FULL FIELD STUDY OF STRAIN DISTRIBUTION NEAR THE CRACK TIP IN THE FRACTURE OF SOLID PROPELLANTS VIA LARGE STRAIN DIGITAL IMAGE CORRELATION AND OPTICAL MICROSCOPY

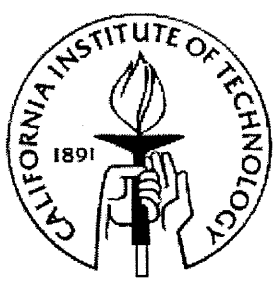
Thesis by

Javier Gonzalez

In Partial Fulfillment of the Requirements

for the Degree of

Aeronautical Engineer



California Institute of Technology

Pasadena, California

1997

(submitted January 14, 1997)

To my parents, my sister Ana and my brothers Pedro, Alejandro and Santiago.

ACKNOWLEDGMENTS

I wish to thank Professor W. G. Knauss for his support and advice and acknowledge the valuable help of my classmates: Demirkan Coker, Pradeep Guduru, Bibhuti Patel, Samudrala Omprakrash, Sangwook Lee, Eduardo Repetto, Sandeep Sane and Raul Radovitzky for the valuable insight they gave me through discussions on various segments of my research topic. It has been of great help to discuss also my project with the Post doctoral scholars Dr. Allan Zhong, Dr. Anne Gelb and Dr. Alfons Noe. I would like to thank Professor Ravichandran and Professor Shepherd for being part of the thesis committee and giving me valuable advice in my research topic. Finally I would like to thank the doctoral student Hongbing Lu for teaching me the digital image correlation procedure.

ABSTRACT

A full field method for visualizing deformation around the crack tip in a fracture process with large strains is developed. A digital image correlation program (DIC) is used to incrementally compute strains and displacements between two consecutive images of a deformation process. Values of strain and displacements for consecutive deformations are added, this way solving convergence problems in the DIC algorithm when large deformations are investigated. The method developed is used to investigate the strain distribution within 1 mm of the crack tip in a particulate composite solid (propellant) using microscopic visualization of the deformation process

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Particulate composites are widely used in engineering. In the automotive industry carbon black filled rubbers are used in tires. Many injection molded materials are filled with small particles, while other rigid polymers are toughened through the addition of rubber particles. Solid propellant rocket fuels are physical mixtures of ammonium perclorate and aluminum powder often in multimodal size distribution, bonded together by a rubber phase called the matrix. Failure in all these materials is heavily dependent upon the interaction between the particles and the matrix, specifically on the separation of particles and binder. Failure is also dependent upon the volume ratio of particles to matrix, which is typically close to 75%. In the sequel we examine the failure progression in a solid propellant Triokol TPH 1011. Application of continuum mechanics to the stress/strain analysis of structures made of these types of materials typically invoke macroscopically homogeneous material performance, even though at the scale level of the particles deformations are anything but homogeneous. We shall see that inhomogeneous deformations occur at a size scale substantially larger than the largest particle, and that the failure process is directly dependent upon these micro-structural deformations. Measuring large deformation strains over small domains of tens to hundreds of microns is not a trivial matter. Imprinted grids tend to serve well at a size scale just above what is required here. Determining the micromechanical deformation with the aid of optical microscopy, e.g. on the tip of a macroscopic crack, implies the need to extend the presently available tools of strain measurements. In principle the digital image correlation method [Sutton, 1986; Vendroux/Knauss, 1994] is ideal for this purpose except that it is not suitable if the deformations are too large. For deformations where

strains of 10% are reached, the Correlation algorithm fails to converge. Strains of 50% to 100% are typical for crack propagation problems in solid propellants. Accordingly we develop and examine here an incremental application that follows the deformation history. This development is addressed first in Section 2, followed by a discussion of the experimental setup and arrangement to define fields around a slowly growing crack. The method developed is used to quantitatively describe deformation on cracked and uncracked specimens of solid propellant TPH 1011. Particular interest is devoted to the inhomogeneity of the material.

2. LARGE DEFORMATION DIGITAL IMAGE CORRELATION (LD-DIC).

To compute strain and displacement fields in large deformations, the Digital Image Correlation (DIC) program cannot be applied in a straightforward manner, as is done for small strains. In this section a method is presented by which the total deformation is subdivided into smaller deformation increments, each of which can be processed by DIC. The results of the DIC program over the small deformation increments are then combined to compute the strain distribution for the large deformation.

2. 1. THE DIC PROGRAM

Developed by Sutton [Sutton, 1986] and improved by Vendroux and Knauss [Vendroux and Knauss, 1994], the Digital Image Correlation (DIC) program is used to measure the displacement field and its gradients from images of an undeformed and deformed body. These images are gray levels images consisting of a grid of pixels, (typically 640 by 480) with gray level ranging from 0 to 255. In this way the images represent a surface in which the heights at grid points represent an associated gray level distribution.

Let X be the mapping of the undeformed configuration onto the deformed configuration so that a material point is represented in the undeformed configuration at the coordinates (x, y) and has an associated gray level value f(x,y). The same material point in the deformed configuration is represented at $(\tilde{x},\tilde{y})$ with a gray level of $g(\tilde{x},\tilde{y})$. It is assumed that the deformation does not

significantly modify the gray level, i.e. f(x,y)≈g($\tilde{x},\tilde{y}$). If one assumes that the deformation is such that the topology (profile pattern) after deformation is uniquely related to that before the deformation, one may determine the deformations (displacement and their gradients) through a correlation between the two pattern images. Let a material point be represented by G(x,y), where x,y are its coordinates in the undeformed configuration. Similarly the same point is represented by $\tilde{G}(\tilde{x},\tilde{y})$ in the deformed configuration, where $\tilde{x},\tilde{y}$ are the coordinates of the material point in the deformed configuration. (Fig 1)

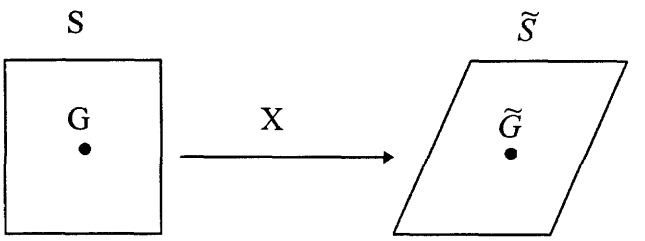


Figure 1. Mapping X.

Define X as the mapping from the undeformed to the deformed configuration

$$X: \quad R^2 -> R^2$$

$$G -> \tilde{G} = X(G) / \ g(\tilde{x},\tilde{y}) = f(x, y). \qquad (1)$$

Relation (1) can be written as

$$\tilde{x} = x + u(x, y)$$
$$\tilde{y} = y + v(x, y), \qquad (2)$$

where u and v are the displacements of G in the Lagrangian reference frame. Let $\tilde{G}_0$ be the image of G_0 through the deformation X. Let S be a neighborhood of G_0 that is mapped onto the set $\tilde{S}$ such that $\tilde{S}$ is a neighborhood of $\tilde{G}_0$. Considering this neighborhood S to be small, the two configurations of the deformation are related by

$\forall\ \tilde{G}(\tilde{x},\tilde{y})$, $\exists$ G(x, y) such that

$$\tilde{x} = x + u(x_0,y_0) + u_x|_{(x_0,y_0)}(x - x_0) + u_y|_{(x_0,y_0)}(y - y_0)$$
$$\tilde{y} = y + v(x_0,y_0) + v_x|_{(x_0,y_0)}(x - x_0) + v_y|_{(x_0,y_0)}(y - y_0). \qquad (3)$$

These equations define a new local mapping X_l around G_0. At this point we introduce the least square correlation coefficient C.

$$C = \frac{\iint_S (f(G) - g(X_l(G)))^2 dS}{\iint_S f^2(G) dS}. \qquad (4)$$

The present correlation method minimizes this correlation coefficient C. It will be a minimum when the parameters of the mapping X_l

$$u_0 = u(x_0, y_0) \qquad v_0 = v(x_0, y_0)$$

$$u_{0,x} = \frac{\partial u}{\partial x}(x_0, y_0) \qquad v_{0,x} = \frac{\partial v}{\partial x}(x_0, y_0)$$

$$u_{0,y} = \frac{\partial u}{\partial y}(x_0, y_0) \qquad v_{0,y} = \frac{\partial v}{\partial y}(x_0, y_0) \tag{5}$$

are exact, i.e. C is then identically zero. Thus the displacements and displacement gradients of the deformation at the point of interest are obtained in the process of minimizing C. In the present application (Pixel location), the definition of the least square coefficient is discretized and integration signs are replaced by summation signs, so that one has

$$C = \frac{\sum_{G_p \in S} (f(G_p) - g(X_l(G_p)))^2}{\sum_{G_p \in S} f^2(G_P)}. \tag{6}$$

Since G_p is a discrete (pixel) point in the undeformed configuration, its value of gray level f(G_p) is directly defined. However, in the deformed configuration the same material point may not be represented by one of the discrete points of the (pixel) grid $\tilde{G}$=X$_l$(G_p), and its gray level is thus not known. It is therefore necessary to deal with the gray level assignment through interpolation around the points in the deformed configuration. This is done by fitting a bicubic spline to a domain of 40 pixels around the points of interest.

2. 1. 1. Optimization scheme

The numerical problem is to minimize the correlation coefficient with respect to the deformation parameters of X_I, i.e. u_0, v_0, $u_{0,x}$, $u_{0,y}$, $v_{0,x}$, $v_{0,y}$. Let these parameters define a six-dimensional space D such that

$$D = \{\mathbf{P} \in R^6 \mid \mathbf{P}(u_0, v_0, u_{0,x}, u_{0,y}, v_{0,x}, v_{0,y})\}. \tag{7}$$

If $\mathbf{P}_0$ is a vector in D and **P** is the vector solution that minimizes (6), C(**P**) can be written as a truncated Taylor series around $\mathbf{P}_o$

$$C(\mathbf{P})=C(\mathbf{P}_o)+\nabla C(\mathbf{P}_o)^T(\mathbf{P}-\mathbf{P}_o)+1/2(\mathbf{P}-\mathbf{P}_o)^T \nabla \nabla C(\mathbf{P}_o)(\mathbf{P}-\mathbf{P}_o). \tag{8}$$

Since **P** makes C a minimum, it follows that $\nabla C(\mathbf{P})=0$, thereby taking the gradient of (8) results in

$$\nabla \nabla C(\mathbf{P}_o)(\mathbf{P}-\mathbf{P}_o) = -\nabla C(\mathbf{P}_o). \tag{9}$$

P can be deduced from (9) if (8) is a good approximation. This is true if **P** is sufficiently close to $\mathbf{P}_o$. The problem can then be solved by the Newton-Raphson optimization algorithm.

2. 2. LIMITS IN THE DIC PROGRAM

Before attempting to apply the DIC program to determine the strains and displacements, it was tested on known deformations in order to establish the largest strain which the program allows. A test was performed on silicone specimens splattered with microscopic speckles to provide the random gray level distribution for the DIC program to identify. The speckles were generated with an airbrush to match the scale of the surface fractures in the solid propellant specimens to be studied later.

2. 2. 1. Convergence depending on strain level

The problem in applying the DIC program to compute strain distributions in a large deformation process is the failure of convergence of the DIC algorithm if the deformation is too large. To test the actual limits on the strain level for convergence of the program, a test on a homogeneous silicone rubber stretched in the y direction was performed. The resultant undeformed and deformed images for stretches from 0% to 40% were compared by the DIC program. For each deformation the strains and displacements were computed at 300 points. The fraction of points at which the minimization process converged is presented in Figure 2 as a function of the Lagrangian strain. For deformations larger than 10% there is a serious decrease in the successful points of convergence. For the purpose of studying cracked solid propellants, where strains in excess of 30% need to be measured, the applicability of the standard DIC method is therefore seriously compromised. Thus a new analysis tool must be developed.

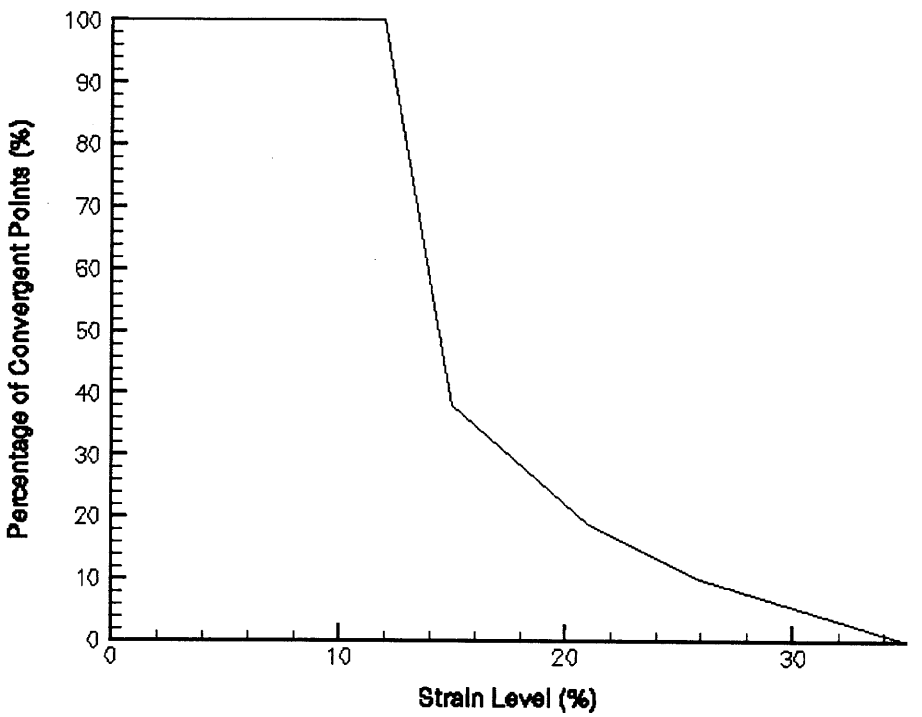


Figure 2. Convergence test for DIC program.

2. 3. PROPOSED SOLUTION

The largest deformation for which the DIC program provides an acceptable result is for a deformation corresponding to a principal strain of about 10%. For larger deformations, we apply the method incrementally through a set of deformations defined in consecutive deformation stages such that a strain greater than 10% is not reached in any increment. Once the strain and displacement maps for the incremental deformations are obtained, continuum concepts are used to construct

the overall deformation. Four schemes for adding strain and displacement fields for the deformation are investigated to single out the most accurate. These schemes are presented in the following section.

2. 4. ADDITION OF STRAIN AND DISPLACEMENT FIELDS METHODS

The general problem can be outlined with the help of Figure 3. For a simple deformation process we establish three pictures of the body, each associated with the configurations 1, 2 and 3 of the sequential deformation

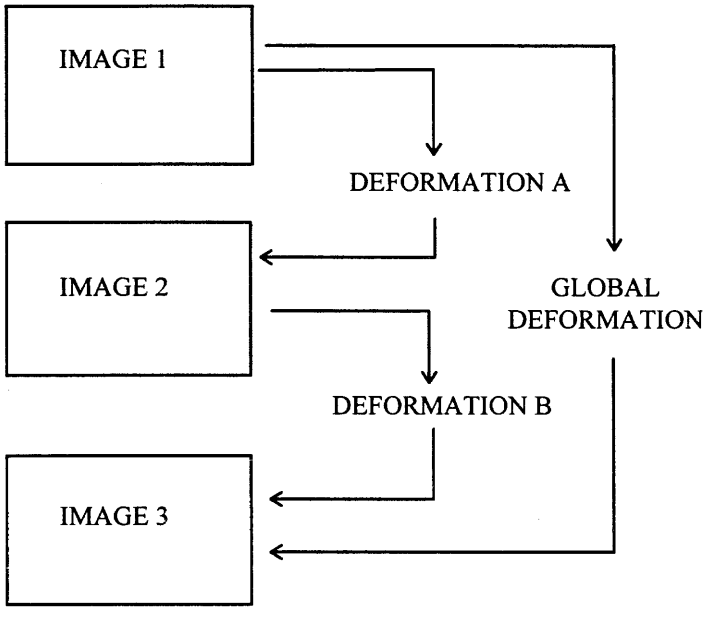


Figure 3. Large Deformation DIC step.

Between image 1 and 2 (deformation A) and between images 2 and 3 (deformation B), the program is successful in giving the deformation fields. However, the strains between images 1 and 3 (Global deformation) are larger than those that lead to the convergence of the correlation. We determine the deformation fields for the global deformation, corresponding to image 3 by using the results that the DIC program provides for the deformations A and B. The DIC strain and displacement maps are discrete representations. The strains and displacements are only computed at a set of pixels forming a grid over the undeformed configuration. A point located in the undeformed reference frame at a pixel which lies in this grid while undergoing displacements in deformation A is not likely to end up as a pixel point in the grid over which the correlation process is performed in the beginning of the second deformation step (B). In order to refer both deformation increments to a common (Lagrangian) reference frame, it will be necessary to interpolate the positions of gray level features for the second deformation relative to the first pixel location. One feature to emphasize in this discussion is that the DIC program calculates the large deformation parameters in a Lagrangian setting. The methods used are:

2. 4. 1. Method 1:

The first method proposed to add the strain and displacement maps from consecutive deformations (A and B) in order to reconstruct the global deformation uses the fact that for large strain deformations, one can compute the deformation gradient tensor for the global deformation by multiplying the deformation gradient tensors of the other two deformations (A and B), giving the expression

$$\tilde{\tilde{F}}_{global} = \tilde{\tilde{F}}_B \tilde{\tilde{F}}_A . \tag{10}$$

A discrete set of particles, H_i , are represented on configuration 1 by the rectangular grid of points, G_i . Those particles are also represented after deformation A by the points $\tilde{G}_i$ in the configuration 2. The results that the DIC program yields for this process are the displacements $u^A{}_i$ and $v^A{}_i$, and the displacement gradients $u_x{}^A{}_i$, $v_y{}^A{}_i$, $u_y{}^A{}_i$ and $v_x{}^A{}_i$ .These values are presented in a Lagrangian setting, that is, with respect to configuration 1. On the configuration 2, the discrete set of material point, H_i, are represented by

$$\tilde{G}_i = G_i + \tilde{u}_i^A . \tag{11}$$

During the second deformation, a different set of particles, J_j, represented on configuration 2 by the rectangular grid of points $\tilde{K}_j$ are mapped onto the set $\tilde{\tilde{K}}_j$ in configuration 3. The set of points $\tilde{\tilde{K}}_j$ can be represented in a Lagrangian setting as

$$\tilde{\tilde{K}}_j = \tilde{K}_j + \tilde{u}_j^B \tag{12}$$

Figure 4 illustrates the process.

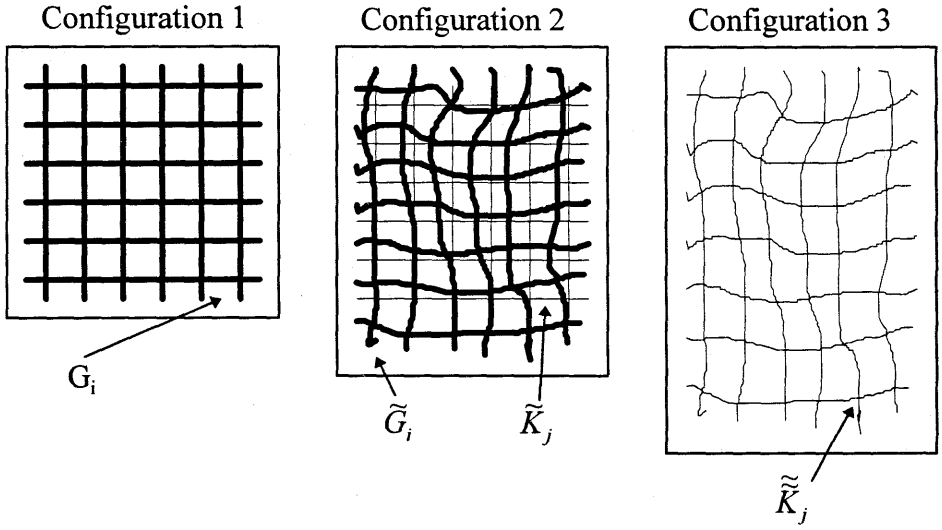


Figure 4. Interpolation process.

Since the global deformation must be expressed in a Lagrangian frame, interpolation of the results of deformation B on the particles J_j is required to obtain the displacements and displacements gradients $u^B{}_i$, $v^B{}_i$, $u_x{}^B{}_i$, $v_y{}^B{}_i$, $u_y{}^B{}_i$ and $v_x{}^B{}_i$ of the particles H_i during deformation B. This is done by fitting a bilinear surface to the four closest points $\tilde{G}_i$ to the point $\tilde{K}_j$ and evaluating it at $\tilde{K}_j$. Then, invoking the tensorial relation (12) we can derive expressions for the displacement gradients of the global deformation by

$$u_x{}^{global} = u_x{}^A + u_x{}^B + u_x{}^A u_x{}^B + u_y{}^B v_x{}^A$$

$$v_y{}^{global} = v_y{}^A + v_y{}^B + v_y{}^A v_y{}^B + v_x{}^B u_y{}^A$$

$$u_y^{global} = u_y^A + u_y^B + u_y^A u_x^B + u_y^B v_y^A$$
$$v_x^{global} = v_x^A + v_x^B + u_x^A v_x^B + v_y^B v_x^A \qquad (13)$$

while the displacements are simply

$$u^{global} = u^A + u^B$$
$$v^{global} = v^A + v^B . \qquad (14)$$

The next step is to construct the Lagrangian strain tensor according to the definition:

$$\varepsilon_{xx} = u_x^{global} + \frac{1}{2}\left\{u_x^{global2} + v_x^{global2}\right\}$$
$$\varepsilon_{yy} = v_y^{global} + \frac{1}{2}\left\{u_y^{global2} + v_y^{global2}\right\}$$
$$\varepsilon_{xy} = \frac{1}{2}\left\{u_y^{global} + v_x^{global}\right\} + \frac{1}{2}\left\{u_x^{global} u_y^{global} + v_x^{global} v_y^{global}\right\} . \qquad (15)$$

2. 4. 2. Method 2

The second method computes the displacements for the global deformation like in the first method. Once the displacement field is obtained, one differentiates it numerically to obtain the displacement gradients, according to the formulas:

$$u_x = \frac{u2 - u1}{\Delta x}$$
$$v_y = \frac{v3 - v1}{\Delta y}$$
$$u_y = \frac{u3 - u1}{\Delta y}$$
$$v_x = \frac{v2 - v1}{\Delta x}. \qquad (16)$$

From which we compute the Lagrangian strains.

2. 4. 3. Method 3

The third scheme is basically the same as the second except that when finding the displacements of the deformation B, the interpolation process is performed using six neighboring points instead of four. This allows the appearance of second-order terms in the interpolating functions.

2. 4. 4. Method 4

In the last scheme the interpolation process for the displacements is changed to a least square fitting with respect to 16 neighboring points. The biquadratic surface obtained in this way is evaluated at a pixel point. The surface is differentiated to evaluate the required derivatives at the same point.

2. 5. CALIBRATION OF METHODS FOR ADDITION OF FIELDS

A specimen of an homogeneous silicone rubber without a crack and coated with microscopic speckles was strained sequentially in the y direction up to a Lagrangian strain of 80%. A sequence of 15 images of the process was taken so that the global deformation was divided into 14 sub deformations of 3-4% Lagrangian strain each. The resultant deformations were added by using the four methods, and the results were compared with the optically measured Lagrangian strain at every step. The results of this test series are presented in Figure 5, which shows very similar characteristics for each of the four methods. Figure 6 shows the difference of the results of each method and the optically measured Lagrangian strains. From this figure we can conclude that method 1 deviates the least from the optically measured strain. The maximum deviation occurs at a strain value of 40%, and yields 1% strain difference between the optical and the DIC Lagrangian strain.

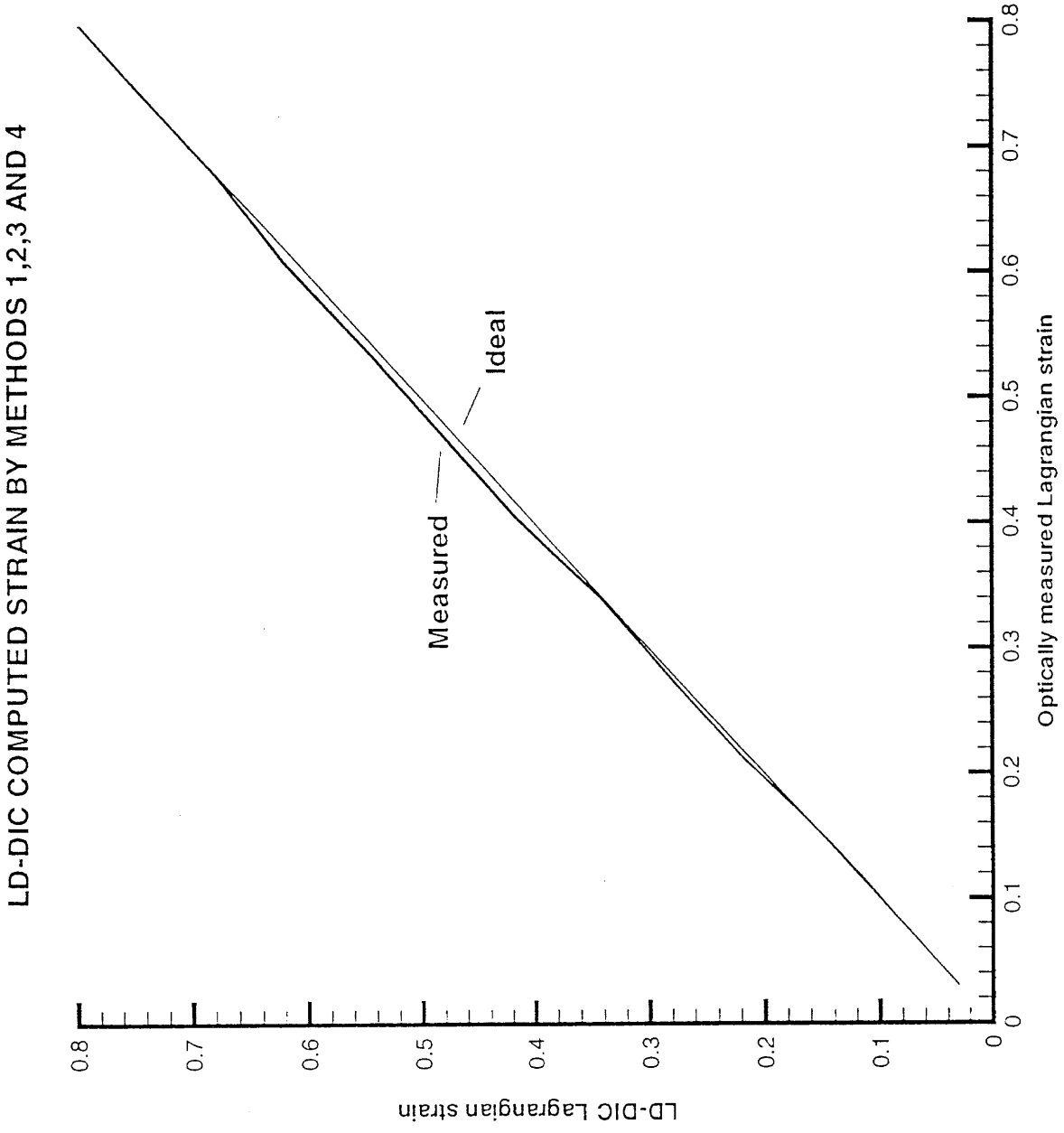

LD-DIC COMPUTED STRAIN BY METHODS 1,2,3 AND 4
Ideal
Measured
LD-DIC Lagrangian strain
Optically measured Lagrangian strain
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

Figure 5.

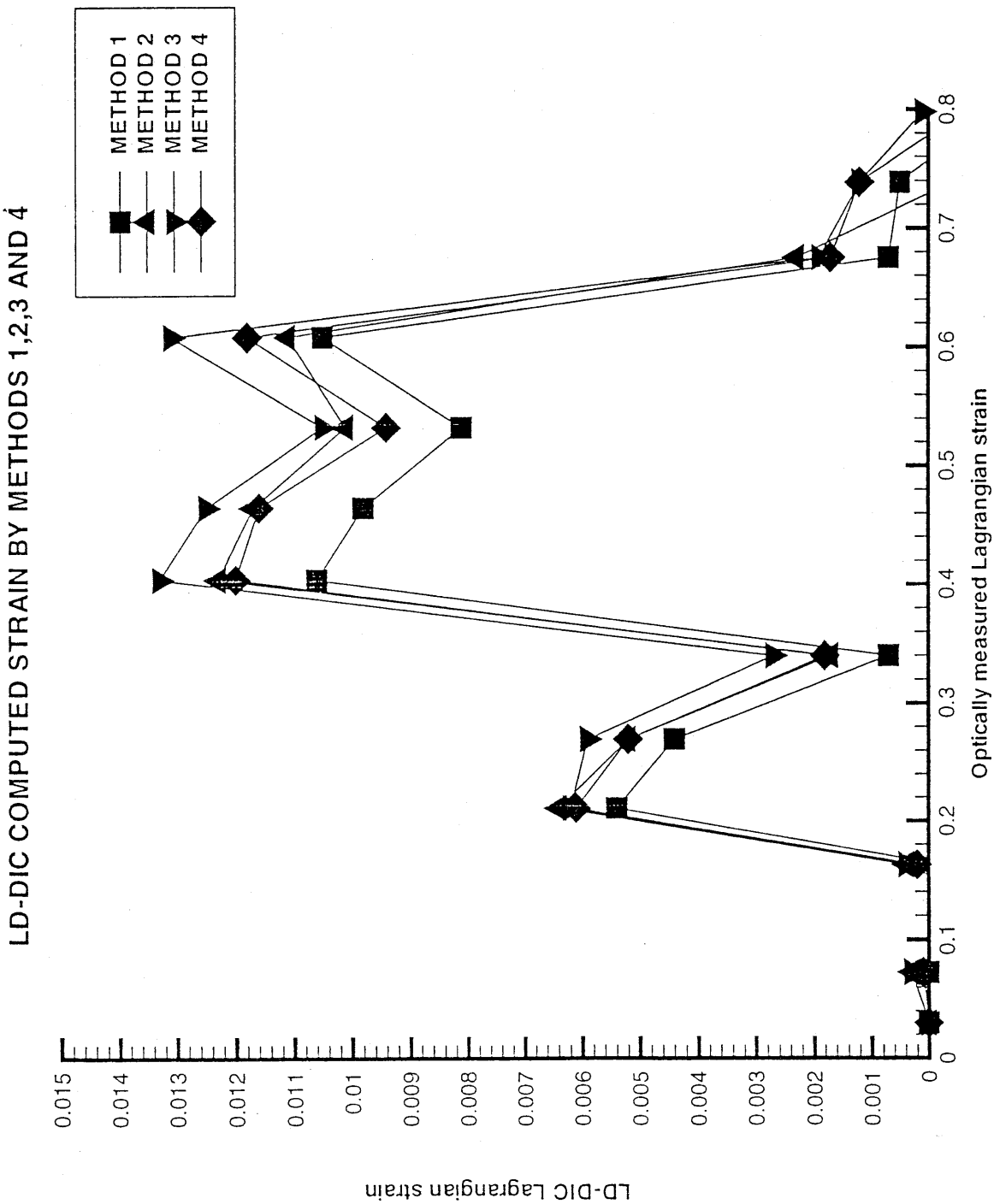

LD-DIC COMPUTED STRAIN BY METHODS 1,2,3 AND 4
METHOD 1
METHOD 2
METHOD 3
METHOD 4
LD-DIC Lagrangian strain
0.015
0.014
0.013
0.012
0.011
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Optically measured Lagrangian strain

Figure 6.

3. APPLICATION OF THE LARGE DEFORMATION DIC METHOD TO THE CRACK OPENING PROBLEM IN SOLID PROPELLANT

The Large Deformation Digital Image Correlation method is used to obtain Lagrangian strain distributions within 1 mm of a crack tip in a solid propellant TPH 1011 specimen.

3. 1. EXPERIMENTAL SETUP

During this experiment, a cracked specimen of solid propellant TPH 1011 is loaded with a constant strain rate in the direction perpendicular to the crack. The crack, initiated with a razor blade, opens with an increase in global strain level. The crack opening process is monitored at a microscopic level. Six digital images of 640 x 480 pixels, representing 3 mm x 4 mm of the specimen surface are obtained, one every 10 seconds. These images of the specimen surface are taken for far field Lagrangian E_{yy} strains of 0%, 1%, 2%, 3%, 4% and 5%. These six images are associated with six different deformation configurations. They also define five intermediate deformations. Using the Large Deformation Digital Image Correlation (LD-DIC) method, the displacement and displacement gradient fields for the global deformation are constructed from fields corresponding to the intermediate deformations. A schematic of the experimental setup is shown in Figure 7. The equipment used to prescribe the desired deformation consists of the following: a strain stage (Figure 8), a positioning stage (Figure 9), a stepping motor (Figure 10), and a controller for the positioning stage (Figure 11). The instruments used for the visualization of the process are a Nikon Metallurgical microscope (Figure 12), a CCD camera (Figure 13) and a personal computer with

a frame grabber unit. Finally the images of the experiment are processed by a Sun workstation.

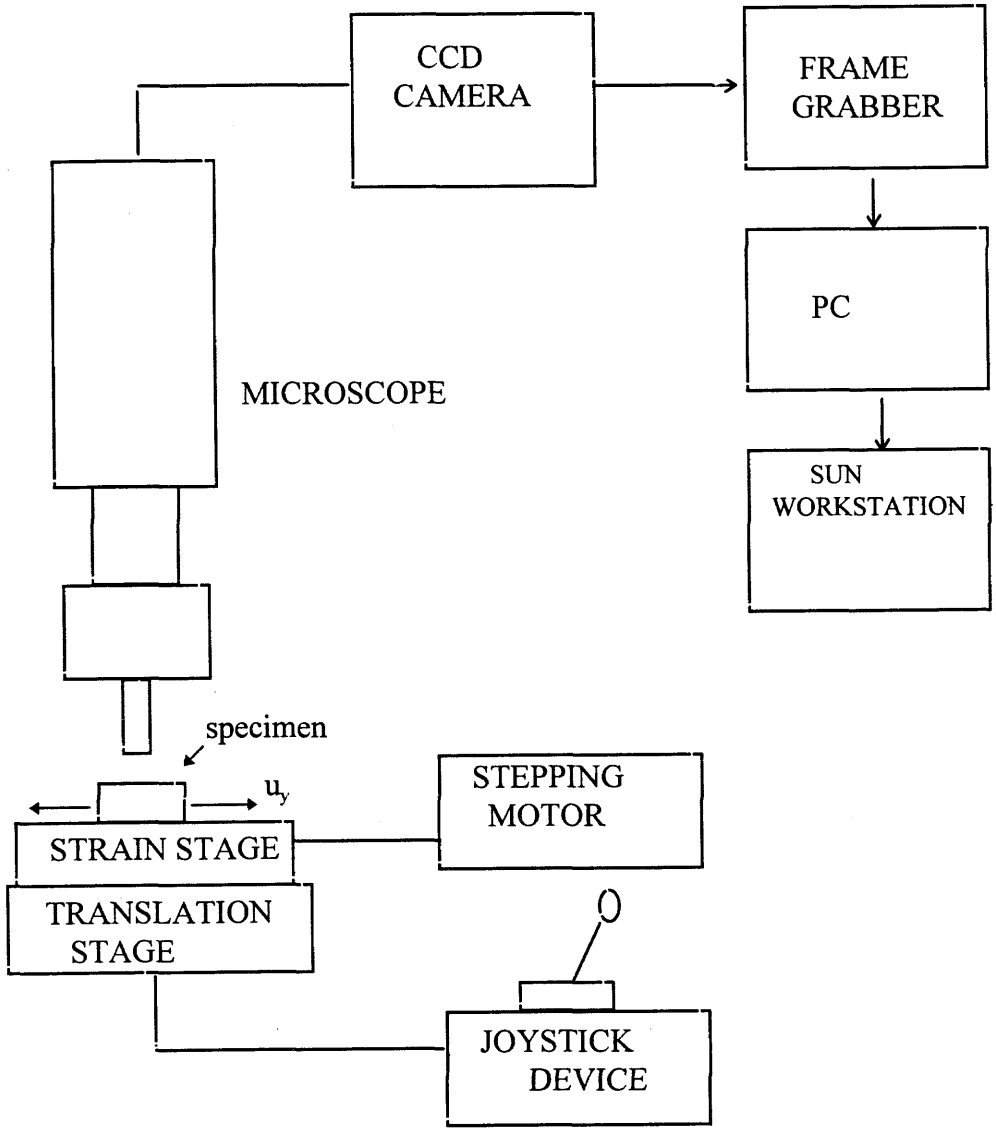


Figure 7. Experimental setup schematic.

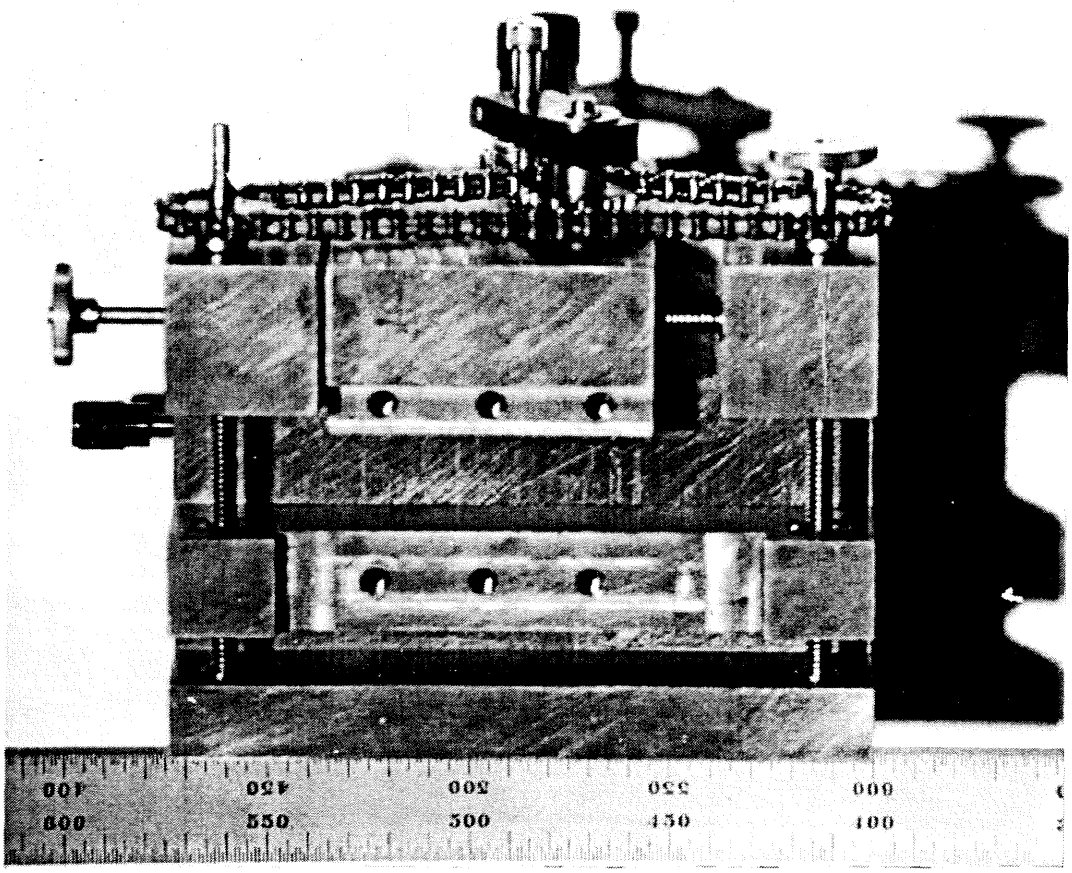


Figure 8. Strain stage.

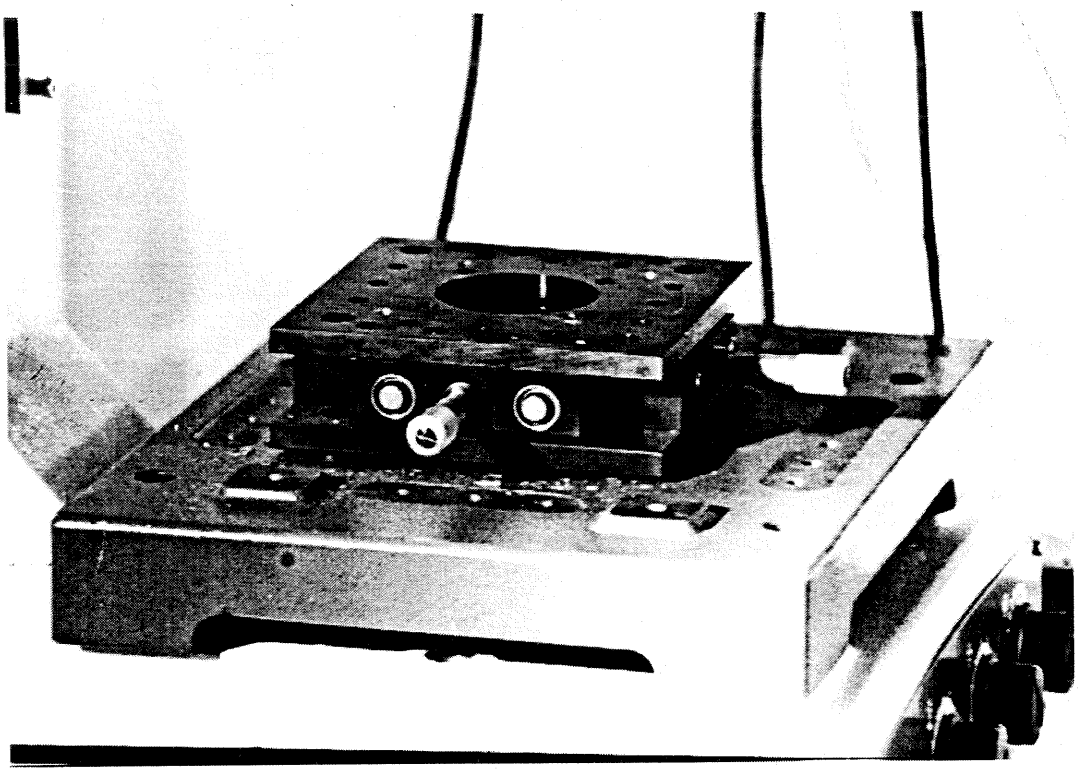


Figure 9. Positioning stage.

