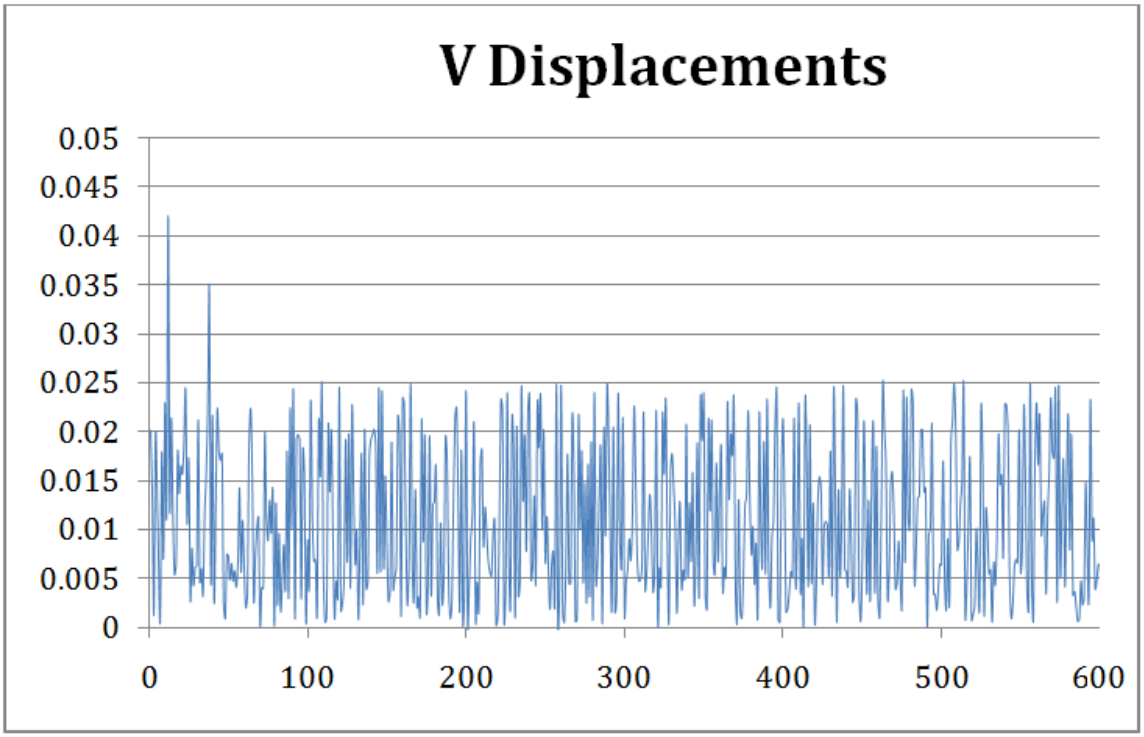


Figure 3.16 Scatter Diagram of v for a Sample with 0.10 µM PGE2

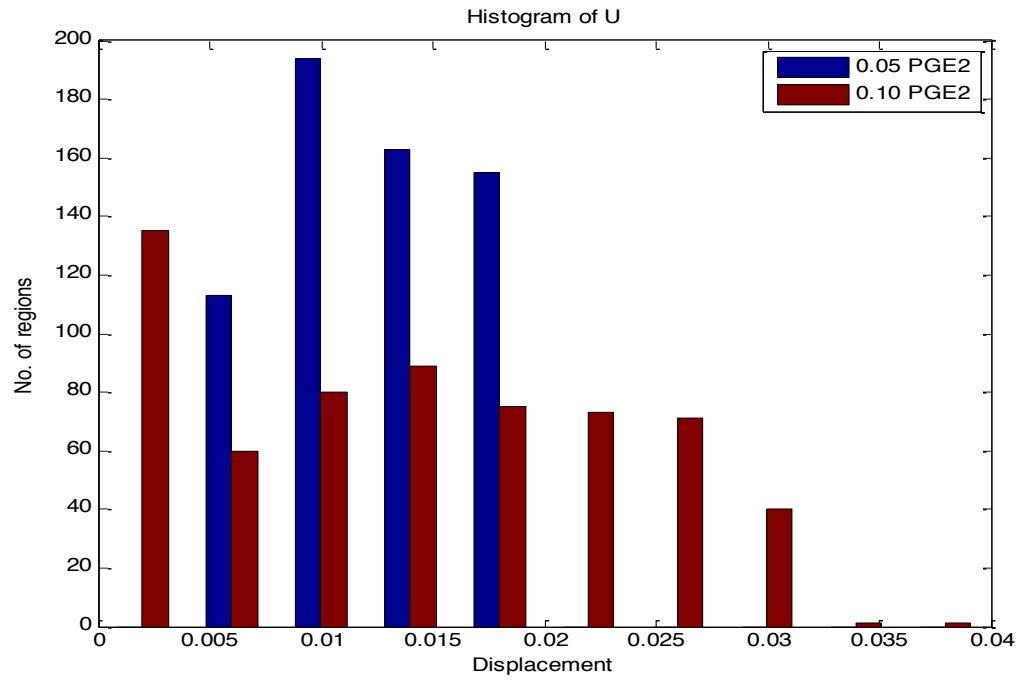


Figure 3.17 Histogram of u for Samples with 0.05 µM PGE2 and 0.10 µM PGE2

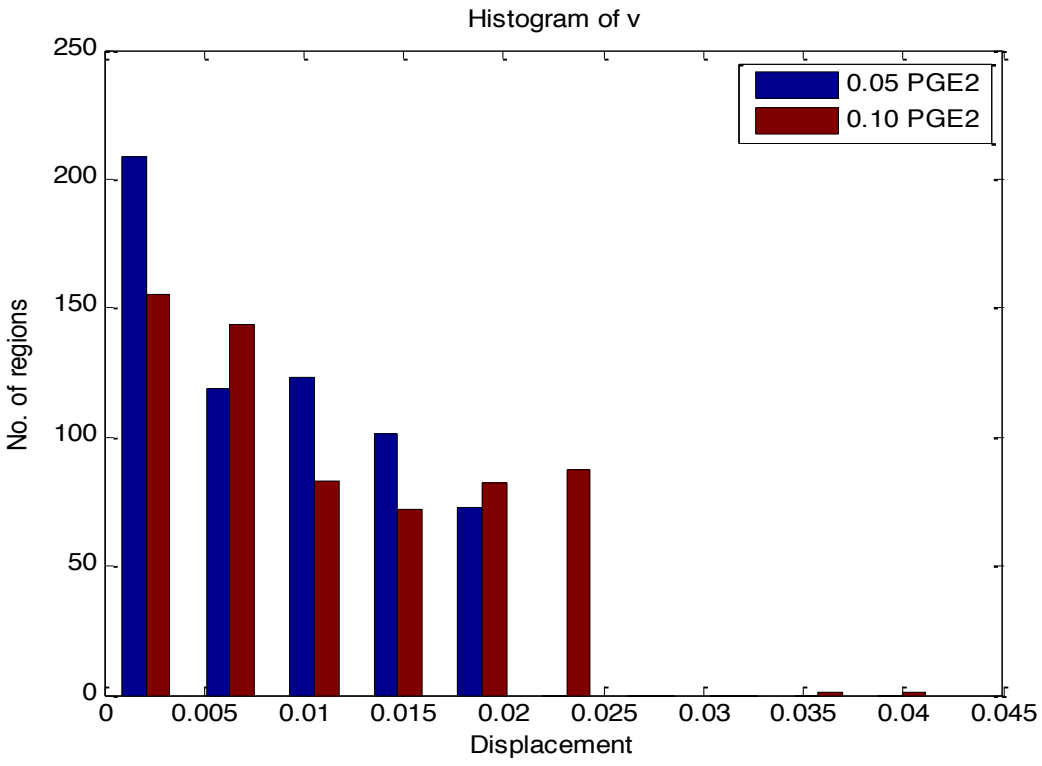


Figure 3.18 Histogram of v for Samples with 0.05 µM PGE2 and 0.10 µM PGE2

Analysis of the scatter diagrams and the histograms for the displacements indicates that there is an increase in the displacement magnitude and variance with an increase in the concentration of PGE2, which merits future investigation.

The Matlab statistical package was used for all statistical analyses of the data. Student's t-tests were performed on the parameters testing the following null hypothesis.

$$H_0: The\ mean\ value\ of individual\ parameters\ \left(u, v, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial v}{\partial x}\ and \frac{\partial v}{\partial y}\right) equals\ zero.$$

Table 3.1 Students t-tests for Parameters at Different PGE2 Concentrations

PGE2	0.05 uM	
Parameter	p-value	null hypothesis
u	>0.025	***rejected***
v	>0.025	***rejected***
du/dx	<0.025	*not rejected*
du/dy	<0.025	*not rejected*
dv/dx	<0.025	*not rejected*
dv/dy	<0.025	*not rejected*

PGE2	0.10 uM	
Parameter	p-value	null hypothesis
u	>0.025	***rejected***
v	>0.025	***rejected***
du/dx	<0.025	*not rejected*
du/dy	<0.025	*not rejected*
dv/dx	<0.025	*not rejected*
dv/dy	<0.025	*not rejected*

4. Discussion

The main objective of this study was to create a tool to measure and compare the strains developed in a soft substrate due to the contractile stresses exerted by cells cultured on the surface and to test, using this tool, if there was a change in the stresses applied by osteoblast-like MC3T3-E1 cells on addition of PGE2. An image-processing program was developed using Levenberg-Marquardt algorithm to compare the strains in the substrate. The results indicate that the addition of PGE2 caused a change in the location of the fluorescent markers but did not change the strain magnitudes in the substrate. Lack of changes in the strains show that the stresses applied by the MC3T3-E1 cells on the substrate did not change on addition of extracellular PGE2. The changes in the locations of the beads (indicated by significant displacement values) show that there were changes in the distribution of the stresses, which is consistent with the changes in the cell morphology on addition of PGE2. The magnitudes and variance of the displacements increase with an increase in the concentration of PGE2.