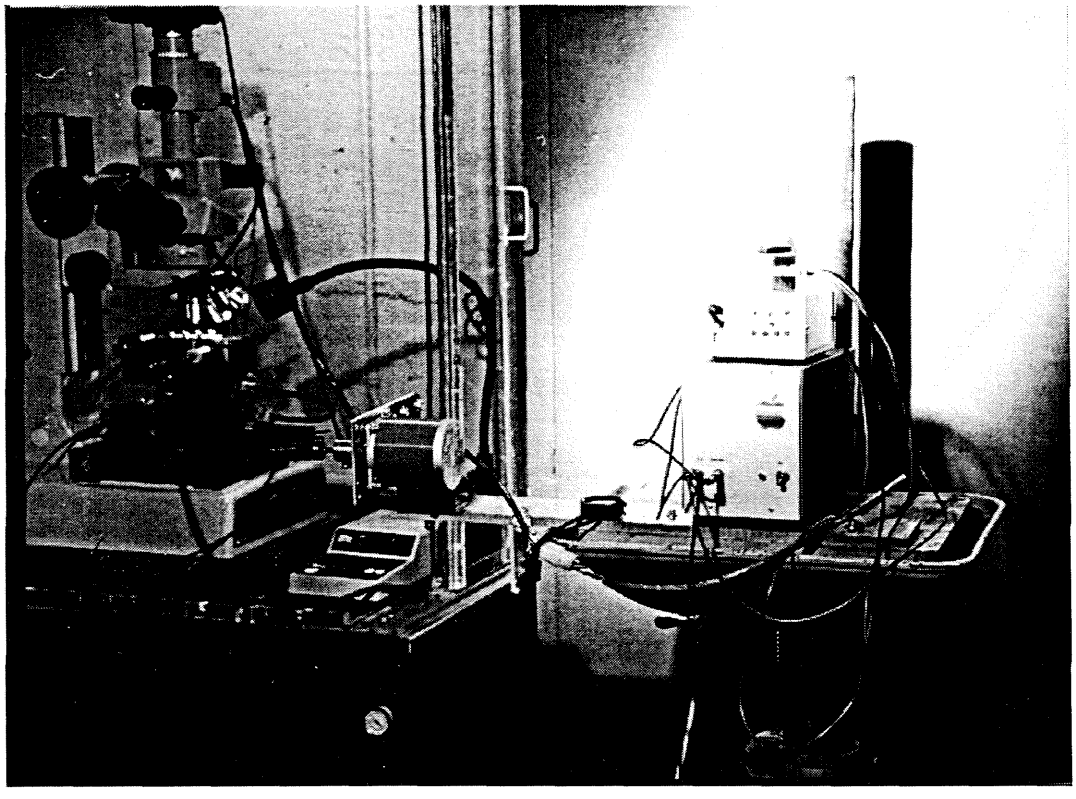


Figure 10. Stepping motor.

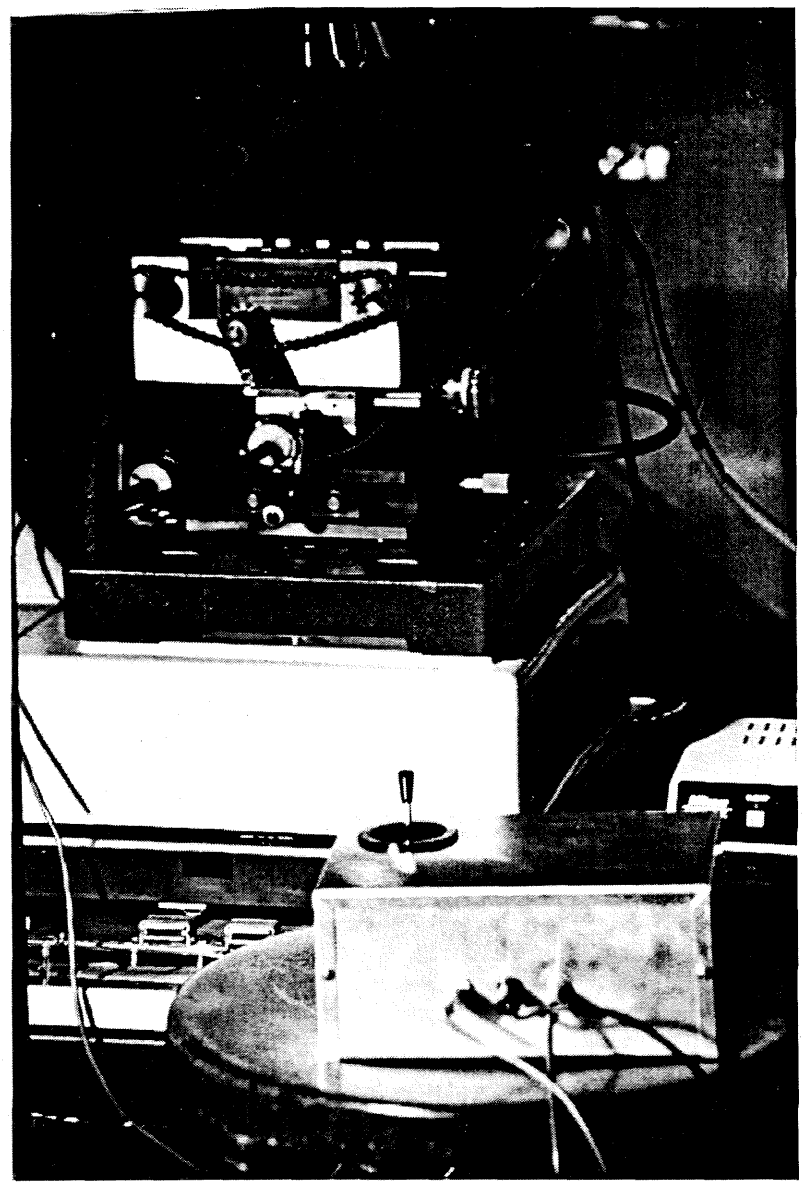


Figure 11. Controller for the positioning stage.

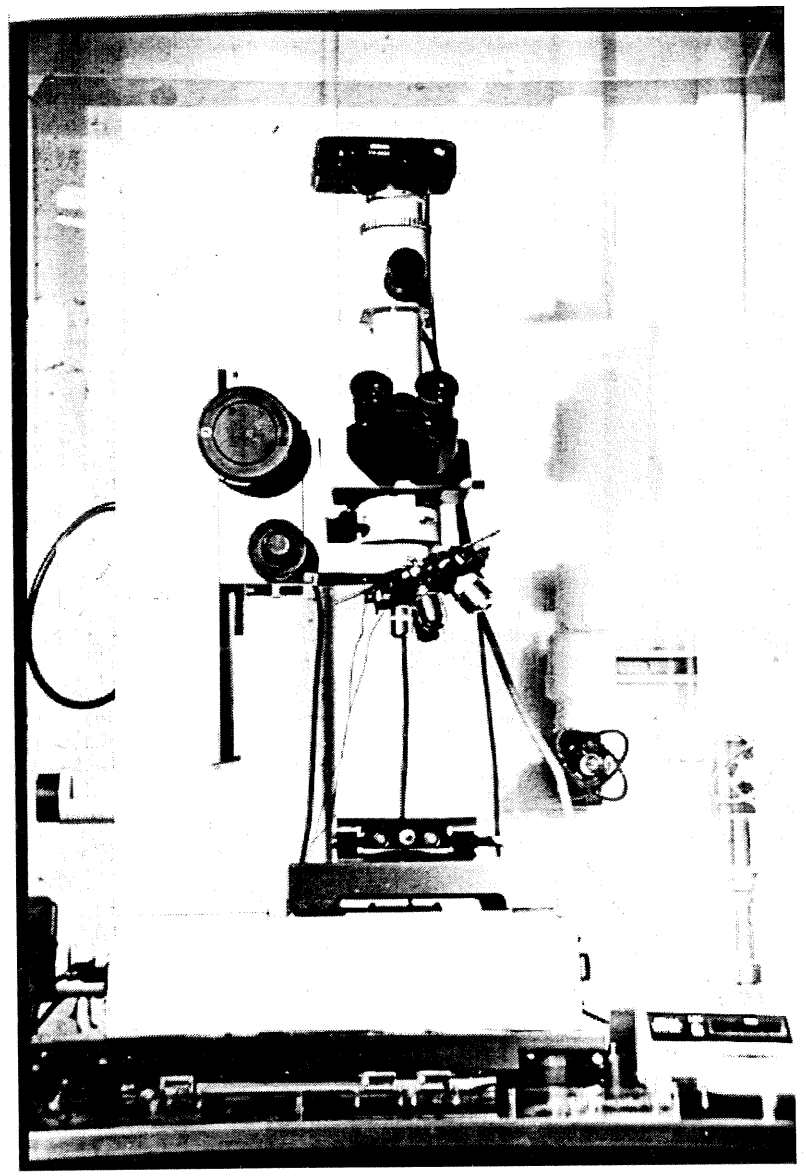


Figure 12. Microscope

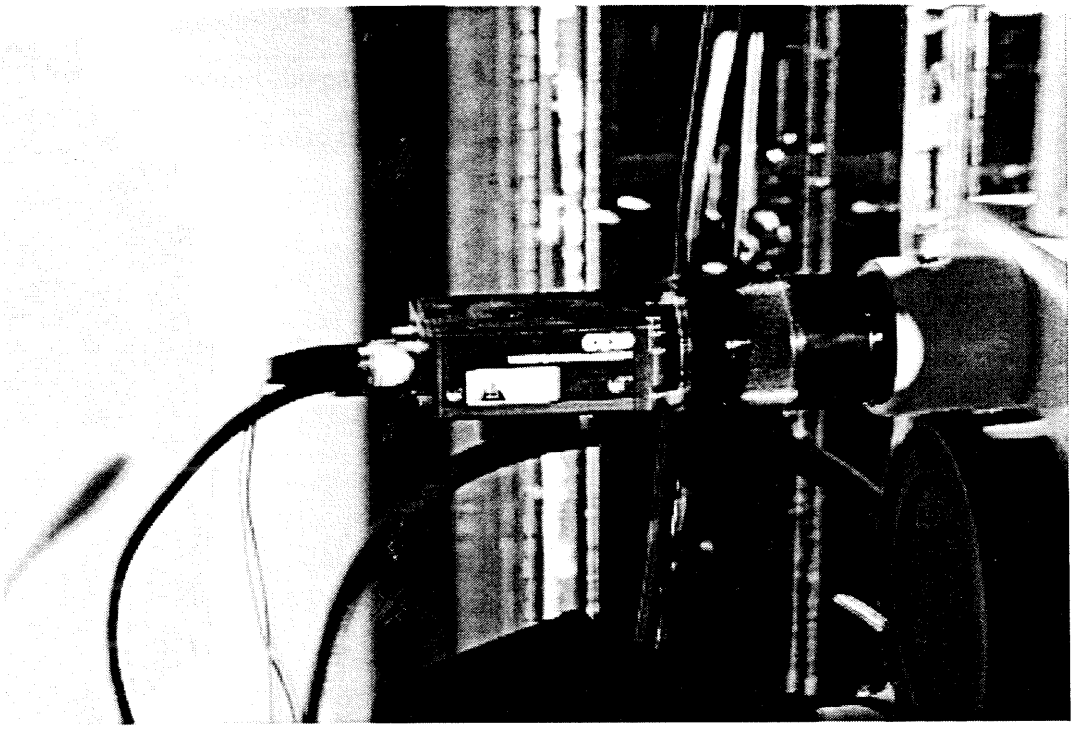


Figure 13. CCD Camera.

3. 2. SOLID PROPELLANT SPECIMEN

The material under study is the solid propellant TPH 1011. This material contains particles of ammonium perclorate, which acts as oxidizer embedded in a rubber matrix that provides the carbon for the combustion. In order to control the rate of burning, the material also contains aluminum particles.

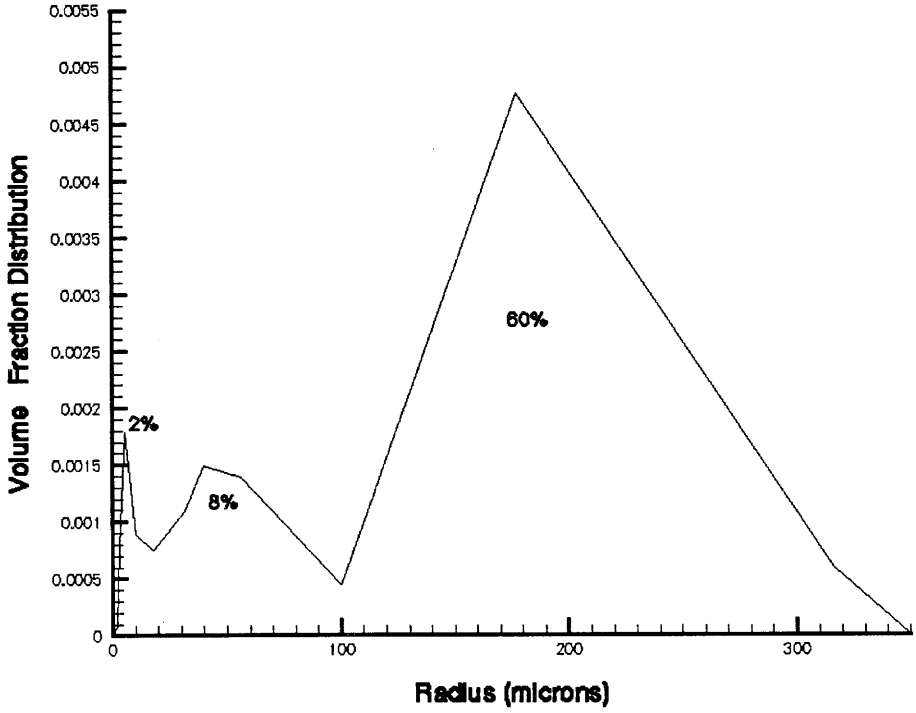


Figure 14. Volume fraction distribution of particles.

For structural analysis, the material is modeled as a viscoelastic particulate composite material with grain size between 10 microns and 600 microns in

diameter. The matrix is a very soft rubber with a Young's modulus of elasticity of 0.1MPa, while the Young's modulus of elasticity of the aluminum particles is 70GPa. The Young's modulii of elasticity of the ammonium perclorate and the aluminum are suficiently large as to model the particles as rigid when they are compared with the rubber matrix. The material is a filled elastomer containing solid particles on a microscopic scale. The volume fraction of the particles is close to 70%. The grain size of the particles is of great importance in interpreting the results obtained by the subsequent experiments.

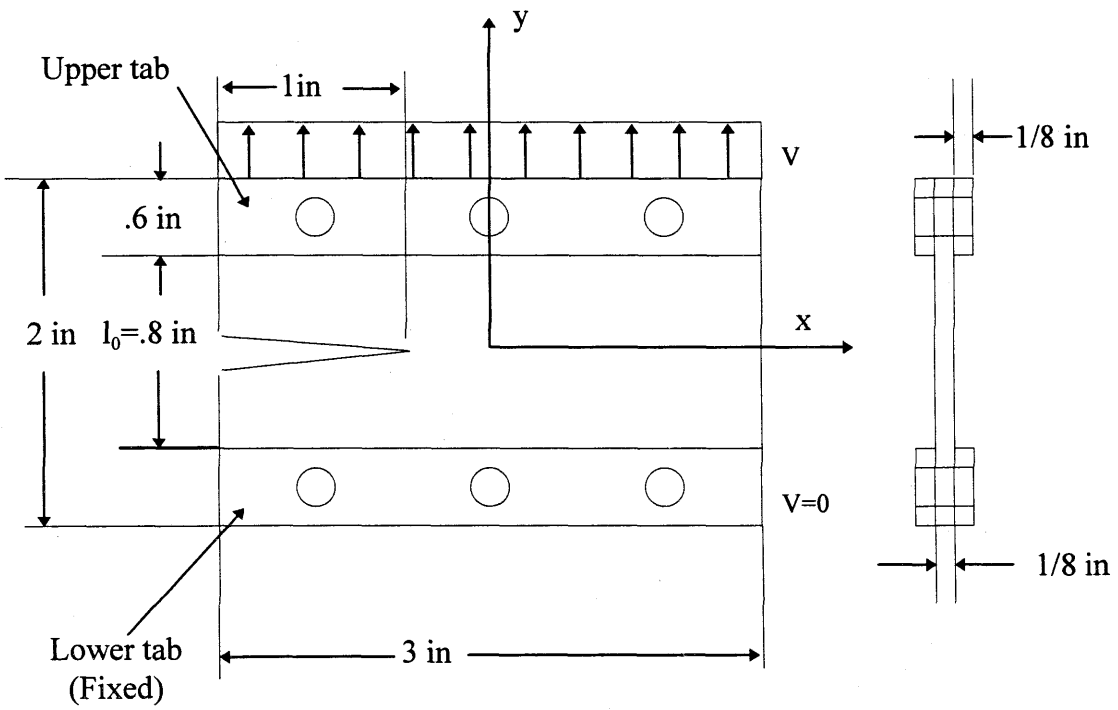


Figure 15. Solid propellant specimen.

To the naked eye, the material looks like dark gray rubber, the texture of which is very similar to erasers at the end of pencils. To study the mode I fracture behavior of this material, 1/8 in thick sheets of the solid propellant are cut into 3 in x 2 in rectangular pieces. Aluminum tabs are attached to the ends to provide a constant displacement boundary condition. The aluminum tabs also ensure that both sides of the specimen remain parallel to each other throughout the deformation (Figure 15). By using a razor blade, a 1 in initial crack is cut in the specimen. This crack opens as the experiment and the crack progress. The surface of the specimen is very irregular under microscopic observation. (Figure 1b in Appendix B). Small dimples of the order of 200 microns in diameter are seen in numbers of 3 to 5 per square millimeter. These dimples are generated during the manufacturing process of the solid propellant sheets. These features play a key role in the fracture process of the material. Most of the damage generated around the crack tip is localized around these dimples.

3. 3 LOADING OF THE SPECIMEN

The strain is applied to the specimen by a prescribed displacement at the boundaries in a manner such that the aluminum tabs always remain parallel. The devices used to load apply the loads to the specimen are:

3. 3. 1. Strain stage

The applied strain is imposed by a straining stage developed at GALCIT (Figure 8). Using set screws, the user can control the displacement of the upper aluminum

tab while the lower aluminum tab remains in its original position (Figure 14). Appendix A contains a set of drawings with the dimension of the strain stage. With the help of a stepping motor[1] (Figure 10), the upper tab velocity can be precisely controlled and therefore the strain rate is accurately prescribed. For the present experiment, the upper tab velocity was set to 0.0008 in/sec. For l_0 equal to 0.8 in (Figure 14), it corresponds to a far field strain of 1%, 2%, 3%, 4% and 5% at times of t = 10 sec, t = 20 sec, t = 30 sec, t = 40 sec and t = 50 sec. These five states are called in what follows steps 1 to 5.

3. 3. 2. Translation stage

As the load is applied to the solid propellant specimen, the position of the observation region relative to the microscope changes. In order to track the same area of the specimen, the position of the specimen under the microscope is controlled by two movable platforms. The first one enables the movement of the specimen in the x and y direction by 10 mm in each direction (Figure 9). It is used to position the crack tip under the microscope before the experiment. This position stage is a Newport Model 405. A second translation stage moves the specimen during the experiment also in the x and y directions. This second position stage is a Newport Model 462. The shift distance of the second stage is 25 mm (1 in) in each direction. As the specimen is loaded, the position of the crack tip moves fast relative to the objective lens of the microscope. For fast and accurate movement of the second translation stage, two electric motors drive this stage as controlled by a GALCIT built joystick device that can be easily used by

[1] The stepping motor used to drive the strain stage is a ASTROSYN Miniangle stepper motor type 34PM-C101. It is capable of applying a torque of 300 in/oz. It is shown in Figure 10.

the operator. The combination of the second translation stage, electric motors and joystick controller are depicted in Figure 11.

3. 3. 3. Translation stage controller

The second translation stage is powered by two 12V electrical motors. They turn two millimetrized screws that control the position of the translation stage in the x and y directions. The motors are controlled by an electronic device operated by a joystick that can prescribe the velocity and direction of the movement of the second stage. The translation stage controller can be seen in Figure 11 and is presented to a greater detail in Appendix E.

3. 4. OBSERVATION AND RECORDING OF THE PROCESS

The process is monitored using an optical microscope, a CCD camera and a frame grabbing unit installed in a PC.

3. 4. 1. Optical microscope

An important feature of the material that determines the length scale of interest is the particle size of the solid propellant. Since this is of 10 - 400 microns in diameter, most of the important characteristics in the fracture process, i.e. inhomogeneous distribution of strains, void formation etc., occurs at this length scale. In addition, the area around the crack tip where significant damage appears during the fracture process has a diameter of about .5 mm to 1 mm. The detection of these features is of main importance for the experiment, and therefore proper

settings in the microscope are chosen to attain the best possible observation. The magnification used for this purpose is 25x. This setting allows a field of view of 3mm x 4 mm, which is optimal for visualizing the addressed features. The microscope used is a Nikon metallurgical microscope HFX - DX (Figure 12).
The light that the microscope provides is not the most adequate for the experiment since it illuminates the specimen from a direction perpendicular to the specimen plane. The images this way obtained don't have enough gray level contrast for the DIC program to work successfully. An alternative method for illuminating the specimens was used with the aid of two halogen lamps. They were manufactured by Sunnex and had a model #710. The two halogen lamps illuminated the specimen from opposite sides of the microscope, at an angle of 30° with respect to the horizontal. With this latter illumination system, the surface roughness could be better observed and therefore, the correlation process between successive images worked better.

3. 4. 2. CCD camera

During the fracture process, the images of the deformed specimen are recorded by a CCD camera installed at the rear side of the microscope (Figure 13). This camera is a Sony XC - 75. The images are digitized and stored as black and white tiff files of 640x480 pixel size and 256 gray levels.

3. 4. 3. Frame grabbing unit and PC

The QuickCapture Human Interface Version V01.03 frame grabbing unit is used to transform to tiff files the images acquired by the CCD camera. This frame

grabbing unit is manufactured by Data Translation, inc. and it is programmed to acquire one image of the deformation process every ten seconds. The unit is installed in the terminal "moscow" . This terminal has a set of cables in the right side of the monitor, out of which the one with label 7 is connected to the CCD camera output.

3. 4. 4. Digital image processing

The tiff image files are transferred to a sun workstation via FTP in order to perform the correlation process. The Large Deformation Digital Image Correlation method is sequentially used for the purpose of finding the Lagrangian strain distribution for a given deformation. The details necessary for obtaining the strain maps for the deformations corresponding to 1%, 2%, 3%, 4% and 5% far field strain are presented in Appendix C.

4. RESULTS AND DISCUSSION

In this chapter, characteristic results obtained by using the Large Deformation Digital Image Correlation method and the experimental procedure described in the previous section, are presented. First, a section is devoted to discussing the inhomogeneity of the solid propellant material. Second, a set of Principal Lagrangian strain distributions for a crack opening deformation are depicted. Also a relation between the crack path during the deformation and the distribution of the strains before crack propagation is provided. Finally, the stress-strain curve of the material is discussed.

4. 1. INHOMOGENEITY OF THE MATERIAL

For structural analysis, solid propellant is modeled as a (particulate) viscoelastic (composite) material. A great number of particles are embedded in a rubber matrix, where the grain size is between 10 and 400 microns in diameter, and the volume fraction 70%. To investigate the inhomogeneity of the material a simple uniaxial test is performed on a specimen of solid propellant without any cracks. Using the straining device, uniaxial tension in the y direction is applied to the specimen (Figure 16).

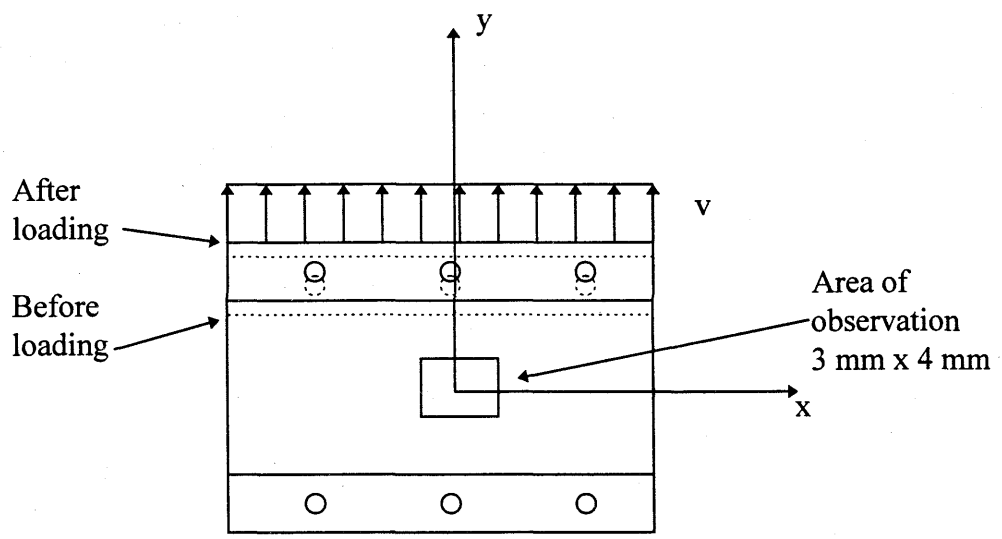


Figure 16. Straining of an uncracked specimen shown before and after loading.

Macroscopically, the specimen is deformed to a Lagrangian strain, $E_{yy} = 0.015$, as calculated from the movement of the upper aluminum tab. The result from the Digital Image Correlation program is an inhomogeneous distribution of strains, with the Lagrangian component E_{yy} shown in Figure 17 . When the material is deforming, the strain map reveals the existence of areas of the order of 400 microns diameter, in which the distribution of the strains is nearly homogeneous and approximately zero. In these regions, the strain is nearly zero, implying that the region acts as a rigid inclusion during the deformation. Some of these regions are as big as 0.5 mm in diameter, as is the case of the area in the lower right corner at (2.2 mm, 0.6 mm) and in the upper left corner. Other rigid areas have a

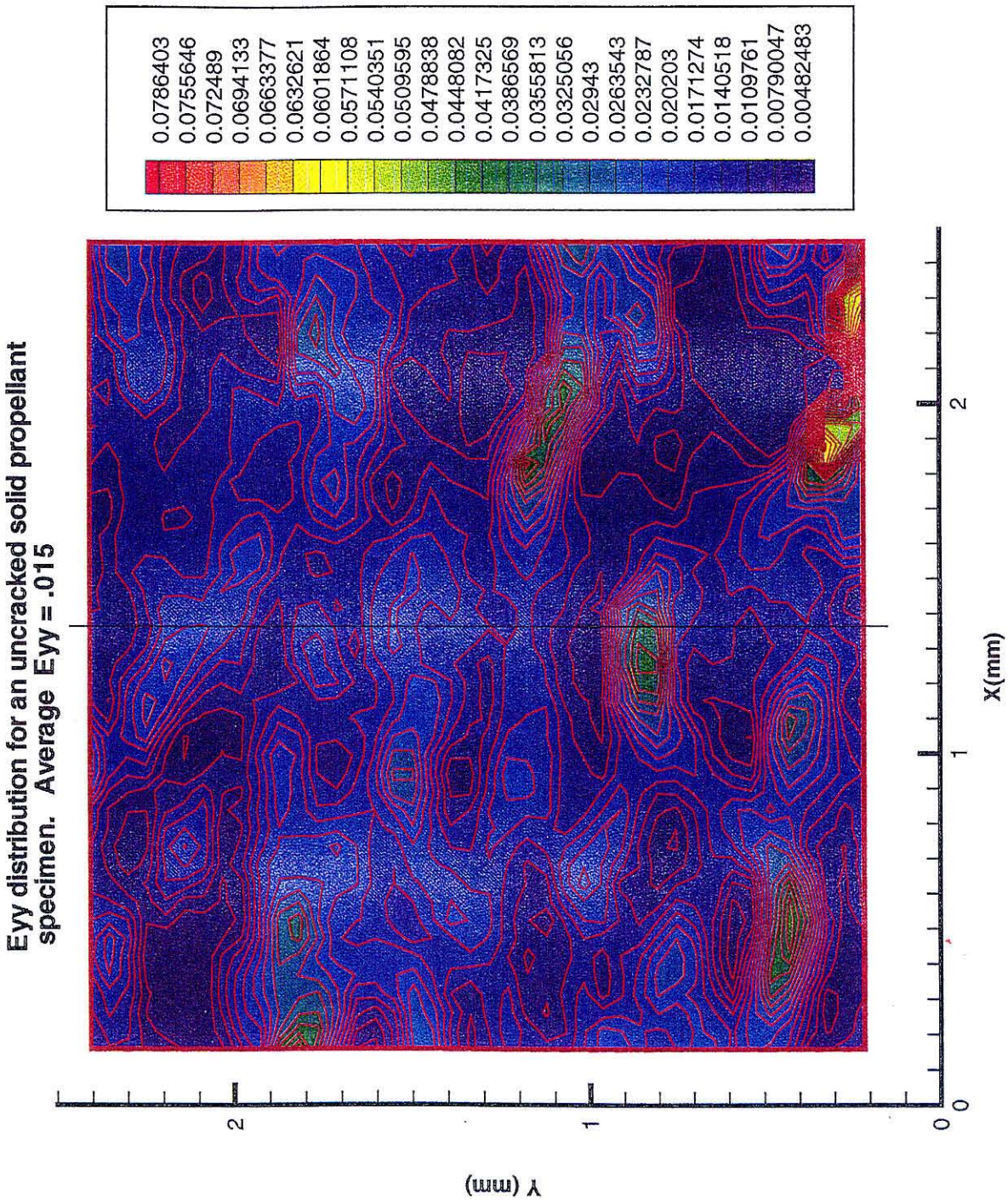

Eyy distribution for an uncracked solid propellant specimen. Average Eyy = .015
0.0786403
0.0755646
0.072489
0.0694133
0.0663377
0.0632621
0.0601864
0.0571108
0.0540351
0.0509595
0.0478838
0.0448082
0.0417325
0.0386569
0.0355813
0.0325056
0.02943
0.0263543
0.0232787
0.020203
0.0171274
0.0140518
0.0109761
0.00790047
0.00482483
Y (mm)
X(mm)
0
1
2

Figure 17.

medium size, like the rigid area located at (1.3 mm, 1 mm), the rigid area at (0.5 mm, 0.5 mm) and the rigid area at (2.1 mm, 1.3 mm). Other small rigid areas appear in great number. It is interesting to observe that in the vicinity of the bigger rigid areas, regions of largest strain localization occur, for example at the bottom and top of the rigid area located at (2.2 mm, 0.6 mm). The rigid area located at (1.3 mm, 1 mm) gives rise to large gradients in the strain distribution underneath it. This is observed by the proximity of the contour lines in the area. Other rigid areas ((0.5 mm, 0.5 mm), (2.1 mm, 1.3 mm), (0.4 mm, 2.2mm)) also show this feature. These rigid areas share a common size of more than 300 microns in diameter. Other smaller rigid areas do not develop these large deformation areas around them.

Another phenomenon of interest is how clusters of rigid areas that are aligned in the direction of the load develop a high strain area between them. Examples of these high strain areas are seen at (2 mm, 1.2 mm), (0.5mm, 0.4mm) and at (0.2 mm, 1.9 mm), where over and underneath the high strain areas are rigid areas. Some insight on the strain distribution can be gained by presenting the strain distribution along a line. If we draw a line (x=1.36 mm) across Figure 17 five rigid domains are crossed at (y=0.2mm), (y=0.5mm), (y=1mm), (y=1.85mm) and (y=2.35mm). At (y=1.65mm) there is another minimum in the strain level that corresponds to a smaller rigid region. The strain pattern along this line is shown in Figure 18. We observe that for a macroscopic strain level of E_{yy}= 0.015 the distribution of strains takes values as high as E_{yy}= 0.04 and as low as E_{yy}= 0.0025. If we assume that the particles are practically rigid when compared with the matrix, a rough calculation can be done to predict the strain concentration in the matrix material for the case where the grain volume fraction is 70%.

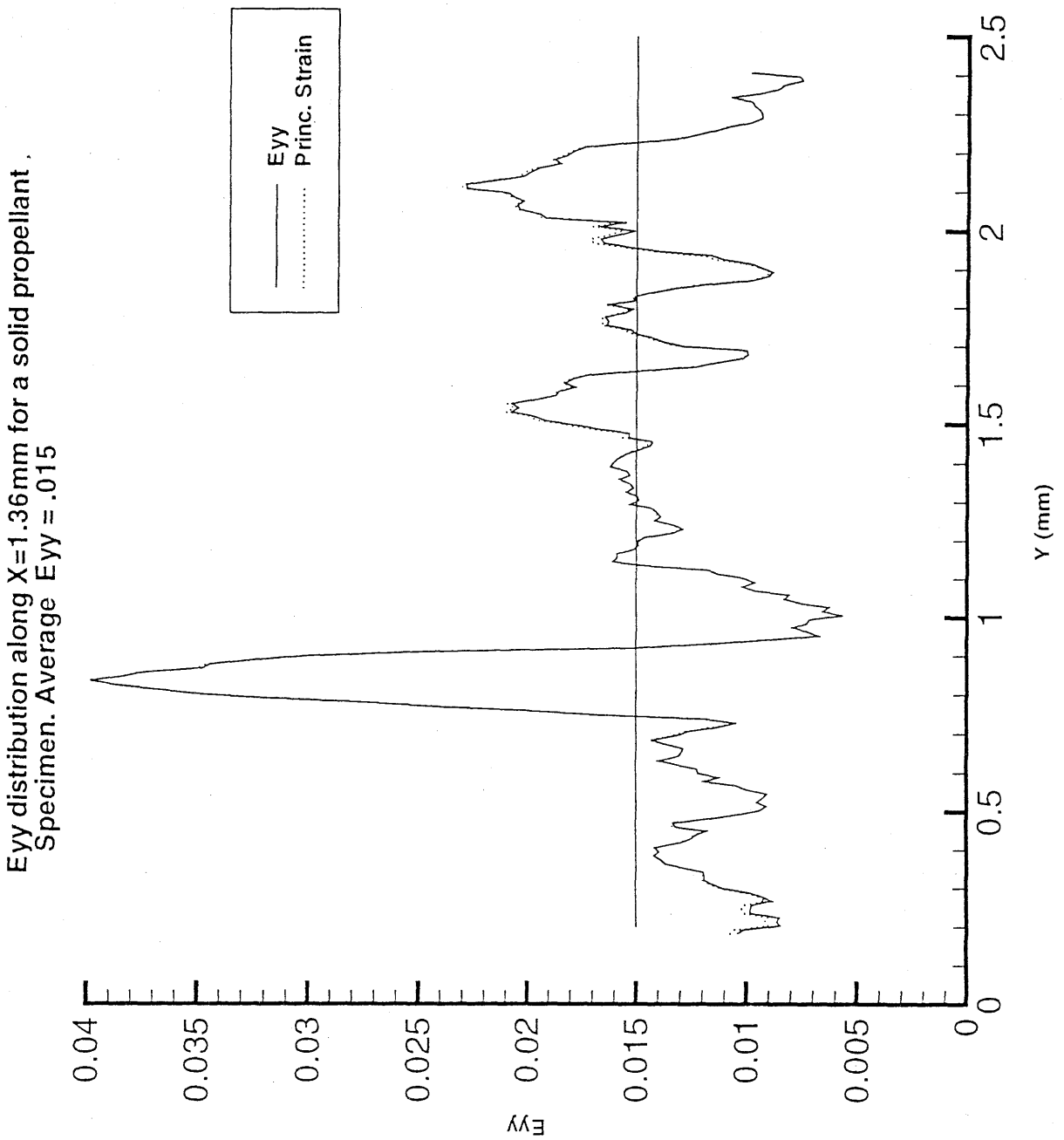

Eyy distribution along X=1.36mm for a solid propellant
Specimen. Average Eyy = .015
Eyy
Princ. Strain
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
Eyy
0
0.5
1
1.5
2
2.5
Y (mm)

Figure 18.

In this case, all the deformation is taken by the matrix. When the volume fraction of particles in the material is computed, it is assumed that a large volume is considered. Under the same assumptions, the average line that crosses the domain occupied by the material, has 70% of its length crossing particle material and 30% crossing matrix. This can be proved by considering an imaginary cylinder center around the line and shrinking its radius close to zero. This means that for a strain of 1% along the line considered, the matrix would have a deformation of 3.3% strain. This corresponds to a 3.3 strain concentration factor. For our case, the macroscopic strain along the line is 1.5% (0.015 strain). The strain in the matrix is 4.9% (0.049 strain) in this case while the particle material remains undeformed. Comparing this rough calculation with the results presented in Figure 18, it is concluded that they are in good agreement, being the strain in the matrix obtained by the experiment 4% (.04 strain).

The average value of the Lagrangian strain component in the y-direction taken over all the points in the contour plot is 0.015.

4. 2. LAGRANGIAN STRAIN DISTRIBUTION AROUND THE CRACK TIP IN SOLID PROPELLANT TPH 1011

In this section, a cracked specimen of solid propellant TPH 1011 is loaded in the direction perpendicular to the crack tip. The load is applied as a prescribed displacement of the specimen boundaries, as seen in Figure 19. The strain rate is constant during the deformation and equals 0.001 1/sec. A series of eight images (Figures 20 to 26) is taken where the far field Lagrangian strain equal to 0%, 1%, 2%, 3%, 4%, 5% and 6%.

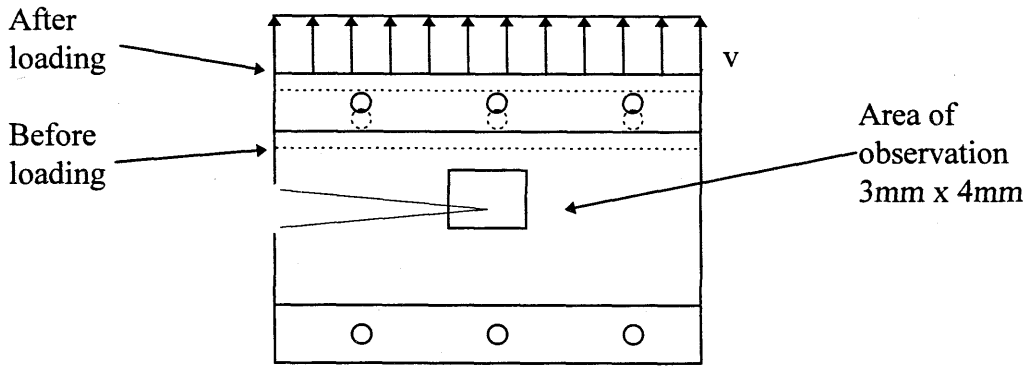


Figure 19. Straining of a cracked specimen before and after loading.

The Large Deformation Digital Image Correlation method (LD - DIC) is used to obtain the strain distribution within 1 mm of the crack tip. Five cases are considered, namely cases for deformations where the far field Lagrangian strain is 1%, 2%, 3%, 4% and 5%.

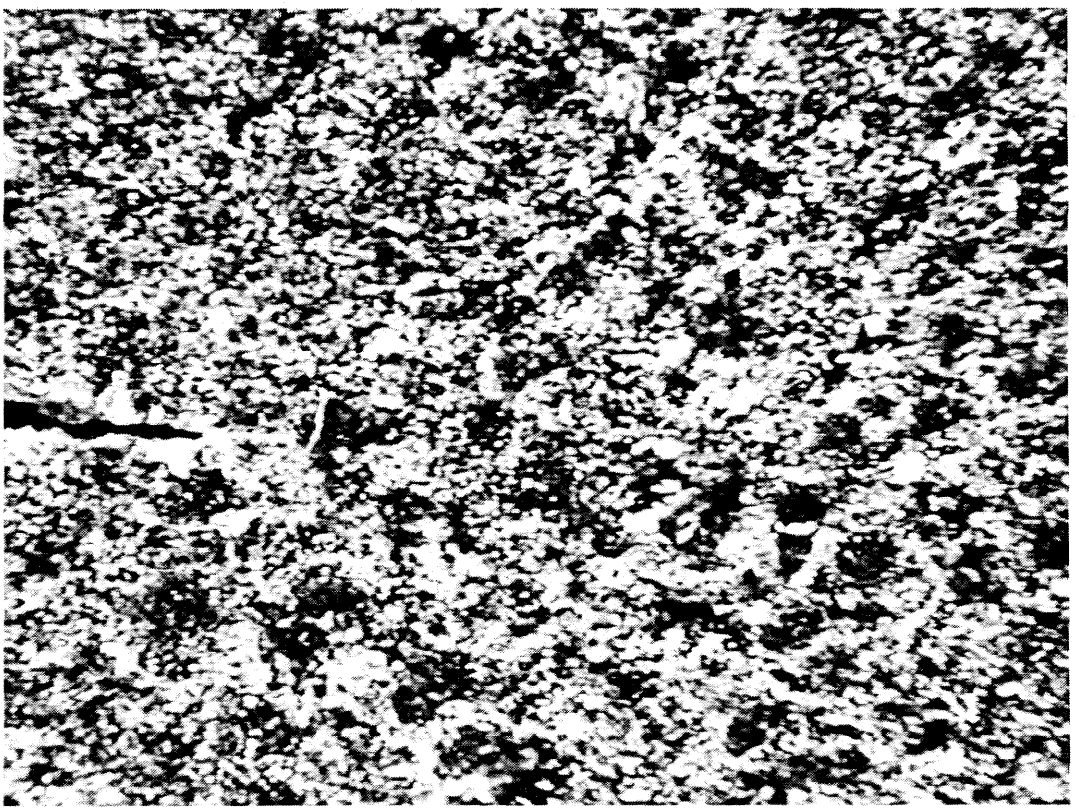


Figure 20. Tiff file of deformation at 0%.

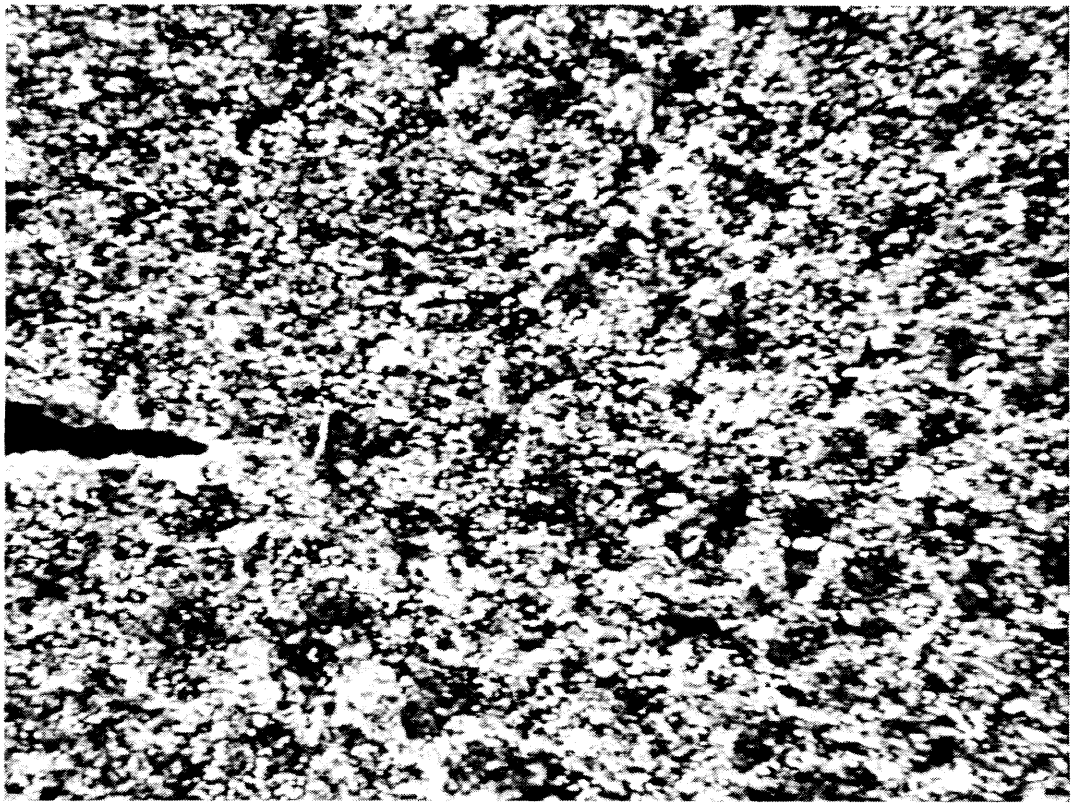


Figure 21. Tiff file of deformation at 1%.

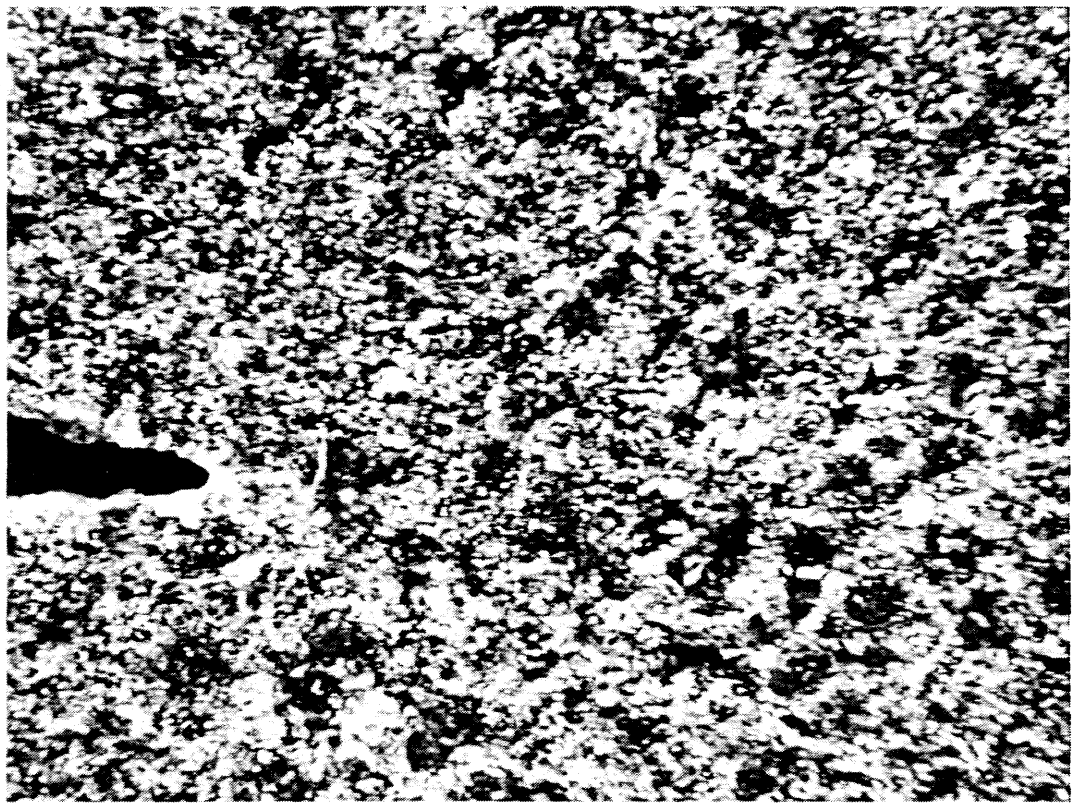


Figure 22. Tiff file of deformation at 2%.

Figure 23. Tiff file of deformation at 3%.

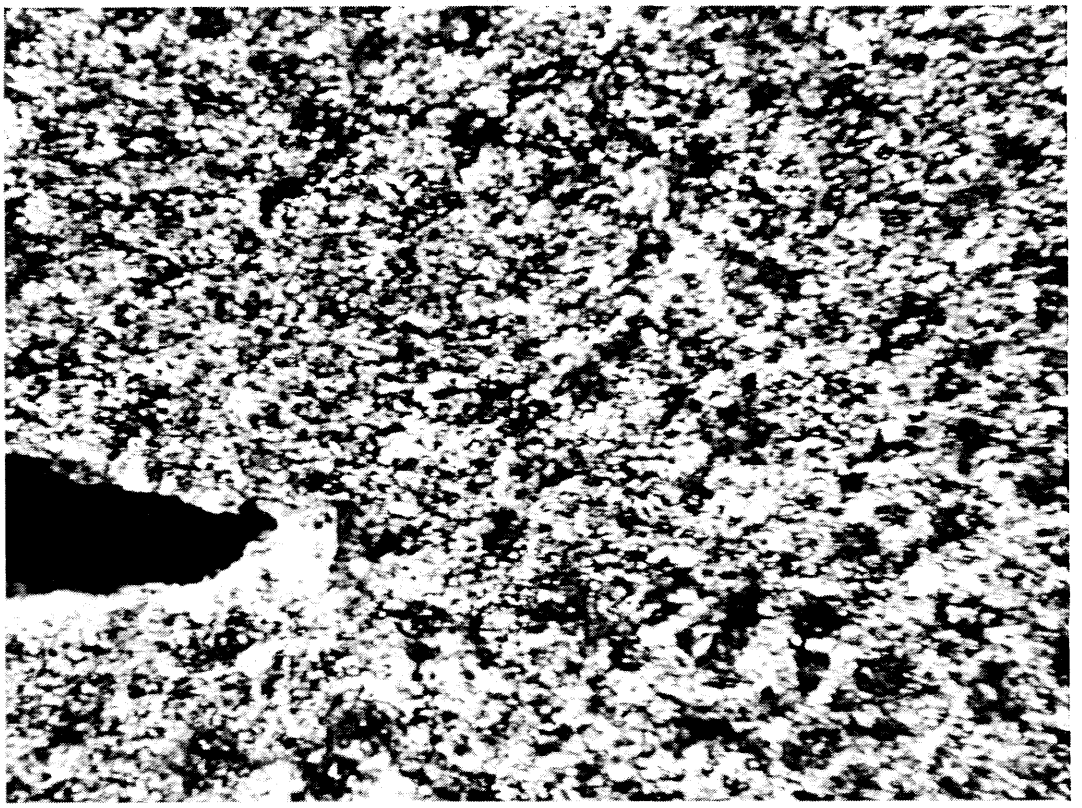


Figure 24. Tiff file of deformation at 4%.

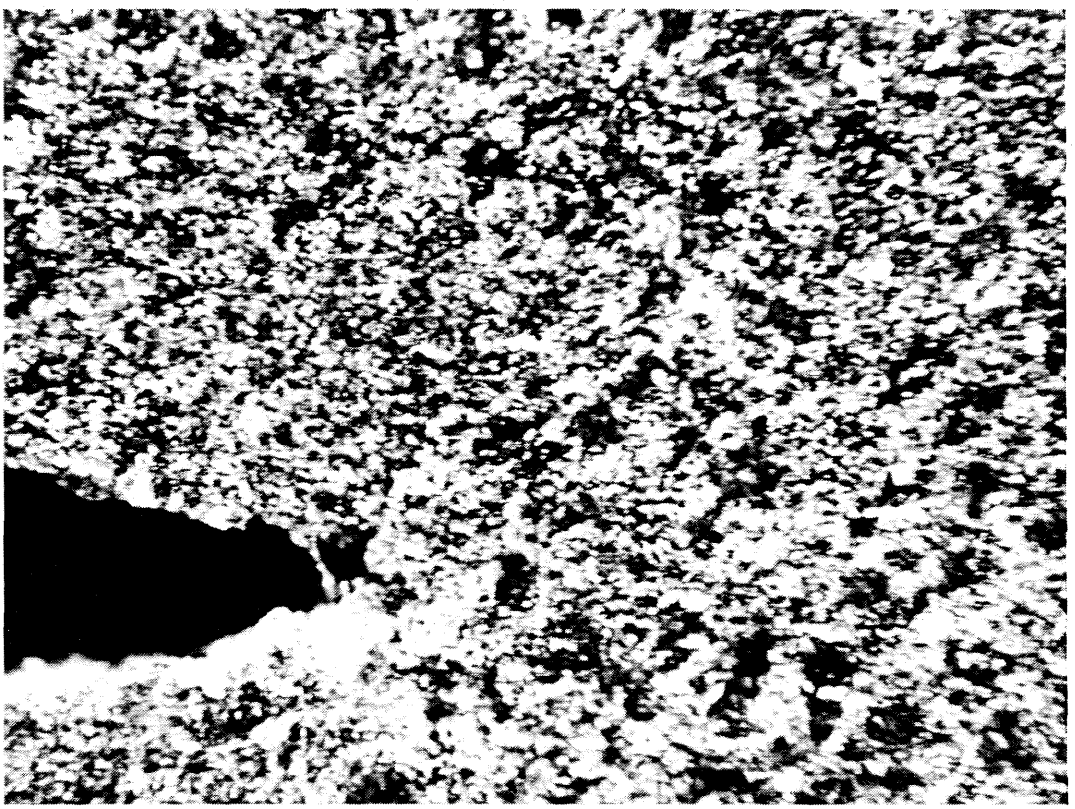


Figure 25. Tiff file of deformation at 5%.

Figure 26. Tiff file of deformation at 6%.

The results obtained using the LD - DIC are presented as color contour plots representing the maximum and minimum principal Lagrangian strains.

Two color codes have been chosen to represent the strain levels, one for all the maximum principal strain plots and the other for all the minimum principal strain plots. The color codes have been scaled to represent the deformation in the fifth step to the greatest detail. Because of this, the contour plots in the first steps do not have a very large contrast. In the maximum principal strain plots there are 25 strain levels represented by colors ranging from dark blue (0% strain) to red (24% strain). In the minimum principal strain plots the scale goes from dark blue (-1% strain) to red (6% strain). This way the growth of the strain levels during the five steps of the deformation can be observed. Also the principal directions are represented as small line segments at every point where the correlation has been performed. In the contour plots corresponding to maximum principal strains, the line segments point towards the maximum principal directions. For the contour plots corresponding to minimum principal strains, the segments point towards the minimum principal direction. The data is presented as principal strains rather than the usual components of the Lagrangian strain (E_{xx}, E_{yy} and E_{xy}), because the crack opening and void formation depend on the local principal strains. This representation method has also the advantage of presenting the directions of maximum deformation at every point.

Only the area of the deformation where the correlation program was successful in producing the correct results is depicted in the contour plots. An area surrounding the crack does not have deformation information. It is represented as a white area and it does not represent the current crack edge after propagation but it is a consequence of the LD - DIC method. As the load level increases from 1% far

field strain (Step 1) to 5% far field strain (Step 5) the area around the crack tip where the information is missing grows bigger. This is caused because every time the LD - DIC method is used to construct a step of the deformation, several rows of information are lost due to the interpolation scheme. Also part of the information around the crack tip is lost because the Digital Image Correlation program doesn't converge in places where new geometrical features appear. The cases of crack propagation and void formation are examples in which black areas suddenly appear and the Digital Image Correlation program does not work. These two later cases can nevertheless be studied by the increase in strain in the region before the features occur.

A Lagrangian description of the deformation is used in the contour plots for the five steps presented. This representation helps to study the strain increase in selected regions.

The Lagrangian representation is more convenient also to represent the crack propagation path and void locations in the undeformed configuration. The crack propagation path is constructed by mapping the upper crack edge, as seen in the image corresponding to the 6^{th} step of the deformation, into the undeformed image. The part of the crack propagation path that has already open is depicted in black color whereas the remaining part of the crack is depicted in gray.
During the 5^{th} step of the deformation there is a void that appears. The contour of the void in the image of the 5^{th} step of the deformation is also projected to the undeformed configuration. The crack propagation path is represented in all the contour plots as a black or gray poly-line going from the crack tip towards the

right. The void is represented also in all the contour plots as a trapezoidal black or gray poly-line around the position (1.1mm, 1.2mm).

One feature of interest that is visible in the contour plots of the deformation is that there is a circular area with 0.5 mm radius around the crack tip where most of the deformation localizes. Figure 27, which shows the maximum principal Lagrangian strains and directions for the first step of the deformation, shows how the deformation around the crack tip is localized in two lobes located on top of each other. The lobe located in the top (Upper lobe) is centered at the crack tip, which in the first three steps of the deformation is at (0.7mm, 1.2mm). This lobe reaches an extreme of more than 12% maximum principal Lagrangian strain and has a radius of 150 microns. The Lower lobe has a similar size and shape. It is centered on the position (0.7mm, 1mm), where it reaches a maximum principal Lagrangian strain of 5%. Surrounding the two lobes there is a region of more than 3% maximum principal Lagrangian strain that extends towards the bottom of the image area. This later region covers the area where the void develops in the fifth step of the deformation. In the area where the void develops, the directions of principal strain are in the horizontal direction, indicating that the void opens in the horizontal direction. This void is visible in Figure 25, that represents the image of the deformation during the fifth step (5% far field strain). Other areas where the deformation concentrates are Area 1, located at (1mm, 2.3mm) and Area 10 (0.9mm, 0.5mm). Out of these five high strain regions(Upper lobe, lower lobe, void region, Area 1 and Area 10), only the upper lobe and the region where the void develops are directly related to the propagation of the crack in subsequent steps.

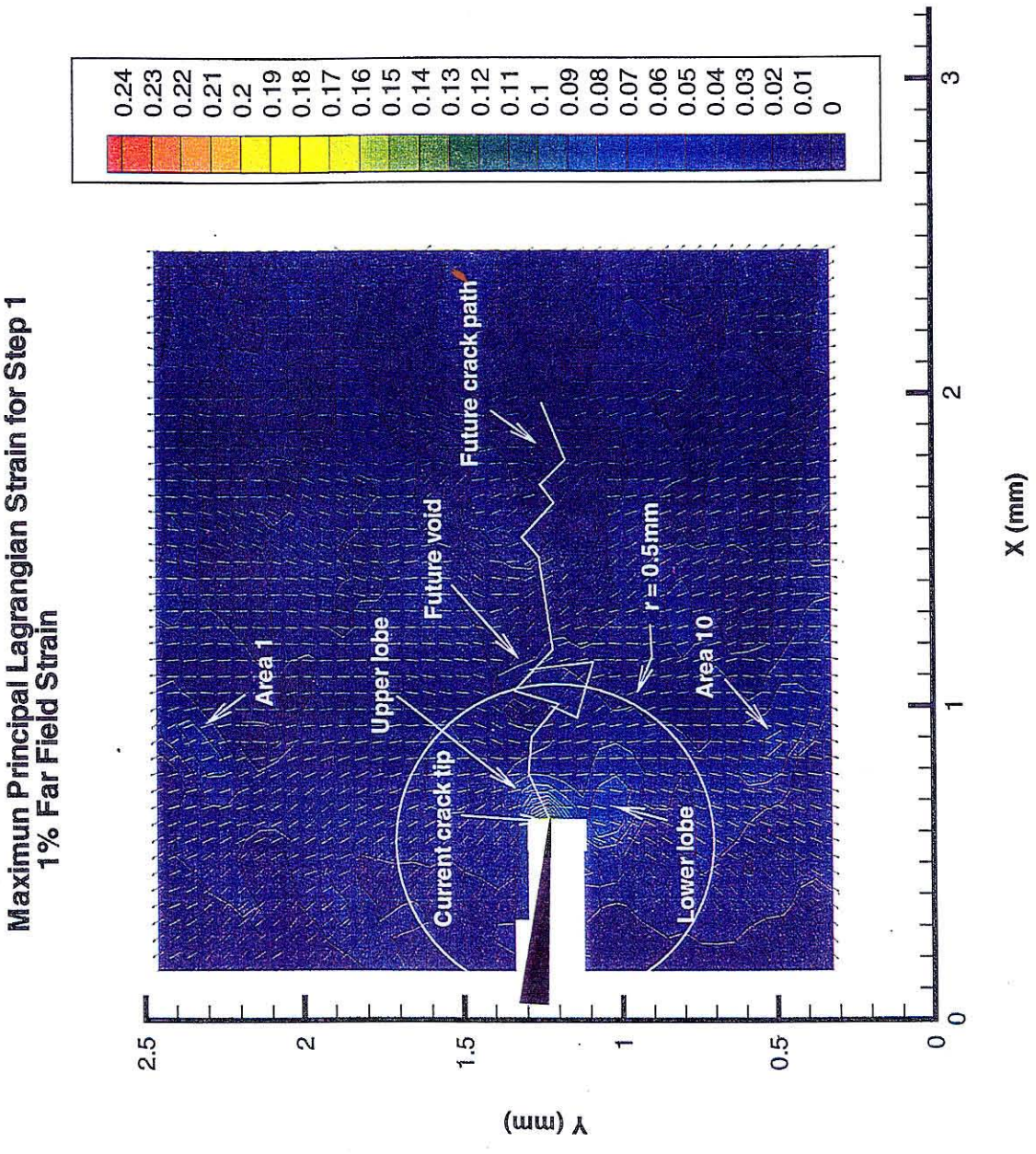

Maximun Principal Lagrangian Strain for Step 1
1% Far Field Strain
0.24
0.23
0.22
0.21
0.2
0.19
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Area 1
Upper lobe
Future void
Future crack path
r = 0.5mm
Area 10
Current crack tip
Lower lobe
2.5
2
1.5
1
0.5
0
Y (mm)
0
1
2
3
X (mm)

Figure 27.

These two regions are within 0.5mm of the crack tip. Note that the upper lobe coincides with the first 0.15mm segment of the crack propagation path. This segment corresponds to the propagation of the crack from the third to the fourth step of the deformation. The orientation of the upper lobe corresponds with the direction of this first crack propagation segment. The strain in the other high strain regions, i.e. area 1 (1mm, 2.3mm) and area 10 (0.9mm, 0.5mm), grows with the far field strain level until the crack has pass them. In the deformation considered, these two areas (area 1 and area 10) did not develop any visible voids.

The strain distribution in Figure 27, corresponding to the maximum principal strain in the first step (1% Far field strain), is very similar to that in Figure 28, corresponding to the maximum principal strain in the second step (2% Far field strain). The positions of the upper and lower are the same in both figures. The maximum values of the maximum principal strain are in the second step 18.5% and 11% for the upper and bottom lobes respectively. Other high strain areas like area 10 at (0.9mm, 0.5mm) and area 1 at (0.9mm, 2.3mm) have very similar shape but their strain levels are close to twice that of their strain level in the first step. It is of interest to note that the upper lobe coincides in position and orientation with the first segment of the crack propagation path, as in the maximum principal strain contour plot for the first step. The distribution of strains in the upper lobe indicates that the crack is going to propagate through it. The main deformation associated with future crack propagation is located within 0.5 mm radius from the crack tip as in the first step.

In Figure 29 representing the maximum principal strains and directions for the third step (3% Far field strain) the crack tip is stationary in the same position as in the previous two steps.

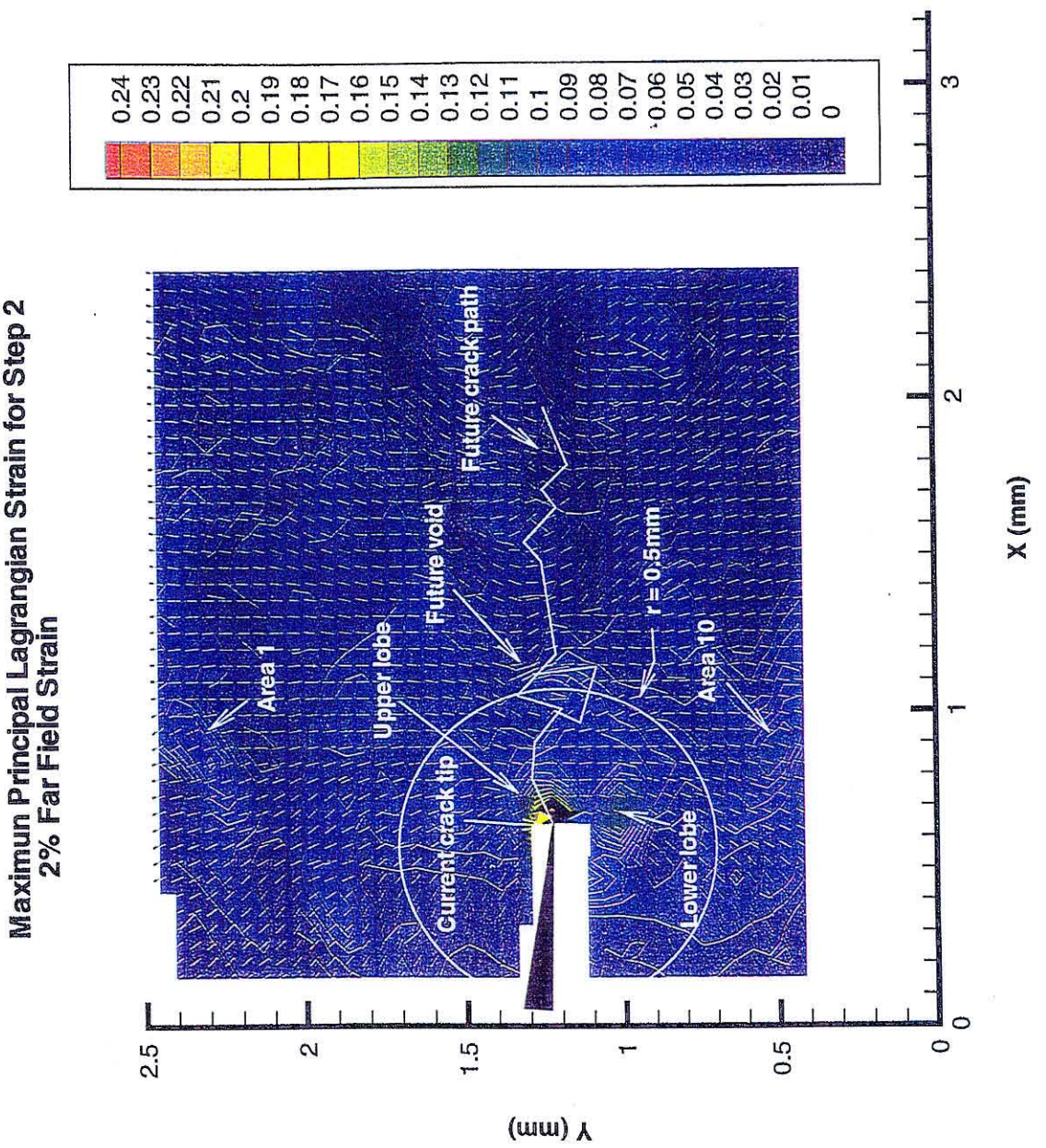

Maximun Principal Lagrangian Strain for Step 2
2% Far Field Strain
0.24
0.23
0.22
0.21
0.2
0.19
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Area 1
Upper lobe
Future void
Future crack path
Current crack tip
r = 0.5mm
Area 10
Lower lobe
2.5
2
1.5
1
0.5
0
Y (mm)
0
1
2
3
X (mm)

Figure 28.

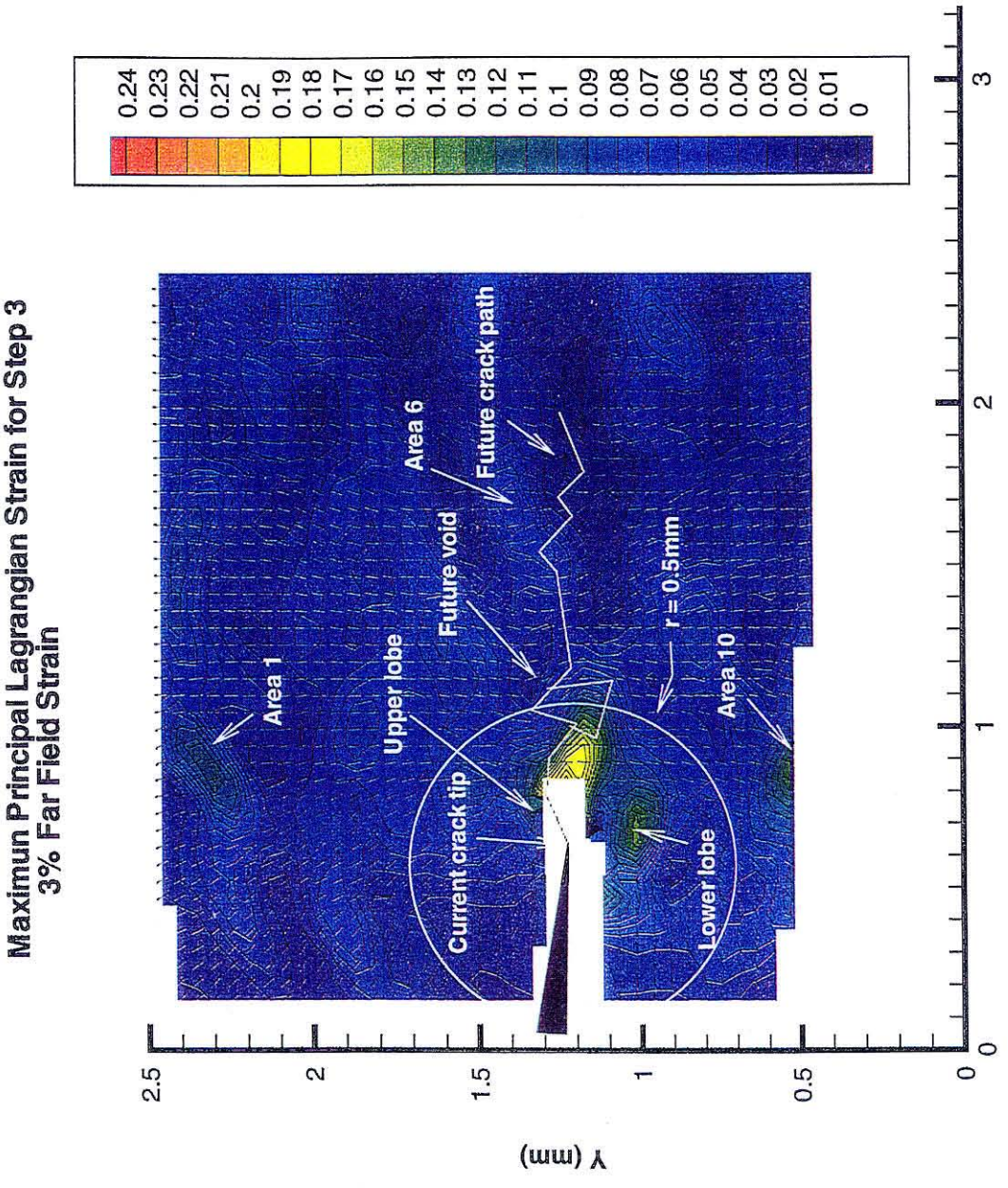

Maximun Principal Lagrangian Strain for Step 3
3% Far Field Strain
0.24
0.23
0.22
0.21
0.2
0.19
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Area 1
Upper lobe
Current crack tip
Future void
Area 6
Future crack path
r = 0.5mm
Area 10
Lower lobe
Y (mm)
X (mm)
2.5
2
1.5
1
0.5
0
1
2
3

Figure 29.

Part of the image around the crack tip is lost not because crack propagation but because of the interpolation scheme in the LD - DIC process. The strain distribution within 0.5mm of the crack tip is noticeably different from that in the second step. The upper lobe that was centered on the crack tip in the first and second step, has moved to cover the area where the void develops (1.1mm, 1.2mm). A high strain region is created this way along the crack propagation path reaching the trapezoidal poly-line that represents the void.

The crack has propagated a distance of 0.15mm in the fourth step of the deformation, corresponding to 4% far field strain (Figure 30). Because of the propagation of the crack, tension is released from the material near the crack faces. An example of this relaxation is seen in the lower lobe, located at (0.7mm, 1mm), where the strain has been reduced to 12% in the fourth step from 14% in the third step.

In the third step, Area 6 is an isolated high strain region. During the fourth step, the strains between Area 6 and the high strain region located where the void develops increase so that the two high strain regions become one. The connection between these two high strain regions is developed along the crack propagation path at (1.4mm, 1.2mm). Nevertheless, the strain distribution in this step indicates that the crack will propagate through the high strain region Area 6 (1.7mm, 1.3mm), which is not true. At (1.55mm, 1.3mm) the crack deviated from this high strain region to take another path. At this point it is important to note that the distance from the crack tip to the high strain Area 6 at (1.7mm, 1.3mm) is very large (0.9mm). The strain distribution around Area 6 may change significantly by the time the crack tip reaches it.

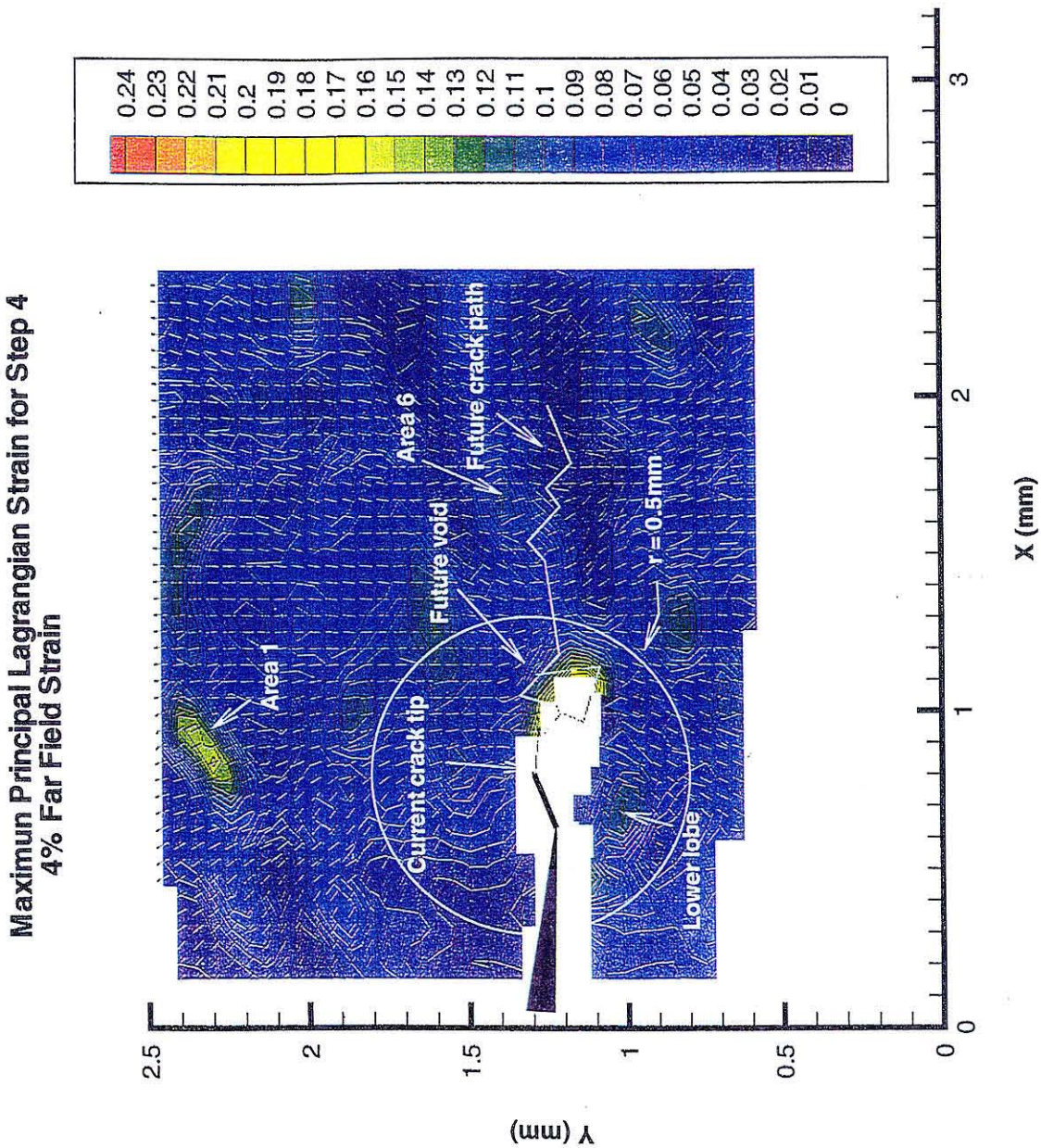

Maximun Principal Lagrangian Strain for Step 4
4% Far Field Strain
0.24
0.23
0.22
0.21
0.2
0.19
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Area 1
Area 6
Future crack path
Future void
Current crack tip
r = 0.5mm
Lower lobe
Y (mm)
X (mm)
2.5
2
1.5
1
0.5
0
0
1
2
3

Figure 30.

In the fifth step of the deformation (Figure 31), the area lost due to the interpolation scheme is very large. Nevertheless the information that can be obtained from Figure 31 is very valuable. In the fifth step of the deformation, corresponding to 5% far field strain, the crack has propagated to the position (1.1mm, 1.3mm) from its previous position at (0.8mm, 1.3mm) in the fourth step. A circle of 0.5mm around the crack tip position includes the high strain regions located at (1.4mm, 1.6mm) and (1.5mm, 1.2mm), which are the highest strain locations of the step.

Using the last five contour plots, the area around the crack tip at every step where the maximum principal strains were larger than 10% have been plotted against the far field strain in Figure 32. A parabolic curve fitting with the additional constrains that at 0% far field strain the value and the slope of the fit are zero is shown in the picture. The curve has been fit to four data points, corresponding to those obtain at 0%, 1%, 2% and 3% far field strain. After this last step no other data point was used to compute the fit because the crack had propagated. In this figure it is appreciated how the increase of the area where the strains are larger than 10% has a parabolic shape when compared to the far field strain. Also it is seen that at the point of propagation the area reaches a maximum ($.09mm^2$) and after propagation the area remains fairly constant.

From the contour plot of the maximum principal strains at 5%, 9 regions are identified where the strains are rather high.

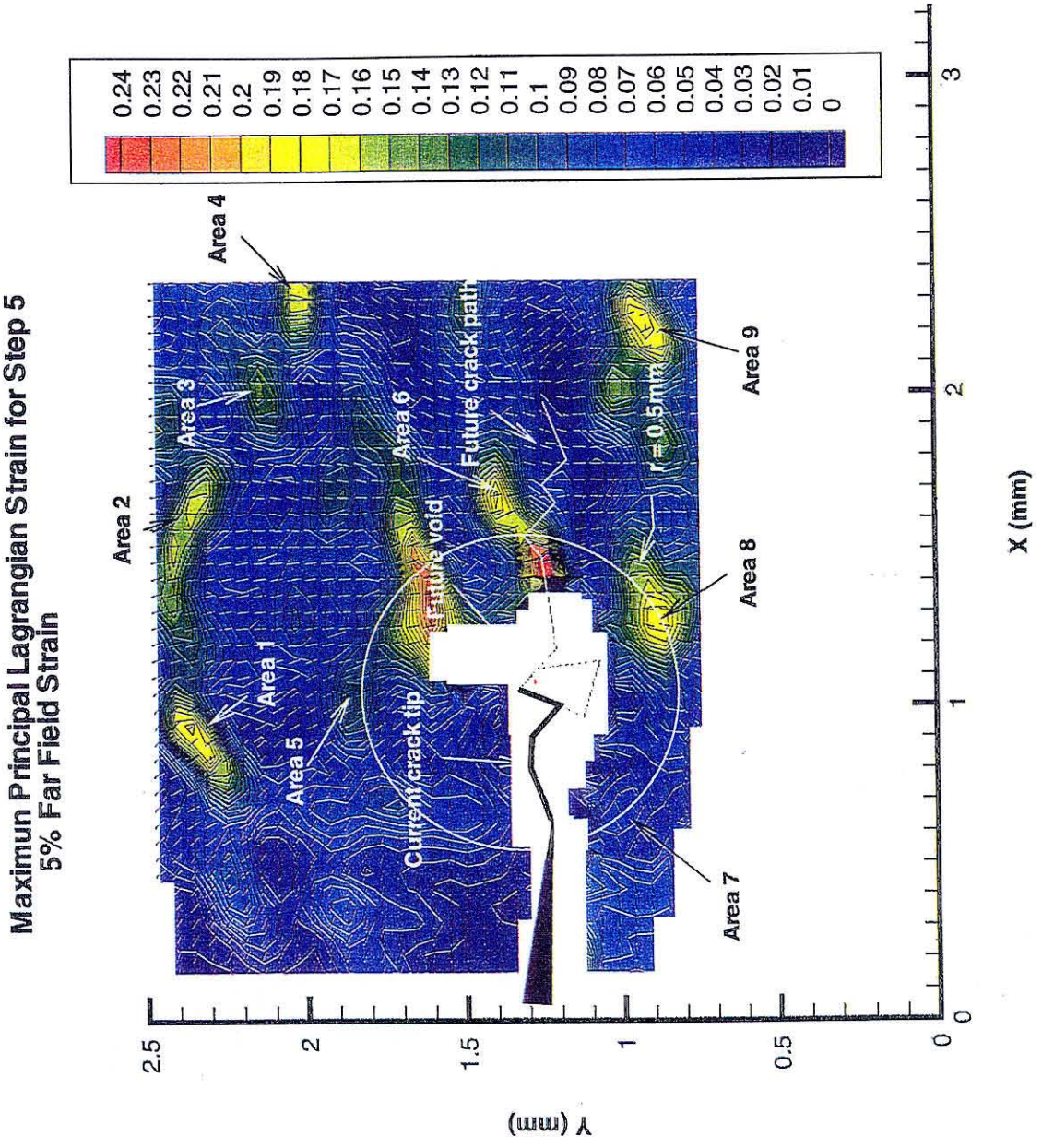

Maximun Principal Lagrangian Strain for Step 5
5% Far Field Strain
0.24
0.23
0.22
0.21
0.2
0.19
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
Area 1
Area 2
Area 3
Area 4
Area 5
Area 6
Area 7
Area 8
Area 9
Current crack tip
Future void
Future crack path
r = 0.5mm
Y (mm)
X (mm)
2.5
2
1.5
1
0.5
0
0
1
2
3

Figure 31.

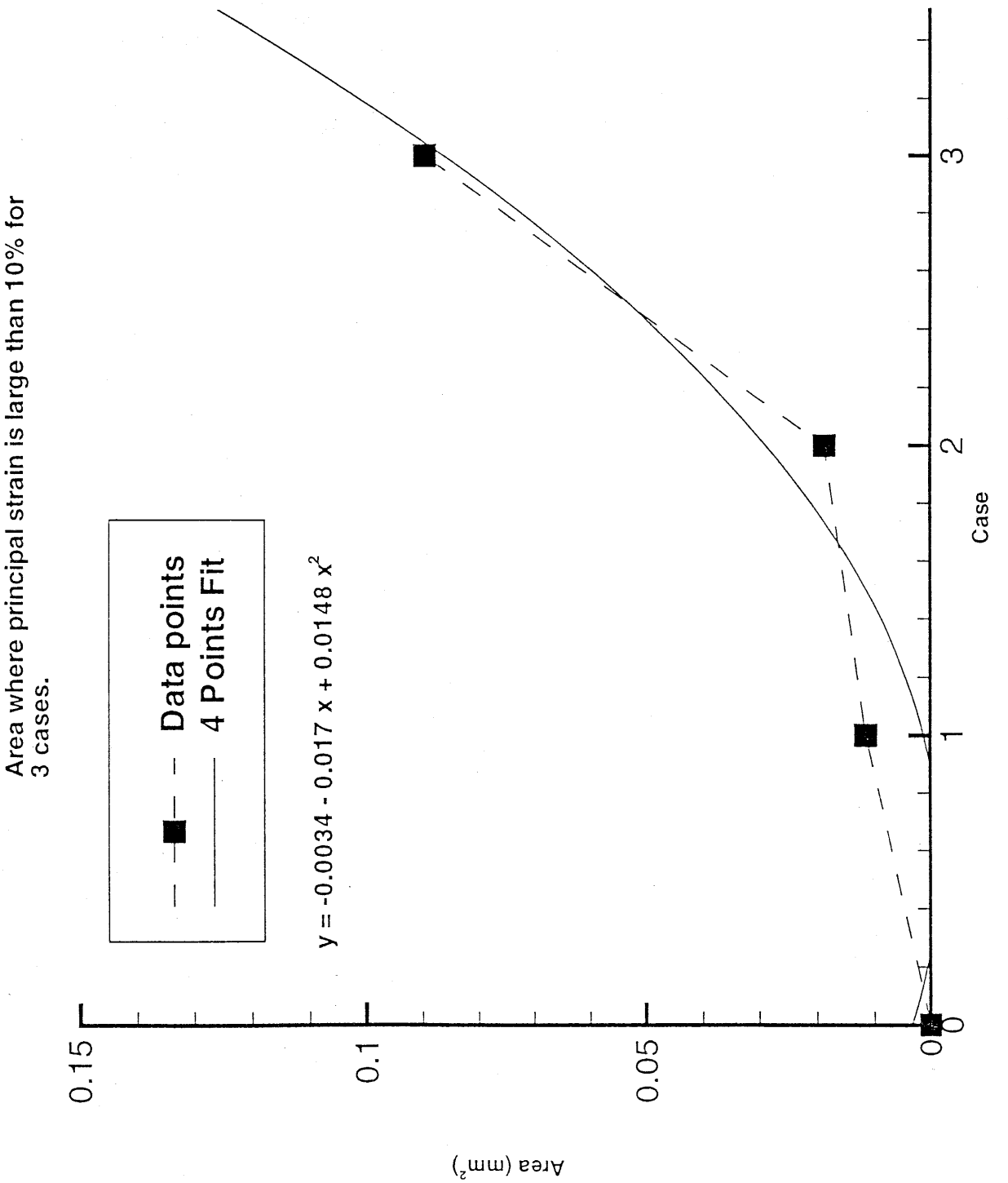

Area where principal strain is large than 10% for 3 cases.
Data points
4 Points Fit
y = -0.0034 - 0.017 x + 0.0148 x²
Area (mm²)
0.15
0.1
0.05
0
0
1
2
3
Case

Figure 32.