The results indicate that the MC3T3-E1 cells respond to extracellular PGE2 with a change in the stress-distribution on the substrate. This indicates that the cells communicate with one another mechanically by a distortion of the substrate. Fluid shear stress induces strains in substrates and is known to induce production of PGE2 by osteoblasts (Bakker et al., 2001), where PGE2 has been observed to auto-amplify its production in osteoblasts (Suda et al., 1998). This

creates a positive feedback loop, where an increase in extracellular PGE2 causes a morphological change as well as further increase in the production of PGE2. In this way, the cells mechanically and biochemically communicate with one another when a shear stress is applied on the surface. Cells apply stronger stresses in harder substrates (Dembo et al., 1999) but the distortions caused by the stresses are more prominent in soft substrates (Reinhart-King et al., 2008). Addition of PGE2 did not change the magnitudes of the stresses but had an impact on the distribution of the stresses as well as the surface pattern. MC3T3-E1 cells have been known to be sensitive to surface pattern (Zhao et al., 2006). These observations indicate that the cells are more sensitive to a change in the substrate pattern and the distribution of stresses than to the magnitude of the stresses. This information will be useful in the design of substrates for tissue engineering studies.

Digital image correlation is increasingly gaining popularity in biomedical and bioengineering fields due to an increasing interest in cell mechanics. Because of significant improvements in digital imaging in microscopy and in increased processing speed available through the new generation computers, image analyses can readily be used to study mechanics at the cellular level. Currently, Newton-Raphson algorithm is the standard optimization algorithm used for digital image correlation methods. This is mainly because the Newton-Raphson method does not need to compute second order differential terms, which increases computation speed and with a good initial guess, convergence is

quite rapid. The Levenberg-Marquardt algorithm gives us another optimization method, which is comparable to the Newton-Raphson method in computation speed and has the added advantage of being more tolerant of initial estimates that are far away from the minima. This digital image correlation tool using the Levenberg-Marquardt algorithm provides a useful tool to measure or compare sub-pixel displacements and displacement gradients.

Cell behavior is sensitive to the mechanical properties of the substrate as well as its surface pattern (Zhao et al., 2006; Dembo et al., 1999). Future investigations can include the behavior of these cells when cultured on compliant anisotropic gel surfaces. The results from this study are relevant to MC3T3-E1 cells only. In addition, the tools developed in this study can be used to measure or compare microscopic deformations in other settings as well.

Cells apply stresses to the substrates in three dimensions, but, in this study, the effects of stresses applied perpendicular to the substrate surface were ignored. Cells exist in a three-dimensional environment increasing their contact with the substrate, which could be significant in the response to externally applied PGE2 or to mechanical stimuli. For future studies, a system where the cells are embedded in a compliant three-dimensional substrate should be used to provide information of the system beyond two dimensions.

6. Conclusions

In this study, a digital image correlation image-analysis tool was developed to compare and analyze strains created on a soft compliant p-NIPA hydrogel substrate by MC3T3-E1 osteoblast-like cells in the presence and absence of prostaglandin E2. The Levenberg-Marquardt algorithm was used to optimize the DIC program and was found to perform in a comparable manner to the commonly used Newton-Raphson algorithm. A polymerization chamber was constructed to prepare p-NIPA gel substrates with an engineered horizontal surface to aid good confocal imaging. The magnitudes of the stresses applied by the cells were observed not to vary significantly in the presence of PGE2 at a 95% confidence level. On addition of PGE2 to the cell culture media, the distribution of the strains in the substrate was observed to change in combination with a change in cell morphology. The degree of this distortion of the substrate was observed to change with the concentration of PGE2. The results suggest that the MC3T3-E1 cells communicate mechanically with one another by means of strains developed in the substrate. Finally, the tools developed in this study can be used to compare and analyze microscopic deformations caused in a substrate in other experimental studies.

References:

1. Skerry, T.M. 2008. The response of bone to mechanical loading and disuse: Fundamental principles and influences on osteoblast/osteocyte homeostasis. *Archives of Biochemistry and Biophysics.* 473: 117-123.
2. Bakker, A.D., Soejima, K., Klein-Nulend and J., Burger, E. 2001. The production of nitric oxide and prostaglandin E2 by primary bone cells is shear stress dependent. *Journal of Biomechanics.* 34: 671-677.
3. Forwood, M.R. 1996. Inducible Cyclo-oxygenase (COX-2) mediates the induction of bone formation by mechanical loading in-vivo. *Journal of Bone and Mineral Research.* 11: 1688-1693.
4. Malone, A.M.D., Batra, N.N., Shivaram, G., Kwon, R.Y., You, L., Kim, C.H., Rodriguez, J., Jair, K. and Jacobs, C.R. 2007. The role of actin cytoskeleton in oscillatory fluid flow induced signaling in MC3T3-E1 osteoblasts. *American Journal of Physiology: Cell Physiology.* 292: C1830-C1836.
5. Ponik, S.M. and Pavalko, F.M. 2004. Formation of focal adhesions on fibronectin promotes fluid shear stress induction of COX-2 and PGE2 release in MC3T3-E1 osteoblasts. *Journal of Applied Physiology.* 97: 135-142.
6. Guignandon, A., Vico, L., Alexandre, C. and Lafage-Proust, M. 1995. Shape changes of osteoblastic cells under gravitational variatons during

parabolic flight – Relationship with PGE2 synthesis. *Cell Structure and Function.* 20: 369-375.

7. Hakeda, Y., Nakatani, Y., Hiramatsu, M., Kurihara, N., Tsunoi, M., Ikeda, E. and Kumegawa, M. 1985. Inductive effects of prostaglandin on alkaline phosphatase in osteoblastic cells clone MC3T3-E1. *Journal of Biochemistry.* 97: 97-104.
8. Chen, E.J., Novakofski, J., Jenkins, W.K. and O'Brien, W.D. Jr. 1996. Young's modulus measurements of soft tissues with application of elasticity imaging. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control.* 43: 191-194.
9. Ghayor, C., Rey, A. and Caverzasio, J. 2005. Prostaglandin-dependent activation of ERK mediates cell proliferation induced by transforming growth factor β in mouse osteoblastic cells. *Bone.* 36: 93-100.
10. Shen, V., Rifas, L., Kohler, G. and Peck, W.A. 1986. Prostaglandins change cell shape and increase intercellular gap junctions in osteoblasts cultured from rat fetal calvaria. *Journal of Bone and Mineral Research.* 1: 243-249.
11. Hughes-Fulford, M. and Lewis, M.L. 1996. Effects of microgravity on osteoblast growth activation. *Experimental Cell Research.* 224: 103-109.
12. Zhao, G., Zinger, O., Schwartz, Z., Wieland, M., Landolt, D. and Boyan, B.D. 2006. Osteoblast-like cells are sensitive to submicron-scale surface structure. *Clinical Oral Implants Research.* 17(3): 258-264.

13. Yang, R.S., Fu, W.M., Wang, S.M., Lu, K.S., Liu, T.K. and Lin-Shiau, S.Y. 1998. Morphological changes induced by prostaglandin E in cultured rat osteoblasts. *Bone.* 22(6): 629-636.
14. Kumegawa, M., Ikeda, E., Tanaka, S., Haneji, T., Yora, T., Sakagishi, Y., Minami, N. and Hiramatsu, M. 1984. The Effects of Prostaglandin E2, Parathyroid Hormone, 1,25 Dihydroxycholecalciferol, and Cyclic Nucleotide Analogs on Alkaline Phosphatase Activity in Osteoblastic Cells. *Calcified Tissue International.* 36: 72-76.
15. Igarashi, K., Hirafuji, M., Adachi, H., Shinoda, H. and Mitani, H. 1994. Role of endogenous PGE2 in osteoblastic functions of a clonal osteoblast-like cell, MC3T3-E1. *Prostaglandins Leukotriens and Essential Fatty Acids.* 50: 169-172.
16. Kajii, T., Suzuki, K., Yoshikawa, M., Imai, T., Matsumoto, A. and Nakamara, S. 1999. Long-term effects of prostaglandin E2 on the mineralization of a clonal osteoblastic cell line (MC3T3-E1). *Archives of Oral Biology.* 44: 233-241.
17. Reinhart-King, C.A., Dembo, M. and Hammer, D.A. 2008. Cell-Cell Mechanical Communication through Compliant Substrates. *Biophysical Journal.* 95: 6044-6051.
18. Dembo, M. and Wang, Y. 1999. Stresses at the cell-to-substrate interface during locomotion of fibroblasts. *Biophysical Journal.* 76: 2307-2316.

19. Meazzini, M.C., Toma, C.D., Schaffer, J.L., Gray, M.L. and Gerstenfeld, L.C. 1998. Osteoblast cytoskeletal modulation in response to mechanical strain *in vitro. Journal of Orthopedic Research.* 16: 170-180.