These regions are: Area 1, at (0.9mm, 2.3mm), Area 2 at (1.6mm, 2.4mm), Area 3 at (2mm, 2.1mm), Area 4 at (2.3mm, 2mm), Area 5 at (1mm, 1.8mm), Area 6 at (1.25mm, 1.6mm), Area 7 at (0.7mm, 1mm), Area 8 at (1.3mm, 0.8mm) and Area 9 at (2.25mm,0.95mm). The largest maximum principal strain at these areas for every step has been obtained and plotted in Figure 33 versus the far field strain. Also the rough calculation performed in Section 4. 1 is shown as a dashed line. This calculation estimated the strain in the matrix in a composite material containing 70% particles and 30% matrix during a deformation. In the plot it can be seen that the increase of strain in the high strain areas is close to linear.

In Figure 34 a distribution of the maximum principal strains along the crack propagation path for the five steps is presented. Also the position of the crack tip at every step of the deformation is included in the figure. The position of the crack tip for the first three steps is represented at x = 0mm. At the fourth and fifth steps the crack tip position is shown at x = 0.15mm and x = 0.4mm, respectively. From this figure it is seen that the strains localize within 0.5mm form the crack tip. In the first three steps, after 0.5mm the strains have decayed to less than half of their values close to the crack tip. In the fourth and fifth steps, the strains have decayed after 0.7mm of distance to half of their values at the crack tip.

In Figure 35 the maximum principal strains are presented in stations at every 0.25mm along the crack tip as a function of the far field strain. Eight points along the crack propagation path have been chosen, points located every 0.25mm from 0.25mm to 2mm.

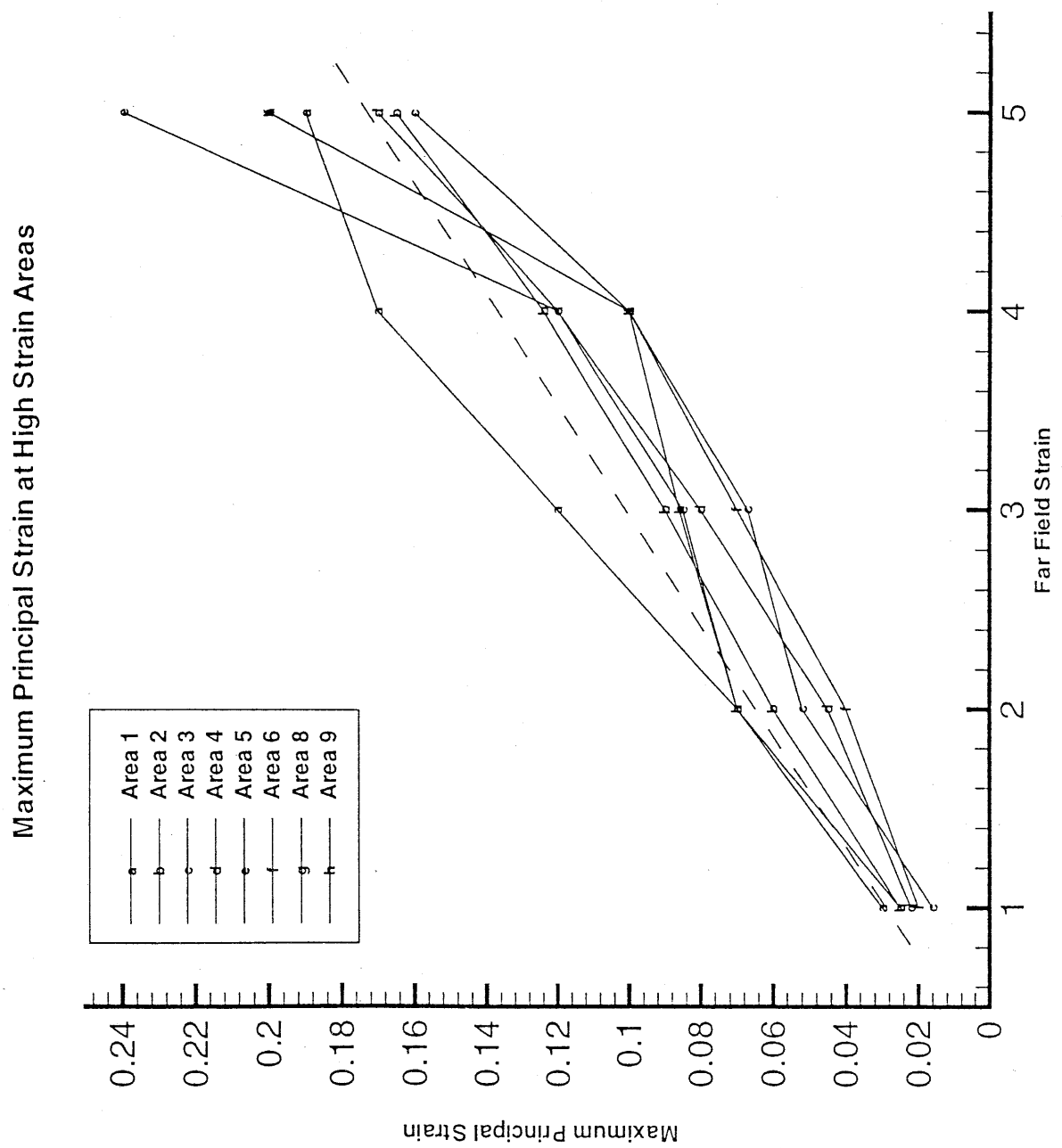

Maximum Principal Strain at High Strain Areas
Area 1
Area 2
Area 3
Area 4
Area 5
Area 6
Area 8
Area 9
Maximum Principal Strain
0.24
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
2
3
4
5
Far Field Strain

Figure 33.

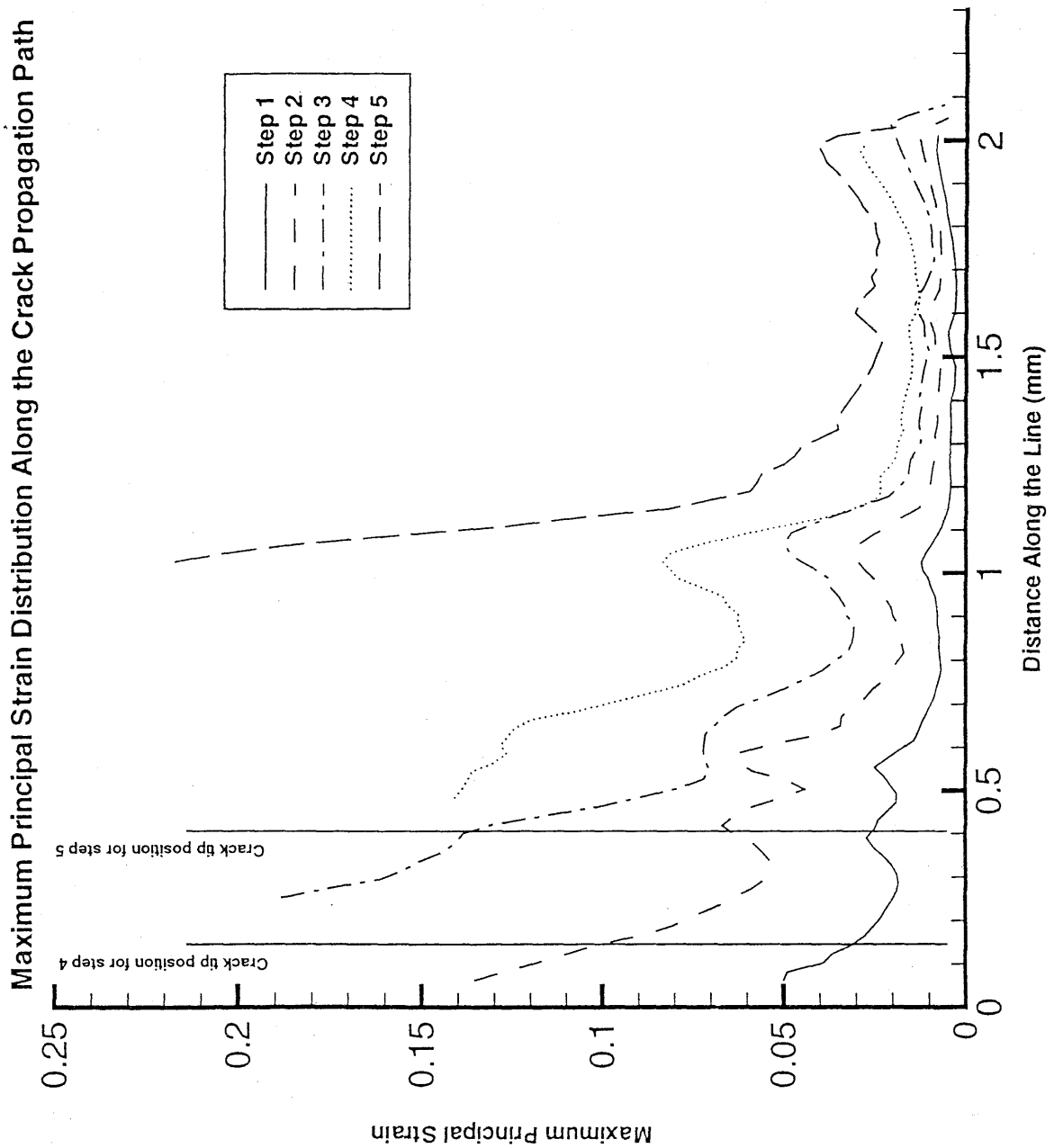

Maximum Principal Strain Distribution Along the Crack Propagation Path
Step 1
Step 2
Step 3
Step 4
Step 5
Crack tip position for step 4
Crack tip position for step 5
Maximum Principal Strain
Distance Along the Line (mm)
0.25
0.2
0.15
0.1
0.05
0
0.5
1
1.5
2

Figure 34.

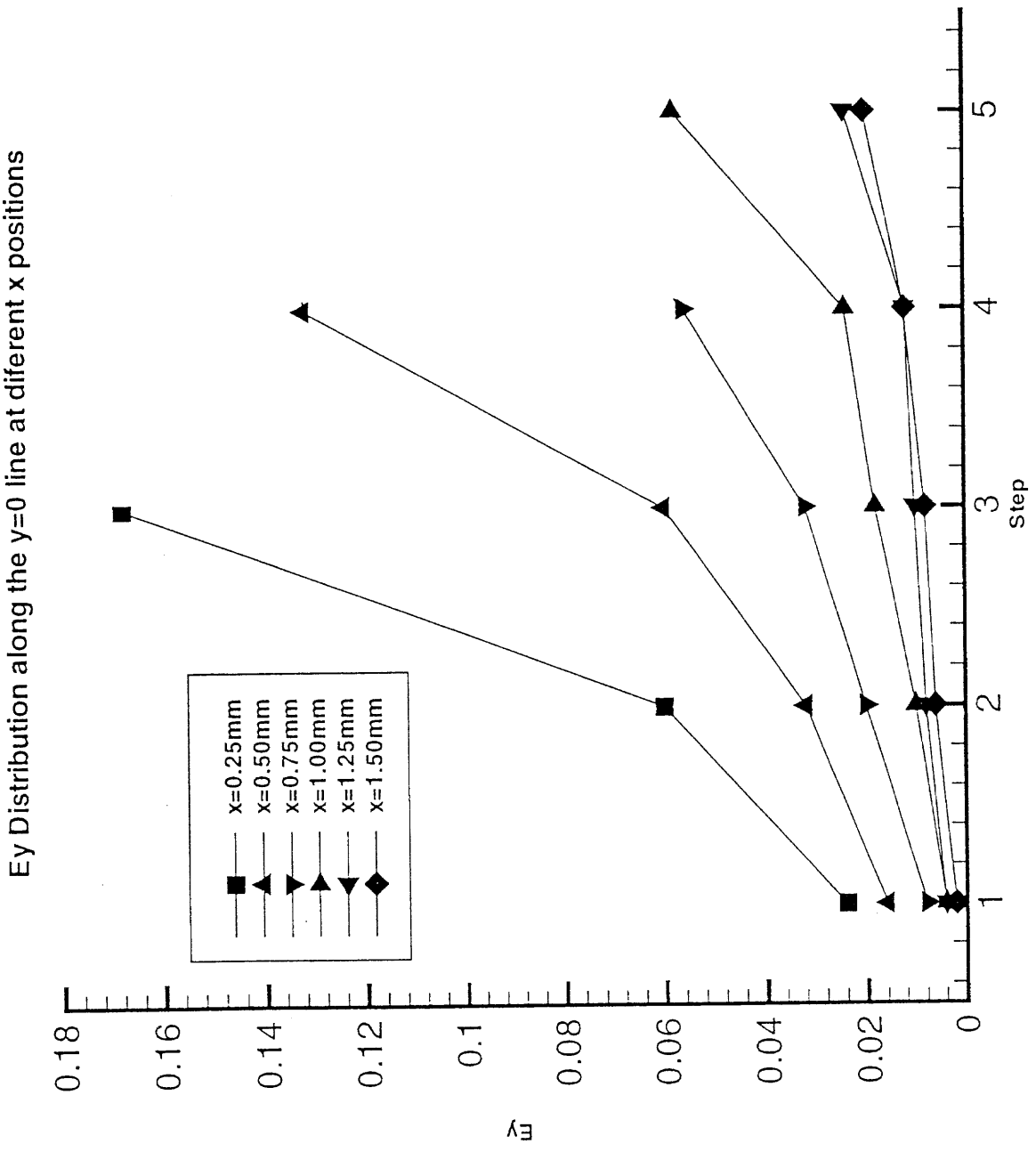

Ey Distribution along the y=0 line at diferent x positions
x=0.25mm
x=0.50mm
x=0.75mm
x=1.00mm
x=1.25mm
x=1.50mm
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Ey
1
2
3
4
5
Step

Figure 35.

The strains for the points located close to the crack tip, at the 0.25mm and 0.5mm stations, grow very fast in a quadratic manner. The strain at other points located further away from the crack tip don't grow that fast with the far field strains. It is interesting to note that it is just after the station at 0.50mm where there is a decrease in the average slope of the curves. This indicates once more that 0.5mm is a key distance in the distribution of the strains from the crack tip.

In figures 36 to 40 the contour plots corresponding to the minimum principal Lagrangian strains for the five steps of the deformation are shown. In Figure 36, corresponding to the first step, it is seen how there is an area of 0.3mm radius around the crack tip where negative strains of less than -1.3% strain localize. The minimum value for the minimum principal strain is located close to the crack tip and corresponds to less than -0.3% strain. There is a region 0.3mm to the right of the crack tip where high positive strain of up to 1.5% strain localizes. This region coincides with the place where the void develops in the fifth step.

One last feature of interest that is observed in the maximum principal strain contour plots, for example in Figure 29 , corresponding to the third step, is that at some regions the principal directions change by almost 90 degrees. Examples of these areas are (1.6mm, 1.1mm), (2.1mm, 1.2mm), (2.2mm, 1.6mm). These areas also coincide with regions where the strains are very small, indicating that they remain rigid during the deformation.

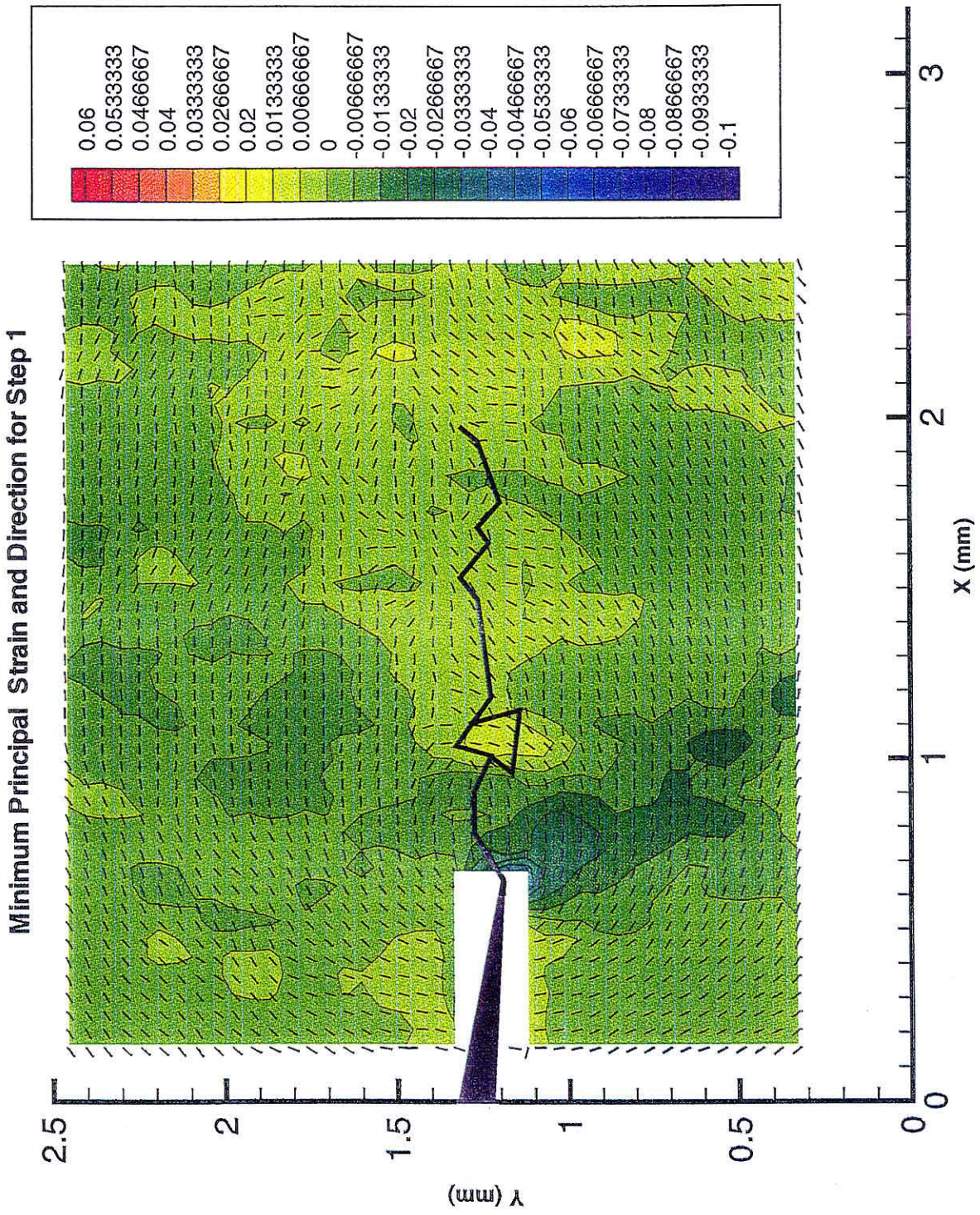

Minimum Principal Strain and Direction for Step 1
0.06
0.0533333
0.0466667
0.04
0.0333333
0.0266667
0.02
0.0133333
0.00666667
0
-0.00666667
-0.0133333
-0.02
-0.0266667
-0.0333333
-0.04
-0.0466667
-0.0533333
-0.06
-0.0666667
-0.0733333
-0.08
-0.0866667
-0.0933333
-0.1
Y (mm)
X (mm)
0
0.5
1
1.5
2
2.5
0
1
2
3

Figure 36.

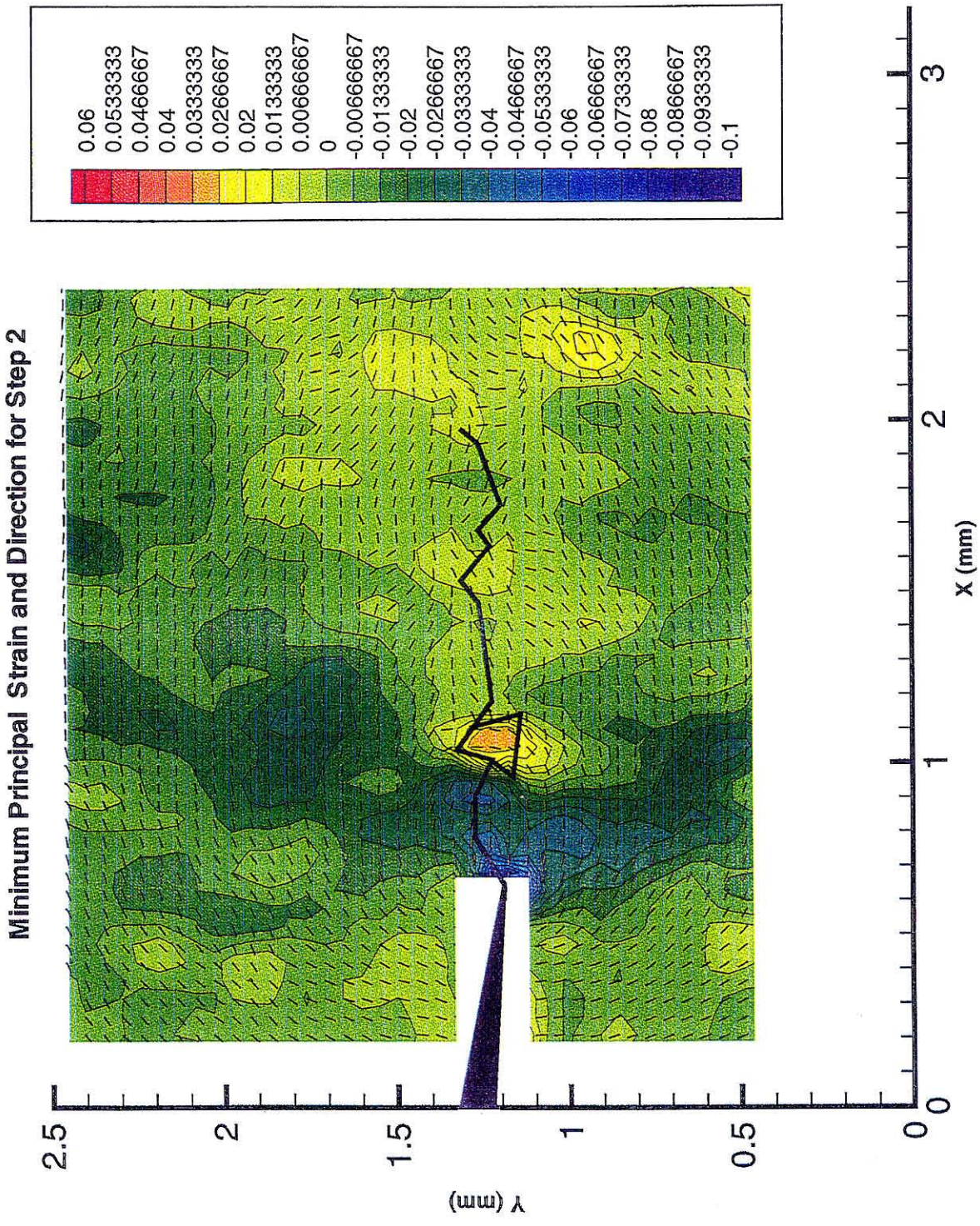

Minimum Principal Strain and Direction for Step 2
0.06
0.0533333
0.0466667
0.04
0.0333333
0.0266667
0.02
0.0133333
0.00666667
0
-0.00666667
-0.0133333
-0.02
-0.0266667
-0.0333333
-0.04
-0.0466667
-0.0533333
-0.06
-0.0666667
-0.0733333
-0.08
-0.0866667
-0.0933333
-0.1
Y (mm)
X (mm)
0
0.5
1
1.5
2
2.5
0
1
2
3

Figure 37.

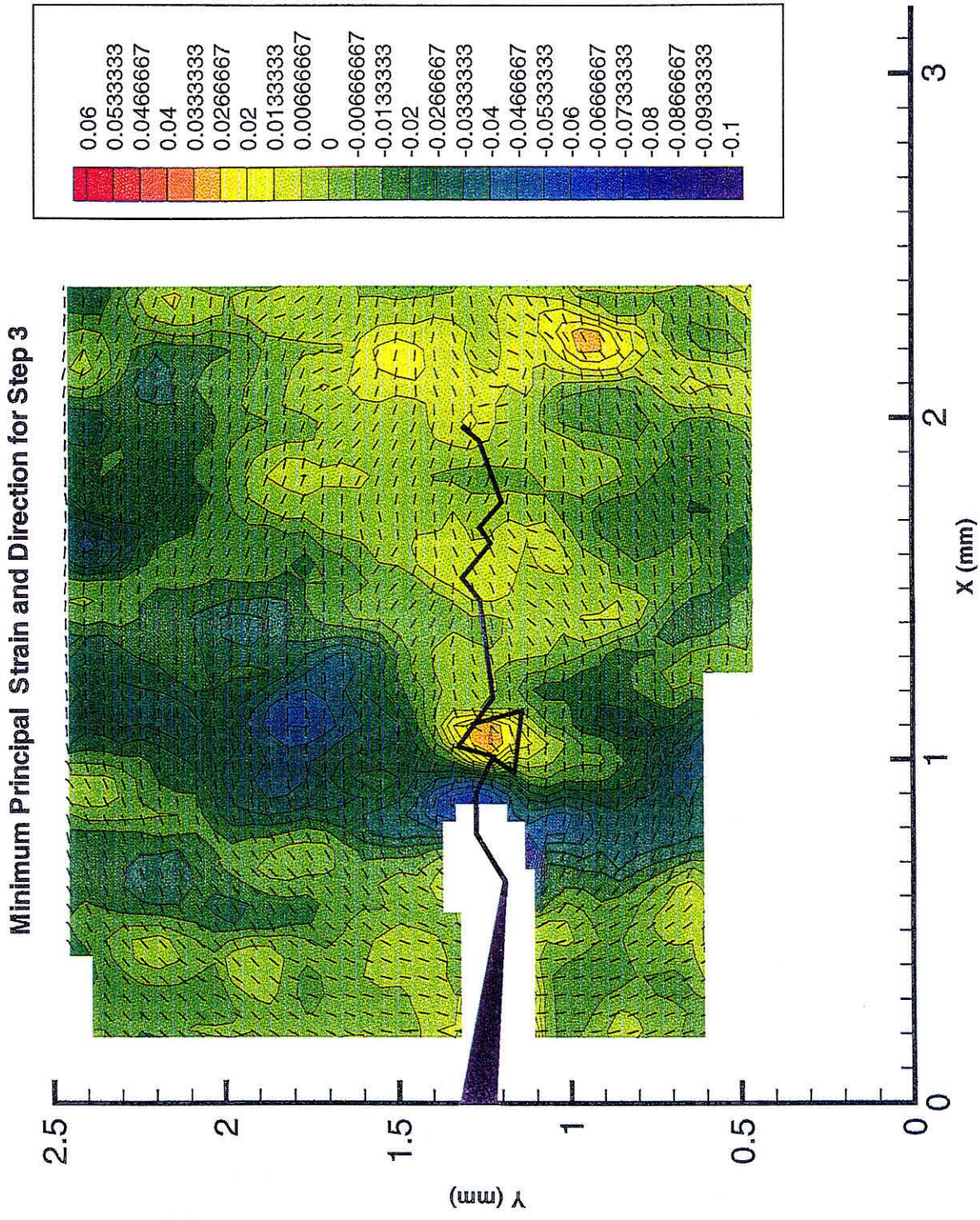

Minimum Principal Strain and Direction for Step 3
0.06
0.0533333
0.0466667
0.04
0.0333333
0.0266667
0.02
0.0133333
0.00666667
0
-0.00666667
-0.0133333
-0.02
-0.0266667
-0.0333333
-0.04
-0.0466667
-0.0533333
-0.06
-0.0666667
-0.0733333
-0.08
-0.0866667
-0.0933333
-0.1
Y (mm)
X (mm)
0
0.5
1
1.5
2
2.5
0
1
2
3

Figure 38.

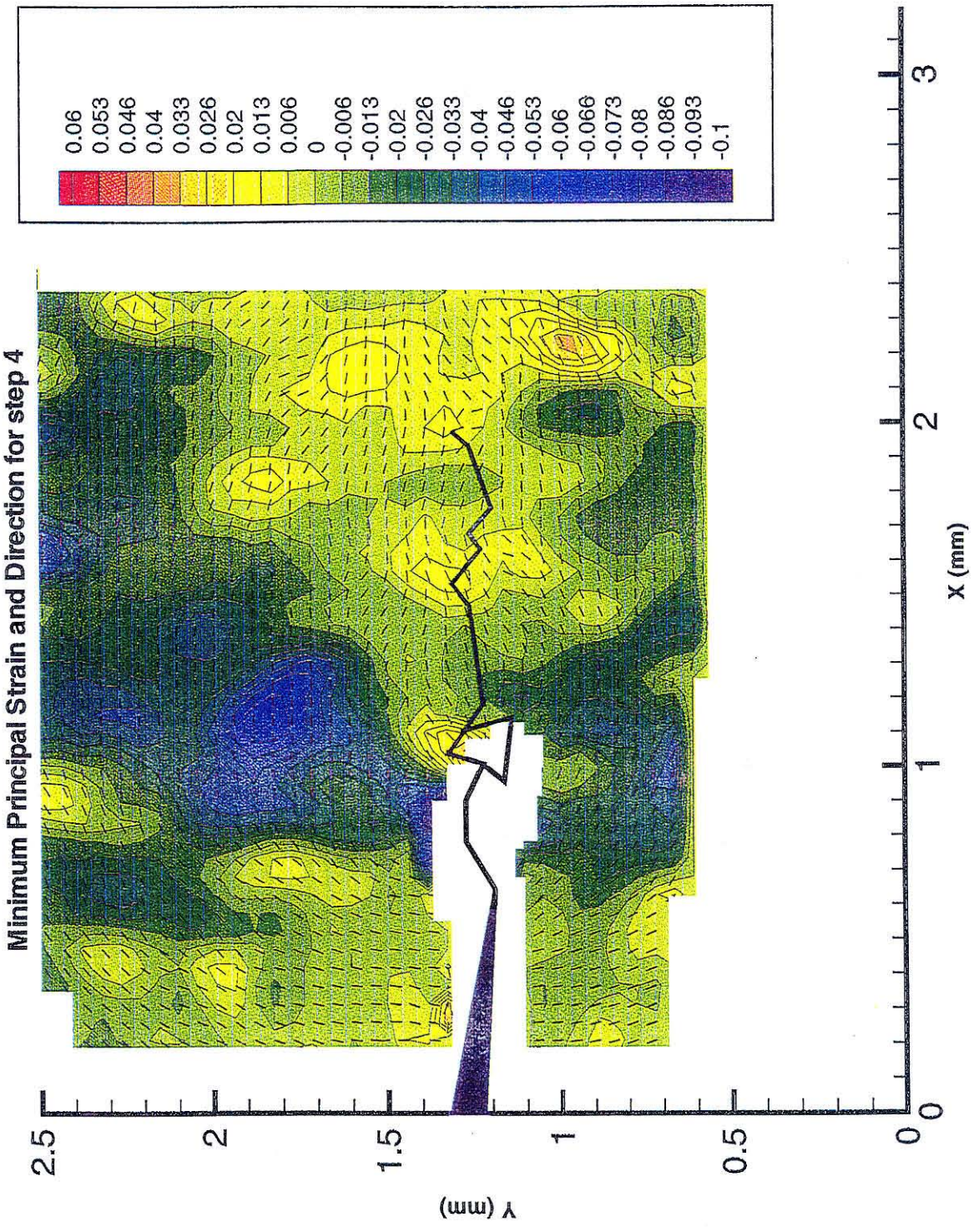

Minimum Principal Strain and Direction for step 4
0.06
0.053
0.046
0.04
0.033
0.026
0.02
0.013
0.006
0
-0.006
-0.013
-0.02
-0.026
-0.033
-0.04
-0.046
-0.053
-0.06
-0.066
-0.073
-0.08
-0.086
-0.093
-0.1
Y (mm)
X (mm)
0
0.5
1
1.5
2
2.5
0
1
2
3

Figure 39.

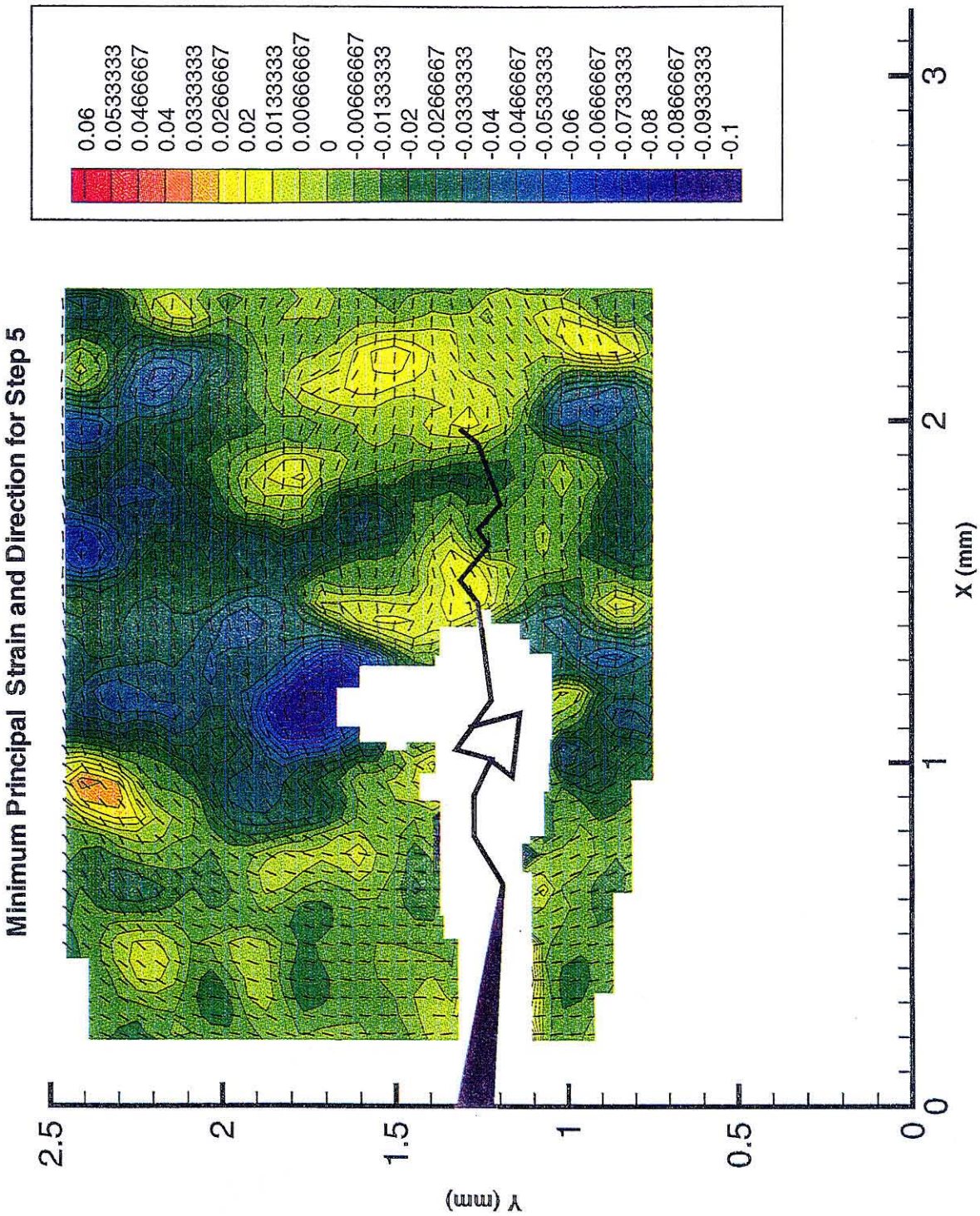

Minimum Principal Strain and Direction for Step 5
0.06
0.0533333
0.0466667
0.04
0.0333333
0.0266667
0.02
0.0133333
0.00666667
0
-0.00666667
-0.0133333
-0.02
-0.0266667
-0.0333333
-0.04
-0.0466667
-0.0533333
-0.06
-0.0666667
-0.0733333
-0.08
-0.0866667
-0.0933333
-0.1
Y (mm)
X (mm)
0
0.5
1
1.5
2
2.5
0
1
2
3

Figure 40.

4. 3. STRESS - STRAIN CURVE FOR SOLID PROPELLANT

A stress-strain curve for solid propellant TPH 1011 is shown in Figure 41. The specimen is of dimensions 5in x 0.5in x 0.125in and are pulled at a constant rate of 0.01 1/min by using an Instron testing machine. The material stiffness is rather low, with an ultimate stress of 0.5 Mpa The material is also ductile, elongating up to 16% of strain. The behavior of the material during the experiment is as follows. Between 0% strain and 3% strain there is a lot of wiggling in the curve and the average modulus is about 4Mpa. At a strain value of 3% the slope increases to 7Mpa and remains fairly constant until a value of 7% strain is attained. From this value on, the slope decreases until it is 0Mpa at 11% strain. Fracture occurs at 16% strain. One should notice that most of the wiggling of the curve is between 0% and 6% strain. From then on the curve is fairly smooth. The curve has a smooth plateau after the deformation has reached a value of 11%. Prior to that value the curve wiggles significantly, particularly in two areas, one close to the origin and the other at a value of 6% strain. This wiggling could indicate the variation of the load due to debonding of the particles. Another characteristic feature of interest is material failure without softening at a value of 16% strain.

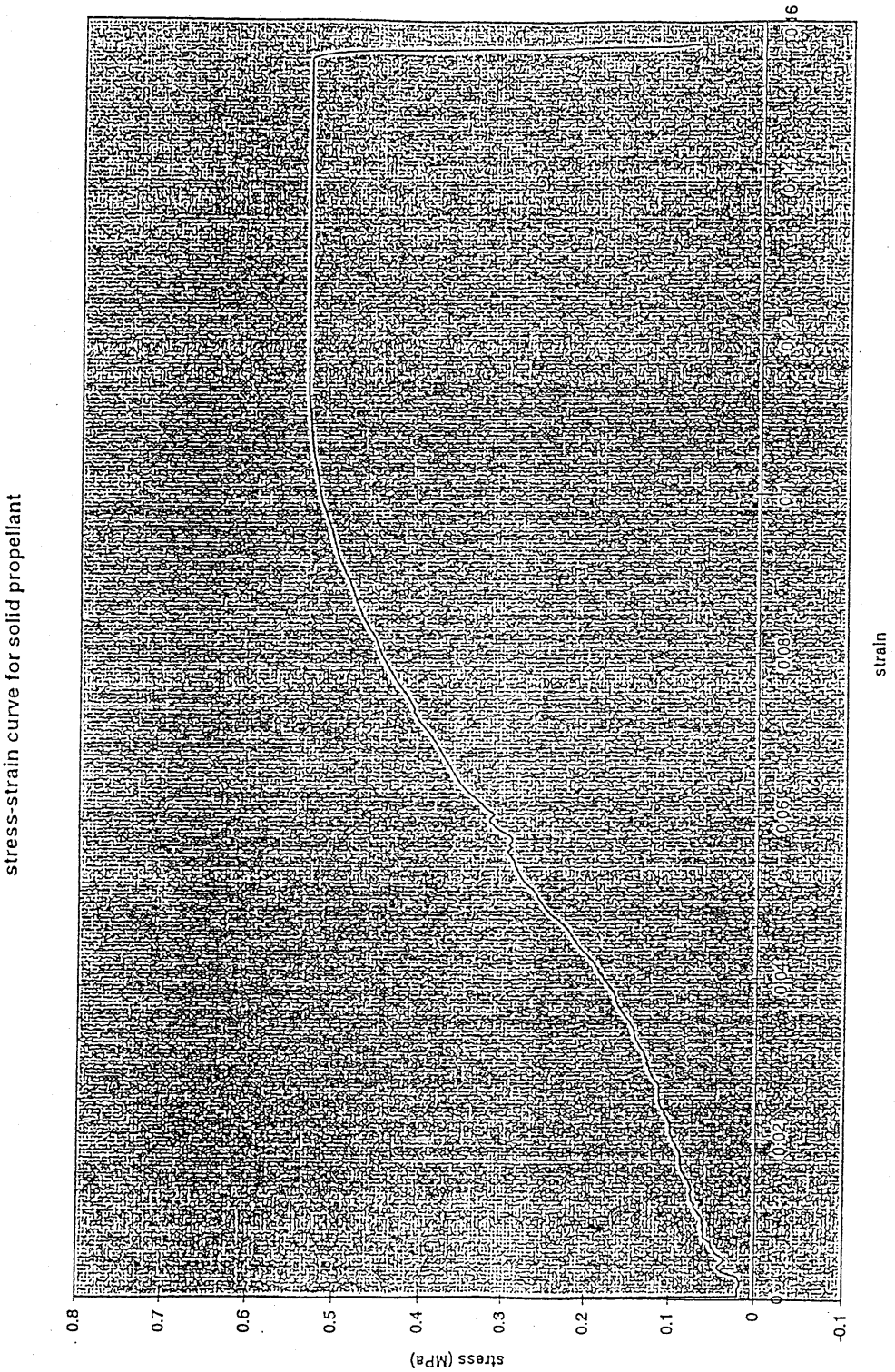

stress-strain curve for solid propellant
stress (MPa)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
strain

Figure 41.

5. CONCLUSIONS

From the results obtained by the microscopic visualization of the fracture process of the solid propellant TPH 1011, it can be concluded that the heterogeneity of the material plays a key role for the crack propagation and distribution of strains around the crack tip.