20. Li, C., Hu, Z. and Li, Y. 1993. Poisson's ratio in polymer gels near the phase transition point. *Physical Review E (Brief Reports).* 48(1): 603-606.

21. Pan, B. Xie, H., Wang, Z., Qian, K. and Wang, Z. 2008. Study on subset size selection in digital image correlation for speckle. *Optics Express.* 16(10): 7037-7048.

22. Lo, C., Wang, H., Dembo, M. and Wang, Y. 2000. Cell movement is guided by the rigidity of the substrate. *Biophysical Journal.* 79: 144-152.

23. Hung, P. and Voloshin, A.S. 2003. In-plane strain measurement by digital image correlation. *Journal of the Brazilian Society of Mechanical Sciences and Engineering.* 25(3): 215-221.

24. Zhang, Z., Kang, Y., Wang, H., Qin, Q., Qiu, Y and Li, X. 2006. A novel coarse-fine search scheme for digital image correlation method. *Measurement.* 39: 710-718.

25. Kuendong, H. 2000. A parametric study of displacement measurements using digital image correlation method. *Korean Society of Mechanical Engineering International Journal.* 14(5): 518-529.

26. Schreier, H.W. and Sutton, M.A. 2002. Systematic errors in digital image correlation due to undermatched subset shape functions. *Experimental Mechanics.* 42(3): 303-310.

27. Bruck, H.A., McNeill, S.R., Sutton, M.A. and Peters III, W.H. 1989. Digital image correlation using Newton-Raphson method of partial differential correction. *Experimental Mechanics.* 29(3): 261-267.

28. Lu. H. and Cary, P.D. 2000. Deformation Measurements by Digital Image Correlation: Implementation of a Second-order Displacement Gradient. *Experimental Mechanics.* 40(4): 393-400.

29. Madsen, K., Nielsen, H.B. and Tingleff, O. 2004. Methods for non-linear least square problems. 2nd Edition.

30. Rieger, R., Hambli, R. and Jennane, R. 2011. Modeling of biological doses and mechanical effects on bone transduction. *Journal of Theoritical Biology.* 274: 36-42.

31. Suda, M., Tanaka, K., Yasoda, A., Natsui, K., Sakuma, Y., Tanaka, I., Ushikubi, F., Narumiya, S. and Nakao, K. 1998. Prostaglandin-E2 (PGE2) autoamplifies its production through EP1 subtype of PGE receptor in mouse osteoblastic MC3T3-E1 cells. *Calcified Tissue International.* 62: 327-331.

A. Matlab code

Matlab code for DIC program with Levenberg-Marquardt Algorithm

```
%% DICLM-Digital Image Correlation Levenberg-Marquardt
% Import the undeformed image into Ogl and the deformed image into Dfd.
%
% Inputs needed from the users
%       Location of the Region of Interest- xLength and yLength
%       Image filenames- Ogl and Dfd.
%
% Outputs for user
%       Final Parameters
%       Final Correlation Coefficient
%       Final deformed image this will be stored in b.DfdImage

%% Initialization
clear
clc
format long;

gauss=[0.00000067   0.00002292  0.00019117  0.00038771  0.00019117
0.00002292  0.00000067;
0.00002292  0.00078633  0.00655965  0.01330373  0.00655965  0.00078633
0.00002292;
0.00019117  0.00655965  0.05472157  0.11098164  0.05472157  0.00655965
0.00019117;
0.00038771  0.01330373  0.11098164  0.22508352  0.11098164  0.01330373
0.00038771;
0.00019117  0.00655965  0.05472157  0.11098164  0.05472157  0.00655965
0.00019117;
0.00002292  0.00078633  0.00655965  0.01330373  0.00655965  0.00078633
0.00002292;
0.00000067  0.00002292  0.00019117  0.00038771  0.00019117  0.00002292
0.00000067];

%% Image  Import
% format- name.type (ex. skin.jpg)
Ogl  = double(imread('spray.png'));

%Dfd  = double(imread('Image.tiff'));
Ogl=Ogl(:,:,1);
Ogl  = conv2(gauss,Ogl);
```

```
Dfd=imresize(Ogl,[234 238]);
%% Choose a box which surrounds an inner region of interest.
yLength=100:150;
xLength=100:150;

%% Begin Program
% Preallocation of the result array
Result = zeros(6,1);
tic

%% Coarse Search
% This is used to find large pixel offsets.
Offset = CoarseSearch1(Ogl, Dfd);
xOffset = Offset(1);
yOffset = Offset(2);

P = [xOffset;yOffset;0;0;0;0]; % the parameter vector [U;V,Ux;Uy;Vx;Vy]

%% Levenberg-Marquardt Iterations
b=LMiter(Ogl,Dfd,P, yLength, xLength);

%% Output Information
disp('The Final Parameters are: ');
disp(b.finalParameters);
disp('The Final Correlation Coefficient is: ');
disp(b.finalCoeff);
toc


function b = CoarseSearch1(OriginalImage,DistortedImage)

% This function performs a coarse search to find the displacement of the
% central position of OriginalImage in the DistortedImage within the
% boundaries chosen by the segments SegmentOriginal and SegmentDistorted

%do normalized cross-correlation and plot the array
c=normxcorr2(OriginalImage,DistortedImage);
%figure, surf(c), shading flat;

% offset found by correlation
[MaxCorrelation, IndexMax] = max(c(:));
[yPeak, xPeak] = ind2sub(size(c),IndexMax(1));
CorrelationOffset = [(xPeak-size(OriginalImage,2))
            (yPeak-size(OriginalImage,1))];
```

```
% relative offset of position of subimages
%RelativeOffset = [(SegmentDistorted(2)-SegmentOriginal(2));
%             (SegmentDistorted(1)-SegmentOriginal(1))];

b=CorrelationOffset;


function c = crosscorr(a,b)

%This function gives the crosscorrelation function between a and b where in
%c=[1-sum(a[:]*b[:]/[{sum(a[:]^2}*{sum(b[:]^2}]^0.5]. The range of this
%function is (0,1). Zero indicates no correlation and one indicates total
%correlation.
num = sum(sum(a.*b));
denom = (sum(sum(realpow(a,2)))*sum(sum(realpow(b,2))))^0.5;
c = 1 - (num/denom);


function b=crossCorrCoeffLM(Ogl,Dfd,dGdu,dGdv, d2Gdu2, d2Gdv2,
d2Gdudv,delX, delY, parameter, xOgl, yOgl, bcs, pp2dir, pp3dir,lambda,updateJ,
J, H1, delSign)

% This function takes two arrays (original image and deformed image, and
% the spline coefficients of the deformed image as input, runs a
% Levenberg-Marquardt on the two images to find the cross-correlation
% coefficient.

% Calculate the Jacobian and the Hessian
if updateJ
package = jacobianHessian(Ogl, Dfd, dGdu, dGdv, d2Gdu2, d2Gdv2, d2Gdudv,
delX, delY);

% the Jacobian
J      = package.jacobian;

% the Hessian
H1       = package.hessian;
end
% Calculate delP (use the backslash to operate the inverse)
H=H1+lambda*eye(size(H1,1));

delP    = delSign*H\J';
```

```
% Update the values in the parameter vector.
P       = parameter+delP;

% Obtain new X and Y coordinates in the deformed image.
xDfd    = xOgl+P(1)+P(3)*delX+P(4)*delY;
yDfd    = yOgl+P(2)+P(5)*delX+P(6)*delY;

vect(1,:) = reshape(yDfd,1,[]);
vect(2,:) = reshape(xDfd,1,[]);
% evaluate the deformed image at the new found points as well as the first
% and second derivatives.

newDfd =fnval(bcs, vect);
bcsFirstDeriv =fnval(pp2dir, vect);
bcsSecondDeriv =fnval(pp3dir, vect);

% Sort evaluated images into their correct variables.
newDfd       = reshape(newDfd,size(Ogl,1),size(Ogl,2));
dGdv(:,:)    = reshape(bcsFirstDeriv(1,:,:)*5,size(Ogl,1),size(Ogl,2));
dGdu(:,:)    = reshape(bcsFirstDeriv(2,:,:)*5,size(Ogl,1),size(Ogl,2));
d2Gdv2(:,:)  = reshape(bcsSecondDeriv(1,:,:)*5,size(Ogl,1),size(Ogl,2));
d2Gdudv(:,:) = reshape(bcsSecondDeriv(2,:,:)*5,size(Ogl,1),size(Ogl,2));
d2Gdu2(:,:)  = reshape(bcsSecondDeriv(4,:,:)*5,size(Ogl,1),size(Ogl,2));

% Calculate new correlation coefficient.
S = crosscorr(newDfd,Ogl);
disp(S);
b=struct('coeff',S,'parameter',P,'delP',delP,'newDfd', newDfd, 'dGdu', dGdu,
'dGdv',dGdv,'d2Gdu2', d2Gdu2,'d2Gdv2', d2Gdv2, 'd2Gdudv', d2Gdudv, 'J', J, 'H',
H);


function b=createHessian(Ogl, Dfd,
dGdu,dGdv,d2Gdu2,d2Gdv2,d2Gdudv,delX,delY)

sizeOgl=size(Ogl,1);

%du's
dudu=secondDerivativeS(Ogl, Dfd, d2Gdu2, dGdu, dGdu, ones(sizeOgl),
ones(sizeOgl));
dudv=secondDerivativeS(Ogl, Dfd, d2Gdudv, dGdu, dGdv, ones(sizeOgl),
ones(sizeOgl));
dududx=secondDerivativeS(Ogl, Dfd, d2Gdu2, dGdu, dGdu, ones(sizeOgl),
delX);
```

```
dududy=secondDerivativeS(Ogl, Dfd, d2Gdu2, dGdu, dGdu, ones(sizeOgl),
delY);
dudvdx=secondDerivativeS(Ogl, Dfd, d2Gdudv, dGdu, dGdv, ones(sizeOgl),
delX);
dudvdy=secondDerivativeS(Ogl, Dfd, d2Gdudv, dGdu, dGdv, ones(sizeOgl),
delY);

%dv's
dvdu=dudv;
dvdv=secondDerivativeS(Ogl, Dfd, d2Gdv2, dGdv, dGdv, ones(sizeOgl),
ones(sizeOgl));
dvdudx=secondDerivativeS(Ogl, Dfd, d2Gdudv, dGdv, dGdu, ones(sizeOgl),
delX);
dvdudy=secondDerivativeS(Ogl, Dfd, d2Gdudv, dGdv, dGdu, ones(sizeOgl),
delY);
dvdvdx=secondDerivativeS(Ogl, Dfd, d2Gdv2, dGdv, dGdv, ones(sizeOgl), delX);
dvdvdy=secondDerivativeS(Ogl, Dfd, d2Gdv2, dGdv, dGdv, ones(sizeOgl), delY);

%dudx's
dudxdu=dududx;
dudxdv=dvdudx;
dudxdudx=secondDerivativeS(Ogl, Dfd, d2Gdu2, dGdu, dGdu, delX, delX);
dudxdudy=secondDerivativeS(Ogl, Dfd, d2Gdu2, dGdu, dGdu, delX, delY);
dudxdvdx=secondDerivativeS(Ogl, Dfd, d2Gdudv, dGdu, dGdv, delX, delX);
dudxdvdy=secondDerivativeS(Ogl, Dfd, d2Gdudv, dGdu, dGdv, delX, delY);

%dudy's
dudydu=dududy;
dudydv=dvdudy;
dudydudx=dudxdudy;
dudydudy=secondDerivativeS(Ogl, Dfd, d2Gdu2, dGdu, dGdu, delY, delY);
dudydvdx=secondDerivativeS(Ogl, Dfd, d2Gdudv, dGdu, dGdv, delY, delX);
dudydvdy=secondDerivativeS(Ogl, Dfd, d2Gdudv, dGdu, dGdv, delY, delY);

%dvdx's
dvdxdu=dudvdx;
dvdxdv=dvdvdx;
dvdxdudx=dudxdvdx;
dvdxdudy=dudydvdx;
dvdxdvdx=secondDerivativeS(Ogl, Dfd, d2Gdv2, dGdv, dGdv, delX, delX);
dvdxdvdy=secondDerivativeS(Ogl, Dfd, d2Gdv2, dGdv, dGdv, delX, delY);

%dvdy's
dvdydu=dudvdy;
```

```
dvdydv=dvdvdy;
dvdydudx=dudxdvdy;
dvdydudy=dudydvdy;
dvdydvdx=dvdxdvdy;
dvdydvdy=secondDerivativeS(Ogl, Dfd, d2Gdv2, dGdv, dGdv, delY, delY);

b=[dudu dudv dududx dududy dudvdx dudvdy...
   ;dvdu dvdv dvdudx dvdudy dvdvdx dvdvdy...
   ;dudxdu dudxdv dudxdudx dudxdudy dudxdvdx dudxdvdy...
   ;dudydu dudydv dudydudx dudydudy dudydvdx dudydvdy...
   ;dvdxdu dvdxdv dvdxdudx dvdxdudy dvdxdvdx dvdxdvdy...
   ;dvdydu dvdydv dvdydudx dvdydudy dvdydvdx dvdydvdy];


function b=jacobianHessian(Ogl, Dfd, dGdu, dGdv, d2Gdu2,d2Gdv2, d2Gdudv,
delX, delY)

dSdu=-(sum(sum(Ogl.*dGdu))/((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^.5))+...
    ((sum(sum(Ogl.*Dfd))*sum(sum(Ogl.^2))*sum(sum(Dfd.*dGdu)))/...
     ((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^1.5));