One conclusion in this experiment is that in the fracture process of the solid propellant TPH - 1011 there is a region around the crack tip of about 0.5mm radius where most of the deformation that is related to the crack propagation path localizes. Outside this 0.5mm region there are other high strain areas but as the load increases, they are not related to the crack path. The only relation to the crack propagation process that these later regions have is crack shielding, were some smaller cracks open and release tension in the material. The deformation within 0.5mm of the crack tip grows much faster than elsewhere. One immediate consequence of the localization of the deformation within 0.5mm from the crack tip is that as the load increases and the crack propagates, the strain distribution in the material changes from load step to load step. Therefore the strain distribution at some step can only predict the crack propagation path within 0.5mm. Outside this area, the strain distribution may change significantly by the time the crack tip reaches the area.

The strains obtained in the experiment performed are surface strains. Nevertheless they not only represent what is happening in the surface but also underneath the surface. One particle that is located underneath the surface affects the distribution of strains in a region of the surface that is close to it.

REFERENCES

1. Sutton, M. A., Cheng, M., Peters, W. H., Chao, Y. J and McNeill, S. R., "Application of an Optimized Digital Correlation Method to Planar Deformation Analysis" , Image Vision Comput. Vol **4**, no 3. pp 143-150 (August 1986)

2. Vendroux, G. and Knauss, W. "Deformation Measurements at the Sub-Micron Size Scale: II. Refinements in the Algorithm for Digital Image Correlation" . GALCIT SM Report 94-5, (APR 1994)

3. Liu, C., "Effect of Predamage on Crack Growth Behavior in a Particulate Composite Material" . Journal of Spacecrafts and Rockets. Vol **32**, no. 3, (May-June 1995).

4. Liu, C., "Crack Growth Behavior in a Composite Propellant with Strain Gradients - Part II" . Journal of Spacecrafts and Rockets. Vol. **27**, no. 6, (Nov.-Dec. 1990)

5. Vratsanos,L. ,Farris, R., "A Predictive Model for the Mechanical Behavior of Particulate Composites. Part II: Comparison of the Model Predictions to Literature Data" . Polymer Engineering and Science, Vol. **33**, no. 22. (November 1993).

6. Farris,. R. J.,"The Character of the Stress-Strain Function for Highly Filled Elastomers" . Trans. Soc. Rheol. Vol **12**, pp 303-314 (1968).

7. Ravichandran, G., Liu., C., "Modeling Constitutive Behavior of Particulate Composites Undergoing Damage" . Int. J. Solids Structures Vol. **32**, no. 6/7, pp. 979-990, (May 1994)

8. Farrys, R. J. And Schapery, R. A. ,"Development of a Solid Rocket Propellant Nonlinear Viscoelastic Constitutive Theory" . Technical Report AFRPL-TR-73-50, Air Force Rocket Propulsion Laboratory, Edwards, CA.

9. Smith,C., Palaniswamy, K., Knauss, W., "The Application of Rate Theory to the Failure of Solid Propellants" . AFRPL-TR- 73-57, (July 1973).

10. Gurtin, M. & Francis, E., Simple Rate-Independent Model for Damage. Journal of Spacecrafts and Rockets, Vol. **18**, no 3, (May-June 1981).

11. Schultheisz, C. & Knauss, W. "The Governing Equations of Moire Interferometry II. Gratings at $\pm 45^{o}$ to the Measurement Axes" . Optics and Lasers in Engineering **20** (1994)

12. Swanson, S. & Christensen, L.,"A Constitutive Formulation for High-Elongation Propellants" . Journal of Spacecrafts and Rockets. Vol **20**, no. 6, (Nov.-Dec 1983).

13. Govindjee, S. &Simo, J., "Transition From Micro-Mechanics to Computationally Efficient Phenomenology: Carbon Black Filled Rubbers Incorporating Mullins' Effect" . J. Mech. Phys. Solids Vol. **40**, no. 1, pp. 213-233, (1992)

14. Christensen, R., "A Critical Evaluation for a Class of Micro-Mechanics Models" . J. Mech. Phys. Solids Vol. **38**, no. 3, pp. 379-404, (1990).

15. Liu,C. "Evaluation of Damage Fields Near Crack Tips in a Composite Solid Propellant" . Journal of Spacecrafts and Rockets. Vol **28**, no. 1, (Jan.-Feb. 1991).

16. Vratsanos,L. ,Farris, R., "A Predictive Model for the Mechanical Behavior of Particulate Composites. Part I: Model Derivation" . Polymer Engineering and Science, Vol. **33**, no. 22. (Nov. 1993).

17. Knauss, W., "Constitutive Behaviour of a Polymer in the Process of Void Formation" . GALCIT SM 92-7, (1992).

18. Knauss, W.,& Losi, G., "Crack Propagation in a Non-Linearly Viscoelastic Solid With Relevance to Adhesive Bond Failure" . GALCIT SM Report 91-16. (1991).

19. Sutton, M. A., Deng, X., Liu, J. And Yang, L., "Determination of Elastic-plastic Stresses and Strains from Measured Surface Strain Data" , Experimental Mechanics pp 99,112.(August 1986)

20. Farris, R. J. And Schapery, R. A.,"Development of a Solid Rocket Propellant Nonlinear Viscoelastic Constitutive Theory" , AFPRL-TR-50, Air Force Rocket Propulsion Laboratory (1973)

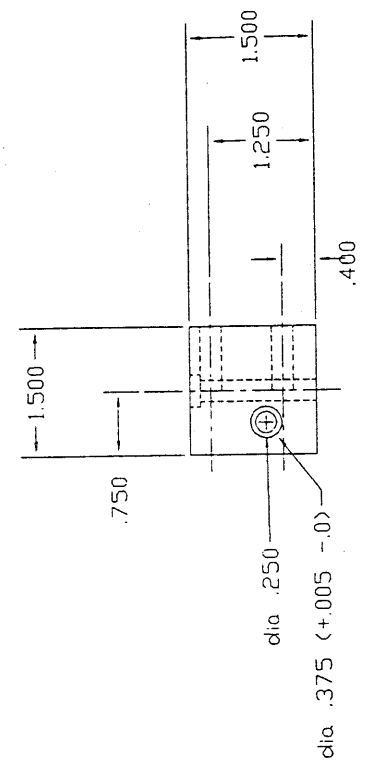
APPENDIX A

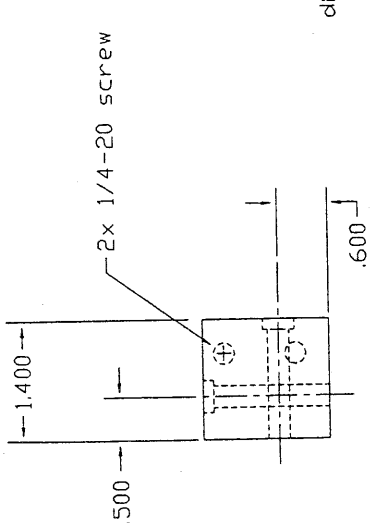
STAINING STAGE

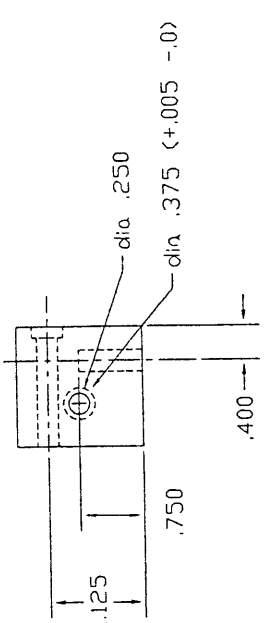
The components used to build the straining stage were purchased in

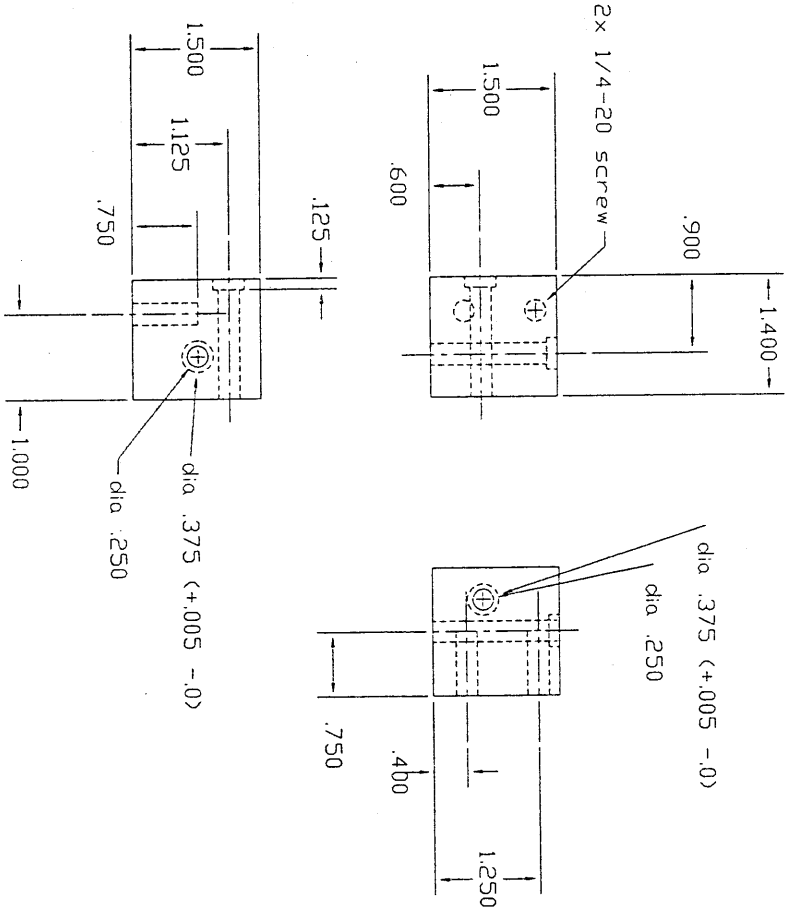
W. N. Berg
499 Ocean Av.
East Rockaway NY 11518
Phone (516) 599-5010

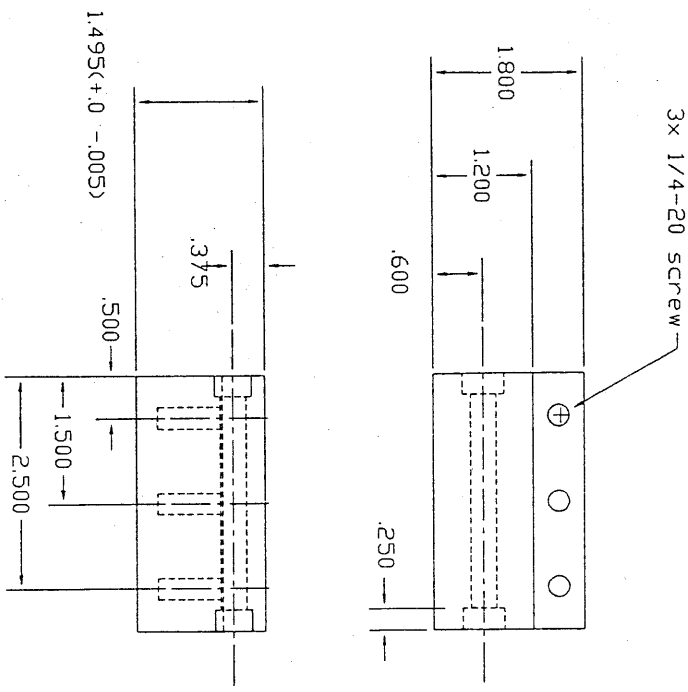
Planes with dimensions of the parts are presented in the next pages.

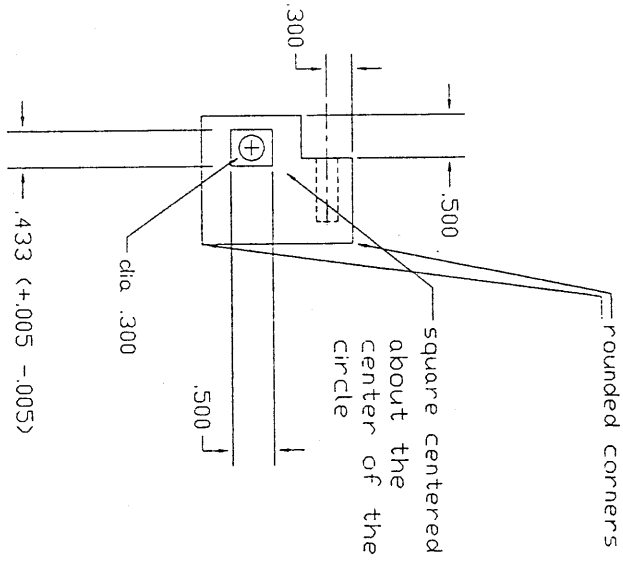

1.500
1.250
1.500
.400
.750
dia .250
dia .375 (+.005 -.0)

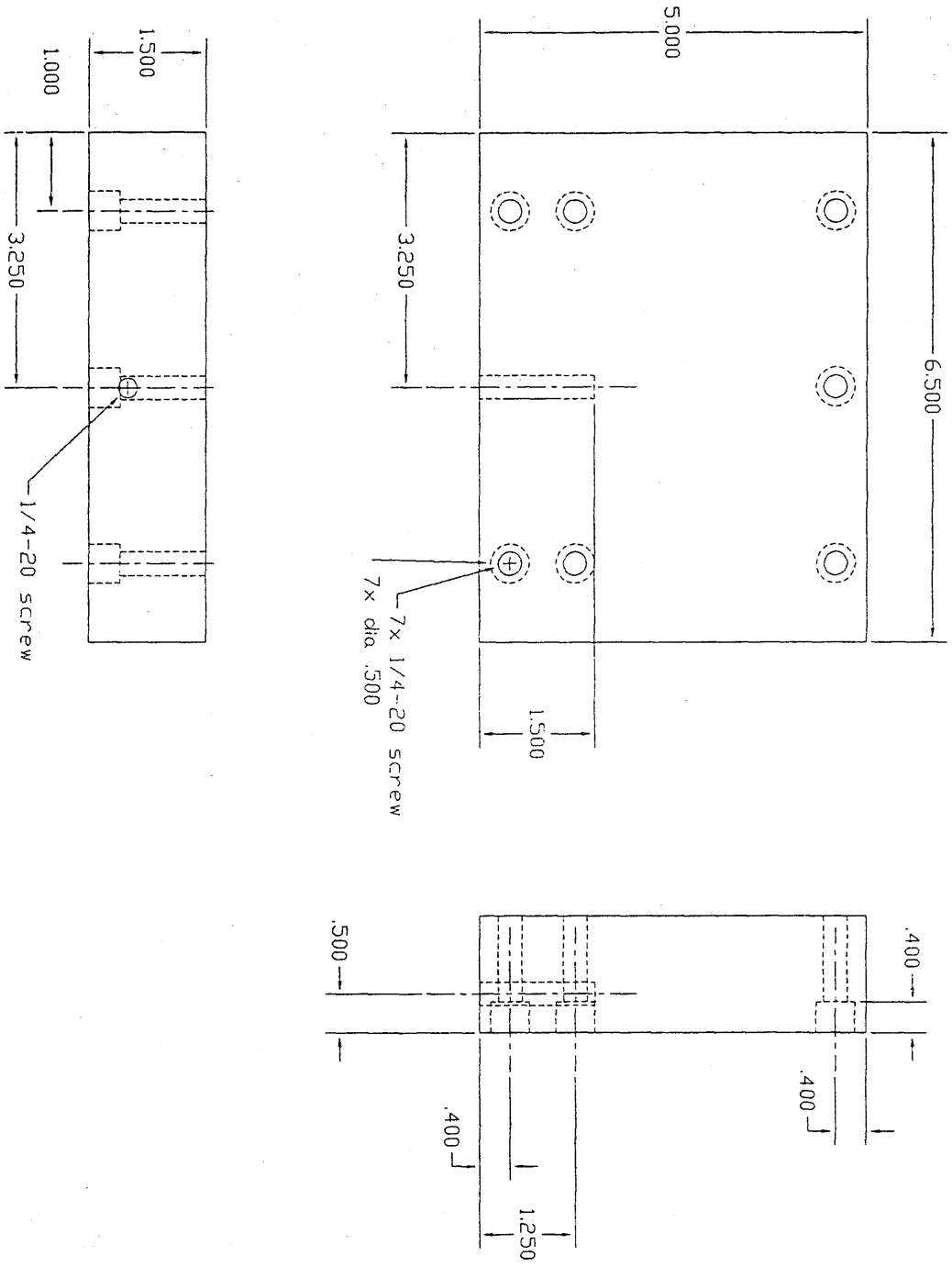

2x 1/4-20 screw
1.400
.500
.600

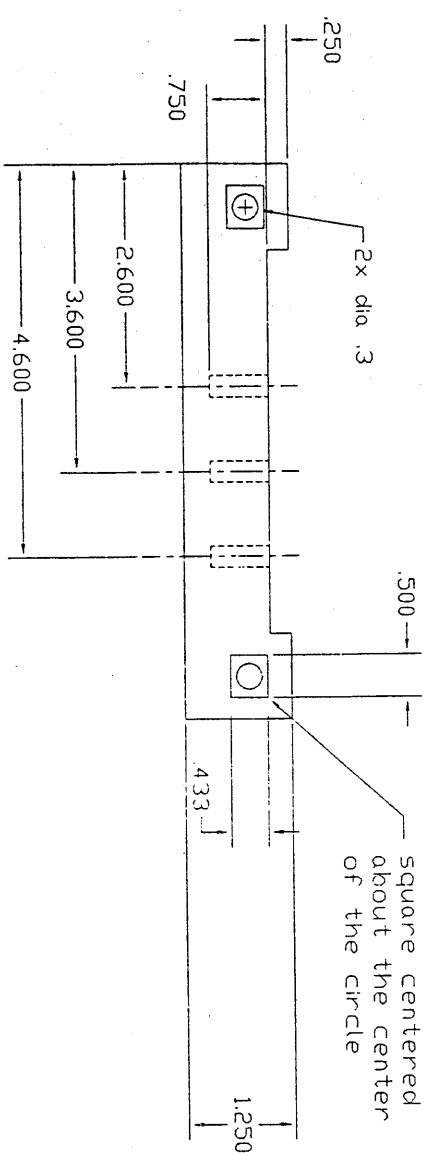

dia .250
dia .375 (+.005 -.0)
.400
.750
1.125

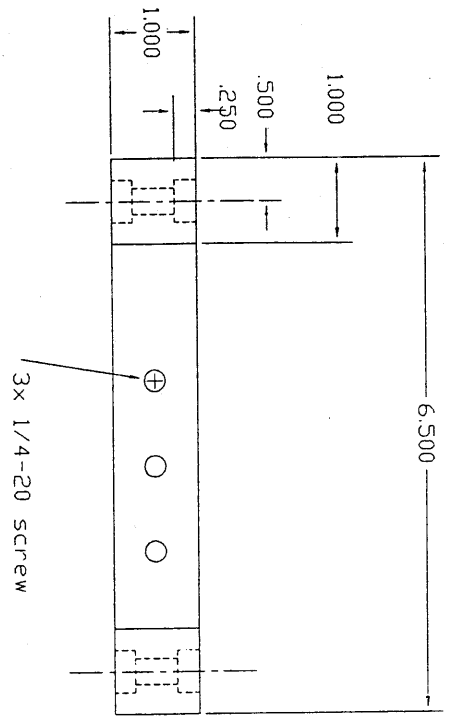

2x 1/4-20 screw
1.500
.600
.900
1.400
1.500
1.125
.750
.125
1.000
dia .250
dia .375 (+.005 -.0)
dia .375 (+.005 -.0)
dia .250
.750
.400
1.250

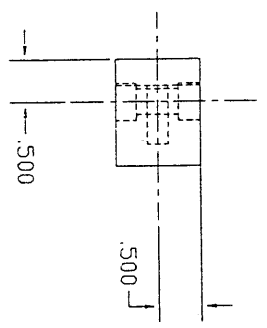

1.495(+.0 -.005)
.375
.500
1.500
2.500
1.800
1.200
.600
.250
3x 1/4-20 screw

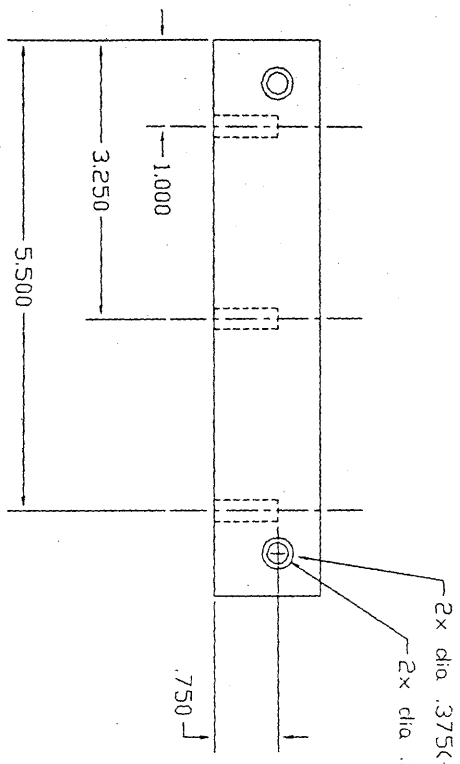

.300
.500
.433 (+.005 -.005)
dia .300
.500
square centered about the center of the circle
rounded corners

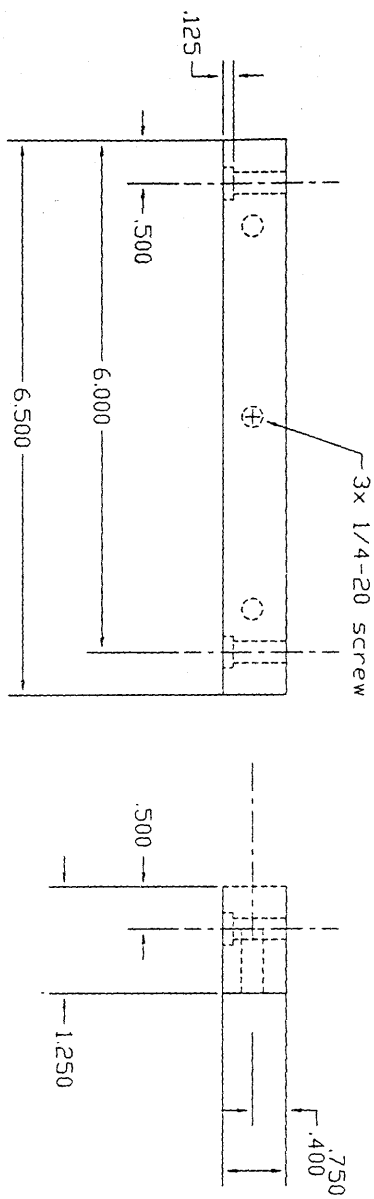

1.500
1.000
3.250
1/4-20 screw
5.000
3.250
6.500
7x 1/4-20 screw
7x dia. .500
1.500
.500
.400
.400
.400
1.250

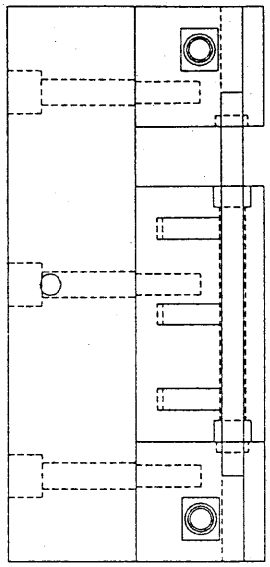

.250
.750
2x dia .3
2.600
3.600
4.600
.500
.433
square centered
about the center
of the circle
1.250

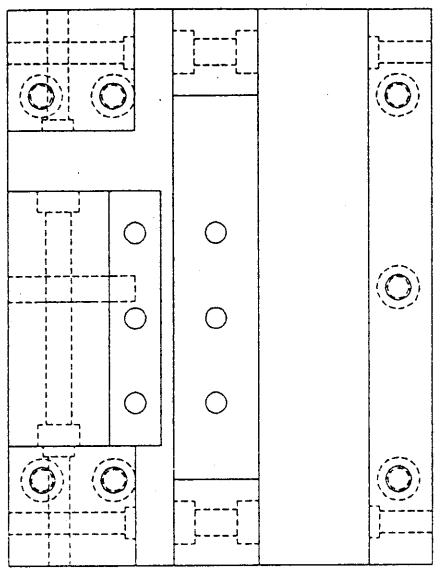

1.000
.500
.250
1.000
6.500
3x 1/4-20 screw

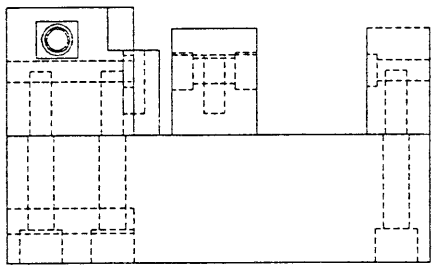

.500
.500


5.500
3.250
1.000
.750
2x dia .375(+.005 -0.0)
2x dia .250


.125
.500
6.000
6.500
3x 1/4-20 screw
.500
1.250
.750
.400







APPENDIX B

GENERAL FEATURES OF THE FRACTURE PROCESS OF THE SOLID PROPELLANT TPH 1011 RECORDED BY OPTICAL MICROSCOPY

A series of 34 pictures were taken to study the fracture process of a solid propellant in which the crack is running from top to button. The applied strain rate of the deformation is constant and equal to 0.001 1/sec. In Figures 1b to 34b it can be seen that the solid propellant material appears as a dark gray irregular surface. We can also detect some dimples on the surface of the material caused during the manufacturing process. These dimples have the diameter of 80 to 600 microns and they are not very deep. Their aspect is like that of holes in the material. Later on, as the fracture process develops, it will be difficult to distinguish these features from the holes that will occur near the crack tip. One important thing about these pictures is the orientation of the light: from top to button. It causes black shadows on the upper parts of the features that penetrate into the material. Example of these features are dimples, holes and cracks. The orientation of the light also creates white lines on the lower parts of these features. Therefore, the crack tip, which is oriented to the button, has light contours. The crack propagation mechanism can be summarized as follows. Initially, the increase in specimen tension causes the crack opening angle to increase (Fig 1b to 5b). At the same time there is a area around the crack tip in which damage localizes. This damage area can be seen on the button and to the left of the crack tip in Figure 6b, as an increase of white small features that indicates that light is shinning perpendicular to these areas because new small crack are opening. We also observe some irregularities in the contour of the crack near the described area. Figures 7b and 8b show how holes develop in the damage area along with the appearance of more white features on the damaged area, which indicates a high density of damage. Finally, in Figure 9b the crack runs through the damage area. The initial dimples on the surface sometimes play an important role for the propagation of the crack. This can be concluded from Figures 9b, 10b and 11b. At a certain point the crack has run through the damaged area and then stops. Next, the increase in tension in the material results in a growth of the crack opening angle and the damage area enlarges. In Figures 13b to 17b

we observe how the crack tip is stationary while damage is developing to its right. During these damage process two holes develop and grow along with the tension in the area. During Figures 15b, 16b and 17b the two holes coalesce releasing tension in the material. From Figures 16b and 17b the crack opening angle decreases while the tension is increasing. This is because the opening of the holes releases tension in the area. In Figure 18b the crack runs through the damage area connecting the hole. In Figures 18b, 19b, 20b and 21b the crack stops while a hole develops in the damaged area. In Figures 22b and 23b a flank of the crack coalesces with the hole. This indicates that in the fracture process of solid propellant TPH 1011, the crack doesn't follow a straight line like expected in a homogeneous material. The inhomogeneities in the material causes the local stress to vary in a random fashion, so the failure sites in the material do not necessarily coincide with the maximum local stress location. In this process we can see the trend of increasing crack opening angle coupled with the generation of damage, developing of holes and decreasing of crack opening angle, crack advancing through the damage area by coalesce with holes and finally stopping of the crack growth. This cycle repeats and after some of these cycles the damage area is very big and the crack becomes unstable. In the last figure it is represented the length scale, one division represents 1/16 in.

Figure 1b.

Figure 2b.

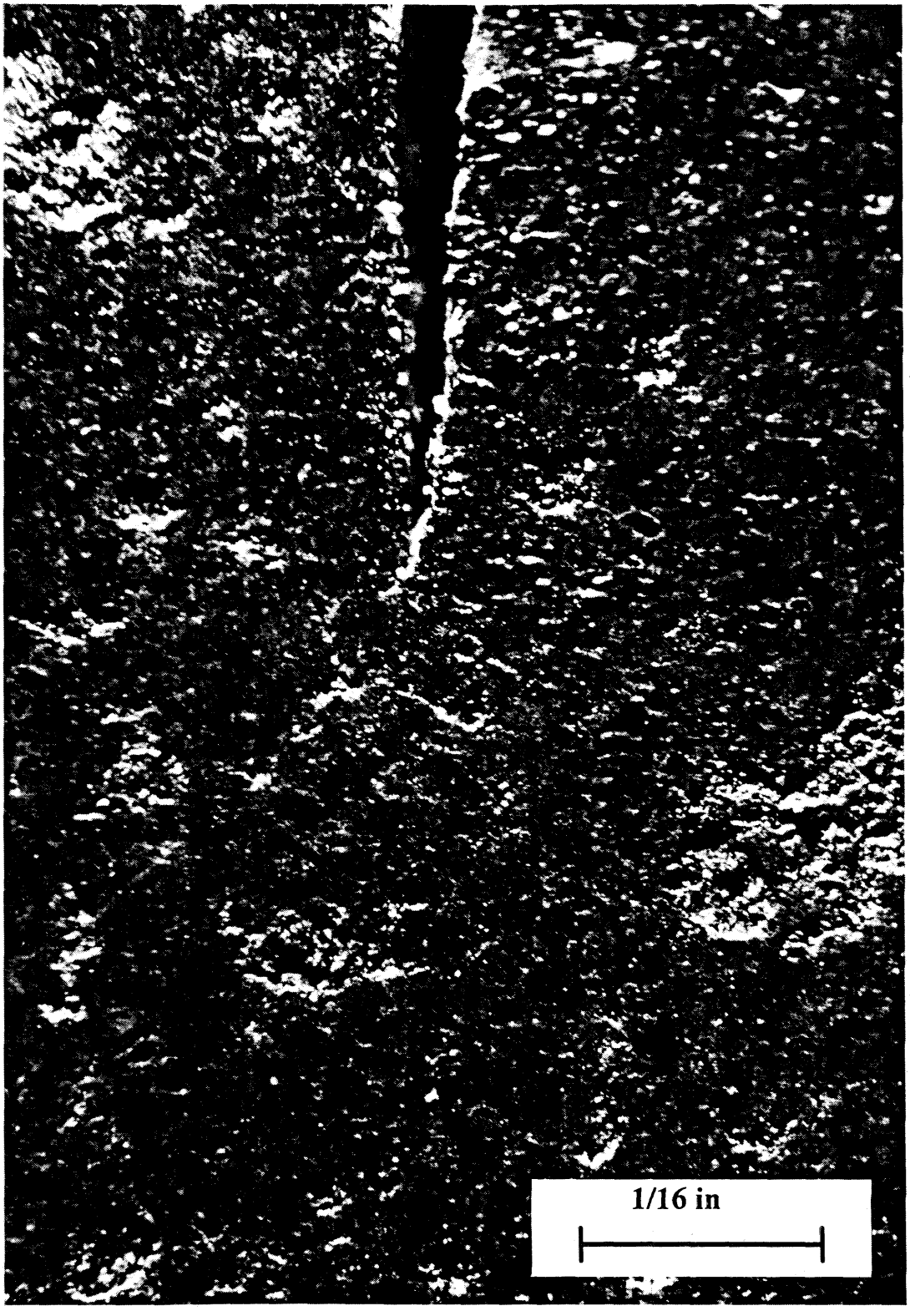


Figure 3b.

Figure 4b.

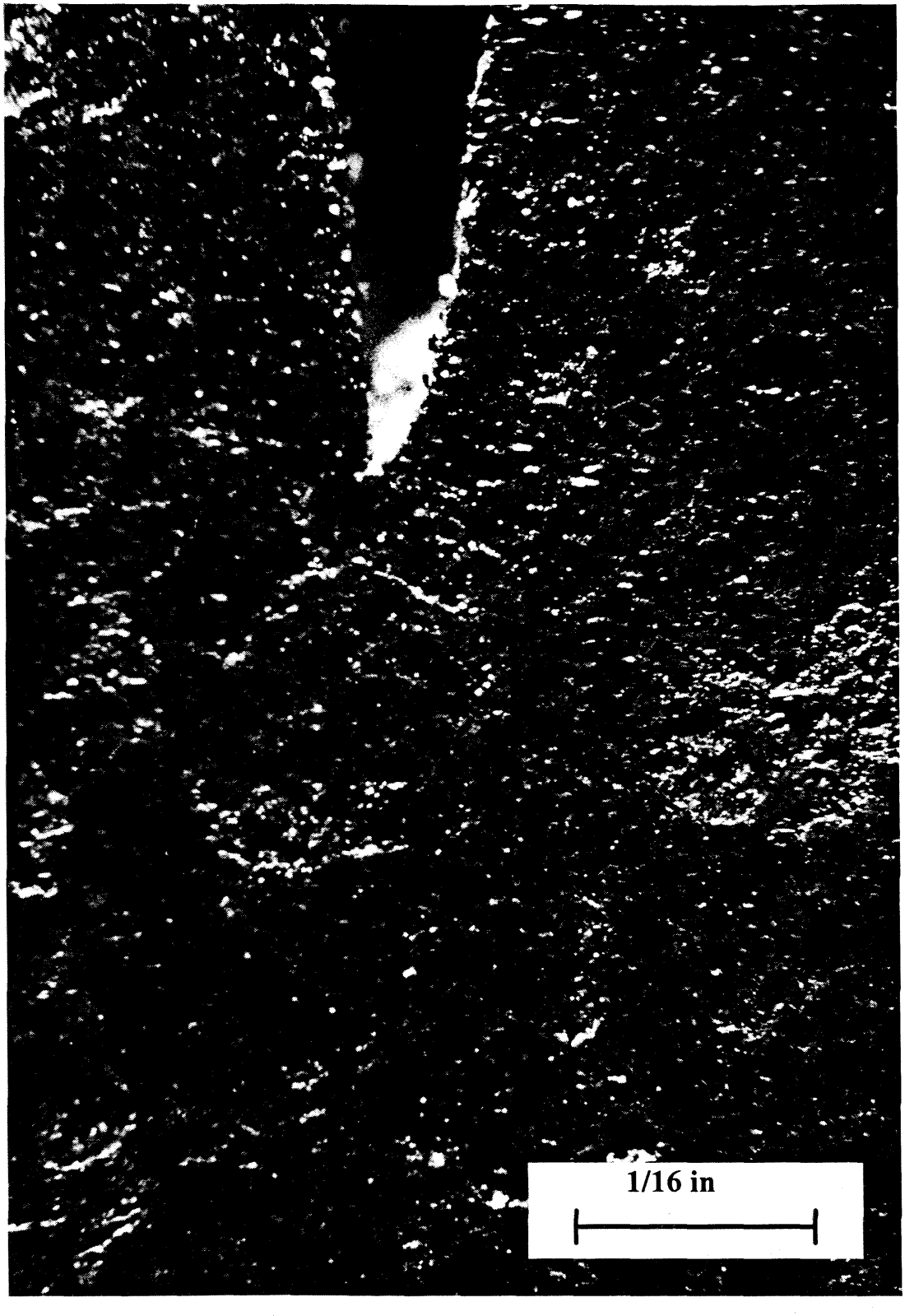


Figure 5b.

Figure 6b.

Figure 7b.

Figure 8b.

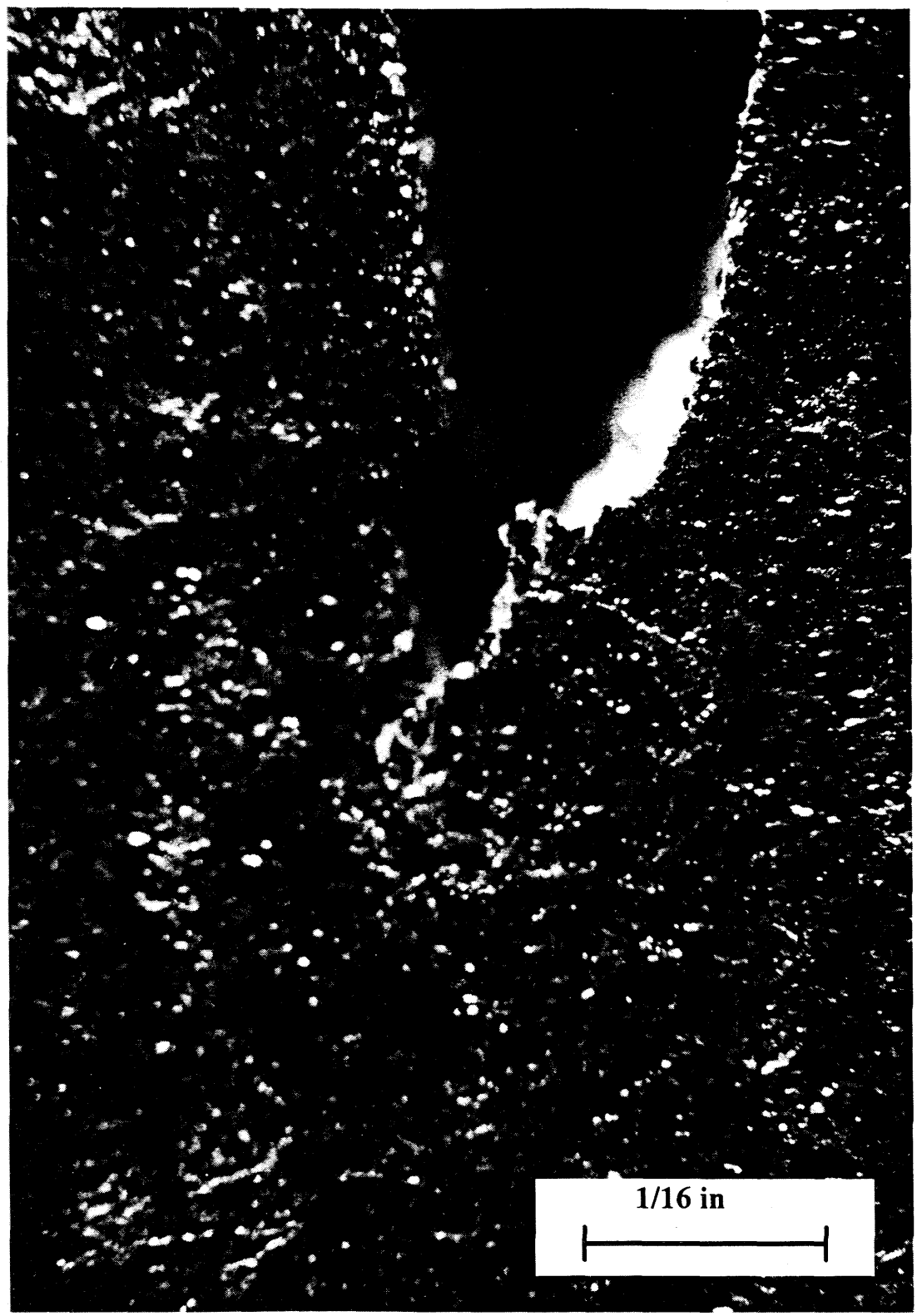


Figure 9b.

Figure 10b.

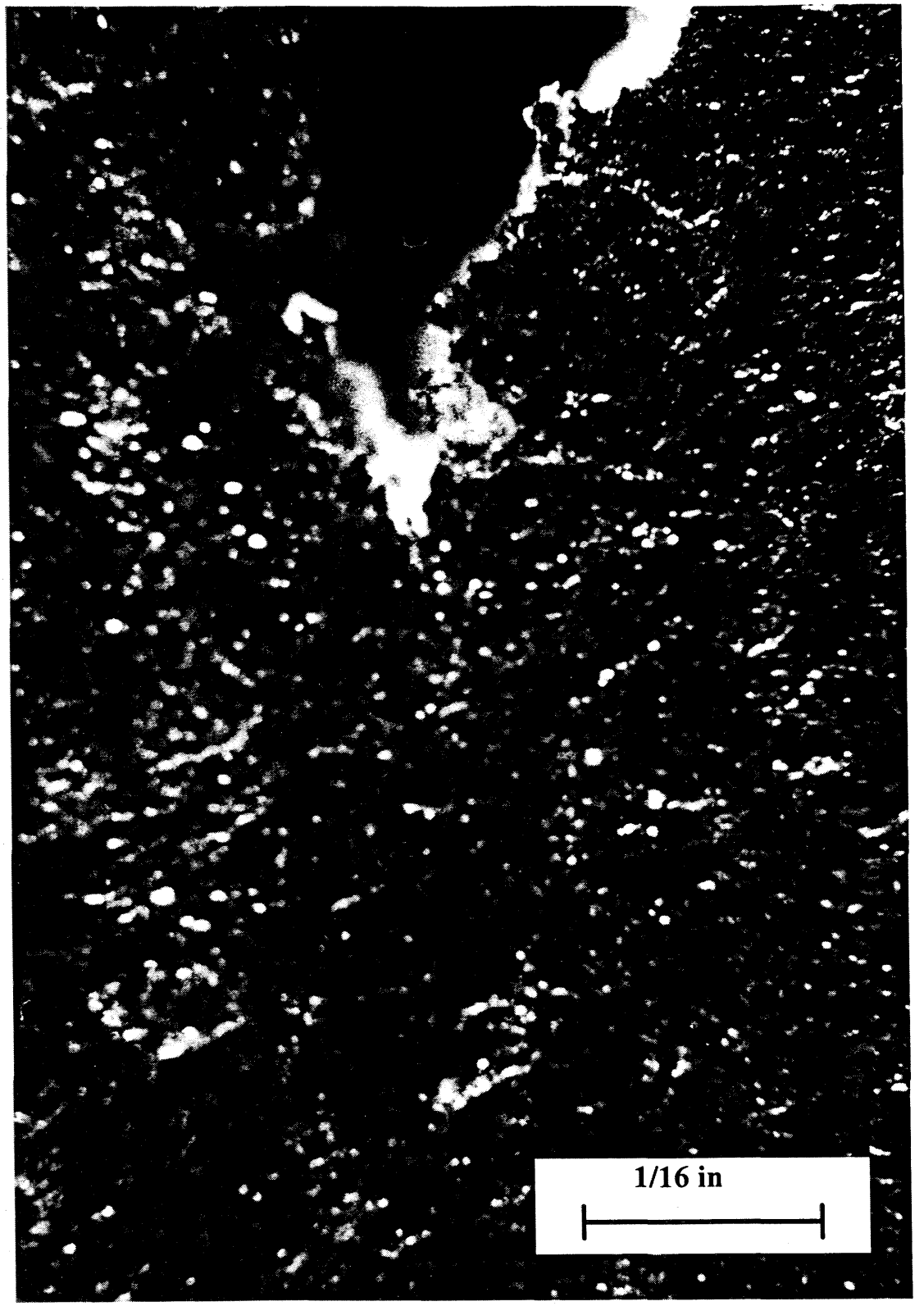


Figure 11b.

Figure 12b.

Figure 13b.

Figure 14b.

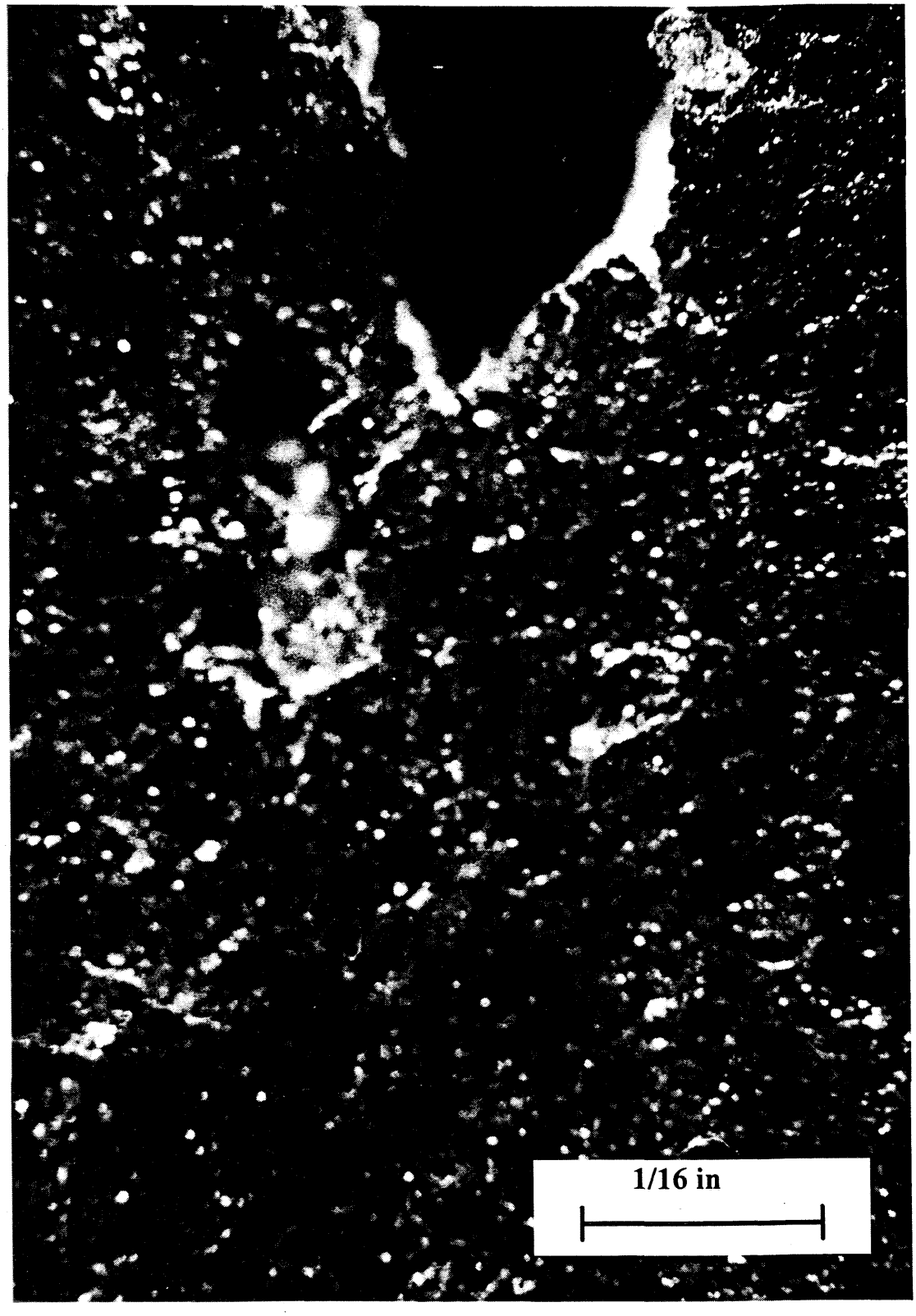


Figure 15b.

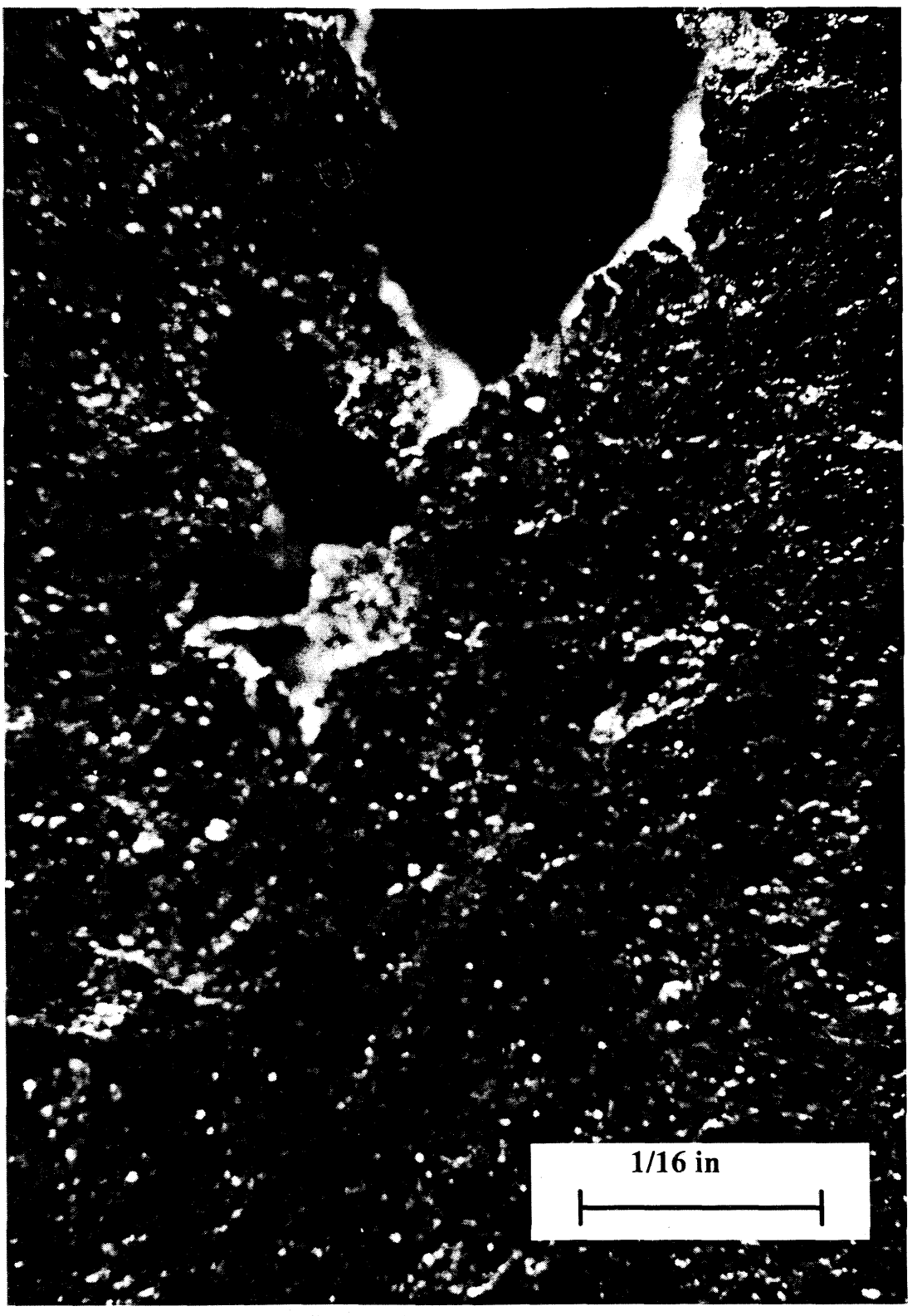


Figure 16b.

Figure 17b.

Figure 18b.

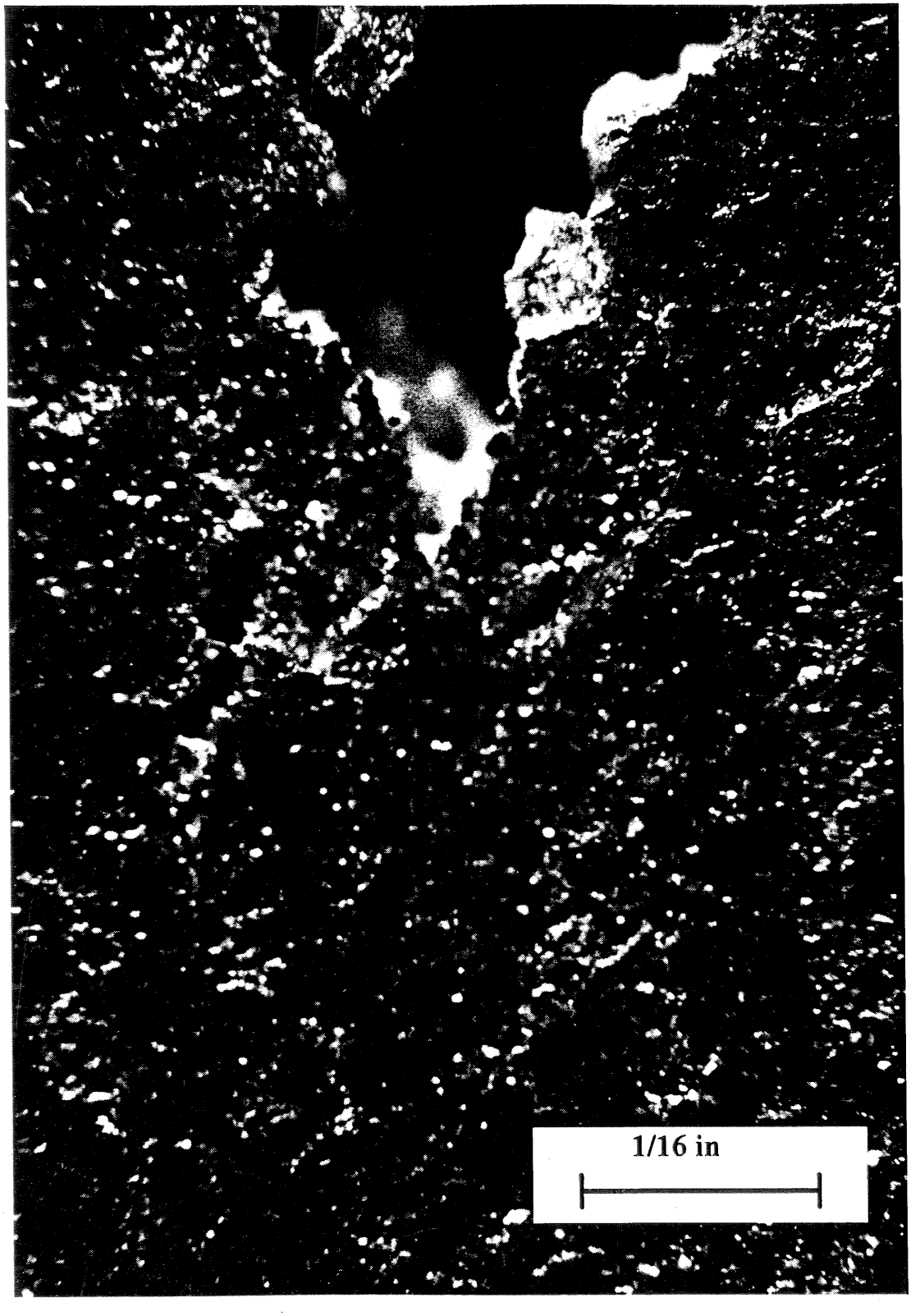


Figure 19b.

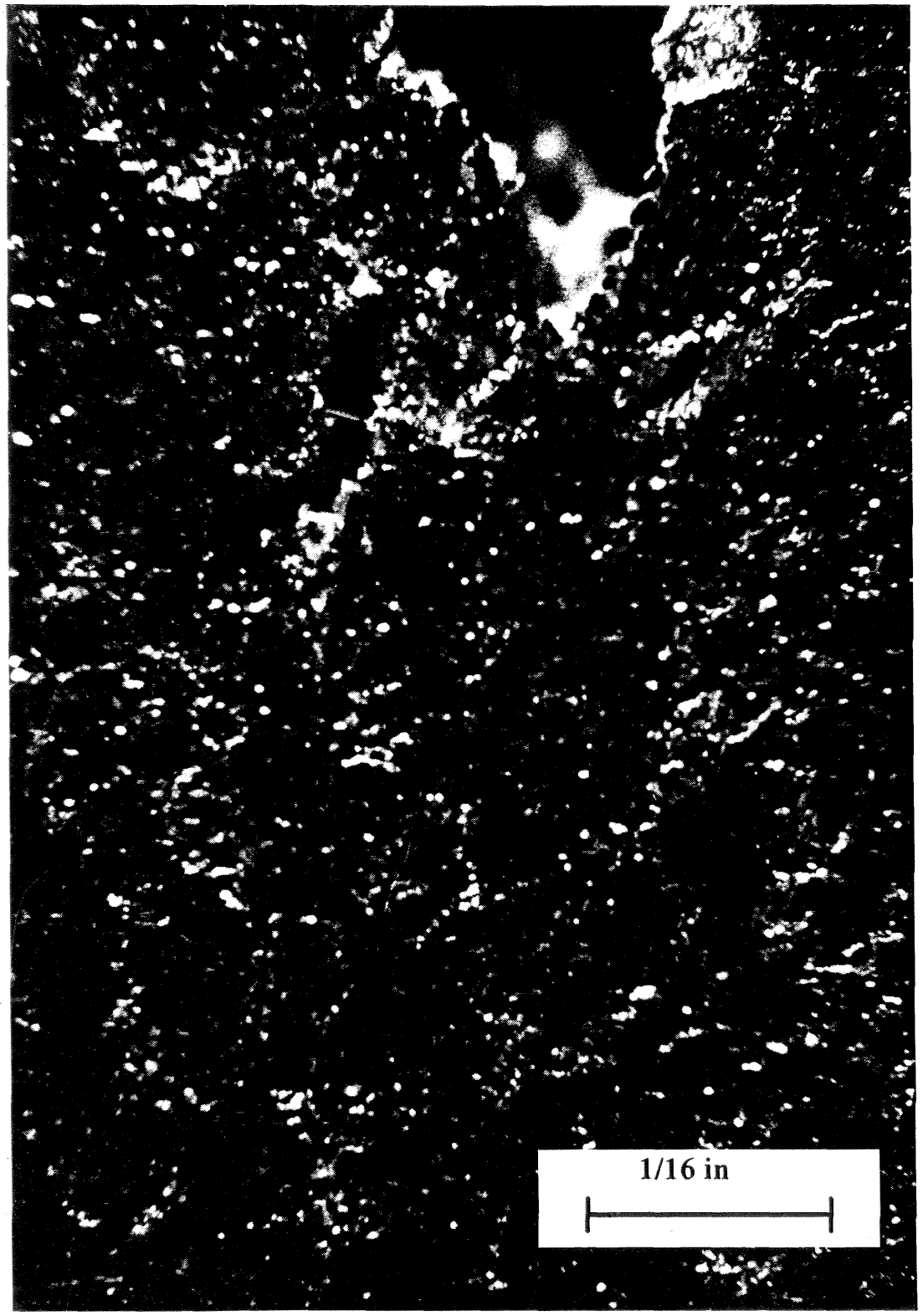


Figure 20b.

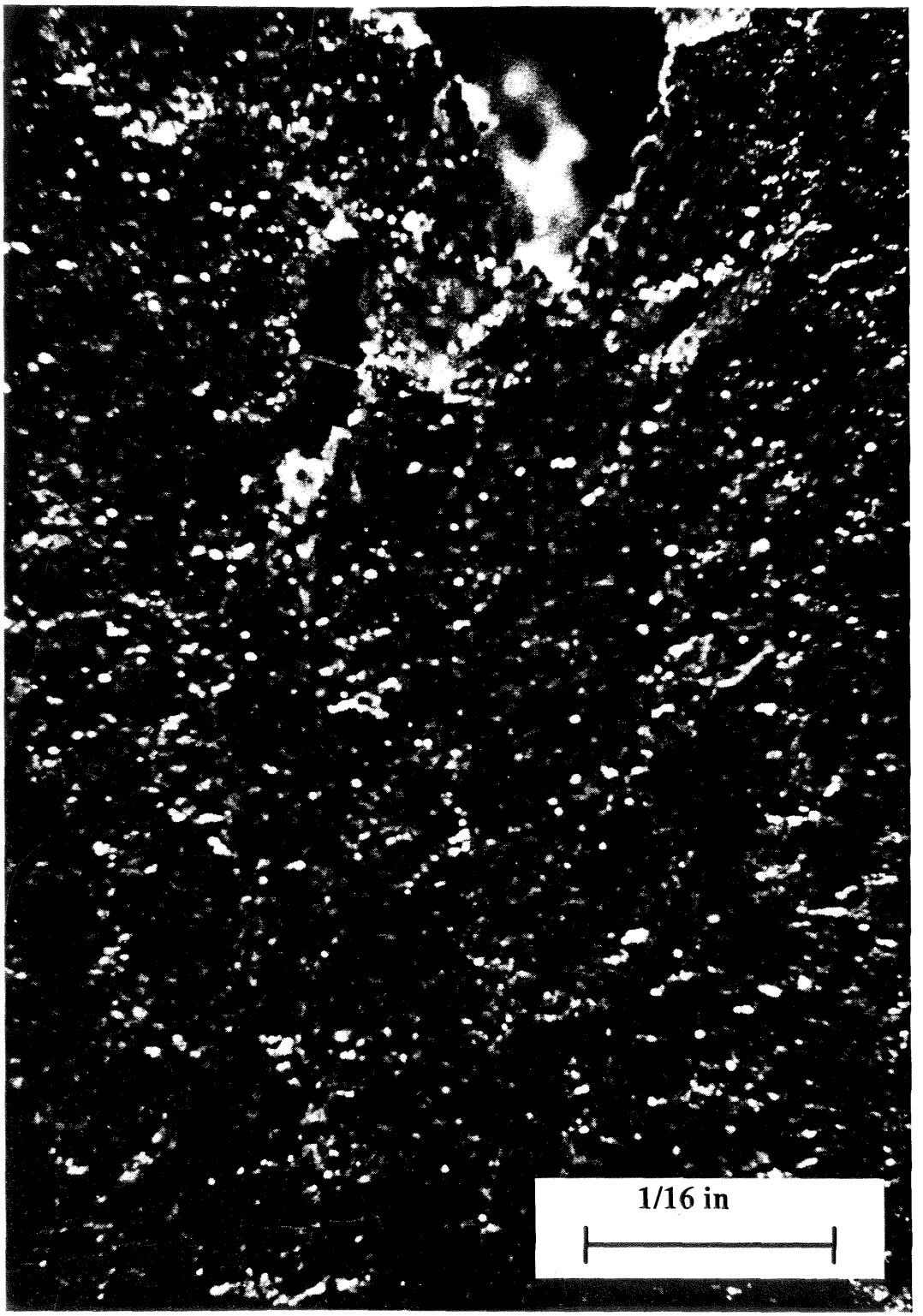


Figure 21b.

Figure 22b.

Figure 23b.

Figure 24b.

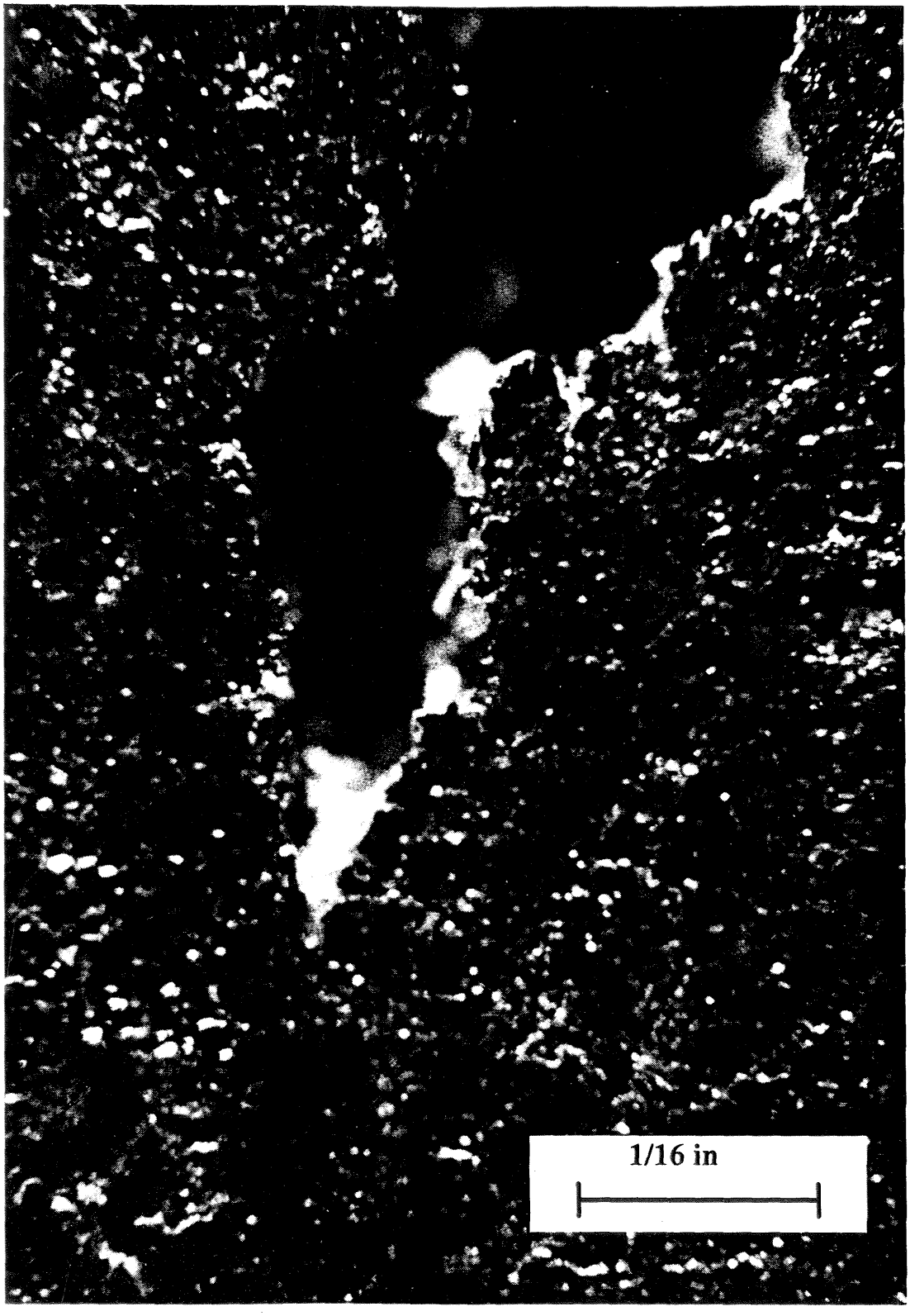


Figure 25b.

Figure 26b.

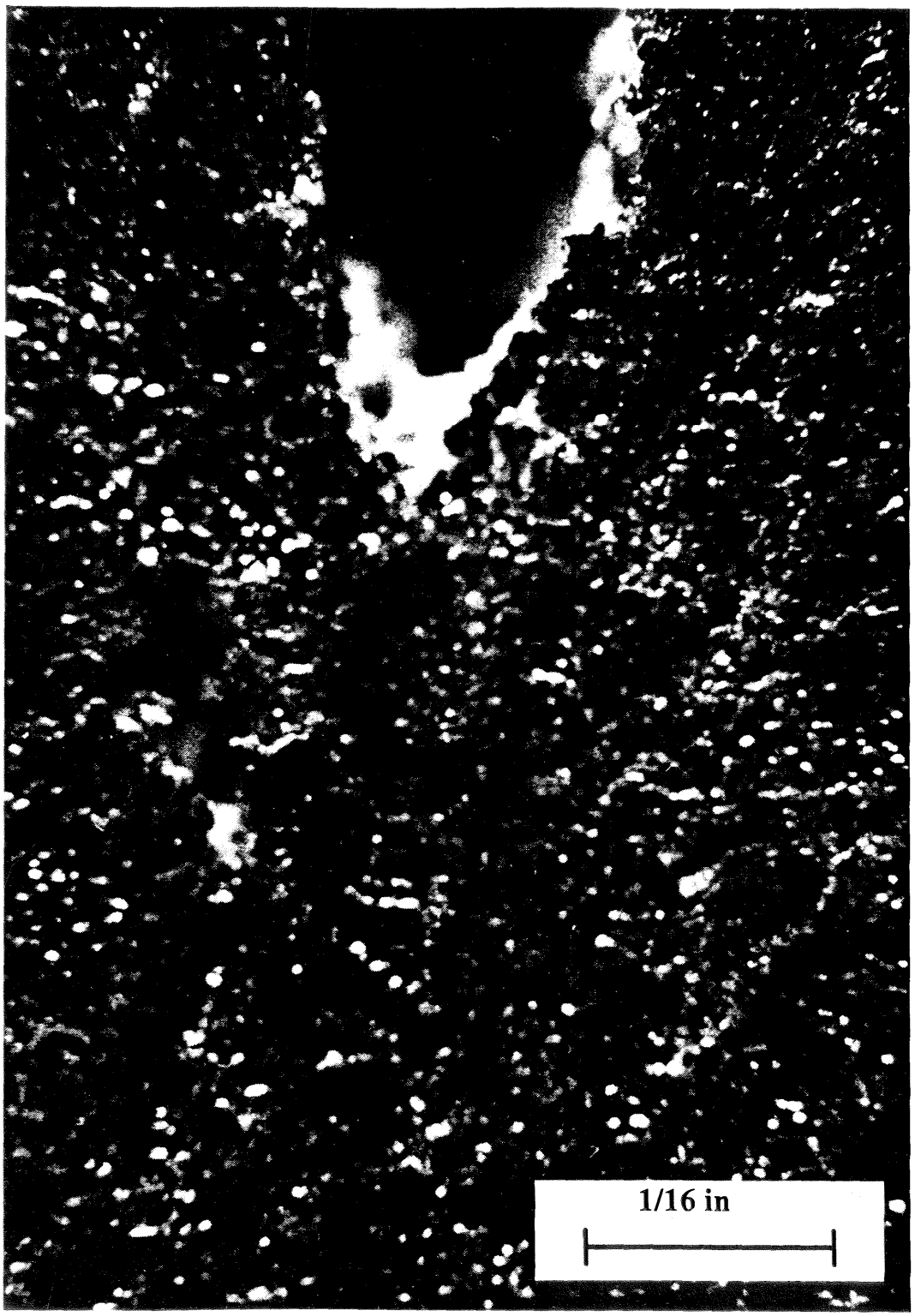


Figure 27b.

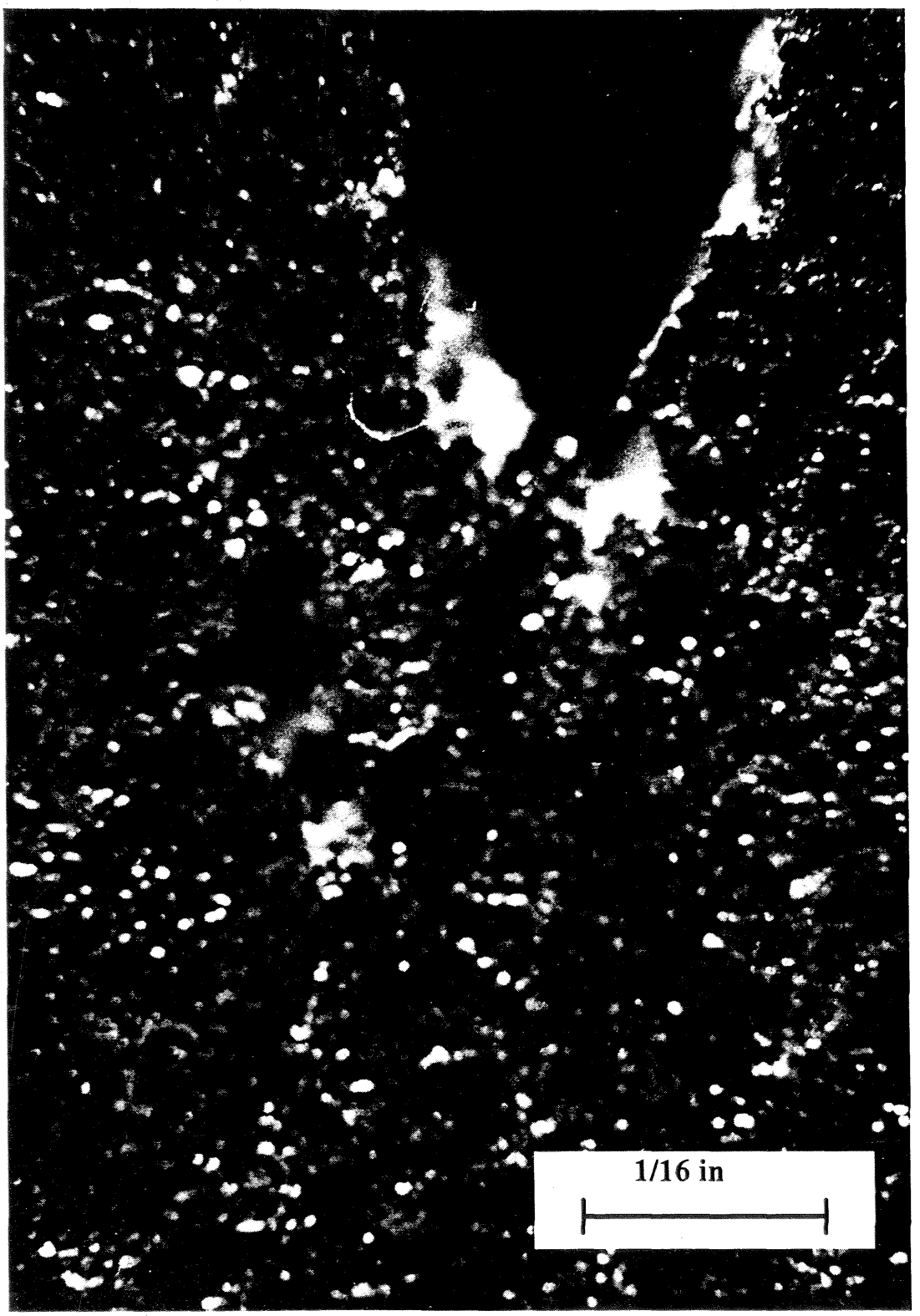


Figure 28b.

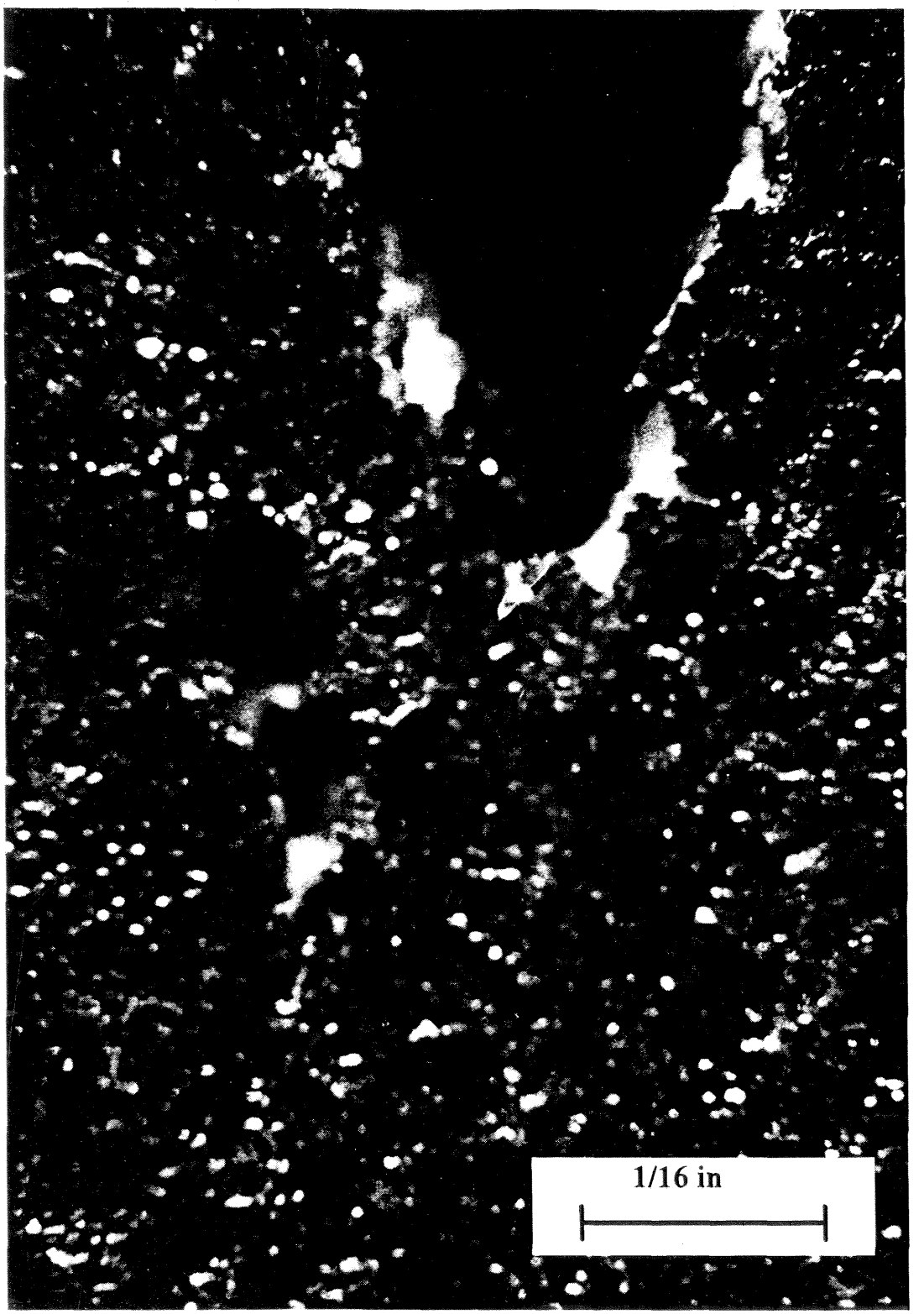


Figure 29b.

Figure 30b.

Figure 31b.

Figure 32b.

Figure 33b.

Figure 34b.

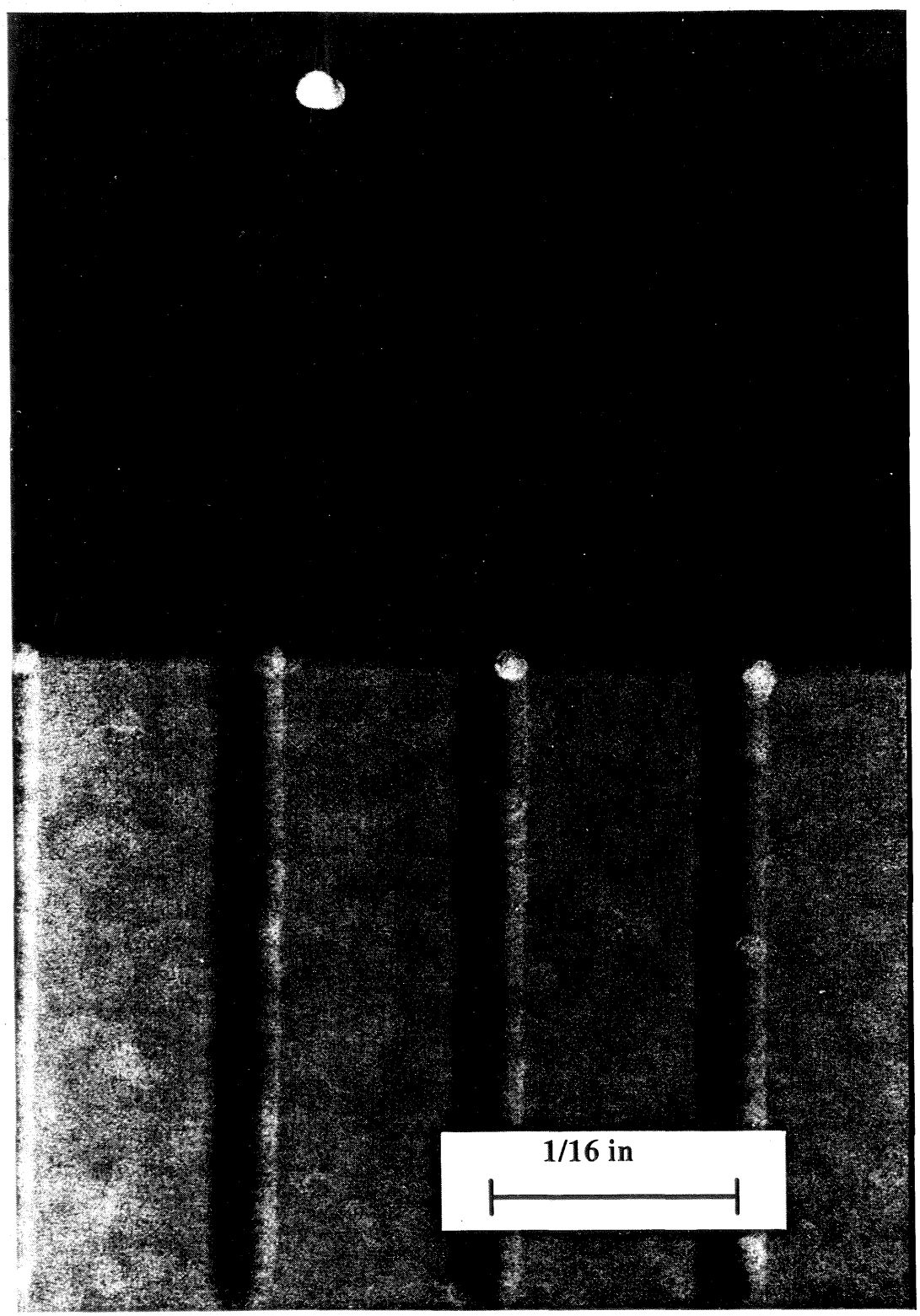


Figure 35b.

APPENDIX C

PROGRAM LISTING

For a simple deformation process, we establish three pictures of the body each associated with the configurations 1, 2 and 3 of the sequential deformations. The DIC program is successful in giving the deformation fields between image 1 and 2 (deformation A) and between images 2 and 3 (deformation B). However, the strains between images 1 and 3 (global deformation) are larger than those that lead to the convergent of the correlation algorithm. We determine the deformation fields for the global deformation, corresponding to images 1 and 3, by using the results the DIC program provides for the deformations A and B. The Large Deformation Digital Image Correlation method is applied in two steps:

1) In the first step, strain and displacement fields for the deformation between two consecutive images are computed by using the DIC program previously described. The DIC program requires the input file (**jan1096.inp**), containing parameters for the correlation algorithm, gray level image files of the deformed and the undeformed body (**hong0.gray** and **hong1.gray**), a file with the pixel coordinates of the undeformed image, in which strains and displacements are calculated, (**unknown.dat**). The output file of the DIC program, which contains the results of the correlation procedure is named **rawdata.dat**. The DIC program computes the strains and displacement at every point prescribed in the file unknown.dat, using the results from the previous point as a initial guess. For the first point, a initial guess is provided by the initial guess file (**guess.dat**). This method requires that all the points are next to the previous one. Since in our problem there are discontinuities in the domain (cracks), the correlation algorithm is not able to converge at some of these discontinuities, and therefore no initial guess is available for the next points. To overcome this difficulty, the DIC program was applied as seen in Figure 1c, and second performing the correlations of the points along the initiation strip A in Figure 1c. As there are no discontinuities along this initiation path (Strip A), the DIC

program is successful in determining the right results. The strain and displacement values obtained along the strip A are used as initial guess for the series of strips B_1, B_2, ..B_5 depicted in Figure 1c. The procedure to compute strain and displacement fields for two images (**hong0.gray** and **hong1.gray**) is the following:

a) Measure the displacement of the initial point with the help of an image editor (xv), and put this value in the file **guess.dat**.

b) Compile the program **2dpre10.f** that will create the file **unknown.dat** containing the pixel coordinates of the points along the strip A (Fig 1c).

c) Compile the program **2dpre111.f** that will generate the files **data1.dat**, **data2.dat**... containing the pixel coordinates of the points along the strips B (Fig 1c).

d) Run the c-shell file **2drun** to sequentially operate the DIC program.

e) The strain and displacement fields will be stored in the file **101.dat**, which is a copy of **rawdata.dat**.

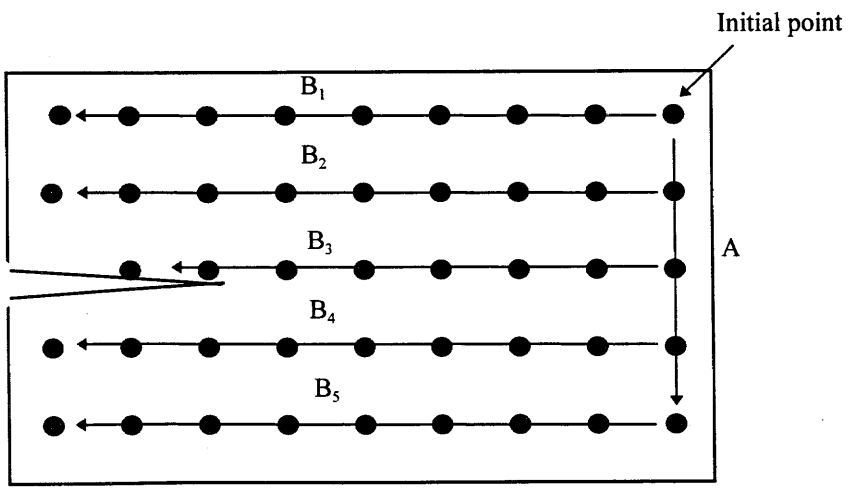


Figure 1c: Arrangement of points.

In the same way the strain and displacement fields for deformation B are computed, and the results are stored in **102.dat**.