dSdv=-(sum(sum(Ogl.*dGdv))/((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^.5))+...
    ((sum(sum(Ogl.*Dfd))*sum(sum(Ogl.^2))*sum(sum(Dfd.*dGdv)))/...
     ((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^1.5));

dSdudx=-
(sum(sum(Ogl.*(dGdu.*delX)))/((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^.5))+...
    ((sum(sum(Ogl.*Dfd))*sum(sum(Ogl.^2))*sum(sum(Dfd.*(dGdu.*delX))))/...
     ((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^1.5));

dSdudy=-
(sum(sum(Ogl.*(dGdu.*delY)))/((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^.5))+...
    ((sum(sum(Ogl.*Dfd))*sum(sum(Ogl.^2))*sum(sum(Dfd.*(dGdu.*delY))))/...
     ((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^1.5));

dSdvdx=-
(sum(sum(Ogl.*(dGdv.*delX)))/((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^.5))+...
    ((sum(sum(Ogl.*Dfd))*sum(sum(Ogl.^2))*sum(sum(Dfd.*(dGdv.*delX))))/...
     ((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^1.5));

dSdvdy=-
(sum(sum(Ogl.*(dGdv.*delY)))/((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^.5))+...
    ((sum(sum(Ogl.*Dfd))*sum(sum(Ogl.^2))*sum(sum(Dfd.*(dGdv.*delY))))/...
     ((sum(sum(Ogl.^2))*sum(sum(Dfd.^2)))^1.5));
```

```
hessian=createHessian(Ogl,Dfd,dGdu,dGdv,d2Gdu2,d2Gdv2,d2Gdudv,delX,delY
);

jacobian(1)=dSdu;
jacobian(2)=dSdv;
jacobian(3)=dSdudx;
jacobian(4)=dSdudy;
jacobian(5)=dSdvdx;
jacobian(6)=dSdvdy;

b=struct('jacobian', jacobian, 'hessian', hessian);



function b=LMiter(firstRegIntOgl,RegIntDfd, prevP, yLength, xLength)

% These values are used to select a smaller portion inside of the first
% region of interest.
regionOgl(:,2) = yLength;
regionOgl(:,1) = xLength;

% Obtain The inner region of interest for the original image, this image
% will never change.
Ogl     = firstRegIntOgl(regionOgl(:,2),regionOgl(:,1));

% Obtain list of points which are locations of X and Y in the original image.
xOgl    = min(min(regionOgl(:,1))):max(max(regionOgl(:,1)));
yOgl    = min(min(regionOgl(:,2))):max(max(regionOgl(:,2)));

% create a meshgrid of points.
[xx,yy] = meshgrid(xOgl, yOgl);

% create deltaX and deltaY which is the distance from the center of the
% image
deltaX  =(-floor(size(Ogl,1)/2):floor(size(Ogl,1)/2));
deltaY  =(-floor(size(Ogl,2)/2):floor(size(Ogl,2)/2));

% Create vectors containing distances from center of original image to be
% used in calculation of jacobian and hessian matrices.
delX    = repmat(deltaX,size(Ogl,1),1);
delY    = repmat(deltaY',1,size(Ogl,2));

% Use csapi to obtain structure to which be differentiated to obtain all dG
```

```
% images.
bcs     = csapi({1:size(RegIntDfd,1),1:size(RegIntDfd,2)}, RegIntDfd);

% Obtain derivatives of G. pp2Dir=first derivative pp3Dir=second derivative
pp2dir  = fndir(bcs,[1 0;0 1]);
pp3dir  = fndir(fndir(bcs,[1 0;0 1]),[1 0;0 1]);

% Calculate X and Y locations in the deformed image from locations from
% original image and parameter vector.
xDfd    = xx+prevP(1)+prevP(3)*delX+prevP(4)*delY;
yDfd    = yy+prevP(2)+prevP(5)*delX+prevP(6)*delY;

% Reshapes the points into vectors (faster).
vect(1,:) = reshape(yDfd,1,[]);
vect(2,:) = reshape(xDfd,1,[]);

% Evaluate structures obtained from csapi at values from xDfd and yDfd
% to obtain deformed image and derivatives.
Dfd           = fnval(bcs, vect);
bcsFirstDeriv  = fnval(pp2dir, vect);
bcsSecondDeriv = fnval(pp3dir, vect);

% Sort derivatives into their variables from evaluated functions and
% reshape them back into 2D images.
prevDfd         = reshape(Dfd,size(Ogl,1),size(Ogl,2));
prevdGdv(:,:)    = reshape(bcsFirstDeriv(1,:,:),size(Ogl,1),size(Ogl,2));
prevdGdu(:,:)    = reshape(bcsFirstDeriv(2,:,:),size(Ogl,1),size(Ogl,2));
prevd2Gdv2(:,:)  = reshape(bcsSecondDeriv(1,:,:),size(Ogl,1),size(Ogl,2));
prevd2Gdudv(:,:) = reshape(bcsSecondDeriv(2,:,:),size(Ogl,1),size(Ogl,2));
prevd2Gdu2(:,:)  = reshape(bcsSecondDeriv(4,:,:),size(Ogl,1),size(Ogl,2));

%Calculate the correlation value for the initial guess
prevS    = crosscorr(prevDfd,Ogl);
lambda   = 0.01;
updateJ  = 1;
H        = zeros(6);
J        = zeros(1,6);
delSign  = -1;

% Calculate jacobian and hessian, increment parameters, obtain new set of
% points to calculate new deformed image, and obtain new correlation coeff.
package = crossCorrCoeffLM(Ogl, prevDfd, prevdGdu, prevdGdv, prevd2Gdu2,
prevd2Gdv2, prevd2Gdudv, delX, delY, prevP,xx,yy, bcs, pp2dir, pp3dir,
lambda,updateJ, J, H, delSign);
```

```
% Set limits for levenberg-marquardt.
diffCoeff = 0.0001;    % This limit is set for the difference between two coeffs
                       % if the change in coeffs is less than this the final
                       % parameters are acceptable.
count=1;
setError=.02;
while(cond(1))

   % This outputs parameters every 100 iterations to allow the user to see
   % if they chose a good region of interest.
   if count==100
      disp('Parameters');
      disp(package.parameter);
      disp('Value of S');
      disp(package.coeff);
      count=0;
   end

   count=count+1;

   if prevS<package.coeff         % Check if S(i-1) is less than S(i)

      if lambda<1000
      lambda=lambda*10;
      end
      updateJ=0;
      delSign=-1;
      package = crossCorrCoeffLM(Ogl, prevDfd, prevdGdu, prevdGdv,
prevd2Gdu2, prevd2Gdv2, prevd2Gdudv, delX, delY, prevP,
xx,yy,bcs,pp2dir,pp3dir,lambda,updateJ,package.J,package.H,delSign);
   else                          % If S(i-1) is not less than S(i) then
                                 % take P(i) as previous parameters and
                                 % continue.

                                       % If the program is here
                                       % then S(i) continues to be
                                       % less than S(i-1).
      if lambda>.0001
      lambda=lambda/10;
      end

      updateJ=1;
      prevS = package.coeff;
```

```
        prevP = package.parameter;
        delSign=1;
        prevDfd=package.newDfd;
        prevdGdu=package.dGdu;
        prevdGdv=package.dGdv;
        prevd2Gdudv=package.d2Gdudv;
        prevd2Gdu2=package.d2Gdu2;
        prevd2Gdv2=package.d2Gdv2;
        package = crossCorrCoeffLM(Ogl, package.newDfd, package.dGdu,
package.dGdv, package.d2Gdu2, package.d2Gdv2, package.d2Gdudv, delX,
delY, package.parameter,
xx,yy,bcs,pp2dir,pp3dir,lambda,updateJ,package.J,package.H, delSign);

    end

     if (prevS-package.coeff) < diffCoeff       % Check the difference
                                                % between the previous
                                                % coeff and the current
                                                % coeff. Check to see if
                                                % the change is large
                                                % enough to continue.
         if package.coeff < setError
         finalCoeff      = package.coeff;
         finalParameters = package.parameter;
         break
         end
     end

end
toc;
b=struct('finalCoeff', finalCoeff, 'finalParameters', finalParameters, 'OglImage',
Ogl, 'DfdImage', package.newDfd);


function b=secondDerivativeS(F, G, d2Gboth, dG1, dG2, wrt1, wrt2)

denom=(sum(sum(F.^2))*sum(sum(G.^2)));

b=((sum(sum(F.*d2Gboth.*wrt1.*wrt2))*denom^(-.5))...
   ...
    -((sum(sum(F.*dG1.*wrt1)))*(denom^(-1.5))...
    *(sum(sum(F.^2))*(sum(sum(G.*dG2.*wrt2)))))...
    ...
    -((sum(sum(F.*dG2.*wrt2)))*(denom^(-1.5))...
```

```
    *(sum(sum(F.^2))*(sum(sum(G.*dG1.*wrt1))))...
    ...
   -((sum(sum(F.*G)))*(denom^(-1.5))...

*(sum(sum(F.^2))*(sum(sum((dG1.*dG2.*wrt1.*wrt2)+(d2Gboth.*G.*wrt1.*wrt2))))
))...
    ...
    +(3*(sum(sum(F.*G)))*(sum(sum(F.^2))*sum(sum(G.*dG1.*wrt1)))...
    *(denom^(-2.5))*(sum(sum(F.^2))*sum(sum(G.*dG2.*wrt2)))));
```

Matlab code for DIC program with Newton-Raphson Algorithm

```
%% 2-D Image Deformation Program Using The Newton-Raphson Method
% This program will simulate a random deformation to an original image.
% First the program will obtain large offsets by using a coarse search
% method. Once the coarse search is finished smaller translations,
% stretches, and rotations will be obtained using the Newton-Raphson
% iterative reduction method.

%% Initialization
% Here the program is initialized clearing previous variables, the screen
% and setting variable outputs to long style.
clear
clc
format long;

%% Image  Import
% Here the image is being imported from a file previously created.
Ogl  =
double(imread('C:\Users\rhk04004\Documents\MATLAB\DIC5\DICImage2.tif'));
Ogl1 = Ogl(:,:,1);

%% Rotation
% This section will rotate the image to simulate the image being deformed.
% This line of code actually implements the rotation. Once the rotation is
% obtained further image cropping is done.
Ogl_def = imrotate(Ogl1,1, 'bicubic','crop');
Dfd=Ogl_def(156:356,156:356);
Ogl=Ogl1(156:356,126:326);

%% Stretching
% This section will be used to stretch the image to simulate deformation of
```

```
% the original image.
Dfd1 = imresize(Dfd,[211 211]);
Ogl  = double(Ogl(1:201,1:201));
Dfd2  = double(Dfd1(1:201,1:201));

%% Translation
% This section will be used to translate the original image to simulate
% deformation of the image.
Dfd = zeros(206);
Dfd(6:206,6:206) = Dfd2;

%% Coarse Search
% This part of the program is used to find large translations between the
% original and deformed images. The large translation is located using full
% image correlation. Using the initial offsets the first parameter vector
% can be made. The offsets are entered to correct the deformed image and to
% get the two images close to eachother. The syntax of the parameter vector
% is [U;V,Ux;Uy;Vx;Vy].

Offset = CoarseSearch1(Ogl, Dfd);
xOffset = Offset(1);
yOffset = Offset(2);

parameter = [xOffset;yOffset;0;0;0;0];

%% Newton-Raphson Reduction
% The original and deformed images are then entered into the function
% containing the Newton-Raphson reduction along with the initial parameter
% vector.
b=splineCorrelationwasworking(Ogl,Dfd,parameter);

%% Conclusion
% Once the Newton-Raphson reduction is finished the final parameters will
% be displayed to the screen along with the correlation coefficient which
% applied to them.
disp('The Final Parameters are: ');
disp(b.finalParameters);
disp('The Final Correlation Coefficient is: ');
disp(b.finalCoeff);
```

```
function b=splineCorrelationwasworking(firstRegIntOgl,RegIntDfd, prevP)
chooseSize=    40;
numPoints      = 10;                    % Choose how much to interpolate by.
interpby       = 1/numPoints;           % Calculate distance between points.
regionOgl(:,1) = floor(size(firstRegIntOgl,1)/2)-
chooseSize/2:floor(size(firstRegIntOgl,1)/2)+chooseSize/2;
regionOgl(:,2) = floor(size(firstRegIntOgl,2)/2)-
chooseSize/2:floor(size(firstRegIntOgl,2)/2)+chooseSize/2;                % Choose
an inner region to use
                                % because edges of deformed
                                % image may become distorted
                                % with csapi.

sizex_range = max(regionOgl(:,1))-min(regionOgl(:,1)); % number of inbetween
points
sizey_range = max(regionOgl(:,1))-min(regionOgl(:,1)); %#ok

Ogl1    = firstRegIntOgl(regionOgl(:,2),regionOgl(:,1));   % Obtain inner Ogl
                                % (this will never change)

                                % Interpolate Original Image
Ogl     = csapiFunction(Ogl1,regionOgl(:,1),regionOgl(:,2), interpby);

% Obtain list of points which are locations of X and Y in the interpolated
% Original image.
xOgl    = min(min(regionOgl(:,1))):interpby:max(max(regionOgl(:,1)));
yOgl    = min(min(regionOgl(:,2))):interpby:max(max(regionOgl(:,2)));

[xx,yy] = meshgrid(xOgl, yOgl);

% create deltaX and deltaY which is the distance from the side of the image
% where the first point will have a distance of zero from the side of the
% image.
deltaX =(0:interpby:size(Ogl1,1)-1);
deltaY =(0:interpby:size(Ogl1,2)-1);  % verify that this and deltaX are not inverted
% Create vectors containing distances from side of original image to be
% used in calculation of jacobian and hessian matrices.
delX   = repmat(deltaX,size(Ogl,1),1);
delY   = repmat(deltaY',1,size(Ogl,2)); %verify this as well

% Use csapi to obtain structure to which be differentiated to obtain all dG
% images.
```

```
bcs     = csapi({1:size(RegIntDfd,1),1:size(RegIntDfd,2)}, RegIntDfd);

% Obtain derivatives of G. pp2Dir=first derivative pp3Dir=second derivative
pp2dir  = fndir(bcs,[1 0;0 1]);
pp3dir  = fndir(fndir(bcs,[1 0;0 1]),[1 0;0 1]);

% Calculate X and Y locations in the deformed image from locations from
% original image and parameter vector.
Dfd     = zeros(size(Ogl,2));                       %#ok
bcsFirstDerivative = zeros(2,size(Ogl,1),size(Ogl,2));  %#ok
bcsSecondDerivative = zeros(4,size(Ogl,1),size(Ogl,2)); %#ok

dGdu    = zeros(size(yOgl,2));
dGdv    = zeros(size(yOgl,2));
d2Gdu2  = zeros(size(yOgl,2));
d2Gdv2  = zeros(size(yOgl,2));
d2Gdudv = zeros(size(yOgl,2));
xDfd    = zeros(1,size(xOgl,2));    %#ok
yDfd    = zeros(1,size(yOgl,2));    %#ok

xDfd    = xx+prevP(1)+prevP(3)*delX+prevP(4)*delY;
yDfd    = yy+prevP(2)+prevP(5)*delX+prevP(6)*delY;

vect(1,:) = reshape(yDfd,1,[]);
vect(2,:) = reshape(xDfd,1,[]);

% Evaluate structures obtained from csapi to obtain deformed image and
% derivatives
Dfd=fnval(bcs, vect);
bcsFirstDeriv  = fnval(pp2dir, vect);
bcsSecondDeriv = fnval(pp3dir, vect);

% Sort derivatives into their variables from evaluated functions.
Dfd= reshape(Dfd,size(Ogl,1),size(Ogl,2));
dGdv(:,:)    = reshape(bcsFirstDeriv(1,:,:),size(Ogl,1),size(Ogl,2));
dGdu(:,:)    = reshape(bcsFirstDeriv(2,:,:),size(Ogl,1),size(Ogl,2));
d2Gdv2(:,:)  = reshape(bcsSecondDeriv(1,:,:),size(Ogl,1),size(Ogl,2));
d2Gdudv(:,:) = reshape(bcsSecondDeriv(2,:,:),size(Ogl,1),size(Ogl,2));
d2Gdu2(:,:)  = reshape(bcsSecondDeriv(4,:,:),size(Ogl,1),size(Ogl,2));

%Calculate the correlation value for the initial guess
prevS  = crosscorr(Dfd,Ogl);

% Calculate jacobian and hessian, increment parameters, obtain new set of
```

```
% points to calculate new deformed image, and obtain new correlation coeff.
package = crossCorrCoeffNew2wasworking(Ogl, Dfd, dGdu, dGdv, d2Gdu2,
d2Gdv2, d2Gdudv, delX, delY, prevP,xx,yy, bcs, pp2dir, pp3dir, sizex_range);

% Set limits for newton raphson method
diffCoeff = 0.0001;    % This limit is set for the difference between two coeffs
                       % if the change in coeffs is less than this the final
                       % parameters are acceptable.
setError=0.0001;
                       % the Newton-Raphson reduction iterations
count=1;
while(cond(1))
   if count==10
      disp('Parameters');
      disp(package.parameter);
      disp('Value of S');
      disp(package.coeff);
      count=0;
   end
   count=count+1;
   if prevS<package.coeff          % Check if S(i-1) is less than S(i)
      doubleCheck=package.coeff;  % if it is calculate S(i+1).
      package = crossCorrCoeffNew2wasworking(Ogl, package.newDfd,
package.dGdu, package.dGdv, package.d2Gdu2, package.d2Gdv2,
package.d2Gdudv, delX, delY, package.parameter,
xx,yy,bcs,pp2dir,pp3dir,sizex_range);
      if doubleCheck<package.coeff      % Check if S(i) is less than S(i+1)
                              % if it is end the program, correlation
                              % is getting worse.
         finalCoeff = prevS;
         finalParameters = prevP;
         break
      else                      % If S(i+1) is less than S(i-1) and
                              % the program is at this point that
                              % means S(i+1) is also less than S(i)
                              % so take P(i+1) as previous parameters
                              % and continue.
         if prevS==package.coeff     % if S(i-1)==S(i+1) it means the
                                 % iterations are oscillating
                                 % therefore take that answer.
            finalCoeff      = prevS;
            finalParameters = package.parameter;
            break
         end
```

```
        prevS = package.coeff;
        prevP = package.parameter;
      end

   else                        % If S(i-1) is not less than S(i) then
                               % take P(i) as previous parameters and
                               % continue.
      if (prevS-package.coeff) < diffCoeff    % Check the difference
                                   % between the previous
                                   % coeff and the current
                                   % coeff. Check to see if
                                   % the change is large
                                   % enough to continue.
        if package.coeff < setError
        finalCoeff      = package.coeff;
        finalParameters = package.parameter;
        break
        end
      end
                                   % If the program is here
                                   % then S(i) continues to be
                                   % less than S(i-1).
      prevS = package.coeff;
      prevP = package.parameter;
      package = crossCorrCoeffNew2wasworking(Ogl, package.newDfd,
package.dGdu, package.dGdv, package.d2Gdu2, package.d2Gdv2,
package.d2Gdudv, delX, delY, prevP, xx,yy,bcs,pp2dir,pp3dir,sizex_range);
   end
end

% for i=1:10
%     package = crossCorrCoeffNew2(Ogl, package.newDfd, package.dGdu,
package.dGdv, package.d2Gdu2, package.d2Gdv2, package.d2Gdudv, delX,
delY, prevP,deltaX, deltaY, xx,yy,bcs,pp2dir,pp3dir);
%     prevP=package.parameter;
%     coeffs(i)=package.coeff;
%     Ps(:,i)=(package.parameter);
% end
toc;
b=struct('finalCoeff', finalCoeff, 'finalParameters', finalParameters, 'OglImage',
Ogl, 'DfdImage', package.newDfd);
```

```
function b=crossCorrCoeffNew2wasworking(Ogl,Dfd,dGdu,dGdv, d2Gdu2,
d2Gdv2, d2Gdudv,delX, delY, parameter, xOgl, yOgl, bcs, pp2dir, pp3dir,~)

% This function takes two arrays (original image and deformed image, and
% the spline coefficients of the deformed image as input, runs a
% newton-raphson on the two images to find the cross-correlation
% coefficient.

% Calculate the Jacobian and the Hessian
package = jacobianHessian(Ogl, Dfd, dGdu, dGdv, d2Gdu2, d2Gdv2, d2Gdudv,
delX, delY);

% the Jacobian
J       = package.jacobian;

% the Hessian
H       = package.hessian;

% Calculate delP (use the backslash to operate the inverse)
delP    = H\J';

% Update the values in the parameter vector.
P       = parameter+delP;

% Obtain new X and Y coordinates in the deformed image.
xDfd    = xOgl+P(1)+P(3)*delX+P(4)*delY;
yDfd    = yOgl+P(2)+P(5)*delX+P(6)*delY;

vect(1,:) = reshape(yDfd,1,[]);
vect(2,:) = reshape(xDfd,1,[]);
% evaluate the deformed image at the new found points as well as the first
% and second derivatives.

newDfd =fnval(bcs, vect);
bcsFirstDeriv =fnval(pp2dir, vect);
bcsSecondDeriv =fnval(pp3dir, vect);

% Sort evaluated images into their correct variables.
newDfd       = reshape(newDfd,size(Ogl,1),size(Ogl,2));
dGdv(:,:)    = reshape(bcsFirstDeriv(1,:,:)*5,size(Ogl,1),size(Ogl,2));
dGdu(:,:)    = reshape(bcsFirstDeriv(2,:,:)*5,size(Ogl,1),size(Ogl,2));
d2Gdv2(:,:)  = reshape(bcsSecondDeriv(1,:,:)*5,size(Ogl,1),size(Ogl,2));
d2Gdudv(:,:) = reshape(bcsSecondDeriv(2,:,:)*5,size(Ogl,1),size(Ogl,2));
d2Gdu2(:,:)  = reshape(bcsSecondDeriv(4,:,:)*5,size(Ogl,1),size(Ogl,2));
```

```
% Calculate new correlation coefficient.
S = crosscorr(newDfd,Ogl);

b=struct('coeff',S,'parameter',P,'delP',delP,'newDfd', newDfd, 'dGdu', dGdu,
'dGdv',dGdv,'d2Gdu2', d2Gdu2,'d2Gdv2', d2Gdv2, 'd2Gdudv', d2Gdudv);



function c = crosscorr(a,b)

%This function gives the crosscorrelation function between a abd b where in
%c=[1-sum(a[:]*b[:]/[{sum(a[:]^2}*{sum(b[:]^2}]^0.5]. The range of this
%function is (0,1). Zero indicates no correlation and one indicates total
%correlation.
num = sum(sum(a.*b));
denom = (sum(sum(realpow(a,2)))*sum(sum(realpow(b,2))))^0.5;
c = 1 - (num/denom);
```

B. Image Sets Used in Analysis

B.1 Image Sets for 0.05 µM PGE2 Experiment

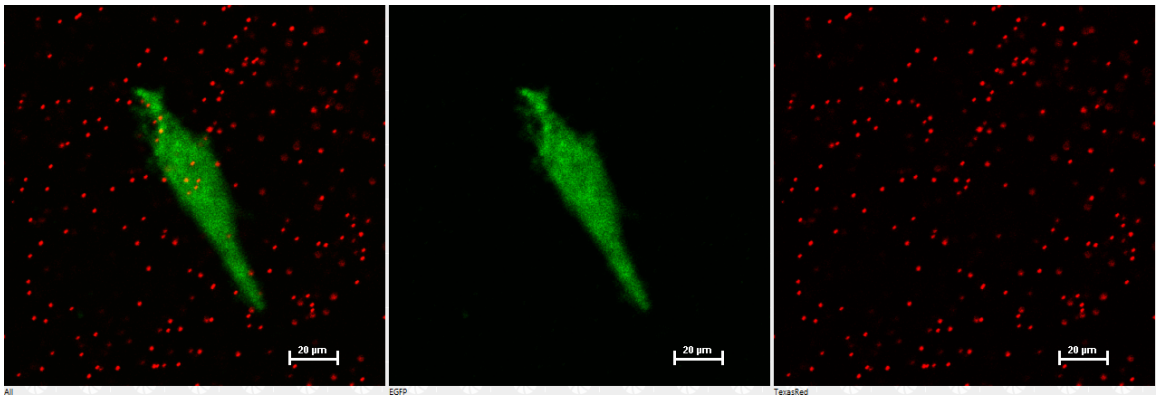


Figure B.11 Cell-Gel System 1 Before Addition of PGE2

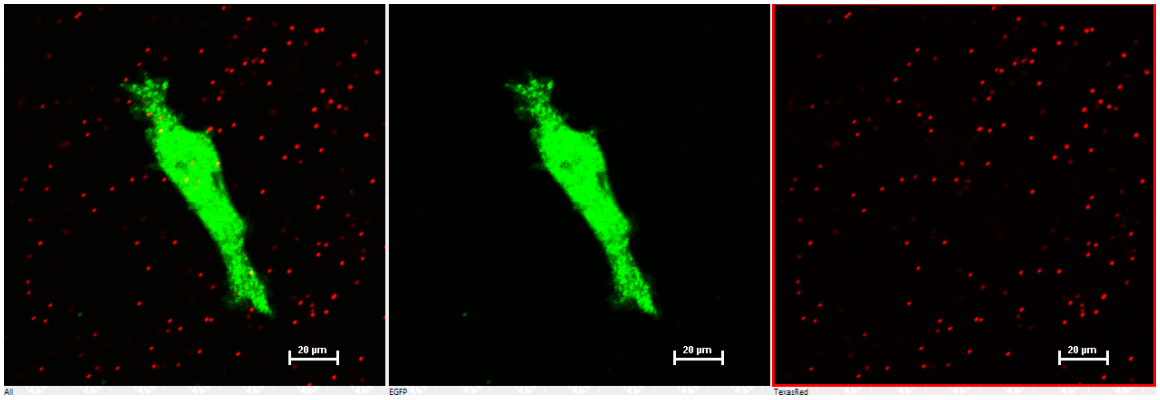


Figure B.2 Cell-Gel System 1 After Addition of 0.05 µM PGE2

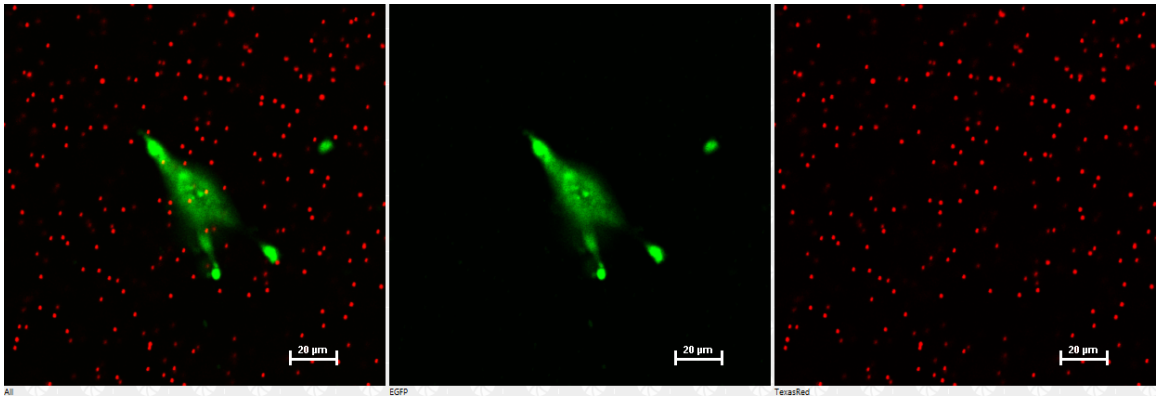


Figure B.3 Cell-Gel System 2 Before Addition of PGE2

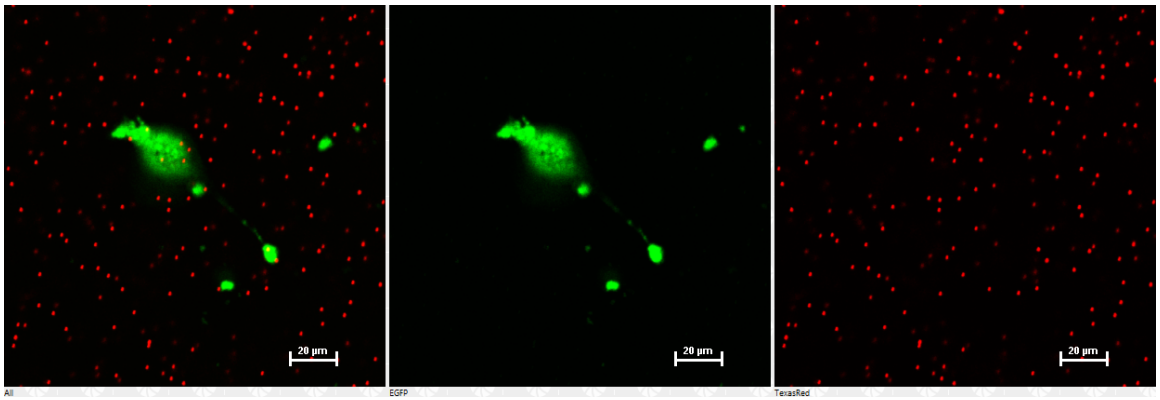


Figure B.4 Cell-Gel System 2 After Addition of 0.05 µM PGE2

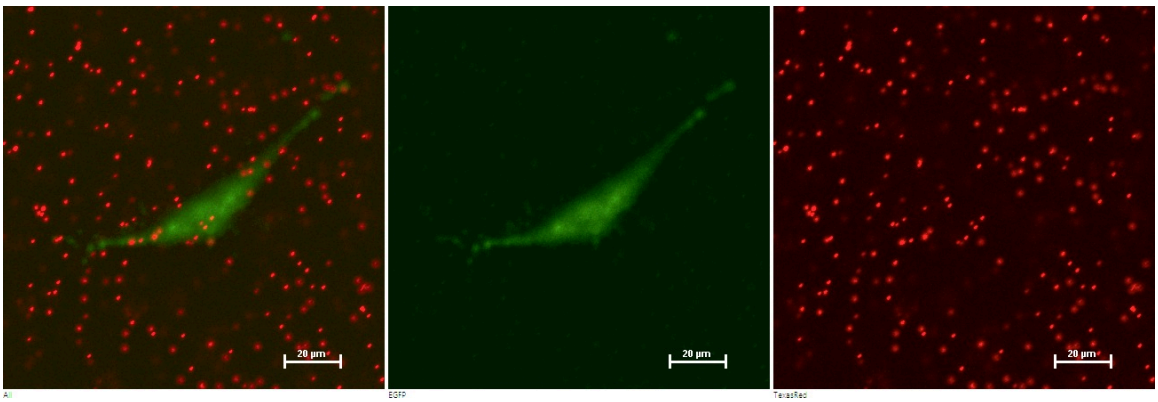


Figure B.5 Cell-Gel System 3 Before Addition of PGE2

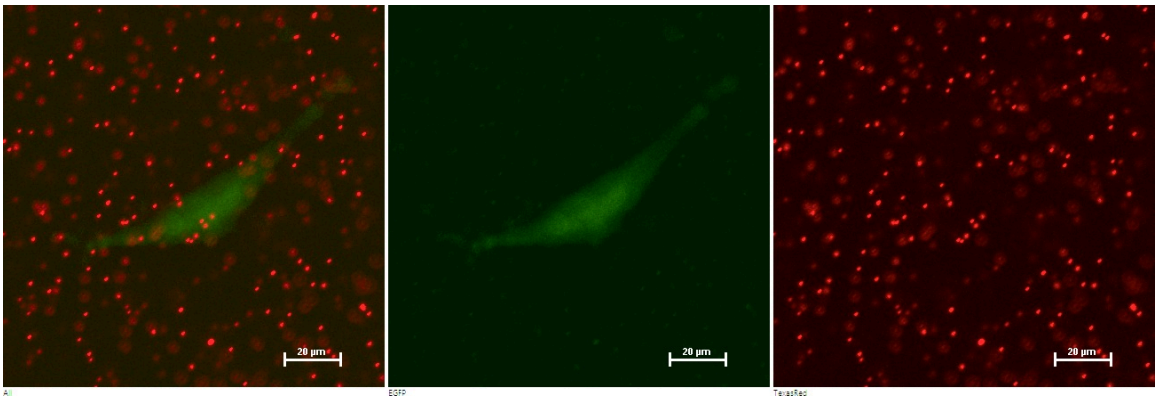


Figure B.6 Cell-Gel System 3 After Addition of 0.05 µM PGE2

B.2 Image Sets for 0.10 µM PGE2 Experiment

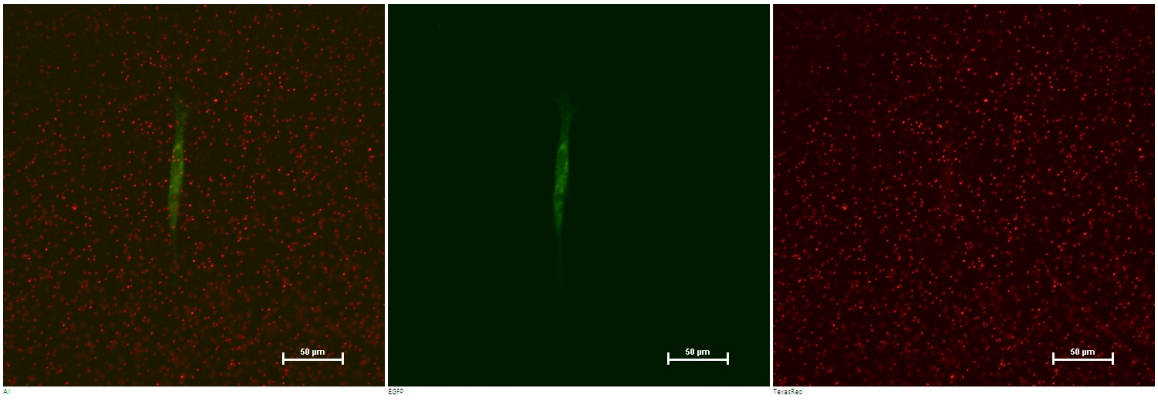


Figure B.7 Cell-Gel System 4 Before Addition of PGE2

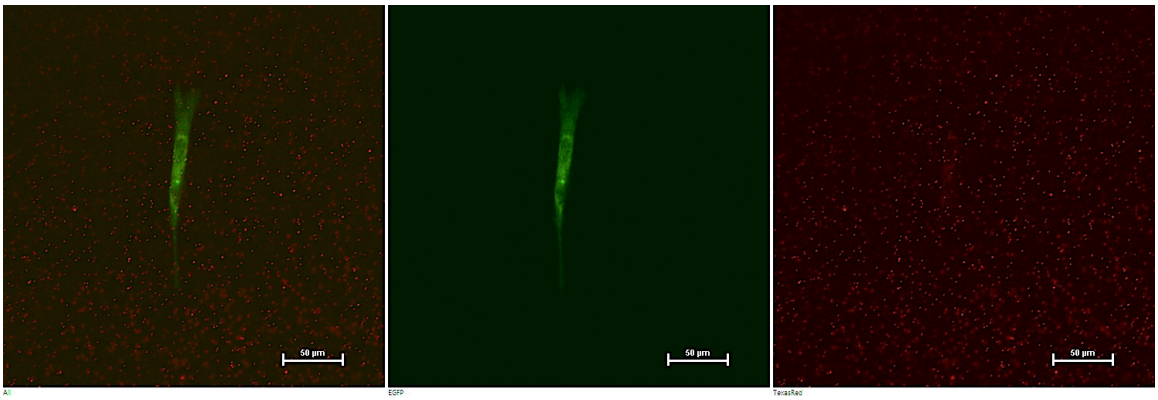


Figure B.8 Cell-Gel System 4 After Addition of 0.10 µM PGE2

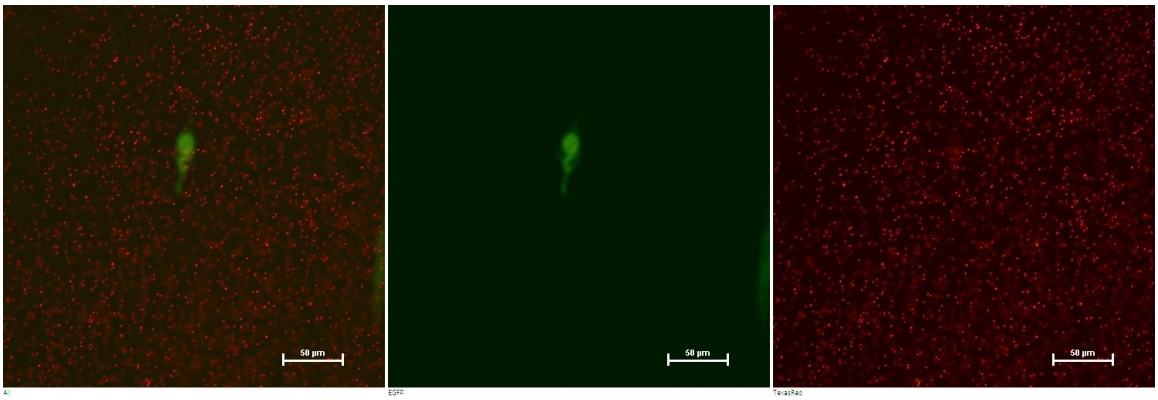


Figure B.9 Cell-Gel System 5 Before Addition of PGE2

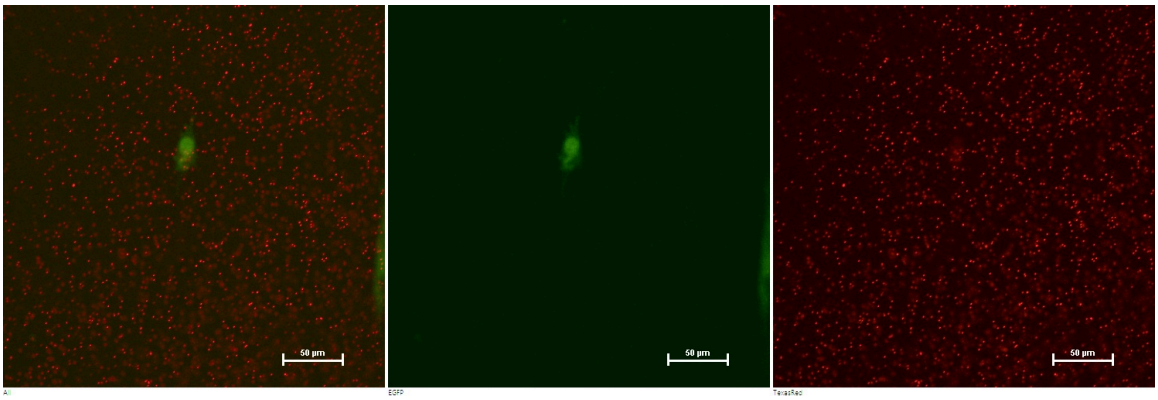


Figure B.10 Cell-Gel System 5 After Addition of 0.10 µM PGE2

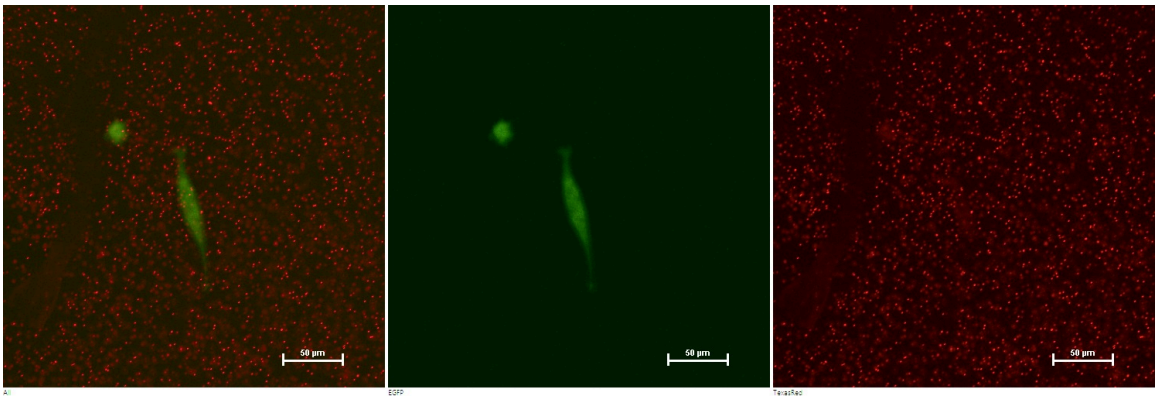


Figure B.11 Cell-Gel System 6 Before Addition of PGE2

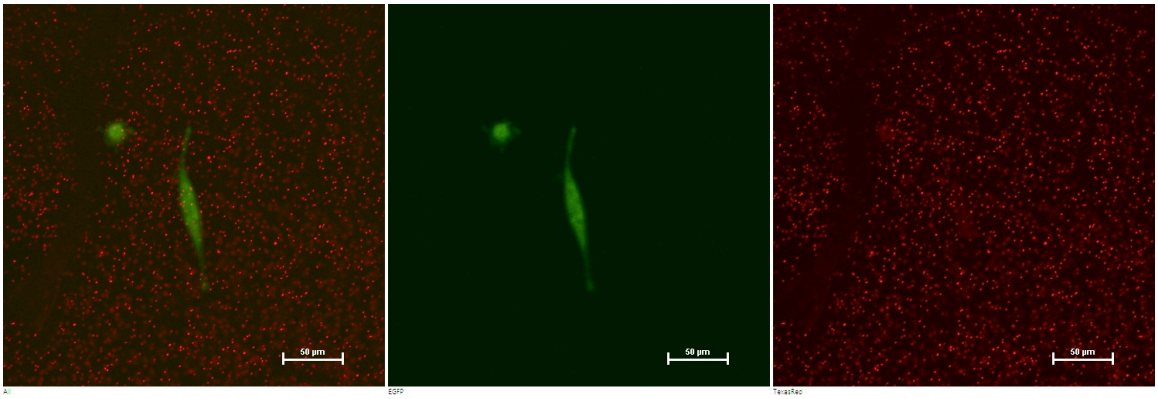


Figure B.12 Cell-Gel System 6 After Addition of 0.10 µM PGE2