2) The second step of the Large Deformation Digital Image Correlation method is to combine the results for deformations A and B to obtain the global deformation. If we assume that the file **101.dat** contains the results from the correlation during deformation A, while **102.dat** contains the same information for deformation B, the task is done with the program **add.f** as follows:

a) Move the file **101.dat** to **uno.dat**.
b) Run the program **101.f** to modify the file **uno.dat**.
c) Move the resultant file from last step (**dat**) to **uno.dat**.
d) Move **102.dat** to **dos.dat**.
e) Run the program **add.f**.

The result of this operations are the strain and displacement fields of the global deformation.

In the following pages it is displayed all these programs.

```
2drun


2dpre10
foreach j (1)
  mv unknow$j.dat unknown.dat
  hong_corr
  mv rawdata.dat pre.dat
  2dpre111
  mv data1.dat unknown.dat
  hong_corr
  mv rawdata.dat 10$j.dat
  foreach i (2 3 4 5 6 7 8 9 10 11 )
    echo data$i.dat
    mv data$i.dat unknown.dat
    hong_corr
    cat rawdata.dat >> 10$j.dat
    end
end
```

```
      program dpre10

      implicit double precision (a-h,o-z)


      open ( unit = 1, file = 'unknow1.dat', status = 'new')
      open ( unit = 11,file =  'guess.dat' , status = 'old')

      i = 0

      do while (.true.)
      i = i + 1

      read(11,*,end=10) u,v

      write(i,*)'3'
      write(i,*)'640  480'
      write(i,'(a10)')'hong0.gray'
      if (i.lt.10) then
      write(i,'(a4,i1,5a)')'hong',i,'.gray'
                  end if
      if (i.ge.10) then
      write(i,'(a4,i2,5a)')'hong',i,'.gray'
                  end if

      write(i,'(11a)')'jan1096.inp'
      write(i,*)u,v,0.,0.,0.,0.
      write(i,*)'211 212'

      do j = 0 , 210
       write(i,*) 260 , 30+j*2
      end do
      close( unit = i )
      end do
   10 continue
      close (unit =11)
       stop
      end
```

```
      program dpre111

      implicit double precision (a-h,o-z)

      character *12 undeformed
      character *12 deformed

      open ( unit = 1, file = 'pre.dat', status = 'old')

      open ( unit = 2, file = 'data1.dat', status = 'new')
      open ( unit = 3, file = 'data2.dat', status = 'new')
      open ( unit = 4, file = 'data3.dat', status = 'new')
      open ( unit = 5, file = 'data4.dat', status = 'new')
      open ( unit = 6, file = 'data5.dat', status = 'new')
      open ( unit = 7, file = 'data6.dat', status = 'new')
      open ( unit = 8, file = 'data7.dat', status = 'new')
      open ( unit = 9, file = 'data8.dat', status = 'new')
      open ( unit = 10, file = 'data9.dat', status = 'new')
      open ( unit = 11, file = 'data10.dat', status = 'new')
      open ( unit = 47, file = 'unknown.dat', status = 'old')


      read (47,*) j
      read (47,*) j,k
      read (47,'(10a)') undeformed
      read (47,'(10a)') deformed
      i = 1

      do while (.true.)
      i = i + 1

      read(1,*,end=10) ix,iy,u,v,ex,ey,exy,ux,uy,vx,vy,junk3

      write(i,*)'3'
      write(i,*)'640  480'
      write(i,'(10a)') undeformed
      write(i,'(10a)') deformed
      write(i,'(11a)')'jan1096.inp'
      write(i,*)u,v,0.,0.,0.,0.
      write(i,*)'45 46'

      do j = 0 , 44
       write(i,*) 470 - j * 10 , iy
      end do
      close( unit = i )
      end do
   10 continue
      close (unit = 47)
      close (unit = 1)
       stop
      end
```

```
      program add1

      implicit double precision (a-h,o-z)

      dimension field1(50,50,22)
      dimension field2(50,50,22)
      dimension field3(50,50,22)

      open ( unit = 1, file = 'uno.dat', status = 'old')
      open ( unit = 2, file = 'dos.dat', status = 'old')
      open ( unit = 3, file = 'dat', status = 'new')


      i0 = 20
      j0 = 60
      imax = 470
      jmax = 460
      istep = 10
      jstep = 10

      xmax = real (imax)
      ymax = real (jmax)
      xmin = real (i0)
      ymin = real (j0)
      xstep = real (istep)
      ystep = real (jstep)

      pu0 = .188d0
      pu1 = .00037d0
      pux0 = -.0014d0
      pux1 = .08d0
      pux2 = .0d0

      do i = 1 , 50
      do j = 1 , 50
        field1(i,j,19) = 4
        field2(i,j,12) = 4
        field3(i,j,19) = 4
        field3(i,j,1) = xmin + xstep * (i - 1)
        field3(i,j,2) = ymin + ystep * (j - 1)
      end do
      end do

c-------------------------------------------------------------
c
c     read information from first field
c
c
      do while (.true.)

      read(1,*,end = 10) i1 , j1 , u1 , v1 , ex1 , ey1 , exy1,
     & ux,vy,uy,vx,eex,eey,eexy,eux,evy,euy,evx,junk3,eu,ev

      i = ( i1 - i0 ) / istep + 1
      j = ( j1 - j0 ) / jstep + 1

      field1(i,j,1)  = i1
      field1(i,j,2)  = j1
      field1(i,j,3)  = u1
      field1(i,j,4)  = v1
      field1(i,j,5)  = ex1
      field1(i,j,6)  = ey1
      field1(i,j,7)  = exy1
      field1(i,j,8)  = ux
      field1(i,j,9)  = vy
      field1(i,j,10) = uy
```

```
      field1(i,j,11) = vx
      field1(i,j,12) = eex
      field1(i,j,13) = eey
      field1(i,j,14) = eexy
      field1(i,j,15) = eux
      field1(i,j,16) = evy
      field1(i,j,17) = euy
      field1(i,j,18) = evx
      field1(i,j,19) = junk3
      field1(i,j,20) = eu
      field1(i,j,21) = ev

      end do
   10 continue

c-----------------------------------------------------------------
c
c     read information from second field
c
c
      do while (.true.)

      read(2,*,end = 20) i1 , j1 , u1 , v1 , ex1 , ey1 , exy1,
     & ux,vy,uy,vx,junk3

      i = ( i1 - i0 ) / istep + 1
      j = ( j1 - j0 ) / jstep + 1

      field2(i,j,1)  = i1
      field2(i,j,2)  = j1
      field2(i,j,3)  = u1
      field2(i,j,4)  = v1
      field2(i,j,5)  = ex1
      field2(i,j,6)  = ey1
      field2(i,j,7)  = exy1
      field2(i,j,8)  = ux
      field2(i,j,9)  = vy
      field2(i,j,10) = uy
      field2(i,j,11) = vx
      field2(i,j,12) = junk3
      end do
   20 continue

c-----------------------------------------------------------------
c
c     main loop for addition of fields
c
c
      do i = 1 , 50
      do j = 1 , 50


      if (field1(i,j,19).lt.4) then    ! if -> 1

      x = field1(i,j,1)
      y = field1(i,j,2)
      u = field1(i,j,3)
      v = field1(i,j,4)
      x = x + u
      y = y + v

c                    if -> 2
       if (x.ge.xmin.and.y.ge.ymin.and.x.le.xmax.and.y.le.ymax) then

      i1 = int((x - xmin)/xstep) + 1
      i2 = i1 + 1
```

```
i3 = i1 + 2
i4 = i1 - 1
j1 = int((y - ymin)/ystep) + 1
j2 = j1 + 1
j3 = j1 + 2
j4 = j1 - 1
x1 = field2(i1,j1,1)
x2 = field2(i2,j1,1)
y1 = field2(i1,j1,2)
y2 = field2(i1,j2,2)
x3 = field2(i3,j1,1)
x4 = field2(i4,j1,1)
y3 = field2(i1,j3,2)
y4 = field2(i1,j4,2)

k1 = field2(i1,j1,12)
k2 = field2(i2,j1,12)
k3 = field2(i1,j2,12)
k4 = field2(i2,j2,12)

k1 = max (k1,k2,k3,k4)

field3(i,j,19) = k1

if (k1.lt.4) then   ! if -> 3

u1 = field2(i1,j1,3) * (x2-x) + field2(i2,j1,3) * (x-x1)
u1 = u1 / xstep
u2 = field2(i1,j2,3) * (x2-x) + field2(i2,j2,3) * (x-x1)
u2 = u2 / xstep
uf = u1*(y2-y) + u2*(y-y1)
uf = uf / ystep
field3(i,j,3) = field1(i,j,3) + uf

v1 = field2(i1,j1,4) * (x2-x) + field2(i2,j1,4) * (x-x1)
v1 = v1 / xstep
v2 = field2(i1,j2,4) * (x2-x) + field2(i2,j2,4) * (x-x1)
v2 = v2 / xstep
vf = v1*(y2-y) + v2*(y-y1)
vf = vf / ystep
field3(i,j,4) = field1(i,j,4) + vf


ux1 = field2(i1,j1,8) * (x2-x) + field2(i2,j1,8) * (x-x1)
ux1 = ux1 / xstep
ux2 = field2(i1,j2,8) * (x2-x) + field2(i2,j2,8) * (x-x1)
ux2 = ux2 / xstep
uxf = ux1*(y2-y) + ux2*(y-y1)
uxf = uxf / ystep
field3(i,j,8) = field1(i,j,8) + uxf

vy1 = field2(i1,j1,9) * (x2-x) + field2(i2,j1,9) * (x-x1)
vy1 = vy1 / xstep
vy2 = field2(i1,j2,9) * (x2-x) + field2(i2,j2,9) * (x-x1)
vy2 = vy2 / xstep
vyf = vy1*(y2-y) + vy2*(y-y1)
vyf = vyf / ystep
field3(i,j,9) = field1(i,j,9) + vyf

uy1 = field2(i1,j1,10) * (x2-x) + field2(i2,j1,10) * (x-x1)
uy1 = uy1 / xstep
uy2 = field2(i1,j2,10) * (x2-x) + field2(i2,j2,10) * (x-x1)
uy2 = uy2 / xstep
uyf = uy1*(y2-y) + uy2*(y-y1)
uyf = uyf / ystep
field3(i,j,10) = field1(i,j,10) + uyf
```

```
 vx1 = field2(i1,j1,11) * (x2-x) + field2(i2,j1,11) * (x-x1)
 vx1 = vx1 / xstep
 vx2 = field2(i1,j2,11) * (x2-x) + field2(i2,j2,11) * (x-x1)
 vx2 = vx2 / xstep
 vxf = vx1*(y2-y) + vx2*(y-y1)
 vxf = vxf / ystep
 field3(i,j,11) = field1(i,j,11) + vxf


 ux = field1(i,j,8)
 vy = field1(i,j,9)
 uy = field1(i,j,10)
 vx = field1(i,j,11)

 field3(i,j,8) = field3(i,j,8) + uxf*ux + uyf*vx
 field3(i,j,9) = field3(i,j,9) + vyf*vy + vxf*uy
 field3(i,j,10) = field3(i,j,10) + uxf*uy + uyf*vy
 field3(i,j,11) = field3(i,j,11) + vxf*ux + vx*vyf

 ex = field3(i,j,8) + .5d0*(field3(i,j,8)*field3(i,j,8)+
& field3(i,j,11)*field3(i,j,11))
 ey = field3(i,j,9) +.5d0*(field3(i,j,9)*field3(i,j,9)+
& field3(i,j,10)*field3(i,j,10))
 exy = .5d0*(field3(i,j,10)+field3(i,j,11))+.5d0*(field3(i,j,8)*
& field3(i,j,10)+
& field3(i,j,9)*field3(i,j,11))

 field3(i,j,5) = ex
 field3(i,j,6) = ey
 field3(i,j,7) = exy

 k1  = field2(i1,j4,12)
 k2  = field2(i1,j1,12)
 k3  = field2(i1,j2,12)
 k4  = field2(i2,j1,12)
 k5  = field2(i4,j1,12)

 k1 =  max(k1,k2,k3,k4,k5)

  if (k1.eq.0.and.field1(i,j,19).eq.0.and.j1.gt.1) then        ! if -> 4


 duxxx = (field2(i4,j1,8)  - 2.d0*field2(i1,j1,8)  +
& field2(i2,j1,8))
 duxxx = duxxx / 2.d0

 duxyy = (field2(i1,j4,8)  - 2.d0*field2(i1,j1,8)  +
& field2(i1,j2,8))
 duxyy = duxyy / 2.d0

 dvxyy = (field2(i1,j4,11) - 2.d0*field2(i1,j1,11) +
& field2(i1,j2,11))
 dvxyy = dvxyy / 2.d0

 dvxxx = (field2(i4,j1,11) - 2.d0*field2(i1,j1,11) +
& field2(i2,j1,11))
 dvxxx = dvxxx / 2.d0


 duyyy = (field2(i1,j4,10) - 2.d0*field2(i1,j1,10) +
& field2(i1,j2,10))
 duyyy = duyyy / 2.d0

 duyxx = (field2(i4,j1,10) - 2.d0*field2(i1,j1,10) +
& field2(i2,j1,10))
```

```
      duyxx = duyxx / 2.d0

      dvyxx = (field2(i4,j1,9)  - 2.d0*field2(i1,j1,9)  +
     & field2(i2,j1,9))
      dvyxx = dvyxx / 2.d0

      dvyyy = (field2(i1,j4,9)  - 2.d0*field2(i1,j1,9)  +
     & field2(i1,j2,9))
      dvyyy = dvyyy / 2.d0


c      write(10,*) i,j,duxxx,duxyy,dvyxx,dvyyy,duyxx,
c     & duyyy,dvxxx,dvxyy,x-x1,y-y1,x,x1


      eux1 = field1(i,j,15)
      evy1 = field1(i,j,16)
      euy1 = field1(i,j,17)
      evx1 = field1(i,j,18)

      eux2 = sqrt(duxxx**2 + duxyy**2 +
     & (pux0+pux1*uxf+pux2*uxf**2)**2)
      evy2 = sqrt(dvyxx**2 + dvyyy**2 +
     & (pux0+pux1*uxf+pux2*uxf**2)**2)
      euy2 = sqrt(duyxx**2 + duyyy**2 +
     & (pux0+pux1*uxf+pux2*uxf**2)**2)
      evx2 = sqrt(dvxxx**2 + dvxyy**2 +
     & (pux0+pux1*uxf+pux2*uxf**2)**2)

      ux = field1(i,j,8)
      vy = field1(i,j,9)
      uy = field1(i,j,10)
      vx = field1(i,j,11)

      eux = sqrt(((1.d0+uxf)*eux1)**2 + ((1.d0+ux )*eux2)**2 +
     & (uyf*evx1)**2 + (vx*euy2)**2)
      euy = sqrt(((1.d0+uxf)*euy1)**2 + ((1.d0+vy )*euy2)**2 +
     & (uy*eux2)**2 + (uyf*evy1)**2)
      evx = sqrt(((1.d0+ux )*evx2)**2 + ((1.d0+vyf)*evx1)**2 +
     & (eux1*vxf)**2 + (vx*evy2)**2)
      evy = sqrt(((1.d0+vyf)*evy1)**2 + ((1.d0+vy )*evy2)**2 +
     & (vxf*euy1)**2 + (uy*evx2)**2)

      field3(i,j,15) = eux
      field3(i,j,16) = evy
      field3(i,j,17) = euy
      field3(i,j,18) = evx

      eex  = sqrt(((1.d0+ux)*eux)**2 + (vx*evx)**2)
      eey  = sqrt(((1.d0+vy)*evy)**2 + (uy*euy)**2)
      eexy = sqrt((eux*.5d0*uy)**2 + (evy*.5d0*ux)**2+
     & (euy*.5d0*(1.d0+ux))**2  + (evx*.5d0*(1.d0+vy))**2 )

      field3(i,j,12) = eex
      field3(i,j,13) = eey
      field3(i,j,14) = eexy
                              else
      field3(i,j,19) = 1.d0

                    end if                  ! if -> 4


      end if                                ! if -> 3
      end if                                ! if -> 2

      end if                                ! if -> 1
```

```
c       field3(i,j,1) = field1(i,j,1)
c       field3(i,j,2) = field1(i,j,2)

        end do
        end do


        do j = 1,41
        do i = 1,46
c        write(3,'(19e15.6)') (field1(i,j,l),l=1,19)
c        write(3,'(12e15.6)') (field2(i,j,l),l=1,12)
        write(3,'(21e15.6)') (field3(i,j,l),l=1,21)
c        write(3,*)' '

        end do
        end do
        close(unit = 1)
        close(unit = 2)
        close(unit = 3)

        stop
        end
```

```
      program add2

      implicit double precision (a-h,o-z)

      dimension field1(50,50,22)
      dimension field2(50,50,22)
      dimension field3(50,50,22)

      open ( unit = 1, file = 'uno.dat', status = 'old')
      open ( unit = 2, file = 'dos.dat', status = 'old')
      open ( unit = 3, file = 'dat', status = 'new')


      i0 = 20
      j0 = 60
      imax = 470
      jmax = 460
      istep = 10
      jstep = 10

      xmax = real (imax)
      ymax = real (jmax)
      xmin = real (i0)
      ymin = real (j0)
      xstep = real (istep)
      ystep = real (jstep)

      pu0 = .188d0
      pu1 = .00037d0
      pux0 = -.0014d0
      pux1 = .08d0
      pux2 = .0d0

      do i = 1 , 50
      do j = 1 , 50
        field1(i,j,19) = 4
        field2(i,j,12) = 4
        field3(i,j,19) = 4
        field3(i,j,1) = xmin + xstep * (i - 1)
        field3(i,j,2) = ymin + ystep * (j - 1)

      end do
      end do

c-----------------------------------------------------------------
c
c     read information from first field
c
c
      do while (.true.)

      read(1,*,end = 10) i1 , j1 , u1 , v1 , ex1 , ey1 , exy1,
     & ux,vy,uy,vx,eex,eey,eexy,eux,evy,euy,evx,junk3,eu,ev

      i = ( i1 - i0 ) / istep + 1
      j = ( j1 - j0 ) / jstep + 1

      field1(i,j,1)  = i1
      field1(i,j,2)  = j1
      field1(i,j,3)  = u1
      field1(i,j,4)  = v1
      field1(i,j,5)  = ex1
      field1(i,j,6)  = ey1
      field1(i,j,7)  = exy1
      field1(i,j,8)  = ux
      field1(i,j,9)  = vy
```

```
      do i = 1 , 50
      do j = 1 , 50


      i1 = i
      i2 = i1 + 1
      i4 = i1 - 1
      j1 = j
      j2 = j1 + 1
      j4 = j1 - 1

      k1 = field3(i1,j1,19)
      k2 = field3(i1,j2,19)
      k3 = field3(i1,j4,19)
      k4 = field3(i2,j1,19)
      k5 = field3(i4,j1,19)

      k1 =  max(k1,k2,k3,k4,k5)

      if (k1.lt.4) then     !  if -> 1b

c
c                                    du**2     Dx**2
c     this derivatives are  duxx = ------- * -------   and so on
c                                    dx**2       2
c

      duxx = (field3(i4,j1,3)  - 2.d0*field3(i1,j1,3)  +
     & field3(i2,j1,3))
      duxx= duxx / 2.d0

      dvxx = (field3(i4,j1,4)  - 2.d0*field3(i1,j1,4)  +
     & field3(i2,j1,4))
      dvxx = dvxx / 2.d0

      dvyy = (field3(i1,j4,4)  - 2.d0*field3(i1,j1,4)  +
     & field3(i1,j2,4))
      dvyy = dvyy / 2.d0

      duyy = (field3(i1,j4,3)  - 2.d0*field3(i1,j1,3)  +
     & field3(i1,j2,3))
      duyy = duyy  / 2.d0


      u = field3(i,j,3)
      v = field3(i,j,4)


      field3(i,j,20) = sqrt((duxx + duyy)**2 + (pu0 + pu1*u)**2 )  !eu
      field3(i,j,21) = sqrt((dvxx + dvyy)**2 + (pu0 + pu1*v)**2 )  !ev

      else
       if (field3(i,j,19).eq.0)  field3(i,j,19) = 1
      end if               !  if -> 1b

      end do
      end do

c**********
c      do j = 1,50
c      do i = 1,50
c      write(30,'(21e15.6)') (field1(i,j,l),l=1,19)
c      write(3,'(12e15.6)') (field2(i,j,l),l=1,12)
c      write(3,'(21e15.6)') (field3(i,j,l),l=1,21)
c      write(3,*)' '
c
```

```
c       end do
c       end do
c       close(unit = 1)
c       close(unit = 2)
c       close(unit = 3)
c
c       stop
c************
c-----------------------------------------------------------------
c
c      computation of errors in gradients of displacements
c
c

       do i = 1 , 50
       do j = 1 , 50


       i1 = i
       i2 = i1 + 1
       j1 = j
       j2 = j1 + 1

       x1 = field3(i1,j1,1)
       x2 = field3(i2,j1,1)
       y1 = field3(i1,j1,2)
       y2 = field3(i1,j2,2)

       k1 = field3(i1,j1,19)
       k2 = field3(i2,j1,19)
       k3 = field3(i1,j2,19)
       k4 = field3(i2,j2,19)

       k1 = max (k1,k2,k3,k4)

       if (k1.lt.1) then     ! if -- 1d

       eu1 = field3(i1,j1,20)
       eu2 = field3(i2,j1,20)
       eu3 = field3(i1,j2,20)
       eu4 = field3(i2,j2,20)

       ev1 = field3(i1,j1,21)
       ev2 = field3(i2,j1,21)
       ev3 = field3(i1,j2,21)
       ev4 = field3(i2,j2,21)


       eux = sqrt( eu2**2 + eu1**2 ) / xstep
       euy = sqrt( eu3**2 + eu1**2 ) / ystep
       evx = sqrt( ev2**2 + ev1**2 ) / xstep
       evy = sqrt( ev3**2 + ev1**2 ) / ystep

       eex = sqrt((eux*(1.d0+ux))**2+(vx*evx)**2)
       eey = sqrt((evy*(1.d0+vy))**2+(uy*euy)**2)
       eexy = sqrt((eux*.5d0*uy)**2+(euy*.5d0*(1.d0+ux))**2+
      $ (evx*.5d0*(1.d0+vy))**2+(evy*.5d0*vx)**2)

       field3(i,j,12) = eex
       field3(i,j,13) = eey
       field3(i,j,14) = eexy
       field3(i,j,15) = eux
       field3(i,j,16) = evy
       field3(i,j,17) = euy
       field3(i,j,18) = evx
```

```
      else
         if (field3(i,j,19).lt.4) field3(i,j,19) = 3
      end if                 !  if -- 1d


      end do
      end do

c-----------------------------------------------------------------------
c
c     print out the results
c
c


      do j = 1 , 41
      do i = 1 , 46

c      write(30,'(21e15.6)') (field1(i,j,l),l=1,21)
c      write(3,'(12e15.6)') (field2(i,j,l),l=1,12)
      write(3,'(21e15.6)') (field3(i,j,l),l=1,21)
c      write(3,*)' '

      end do
      end do
      close(unit = 1)
      close(unit = 2)
      close(unit = 3)

      stop
      end
```

```
      program add3

      implicit double precision (a-h,o-z)

      dimension field1(50,50,22)
      dimension field2(50,50,22)
      dimension field3(50,50,22)

      open ( unit = 1, file = 'uno.dat', status = 'old')
      open ( unit = 2, file = 'dos.dat', status = 'old')
      open ( unit = 3, file = 'dat', status = 'new')


      i0 = 20
      j0 = 60
      imax = 470
      jmax = 460
      istep = 10
      jstep = 10

      xmax = real (imax)
      ymax = real (jmax)
      xmin = real (i0)
      ymin = real (j0)
      xstep = real (istep)
      ystep = real (jstep)

      pu0 = .188d0
      pu1 = .00037d0
      pux0 = -.0014d0
      pux1 = .08d0
      pux2 = .0d0

      do i = 1 , 50
      do j = 1 , 50
        field1(i,j,19) = 4
        field2(i,j,12) = 4
        field3(i,j,19) = 4
        field3(i,j,1) = xmin + xstep * (i - 1)
        field3(i,j,2) = ymin + ystep * (j - 1)
      end do
      end do

c-----------------------------------------------------------------
c
c     read information from first field
c
c
      do while (.true.)

      read(1,*,end = 10) i1 , j1 , u1 , v1 , ex1 , ey1 , exy1,
     & ux,vy,uy,vx,eex,eey,eexy,eux,evy,euy,evx,junk3,eu,ev

      i = ( i1 - i0 ) / istep + 1
      j = ( j1 - j0 ) / jstep + 1

      field1(i,j,1)  = i1
      field1(i,j,2)  = j1
      field1(i,j,3)  = u1
      field1(i,j,4)  = v1
      field1(i,j,5)  = ex1
      field1(i,j,6)  = ey1
      field1(i,j,7)  = exy1
      field1(i,j,8)  = ux
      field1(i,j,9)  = vy
      field1(i,j,10) = uy
```

```
      field1(i,j,11) = vx
      field1(i,j,12) = eex
      field1(i,j,13) = eey
      field1(i,j,14) = eexy
      field1(i,j,15) = eux
      field1(i,j,16) = evy
      field1(i,j,17) = euy
      field1(i,j,18) = evx
      field1(i,j,19) = junk3
      field1(i,j,20) = eu
      field1(i,j,21) = ev

      end do
   10 continue

c-----------------------------------------------------------------
c
c     read information from second field
c
c

      do while (.true.)

      read(2,*,end = 20) i1 , j1 , u1 , v1 , ex1 , ey1 , exy1,
     & ux,vy,uy,vx,junk3

      i = ( i1 - i0 ) / istep + 1
      j = ( j1 - j0 ) / jstep + 1

      field2(i,j,1)  = i1
      field2(i,j,2)  = j1
      field2(i,j,3)  = u1
      field2(i,j,4)  = v1
      field2(i,j,5)  = ex1
      field2(i,j,6)  = ey1
      field2(i,j,7)  = exy1
      field2(i,j,8)  = ux
      field2(i,j,9)  = vy
      field2(i,j,10) = uy
      field2(i,j,11) = vx
      field2(i,j,12) = junk3
      end do
   20 continue

c-----------------------------------------------------------------
c
c     loop for addition of displacements
c

      do i = 1 , 50
      do j = 1 , 50

c      field3(i,j,1) = field1(i,j,1)
c      field3(i,j,2) = field1(i,j,2)

      if (field1(i,j,19).lt.4) then    ! if -> 1a

      x = field1(i,j,1)
      y = field1(i,j,2)
      u = field1(i,j,3)
      v = field1(i,j,4)
      x = x + u
      y = y + v

c                    if -> 2a
       if (x.ge.xmin.and.y.ge.ymin.and.x.le.xmax.and.y.le.ymax) then
```

```
      i1 = int((x - xmin)/xstep) + 1
      i2 = i1 + 1
      j1 = int((y - ymin)/ystep) + 1
      j2 = j1 + 1
      x1 = field1(i1,j1,1)
      x2 = field1(i2,j1,1)
      y1 = field1(i1,j1,2)
      y2 = field1(i1,j2,2)

      k1 = field2(i1,j1,12)
      k2 = field2(i1,j2,12)
      k3 = field2(i2,j1,12)
      k4 = field2(i2,j2,12)

      k1 = max (k1,k2,k3,k4)


      if (k1.lt.4) then        ! if -> 3a


      u1 = field2(i1,j1,3) * (x2-x) + field2(i2,j1,3) * (x-x1)
      u1 = u1 / real (istep)
      u2 = field2(i1,j2,3) * (x2-x) + field2(i2,j2,3) * (x-x1)
      u2 = u2 / real (istep)
      uf = u1*(y2-y) + u2*(y-y1)
      uf = uf / real (jstep)
      field3(i,j,3) = field1(i,j,3) + uf

      v1 = field2(i1,j1,4) * (x2-x) + field2(i2,j1,4) * (x-x1)
      v1 = v1 / real (istep)
      v2 = field2(i1,j2,4) * (x2-x) + field2(i2,j2,4) * (x-x1)
      v2 = v2 / real (istep)
      vf = v1*(y2-y) + v2*(y-y1)
      vf = vf / real (jstep)
      field3(i,j,4) = field1(i,j,4) + vf

      field3(i,j,19) =  0

      else
         field3(i,j,19) = 4
      end if       ! if -> 3a

      else
         field3(i,j,19) = 4
      end if       ! if -> 2a

      end if       ! if -> 1a

      end do
      end do

c**********
c      do j = 1,50
c      do i = 1,50
c      write(30,'(21e15.6)') (field1(i,j,l),l=1,21)
c      write(3,'(12e15.6)') (field2(i,j,l),l=1,12)
c      write(3,'(21e15.6)') (field3(i,j,l),l=1,21)
c      write(3,*)' '
c
c      end do
c      end do
c      close(unit = 1)
c      close(unit = 2)
c      close(unit = 3)
c
```

```
c      stop
c************
c-----------------------------------------------------------------
c
c     computation of  gradients of displacements
c
c


      do i = 1 , 50
      do j = 1 , 50


      i1 = i
      i2 = i1 + 1
      j1 = j
      j2 = j1 + 1
      x1 = field3(i1,j1,1)
      x2 = field3(i2,j1,1)
      y1 = field3(i1,j1,2)
      y2 = field3(i1,j2,2)

      k1 = field3(i1,j1,19)
      k2 = field3(i2,j1,19)
      k3 = field3(i1,j2,19)
      k4 = field3(i2,j2,19)

      k1 = max (k1,k2,k3,k4)

      if (k1.lt.4) then    ! if -> 1c

      uxf = ( field3(i2,j1,3) - field3(i1,j1,3) ) / xstep
      vyf = ( field3(i1,j2,4) - field3(i1,j1,4) ) / ystep
      uyf = ( field3(i1,j2,3) - field3(i1,j1,3) ) / ystep
      vxf = ( field3(i2,j1,4) - field3(i1,j1,4) ) / xstep

      field3(i,j,8) =  uxf
      field3(i,j,9) =  vyf
      field3(i,j,10) =  uyf
      field3(i,j,11) =  vxf

      ex = field3(i,j,8) + .5d0*(field3(i,j,8)*field3(i,j,8)+
     & field3(i,j,11)*field3(i,j,11))
      ey = field3(i,j,9) +.5d0*(field3(i,j,9)*field3(i,j,9)+
     & field3(i,j,10)*field3(i,j,10))
      exy = .5d0*(field3(i,j,10)+field3(i,j,11))+.5d0*(field3(i,j,8)*
     & field3(i,j,10)+
     & field3(i,j,9)*field3(i,j,11))

      field3(i,j,5) = ex
      field3(i,j,6) = ey
      field3(i,j,7) = exy

      else
         if (field3(i,j,19).lt.4) field3(i,j,19) = 2
      end if               !  if -> 1c


      end do
      end do

c-----------------------------------------------------------------
c
c     loop for computation of error in displacements in last
c       interpolation
c
```

```
      Program add4

      implicit double precision (a-h,o-z)

      dimension field1(50,50,22)
      dimension field2(50,50,22)
      dimension field3(50,50,22)
      dimension xx(6)
      dimension yy(6)
      dimension uu(16)
      dimension vv(16)
      dimension a(16,6)
      dimension b(6,6)
      dimension c(6)
      dimension e(6)
      dimension bb(6,6)

      open ( unit = 1, file = 'uno.dat', status = 'old')
      open ( unit = 2, file = 'dos.dat', status = 'old')
      open ( unit = 3, file = 'dat', status = 'new')


      i0 = 20
      j0 = 60
      imax = 470
      jmax = 460
      istep = 10
      jstep = 10

      xmax = real (imax)
      ymax = real (jmax)
      xmin = real (i0)
      ymin = real (j0)
      xstep = real (istep)
      ystep = real (jstep)

      pu0 = .188d0
      pu1 = .00037d0
      pux0 = -.0014d0
      pux1 = .08d0
      pux2 = .0d0

      do i = 1 , 50
      do j = 1 , 50
        field1(i,j,19) = 4
        field2(i,j,12) = 4
        field3(i,j,19) = 4
        field3(i,j,1) = xmin + xstep * (i - 1)
        field3(i,j,2) = ymin + ystep * (j - 1)
      end do
      end do

c-----------------------------------------------------------------
c
c     read information from first field
c
c
      do while (.true.)

      read(1,*,end = 10) i1 , j1 , u1 , v1 , ex1 , ey1 , exy1,
     & ux,vy,uy,vx,eex,eey,eexy,eux,evy,euy,evx,junk3,eu,ev

      i = ( i1 - i0 ) / istep + 1
      j = ( j1 - j0 ) / jstep + 1

      field1(i,j,1)  = i1
```

```
      field1(i,j,2)  = jl
      field1(i,j,3)  = ul
      field1(i,j,4)  = vl
      field1(i,j,5)  = exl
      field1(i,j,6)  = eyl
      field1(i,j,7)  = exyl
      field1(i,j,8)  = ux
      field1(i,j,9)  = vy
      field1(i,j,10) = uy
      field1(i,j,11) = vx
      field1(i,j,12) = eex
      field1(i,j,13) = eey
      field1(i,j,14) = eexy
      field1(i,j,15) = eux
      field1(i,j,16) = evy
      field1(i,j,17) = euy
      field1(i,j,18) = evx
      field1(i,j,19) = junk3
      field1(i,j,20) = eu
      field1(i,j,21) = ev

      end do
   10 continue

c-----------------------------------------------------------------
c
c     read information from second field
c
c

      do while (.true.)

      read(2,*,end = 20) il , jl , ul , vl , exl , eyl , exyl,
     & ux,vy,uy,vx,junk3

      i = ( il - i0 ) / istep + 1
      j = ( jl - j0 ) / jstep + 1

      field2(i,j,1)  = il
      field2(i,j,2)  = jl
      field2(i,j,3)  = ul
      field2(i,j,4)  = vl
      field2(i,j,5)  = exl
      field2(i,j,6)  = eyl
      field2(i,j,7)  = exyl
      field2(i,j,8)  = ux
      field2(i,j,9)  = vy
      field2(i,j,10) = uy
      field2(i,j,11) = vx
      field2(i,j,12) = junk3
      end do
   20 continue

c-----------------------------------------------------------------
c
c     loop for addition of displacements
c

      do i = 1 , 50
      do j = 1 , 50

c      field3(i,j,1) = field1(i,j,1)
c      field3(i,j,2) = field1(i,j,2)

      if (field1(i,j,19).lt.4) then    ! if -> 1a
```

```
      x = field1(i,j,1)
      y = field1(i,j,2)
      u = field1(i,j,3)
      v = field1(i,j,4)
      x = x + u
      y = y + v

c                    if -> 2a
       if (x.ge.xmin.and.y.ge.ymin.and.x.le.xmax.and.y.le.ymax) then

      i1 = int((x - xmin)/xstep)
      i2 = i1 + 1
      i3 = i1 + 2
      i4 = i1 + 3
      j1 = int((y - ymin)/ystep)
      j2 = j1 + 1
      j3 = j1 + 2
      j4 = j1 + 3

      xx(1) = field1(i1,j1,1)
      xx(2) = field1(i2,j1,1)
      xx(3) = field1(i3,j1,1)
      xx(4) = field1(i4,j1,1)
      yy(1) = field1(i1,j1,2)
      yy(2) = field1(i1,j2,2)
      yy(3) = field1(i1,j3,2)
      yy(4) = field1(i1,j4,2)

      k1 = field2(i1,j1,12)
      k2 = field2(i2,j1,12)
      k3 = field2(i3,j1,12)
      k4 = field2(i4,j1,12)
      k5 = field2(i1,j2,12)
      k6 = field2(i2,j2,12)
      k7 = field2(i3,j2,12)
      k8 = field2(i4,j2,12)
      k9 = field2(i1,j3,12)
      k10 = field2(i2,j3,12)
      k11 = field2(i3,j3,12)
      k12 = field2(i4,j3,12)
      k13 = field2(i1,j4,12)
      k14 = field2(i2,j4,12)
      k15 = field2(i3,j4,12)
      k16 = field2(i4,j4,12)

      k1 = max (k1,k2,k3,k4,k5,k6,k7,k8,k9,k10,k11)
      k1 = max (k1,k12,k13,k14,k15,k16)

      if (k1.lt.4.and.i1.ge.1.and.j1.ge.1) then       ! if -> 3a

c      write(*,*) 'here',i,j,k1,i1,j1

      uu(1) = field2(i1,j1,3)
      uu(2) = field2(i2,j1,3)
      uu(3) = field2(i3,j1,3)
      uu(4) = field2(i4,j1,3)
      uu(5) = field2(i1,j2,3)
      uu(6) = field2(i2,j2,3)
      uu(7) = field2(i3,j2,3)
      uu(8) = field2(i4,j2,3)
      uu(9) = field2(i1,j3,3)
      uu(10) = field2(i2,j3,3)
      uu(11) = field2(i3,j3,3)
      uu(12) = field2(i4,j3,3)
      uu(13) = field2(i1,j4,3)
      uu(14) = field2(i2,j4,3)
```

```
      uu(15) = field2(i3,j4,3)
      uu(16) = field2(i4,j4,3)

      vv(1) = field2(i1,j1,4)
      vv(2) = field2(i2,j1,4)
      vv(3) = field2(i3,j1,4)
      vv(4) = field2(i4,j1,4)
      vv(5) = field2(i1,j2,4)
      vv(6) = field2(i2,j2,4)
      vv(7) = field2(i3,j2,4)
      vv(8) = field2(i4,j2,4)
      vv(9) = field2(i1,j3,4)
      vv(10) = field2(i2,j3,4)
      vv(11) = field2(i3,j3,4)
      vv(12) = field2(i4,j3,4)
      vv(13) = field2(i1,j4,4)
      vv(14) = field2(i2,j4,4)
      vv(15) = field2(i3,j4,4)
      vv(16) = field2(i4,j4,4)

      do ii = 1 , 16

      jjj = (ii - 1)/4 + 1
      iii = ii - (jjj - 1)*4
c      write(*,*)ii,jjj,iii
      a(ii,1) = xx(iii)**2
      a(ii,2) = yy(jjj)**2
      a(ii,3) = xx(iii)*yy(jjj)
      a(ii,4) = xx(iii)
      a(ii,5) = yy(jjj)
      a(ii,6) = 1.d0

      end do

      do ii = 1 , 6
      do jj = 1 , 6
          b(ii,jj) = 0.d0
          c(ii) = 0.d0
          e(ii) = 0.d0
          do k = 1 , 16
           c(ii) = c(ii) + a(k,ii) * uu(k)
           b(ii,jj) = b(ii,jj) + a(k,ii) * a(k,jj)
           e(ii) = e(ii) + a(k,ii) * vv(k)
          end do
          bb(ii,jj) = b(ii,jj)
      end do
      end do

      call gaussj(b,6,6,c,1,1)
      call gaussj(bb,6,6,e,1,1)

      uf = c(1) * x**2 + c(2) * y**2 + c(3) *x*y +
     $ c(4) * x + c(5) * y + c(6)

      vf = e(1) * x**2 + e(2) * y**2 + e(3) *x*y +
     $ e(4) * x + e(5) * y + e(6)

      ux = field1(i,j,8)
      vy = field1(i,j,9)
      uy = field1(i,j,10)
      vx = field1(i,j,11)

      uxf = c(4) + 2.d0*c(1)*x + c(3)*y
      vyf = e(5) + 2.d0*e(2)*y + e(3)*x
      uyf = c(5) + 2.d0*c(2)*y + c(3)*x
      vxf = e(4) + 2.d0*e(1)*x + e(3)*y
```

```
      field3(i,j,3) = field1(i,j,3) + uf
      field3(i,j,4) = field1(i,j,4) + vf

      field3(i,j,8)  = ux + uxf + uxf*ux + uyf*vx
      field3(i,j,9)  = vy + vyf + vyf*vy + vxf*uy
      field3(i,j,10) = uy + uyf + uxf*uy + uyf*vy
      field3(i,j,11) = vx + vxf + vxf*ux + vx*vyf

      ex = field3(i,j,8) + .5d0*(field3(i,j,8)*field3(i,j,8)+
     & field3(i,j,11)*field3(i,j,11))
      ey = field3(i,j,9) +.5d0*(field3(i,j,9)*field3(i,j,9)+
     & field3(i,j,10)*field3(i,j,10))
      exy = .5d0*(field3(i,j,10)+field3(i,j,11))+.5d0*(field3(i,j,8)*
     & field3(i,j,10)+
     & field3(i,j,9)*field3(i,j,11))

      field3(i,j,5) = ex
      field3(i,j,6) = ey
      field3(i,j,7) = exy

      do jn = 1 , 4
      do in = 1 , 4
       nn = (jn - 1) * 4 + in
       xc = xx(in)
       yc = yy(jn)
      uc = abs(c(1) * xc**2 + c(2) * yc**2 + c(3) *xc*yc +
     $ c(4) * xc + c(5) * yc + c(6) - uu(nn))
      vc = abs(e(1) * xc**2 + e(2) * yc**2 + e(3) *xc*yc +
     $ e(4) * xc + e(5) * yc + e(6) - vv(nn))
        if (umax.lt.uc) umax = uc
        if (vmax.lt.vc) vmax = vc
      end do
      end do

      field3(i,j,20) = sqrt(umax**2 + (pu0 + pu1*uf)**2)
      field3(i,j,21) = sqrt(vmax**2 + (pu0 + pu1*vf)**2)

      field3(i,j,19) =  0

      else
         field3(i,j,19) = 4
      end if       ! if -> 3a

      else
         field3(i,j,19) = 4
      end if       ! if -> 2a

      end if       ! if -> 1a

      end do
      end do

c**********
      do j = 1,41
      do i = 1,46
c      write(30,'(21e15.6)') (field1(i,j,l),l=1,21)
c      write(3,'(12e15.6)') (field2(i,j,l),l=1,12)
      write(3,'(21e15.6)') (field3(i,j,l),l=1,21)
c      write(3,*)' '

      end do
      end do
      close(unit = 1)
      close(unit = 2)
      close(unit = 3)
```

```
c
      stop
c************
c-----------------------------------------------------------------------
c
c     loop for computation of error in displacements in last
c       interpolation
c

      do i = 1 , 50
      do j = 1 , 50


      i1 = i
      i2 = i1 + 1
      i4 = i1 - 1
      j1 = j
      j2 = j1 + 1
      j4 = j1 - 1

      k1 = field3(i1,j1,19)
      k2 = field3(i1,j2,19)
      k3 = field3(i1,j4,19)
      k4 = field3(i2,j1,19)
      k5 = field3(i4,j1,19)

      k1 =  max(k1,k2,k3,k4,k5)

      if (k1.lt.4) then      !  if -> 1b

c
c                                      du**2     Dx**2
c     this derivatives are  duxx = ------- * -------   and so on
c                                      dx**2       2
c

      duxx = (field3(i4,j1,3)  - 2.d0*field3(i1,j1,3)  +
     & field3(i2,j1,3))
      duxx= duxx / 2.d0

      dvxx = (field3(i4,j1,4)  - 2.d0*field3(i1,j1,4)  +
     & field3(i2,j1,4))
      dvxx = dvxx / 2.d0

      dvyy = (field3(i1,j4,4)  - 2.d0*field3(i1,j1,4)  +
     & field3(i1,j2,4))
      dvyy = dvyy / 2.d0

      duyy = (field3(i1,j4,3)  - 2.d0*field3(i1,j1,3)  +
     & field3(i1,j2,3))
      duyy = duyy / 2.d0


      u = field3(i,j,3)
      v = field3(i,j,4)


      field3(i,j,20) = sqrt((duxx + duyy)**2 + (pu0 + pu1*u)**2 )  !eu
      field3(i,j,21) = sqrt((dvxx + dvyy)**2 + (pu0 + pu1*v)**2 )  !ev

      else
       if (field3(i,j,19).eq.0)  field3(i,j,19) = 1
      end if                !  if -> 1b

      end do
      end do
```

```
c**********
c     do j = 1,50
c     do i = 1,50
c     write(30,'(21e15.6)') (field1(i,j,l),l=1,19)
c     write(3,'(12e15.6)') (field2(i,j,l),l=1,12)
c     write(3,'(21e15.6)') (field3(i,j,l),l=1,21)
c     write(3,*)' '
c
c     end do
c     end do
c     close(unit = 1)
c     close(unit = 2)
c     close(unit = 3)
c
c     stop
c************
c---------------------------------------------------------------------
c
c     print out the results
c
c


      do j = 1 , 41
      do i = 1 , 46

c      write(30,'(21e15.6)') (field1(i,j,l),l=1,21)
c      write(3,'(12e15.6)') (field2(i,j,l),l=1,12)
      write(3,'(21e15.6)') (field3(i,j,l),l=1,21)
c      write(3,*)' '

      end do
      end do
      close(unit = 1)
      close(unit = 2)
      close(unit = 3)

      stop
      end


      SUBROUTINE gaussj(a,n,np,b,m,mp)
      INTEGER m,mp,n,np,NMAX
      double precision a(np,np),b(np,mp)
      Parameter (NMAX=50)
      INTEGER i,icol,irow,j,k,l,ll,indxc(NMAX),indxr(NMAX),ipiv(NMAX)
      double precision big,dum,pivinv
      implicit double precision (a-h,o-z)
      do 11 j=1,n
        ipiv(j)=0
11    continue
      do 22 i=1,n
        big=0.d0
        do 13 j=1,n
          if(ipiv(j).ne.1)then
            do 12 k=1,n
              if (ipiv(k).eq.0) then
                if (dabs(a(j,k)).ge.big)then
                  big=dabs(a(j,k))
                  irow=j
                  icol=k
                endif
              else if (ipiv(k).gt.1) then
                write(*,*) ipiv(k)
                pause 'singular matrix in gaussj'
```

```
            endif
12          continue
          endif
13      continue
        ipiv(icol)=ipiv(icol)+1
        if (irow.ne.icol) then
          do 14 l=1,n
            dum=a(irow,l)
            a(irow,l)=a(icol,l)
            a(icol,l)=dum
14        continue
          do 15 l=1,m
            dum=b(irow,l)
            b(irow,l)=b(icol,l)
            b(icol,l)=dum
15        continue
        endif
        indxr(i)=irow
        indxc(i)=icol
        if (a(icol,icol).eq.0.d0) pause 'singular matrix in gaussj'
        pivinv=1.d0/a(icol,icol)
        a(icol,icol)=1.d0
        do 16 l=1,n
          a(icol,l)=a(icol,l)*pivinv
16      continue
        do 17 l=1,m
          b(icol,l)=b(icol,l)*pivinv
17      continue
        do 21 ll=1,n
          if(ll.ne.icol)then
            dum=a(ll,icol)
            a(ll,icol)=0.d0
            do 18 l=1,n
              a(ll,l)=a(ll,l)-a(icol,l)*dum
18          continue
            do 19 l=1,m
              b(ll,l)=b(ll,l)-b(icol,l)*dum
19          continue
          endif
21      continue
22    continue
      do 24 l=n,1,-1
        if(indxr(l).ne.indxc(l))then
          do 23 k=1,n
            dum=a(k,indxr(l))
            a(k,indxr(l))=a(k,indxc(l))
            a(k,indxc(l))=dum
23        continue
        endif
24    continue
      return
      END

C ---------------------------------------------------------------
```

```
      field1(i,j,10) = uy
      field1(i,j,11) = vx
      field1(i,j,12) = eex
      field1(i,j,13) = eey
      field1(i,j,14) = eexy
      field1(i,j,15) = eux
      field1(i,j,16) = evy
      field1(i,j,17) = euy
      field1(i,j,18) = evx
      field1(i,j,19) = junk3
      field1(i,j,20) = eu
      field1(i,j,21) = ev

      end do
   10 continue

c-----------------------------------------------------------------
c
c     read information from second field
c
c

      do while (.true.)

      read(2,*,end = 20) i1 , j1 , u1 , v1 , ex1 , ey1 , exy1,
     & ux,vy,uy,vx,junk3

      i = ( i1 - i0 ) / istep + 1
      j = ( j1 - j0 ) / jstep + 1

      field2(i,j,1)  = i1
      field2(i,j,2)  = j1
      field2(i,j,3)  = u1
      field2(i,j,4)  = v1
      field2(i,j,5)  = ex1
      field2(i,j,6)  = ey1
      field2(i,j,7)  = exy1
      field2(i,j,8)  = ux
      field2(i,j,9)  = vy
      field2(i,j,10) = uy
      field2(i,j,11) = vx
      field2(i,j,12) = junk3
      end do
   20 continue

c-----------------------------------------------------------------
c
c     loop for addition of displacements
c

      do i = 1 , 50
      do j = 1 , 50

c      field3(i,j,1) = field1(i,j,1)
c      field3(i,j,2) = field1(i,j,2)

      if (field1(i,j,19).lt.4) then    ! if -> 1a

      x = field1(i,j,1)
      y = field1(i,j,2)
      u = field1(i,j,3)
      v = field1(i,j,4)
      x = x + u
      y = y + v

c                    if -> 2a
```

```
      if (x.ge.xmin.and.y.ge.ymin.and.x.le.xmax.and.y.le.ymax) then

      i1 = int((x - xmin)/xstep) + 1
      i2 = i1 + 1
      i3 = i1 + 2
      i4 = i1 - 1
      j1 = int((y - ymin)/ystep) + 1
      j2 = j1 + 1
      j3 = j1 + 2
      j4 = j1 - 1
      x1 = field1(i1,j1,1)
      x2 = field1(i2,j1,1)
      y1 = field1(i1,j1,2)
      y2 = field1(i1,j2,2)
      x3 = field1(i3,j1,1)
      x4 = field1(i4,j1,1)
      y3 = field1(i1,j3,2)
      y4 = field1(i1,j4,2)

      k1 = field2(i4,j1,12)
      k2 = field2(i1,j1,12)
      k3 = field2(i2,j1,12)
      k4 = field2(i1,j4,12)
      k5 = field2(i1,j2,12)
      k6 = field2(i2,j2,12)

      k1 = max (k1,k2,k3,k4,k5,k6)


      if (k1.lt.4.and.i4.ge.1.and.j4.ge.1) then       ! if -> 3a


      u1 = field2(i1,j1,3) * (x2-x) + field2(i2,j1,3) * (x-x1)
      u1 = u1 / real (istep)
      u2 = field2(i1,j2,3) * (x2-x) + field2(i2,j2,3) * (x-x1)
      u2 = u2 / real (istep)
      uf = u1*(y2-y) + u2*(y-y1)
      uf = uf / real (jstep)
      field3(i,j,3) = field1(i,j,3) + uf

      v1 = field2(i1,j1,4) * (x2-x) + field2(i2,j1,4) * (x-x1)
      v1 = v1 / real (istep)
      v2 = field2(i1,j2,4) * (x2-x) + field2(i2,j2,4) * (x-x1)
      v2 = v2 / real (istep)
      vf = v1*(y2-y) + v2*(y-y1)
      vf = vf / real (jstep)
      field3(i,j,4) = field1(i,j,4) + vf

c
c                                      du**2     Dx**2
c     this derivatives are  duxx = ------- * -------   and so on
c                                      dx**2       2
c

      duxx = (field2(i4,j1,3)  - 2.d0*field2(i1,j1,3)  +
     & field2(i2,j1,3)) / xstep**2 * (x - x1)**2
      duxx= duxx / 2.d0

      dvxx = (field2(i4,j1,4)  - 2.d0*field2(i1,j1,4)  +
     & field2(i2,j1,4)) / xstep**2 * (x - x1)**2
      dvxx = dvxx / 2.d0

      dvyy = (field2(i1,j4,4)  - 2.d0*field2(i1,j1,4)  +
     & field2(i1,j2,4)) / ystep**2 * (y - y1)**2
      dvyy = dvyy / 2.d0
```

```
      duyy = (field2(i1,j4,3)  - 2.d0*field2(i1,j1,3)  +
     & field2(i1,j2,3)) / ystep**2 * (y - y1)**2
      duyy = duyy  / 2.d0

c      write(10,*) field3(i,j,3),duxx+duyy

      field3(i,j,3) = field3(i,j,3) + duxx + duyy
      field3(i,j,4) = field3(i,j,4) + dvxx + dvyy
c      field3(i,j,3) =  (duxx + duyy) !/field3(i,j,3)
c      field3(i,j,4) =  (dvxx + dvyy) !/field3(i,j,4)

      field3(i,j,19) =  0

      else
         field3(i,j,19) = 4
      end if      ! if -> 3a

      else
         field3(i,j,19) = 4
      end if      ! if -> 2a

      else
         field3(i,j,19) = 4
      end if      ! if -> 1a

      end do
      end do

c**********
c      do j = 1,50
c      do i = 1,50
c      write(30,'(21e15.6)') (field1(i,j,l),l=1,21)
c      write(3,'(12e15.6)') (field2(i,j,l),l=1,12)
c      write(3,'(21e15.6)') (field3(i,j,l),l=1,21)
c      write(3,*)' '
c
c      end do
c      end do
c      close(unit = 1)
c      close(unit = 2)
c      close(unit = 3)
c
c      stop
c************
c-----------------------------------------------------------------------
c
c     computation of  gradients of displacements
c
c


      do i = 1 , 50
      do j = 1 , 50


      i1 = i
      i2 = i1 + 1
      j1 = j
      j2 = j1 + 1
      x1 = field3(i1,j1,1)
      x2 = field3(i2,j1,1)
      y1 = field3(i1,j1,2)
      y2 = field3(i1,j2,2)

      k1 = field3(i1,j1,19)
      k2 = field3(i2,j1,19)
      k3 = field3(i1,j2,19)
```

```
      k1 = max (k1,k2,k3)

      if (k1.lt.3) then     ! if -> 1c

      uxf = ( field3(i2,j1,3) - field3(i1,j1,3) ) / xstep
      vyf = ( field3(i1,j2,4) - field3(i1,j1,4) ) / ystep
      uyf = ( field3(i1,j2,3) - field3(i1,j1,3) ) / ystep
      vxf = ( field3(i2,j1,4) - field3(i1,j1,4) ) / xstep

      field3(i,j,8)  =  uxf
      field3(i,j,9)  =  vyf
      field3(i,j,10) =  uyf
      field3(i,j,11) =  vxf

      ex = field3(i,j,8) + .5d0*(field3(i,j,8)*field3(i,j,8)+
     & field3(i,j,11)*field3(i,j,11))
      ey = field3(i,j,9) +.5d0*(field3(i,j,9)*field3(i,j,9)+
     & field3(i,j,10)*field3(i,j,10))
      exy = .5d0*(field3(i,j,10)+field3(i,j,11))+.5d0*(field3(i,j,8)*
     & field3(i,j,10)+
     & field3(i,j,9)*field3(i,j,11))

      field3(i,j,5) = ex
      field3(i,j,6) = ey
      field3(i,j,7) = exy

      else
         if (field3(i,j,19).lt.4) field3(i,j,19) = 2
      end if                !  if -> 1c


      end do
      end do

c-----------------------------------------------------------------------
c
c     loop for computation of error in displacements in last
c       interpolation
c

      do i = 1 , 50
      do j = 1 , 50

      i1 = i
      i2 = i1 + 1
      i4 = i1 - 1
      j1 = j
      j2 = j1 + 1
      j4 = j1 - 1


      k1 = field3(i1,j1,19)
      k2 = field3(i1,j2,19)
      k3 = field3(i1,j4,19)
      k4 = field3(i2,j1,19)
      k5 = field3(i4,j1,19)

      k1 =  max(k1,k2,k3,k4,k5)

      if (k1.lt.4) then     !  if -> 1b

c
c                                    d**2u     Dx**2
c     this derivatives are  duxx = ------- * -------   and so on
c                                    dx**2       2
```

```
c

      duxxx = (field3(i4,j1,8)  - 2.d0*field3(i1,j1,8)   +
     & field3(i2,j1,8)) * xstep
      duxxx = duxxx / 6.d0

      dvxxx = (field3(i4,j1,11) - 2.d0*field3(i1,j1,11)  +
     & field3(i2,j1,11)) * xstep
      dvxxx = dvxxx / 6.d0

      dvyyy = (field3(i1,j4,9)  - 2.d0*field3(i1,j1,9)   +
     & field3(i1,j2,9)) * ystep
      dvyyy = dvyyy / 6.d0

      duyyy = (field3(i1,j4,10) - 2.d0*field3(i1,j1,10)  +
     & field3(i1,j2,10)) * ystep
      duyyy = duyyy / 6.d0

      duyxx = (field3(i4,j1,10) - 2.d0*field3(i1,j1,10)  +
     & field3(i2,j1,10)) * xstep
      duyxx = duyxx / 2.d0

      dvyxx = (field3(i4,j1,9) - 2.d0*field3(i1,j1,9)    +
     & field3(i2,j1,9)) * xstep
      dvyxx = dvyxx / 2.d0

      dvxyy = (field3(i1,j4,11) - 2.d0*field3(i1,j1,11)  +
     & field3(i1,j2,11)) * ystep
      dvxyy = dvxyy / 2.d0

      duxyy = (field3(i1,j4,8)  - 2.d0*field3(i1,j1,8)   +
     & field3(i1,j2,8)) * ystep
      duxyy = duxyy / 2.d0

c       eu = field2(i,j,20)    !  error in the displacements from
c       ev = field2(i,j,21)    !  last interpolation


      u = field3(i,j,3)
      v = field3(i,j,4)

c       write(11,*) duxxx,dvyyy,dvxyy,u,v

      field3(i,j,20) = sqrt((duxxx + duyyy + duyxx + duxyy)**2 +
     $ (pu0 + pu1*u)**2 )  !eu
      field3(i,j,21) = sqrt((dvxxx + dvyyy + dvyxx + dvxyy)**2 +
     $ (pu0 + pu1*v)**2 )  !ev


      else
       if (field3(i,j,19).eq.0)  field3(i,j,19) = 1
      end if               !  if -> 1b

      end do
      end do

c**********
c       do j = 1,50
c       do i = 1,50
c       write(30,'(21e15.6)') (field1(i,j,l),l=1,19)
c       write(3,'(12e15.6)') (field2(i,j,l),l=1,12)
c       write(3,'(21e15.6)') (field3(i,j,l),l=1,21)
c       write(3,*)' '
c
c       end do
```

```
c       end do
c       close(unit = 1)
c       close(unit = 2)
c       close(unit = 3)
c
c       stop
c************
c-----------------------------------------------------------------------
c
c      computation of errors in gradients of displacements
c
c


       do i = 1 , 50
       do j = 1 , 50


       i1 = i
       i2 = i1 + 1
       j1 = j
       j2 = j1 + 1

       x1 = field3(i1,j1,1)
       x2 = field3(i2,j1,1)
       y1 = field3(i1,j1,2)
       y2 = field3(i1,j2,2)

       k1 = field3(i1,j1,19)
       k2 = field3(i2,j1,19)
       k3 = field3(i1,j2,19)
       k4 = field3(i2,j2,19)

       k1 = max (k1,k2,k3,k4)

       if (k1.lt.1) then     ! if -> ld

       eu1 = field3(i1,j1,20)
       eu2 = field3(i2,j1,20)
       eu3 = field3(i1,j2,20)

       ev1 = field3(i1,j1,21)
       ev2 = field3(i2,j1,21)
       ev3 = field3(i1,j2,21)

       ux = field3(i,j,8)
       vy = field3(i,j,9)
       uy = field3(i,j,10)
       vx = field3(i,j,11)

       eux = sqrt( eu2**2 + eu1**2 ) / xstep
       euy = sqrt( eu3**2 + eu1**2 ) / ystep
       evx = sqrt( ev2**2 + ev1**2 ) / xstep
       evy = sqrt( ev3**2 + ev1**2 ) / ystep

       write(12,*) eu1,ev1,eux,ux

       eex = sqrt((eux*(1.d0+ux))**2+(vx*evx)**2)
       eey = sqrt((evy*(1.d0+vy))**2+(uy*euy)**2)
       eexy = sqrt((eux*.5d0*uy)**2+(euy*.5d0*(1.d0+ux))**2+
      $ (evx*.5d0*(1.d0+vy))**2+(evy*.5d0*vx)**2)

       field3(i,j,12) = eex
       field3(i,j,13) = eey
       field3(i,j,14) = eexy
       field3(i,j,15) = eux
       field3(i,j,16) = evy
```

```
      field3(i,j,17) = euy
      field3(i,j,18) = evx

      else
         if (field3(i,j,19).lt.4) field3(i,j,19) = 3
      end if                !  if -> ld


      end do
      end do

c-----------------------------------------------------------------------
c
c     print out the results
c
c


      do j = 1 , 41
      do i = 1 , 46

c      write(30,'(21e15.6)') (field1(i,j,l),l=1,21)
c      write(3,'(12e15.6)') (field2(i,j,l),l=1,12)
      write(3,'(21e15.6)') (field3(i,j,l),l=1,21)
c      write(3,*)' '

      end do
      end do
      close(unit = 1)
      close(unit = 2)
      close(unit = 3)

      stop
      end
```

APPENDIX D

ALTERNATIVE ERROR ANALYSIS

An alternative method to compute error estimate in the Large Deformation Digital Image Correlation method is presented. This error estimate is constructed using two sources; the precision of the DIC and the errors generated in the interpolation process. The error estimate in the DIC program is calculated as follows: Two tests were performed: one on solid propellant (TPH 1011) and the other on 40 shore silicone rubber. The silicone specimen was splattered with microscopic speckles to provide the random gray level distribution the DIC program do identify. The speckles were generated with an airbrush to match the scale of the surface fractures in the solid propellant specimens.

Displacement test

The material used for this test was TPH 1011 solid propellant. Several pictures of a specimen without any cuts were taken during a prescribed translation. This translation was applied with the help of a milimetriced screw located on the translation stage of the microscope. Figure 1d shows the difference between the displacement values obtained with the DIC program and those prescribed. Both quantities are expressed in pixels. A linear curve is fitted to the results to estimate the errors of the program computing displacements. Notice that for a displacement smaller than 100 pixels, the error estimate is close to .25 pixel. The error estimate (y) in pixels as a function of the translation (x) in pixels is:

$$y = 0.188 + 0.0037\,x. \qquad (1)$$

Strain test

To determine the precision of the DIC program for determining strains, a globally homogeneous, uniaxial deformation on a homogeneous 40 shore silicone rubber specimen coated with speckles was imposed. With uniaxial tension applied, images of the deformations are taken at different strain levels ranging from 0 to 10%. The distance between two marked points was obtained from photo record of the digitized images and the Lagrangian strain component in the direction of the stretch was determined. For reference purposes we refer to this strain as the optically determined strain. Also the Lagrangian strain was determined by means of the DIC method outlined before. These two strains are compared in Fig 2d and similar to the computation of error estimates in the displacements, an error estimate on the strains is generated as a function of the strain level. The analytical expression of the error estimate in the DIC strains (y) as a function of the optical strain (x) is

$$y = -0.0014 + 0.08\, x\,. \qquad (2)$$

Interpolation error

As explained in section 2. 4. 1 of the tesis, an interpolation process is required over the second deformation. Every time an interpolation is performed an error estimate associated with the interpolation process is computed. This error estimate is computed as follows. For purposes of constructing the interpolating function, four points are considered. The interpolating functions have the form:

$$y = c_1 + c_2\, x + c_3\, y + c_4\, xy. \qquad (3)$$

Considering two more points in the interpolation process will allow the appearance of all the quadratic terms in the interpolating function. i.e.

$$y = c_1+c_2\,x+c_3\,y+c_4\,xy+c_5\,x^2+c_6\,y^2\,. \qquad (4)$$

The difference between the quantities represented in form (3) and form (4) is considered to be the error estimate associated with the interpolation process. These two sources of errors are propagated through the analysis to obtain and error estimate on the Lagrangian strains of the global deformation.

Using these two sources of error, an upper bound of the error in the E_{yy} component of the Lagrangian strain is constructed every time the Large Deformation Digital Image Correlation method is used. ThIS error bound are displayed as A contour map for the LD-DIC methods 1, 2 and 3 (Fig 3, 4 and 5). The error bound represented in these contour plots is very conservative and it has not been used in the analysis.

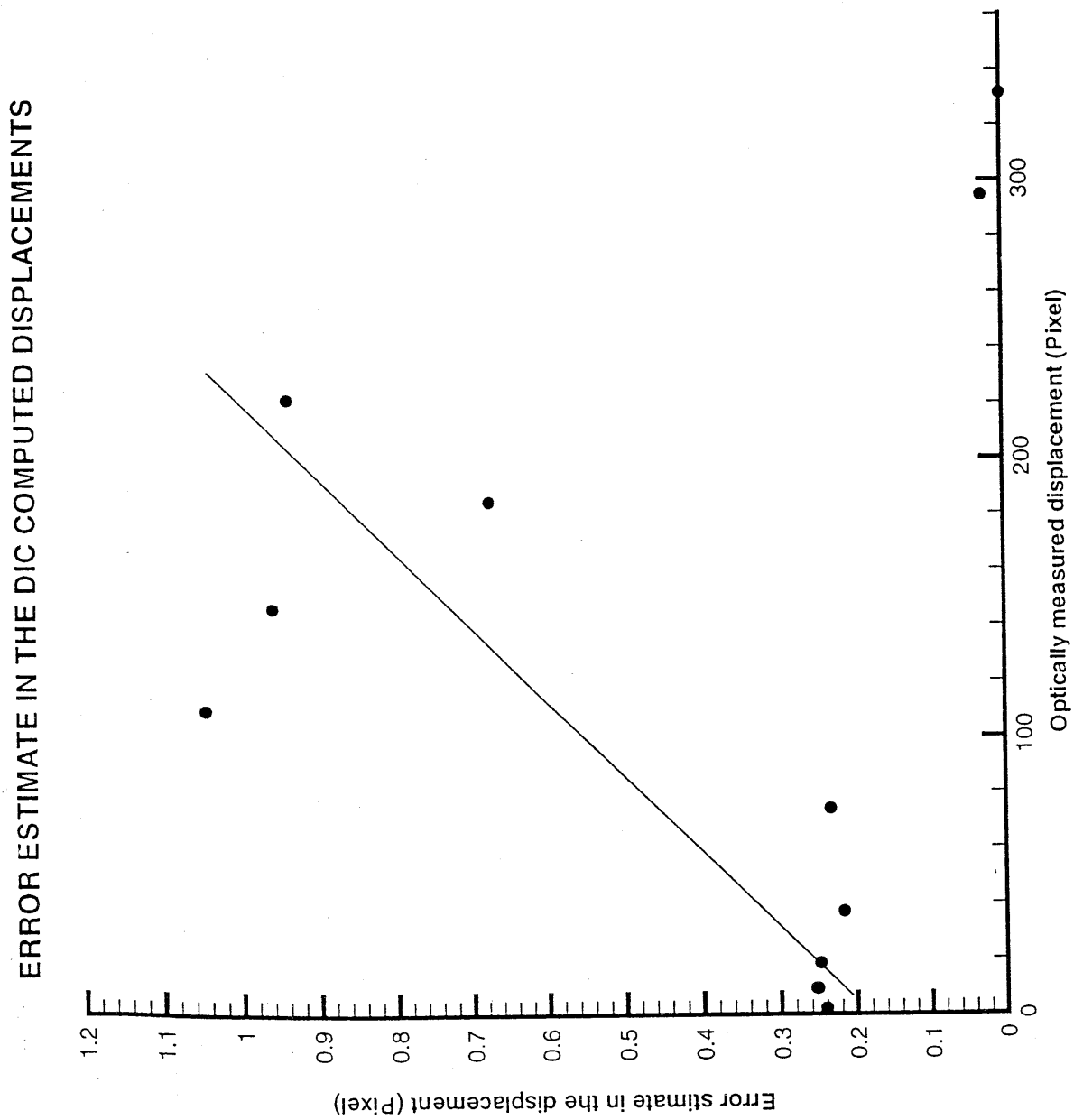

ERROR ESTIMATE IN THE DIC COMPUTED DISPLACEMENTS
Error stimate in the displacement (Pixel)
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
100
200
300
Optically measured displacement (Pixel)

Figure 1d.

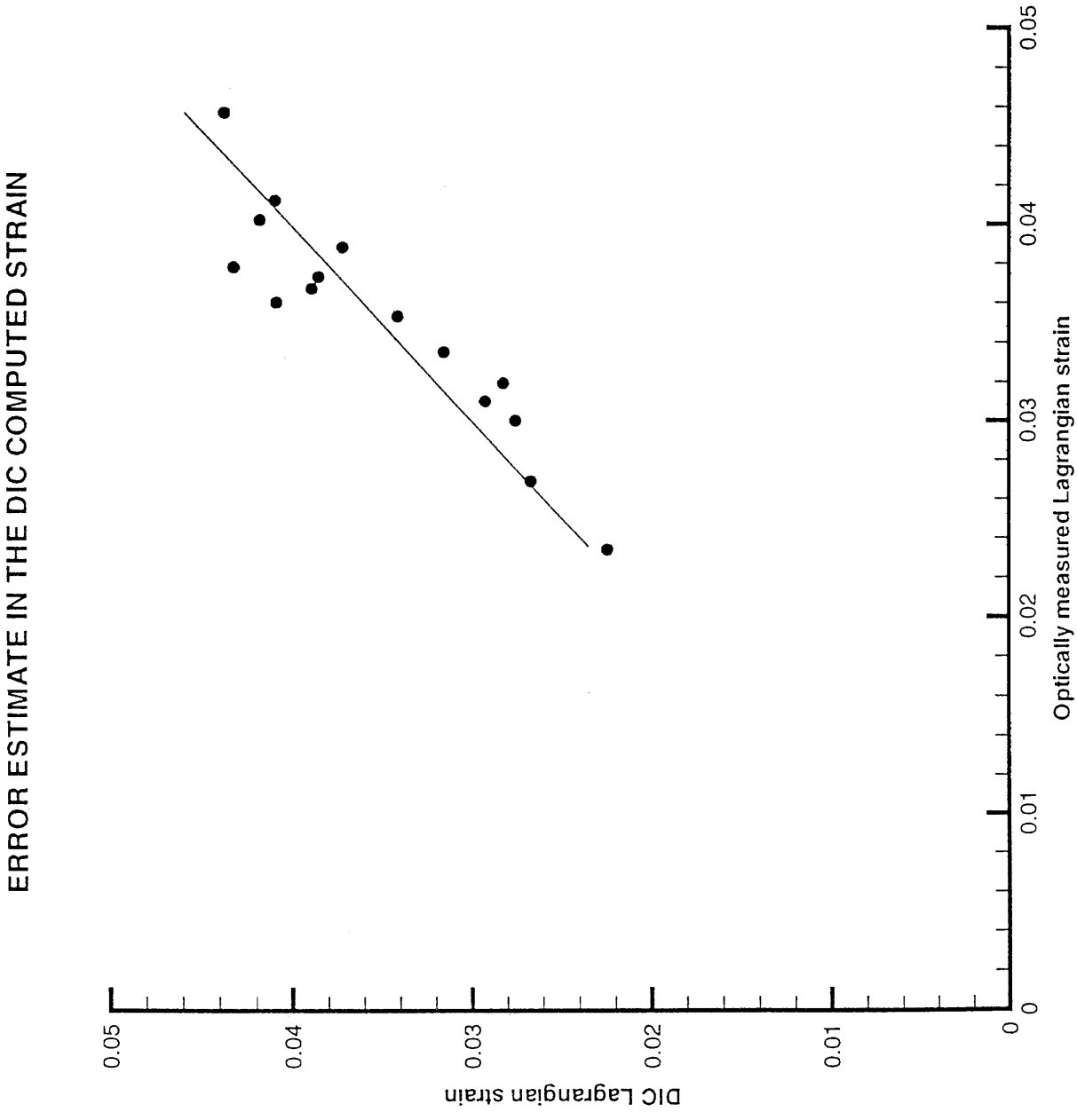

ERROR ESTIMATE IN THE DIC COMPUTED STRAIN
DIC Lagrangian strain
0.05
0.04
0.03
0.02
0.01
0
0.01
0.02
0.03
0.04
0.05
Optically measured Lagrangian strain

Figure 2d.

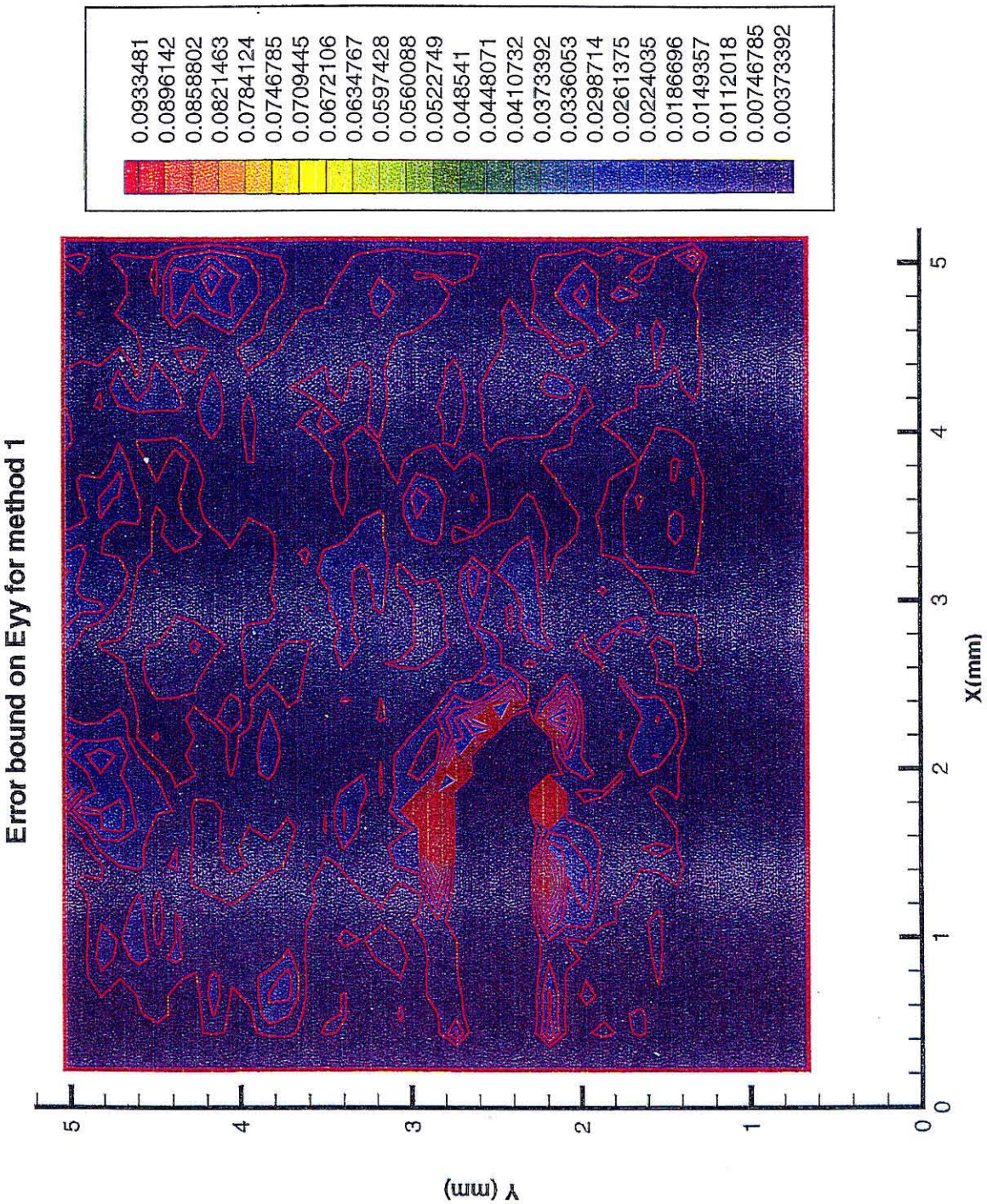

Error bound on Eyy for method 1
0.0933481
0.0896142
0.0858802
0.0821463
0.0784124
0.0746785
0.0709445
0.0672106
0.0634767
0.0597428
0.0560088
0.0522749
0.048541
0.0448071
0.0410732
0.0373392
0.0336053
0.0298714
0.0261375
0.0224035
0.0186696
0.0149357
0.0112018
0.00746785
0.00373392
Y (mm)
X(mm)
0
1
2
3
4
5

Figure 3d.

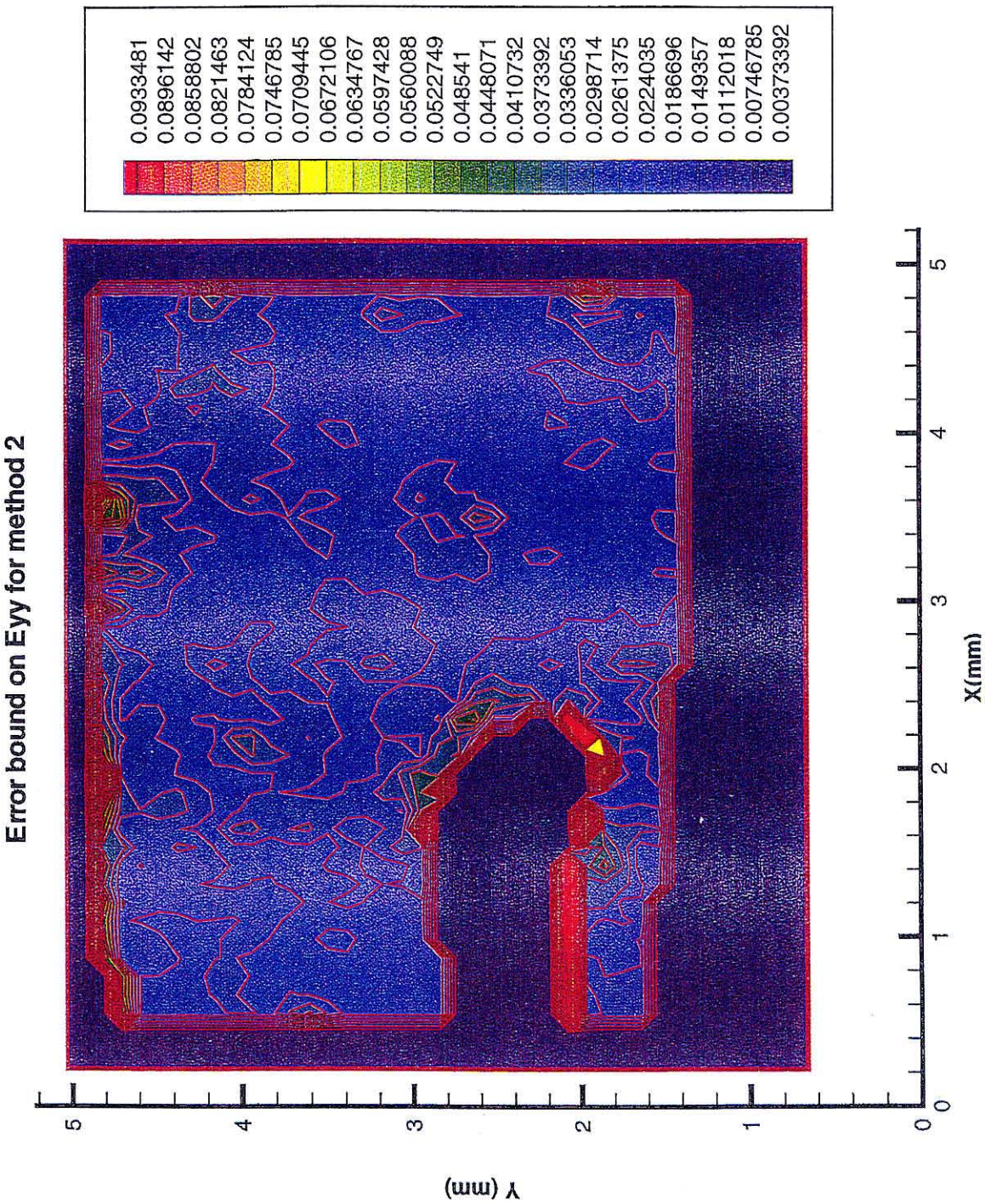

Error bound on Eyy for method 2
0.0933481
0.0896142
0.0858802
0.0821463
0.0784124
0.0746785
0.0709445
0.0672106
0.0634767
0.0597428
0.0560088
0.0522749
0.048541
0.0448071
0.0410732
0.0373392
0.0336053
0.0298714
0.0261375
0.0224035
0.0186696
0.0149357
0.0112018
0.00746785
0.00373392
Y (mm)
X(mm)
0
1
2
3
4
5

Figure 4d.

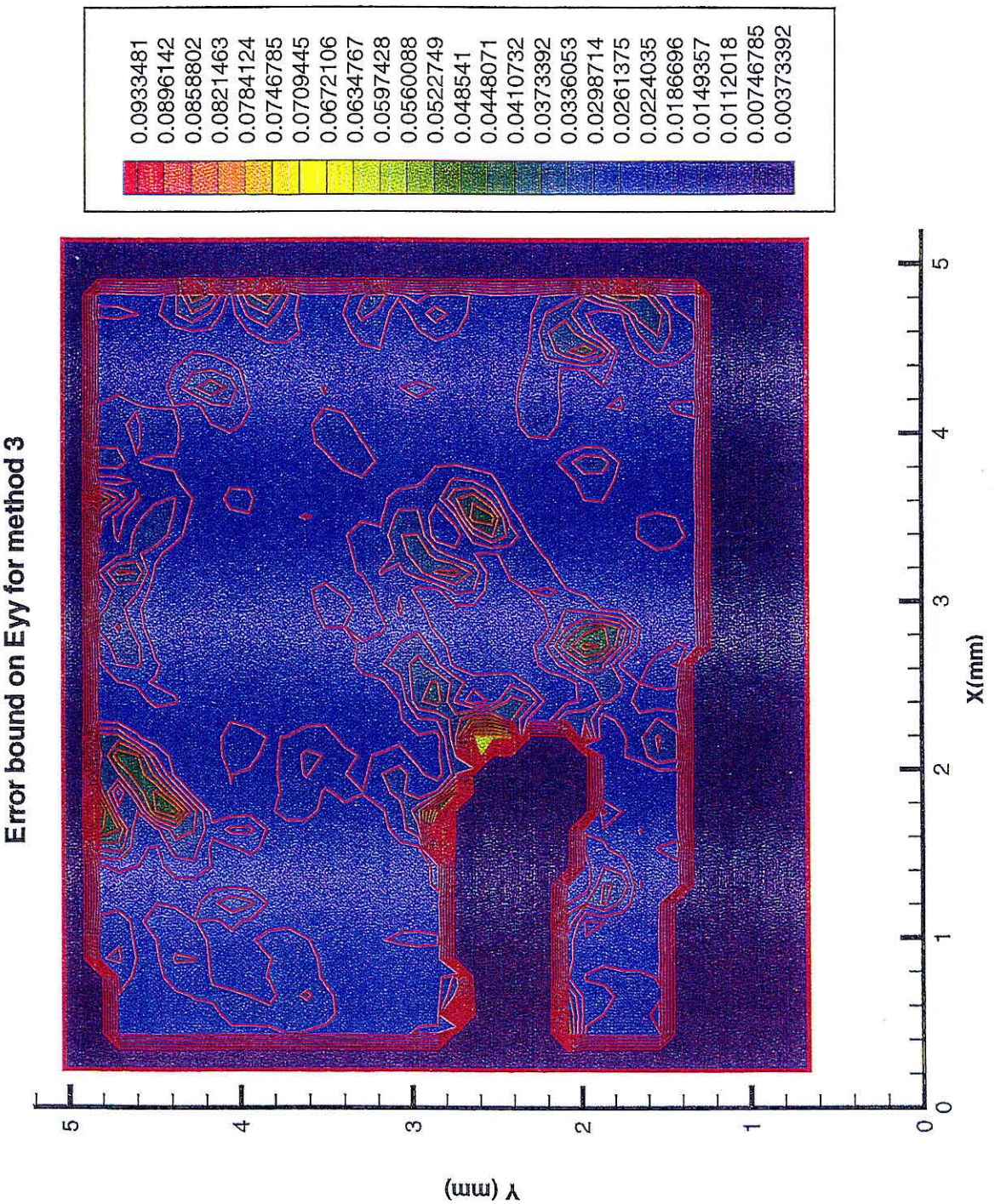

Error bound on Eyy for method 3
0.0933481
0.0896142
0.0858802
0.0821463
0.0784124
0.0746785
0.0709445
0.0672106
0.0634767
0.0597428
0.0560088
0.0522749
0.048541
0.0448071
0.0410732
0.0373392
0.0336053
0.0298714
0.0261375
0.0224035
0.0186696
0.0149357
0.0112018
0.00746785
0.00373392
Y (mm)
X(mm)
0
1
2
3
4
5

Figure 5d.

APPENDIX E

TRANSLATION STAGE CONTROLLER

As loads are applied to the specimen, the crack propagates at a rather fast speed. Since in the experiment it is of importance to visualize the crack tip with the microscope, a joystick device has been built to move the specimen under the microscope objective. This device is capable of powering two electric motors that move the specimen in the x and y direction. The user can easily control the movement by means of a joystick. The electric circuit schematic for one degree of freedom is: