NUMERICAL DIFFERENTIATION OF EXPERIMENTALLY MEASURED DISPLACEMENTS

Tuan Hoang Sy, Bohdana Marvalová

Technical University of Liberec, Czech Republic

1. Introduction

The evaluation of derivatives from discrete experimental data is a common problem which occurs in experimental mechanics. The numerical differentiation process is ill-conditioned and strongly influenced by the data discretization and noise. The problem of approximating the gradient of a function of two variables when noisy data are given involve data smoothing by means of splines and local or global differencing.

We use a non-contacting optical technique, the digital image correlation, for measurements of surface deformation of planar objects. We determine coordinates of grid points marked at the surface from experimental data. It is necessary to differentiate numerically the coordinates of the deformed grid with respect to reference coordinates in order to evaluate strains.

In this paper, we discuss some techniques for the estimation of derivatives in a domain of arbitrary boundary containing holes, cracks or other local concentrators of strains. We develop a scheme for global and local approximation methods, and illustrate the estimation of the derivatives in a complex sample.

The outline of this paper is as follows. In Section 2, we present some mechanical basic equations and a procedure involving interpolation, extrapolation and smoothing data. Then in Section 3 we show estimated strain results of a plane deformed plate consisting of a hole and a crack by available functions in Matlab software. Finally, Section 4 shall give some conclusions.

2. Fundamental theories

2.1. Mechanical constitutive equations

A material point within the body in the reference configuration identified by the position vector **X** occupies the position $\mathbf{x} = \chi(\mathbf{X})$ in the spatial (or deformed) configuration. The function χ describes the static deformation of the body and is a one-to-one, orientation-preserving mapping with suitable regularity properties. The deformation gradient tensor **F** is defined by

$$\mathbf{F} = \mathrm{Grad}\chi \tag{1}$$

where Grad is the gradient operator with respect to X. The Cartesian components of **F** are given by $F_{ij} = \partial x_i / \partial X_j$, where X_i and x_i, i = 1,2,3, are the Cartesian components of X and x, respectively.

The right Cauchy-Green tensor is given by

$$\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F} \tag{2}$$

and the Green-Lagrange strain tensor is calculated from

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}) \tag{3}$$

where **I** is the identity tensor. For full details we refer to Holzapfel (2000).

2.2. Methods of numerical evaluation of derivatives

The main task is to find the best approximate functions of the spatial coordinates *x* and *y* depending on the reference coordinates (X,Y) in such a way that the loss of accuracy is minimal and the smoothness of the result is maximal. We deduce to solve the same problem that it evaluates a fitting surface from one set of N discontinuous given data points

$$z_i = f\left(X_i, Y_i\right) \approx \bar{z}_i \tag{4}$$

where $\bar{z}_i$ is the i-th measured data value corresponding to a position point (X_i,Y_i) and f is an unknown bivariate function estimated from $\sum_i \left\| f\left(X_i, Y_i\right) - \bar{z}_i \right\| = \min$ in the least-square sense.

A fitting function is interpolated through the data points locally, as for a running window. A local method approximates the function f(X,Y) in sub-domain, and does not yield information beyond it boundary. This means especially, that there can be jumps in the approximated function between different points.

Suppose that f is represented by a regular grid of coefficients, given by a vector u, over a local region containing the points (X_i,Y_i). These coefficients may be a regular grid of values of the unknown function f. More generally they denote coefficients of a finite element representation of f. In the interests of computational efficiency, these elements normally have minimal local support, such as bilinear or biquadratic polynomials. Using such representations, the vector of function values $f(X_i,Y_i)$ may be written as **P.u** where **P** is a N×M matrix, N is a number data points in the local and M depends on an order of the chosen polynomial. An approximation to the data in equation (4) may then be determined by finding the vector u that

$$\left\| \mathbf{Pu} - \mathbf{z} \right\|^2 \to \min \tag{5}$$

where $\mathbf{z} = [z_1, z_2, ..., z_n]^T$ is a known value vector consisting of a set of measured data points in the local domain.

Interpolation in local domain

The bilinear or biquadratic polynomials are used in estimations, respectively, as follows

$$f_1(x,y) = a_0 + a_1 x + b_1 y \tag{6-a}$$

$$f_2(x,y) = a_0 + a_1 x + b_1 y + a_2 x^2 + b_2 y^2 + c_2 xy \tag{6-b}$$

therefore **P** and **u** are in a matrix form as

$$\mathbf{P}_1 = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots \\ 1 & x_N & y_N \end{bmatrix} \qquad \mathbf{P}_2 = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2^2 & y_2^2 & x_2 y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & y_N & x_N^2 & y_N^2 & x_N y_N \end{bmatrix} \tag{7-a}$$

$$\mathbf{u}_1 = [a_0, a_1, b_1]^T \qquad \mathbf{u}_2 = [a_0, a_1, b_1, a_2, b_2, c_2]^T \tag{7-b}$$

To determine the deformation tensor **F** in (1), we deduce to define a partial differentiation of *f* with respect to x and y, f_x and f_y, hence

$$f_x = \frac{\partial f}{\partial x} = \frac{\partial \mathbf{P}}{\partial x}\mathbf{u} = \mathbf{P}_x\mathbf{u} \tag{8-a}$$

$$f_y = \frac{\partial f}{\partial y} = \frac{\partial \mathbf{P}}{\partial y}\mathbf{u} = \mathbf{P}_y\mathbf{u} \tag{8-b}$$

where components of P_x and P_y are expressed specifically as

$$\mathbf{P}_{1x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{P}_{2x} = \begin{bmatrix} 0 & 1 & 0 & 2x_1 & 0 & y_1 \\ 0 & 1 & 0 & 2x_2 & 0 & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 2x_N & 0 & y_N \end{bmatrix} \tag{9-a}$$

$$\mathbf{P}_{1y} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{P}_{2y} = \begin{bmatrix} 0 & 0 & 1 & 0 & 2y_1 & x_1 \\ 0 & 0 & 1 & 0 & 2y_2 & x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & 0 & 2y_N & x_N \end{bmatrix} \tag{9-b}$$

Local domains can be chosen as square elements including 9, 16, 25,… data points, see in Fig.1. Estimated results take out only to points inside the element corresponding to red points.

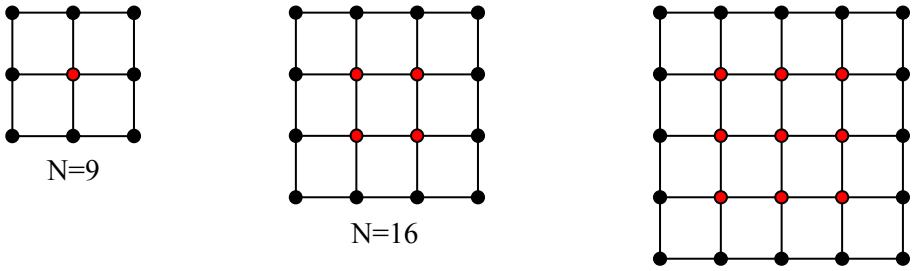

Fig.1 Estimated element types

Note that when the bilinear polynomial is chosen the estimated strains in local are constant and for the chosen biquadratic polynomial the strains obtain as be linear.

Extrapolation to boundaries

After applying the interpolation method for points inside the domain, our next task uses an extrapolation method to extend estimated values of points on the boundaries. Available neighbor points are found by a window which starts at the evaluating point and expands around. This estimation is also based on fitting a chosen polynomial.

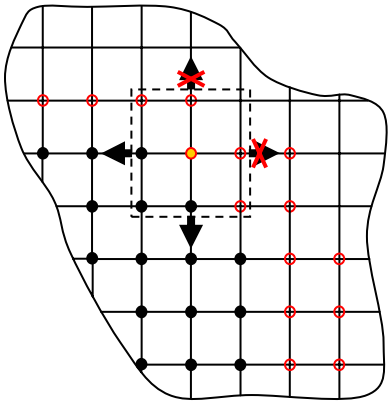

Fig.2 Extrapolation scheme

Smoothing by Moving average

A moving average method smooths data by replacing each data point with the average of the neighboring data points defined within the spans. The smoothing process is given by the equation

$$z_{sm}(i,j)=\frac{1}{(N_x+1)(N_y+1)}\sum_{r=-N_x}^{N_x}\sum_{s=-N_y}^{N_y}z(i+r,j+s) \qquad (10)$$

where $z_{sm}(i,j)$ is the smoothed value for the data point (i,j), N_x and N_y are the number of neighboring data points on either side of $z_{sm}(i,j)$ in the axis x and the axis y.

3. Numerical results

In this section we will show step by step results obtained by application of interpolation, extrapolation and smoothing methods mentioned above to experimental data processing of a deformed plate containing a hole and a crack (Fig.3).

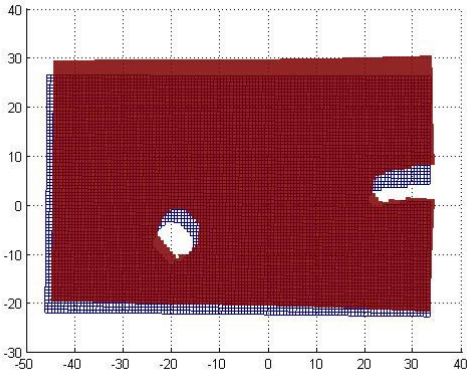

Fig.3 Undeformed and deformed models

First, the interpolation in local domain is effectuated to find the most suitable coefficients by solving eq. (5).

$$\mathbf{u} = \mathbf{P} \setminus \mathbf{z}$$

This procedure is repeated for every data point, alike for a moving window and the partial differentiation components are calculated by eqs. (8) and (9), so that the deformation gradient tensor **F** is achieved. The right Cauchy-Green tensor and the component strains are therefore obtained too in eqs. (2) and (3).

The strain components are estimated by applying the interpolation method in local domain, represented as in Fig. 4.

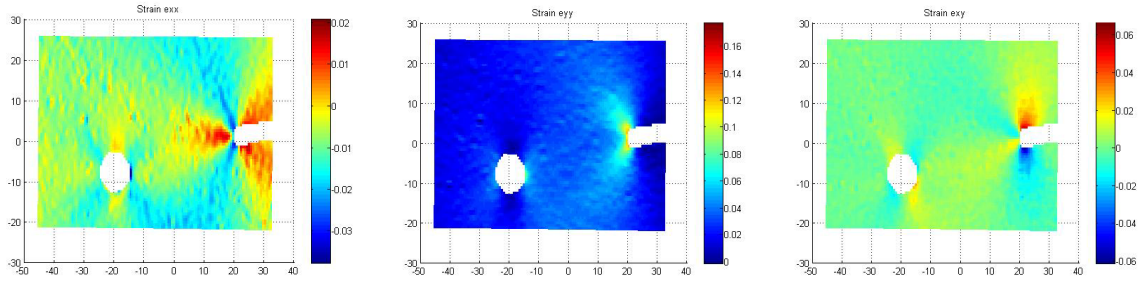

Fig.4 Interpolate strain components in local

Since the lack information supporting of neighbor points that almost strains at points lying near the boundaries is not estimated by the interpolation method, this is also manifested in obtained result areas. It is necessary to use an extrapolation method to expand available strain values to the full domain. The

distribution of achieved strains in the full domain by applying the extrapolation method is displayed in 2 types: surfaces and contour lines, as in Fig.5.

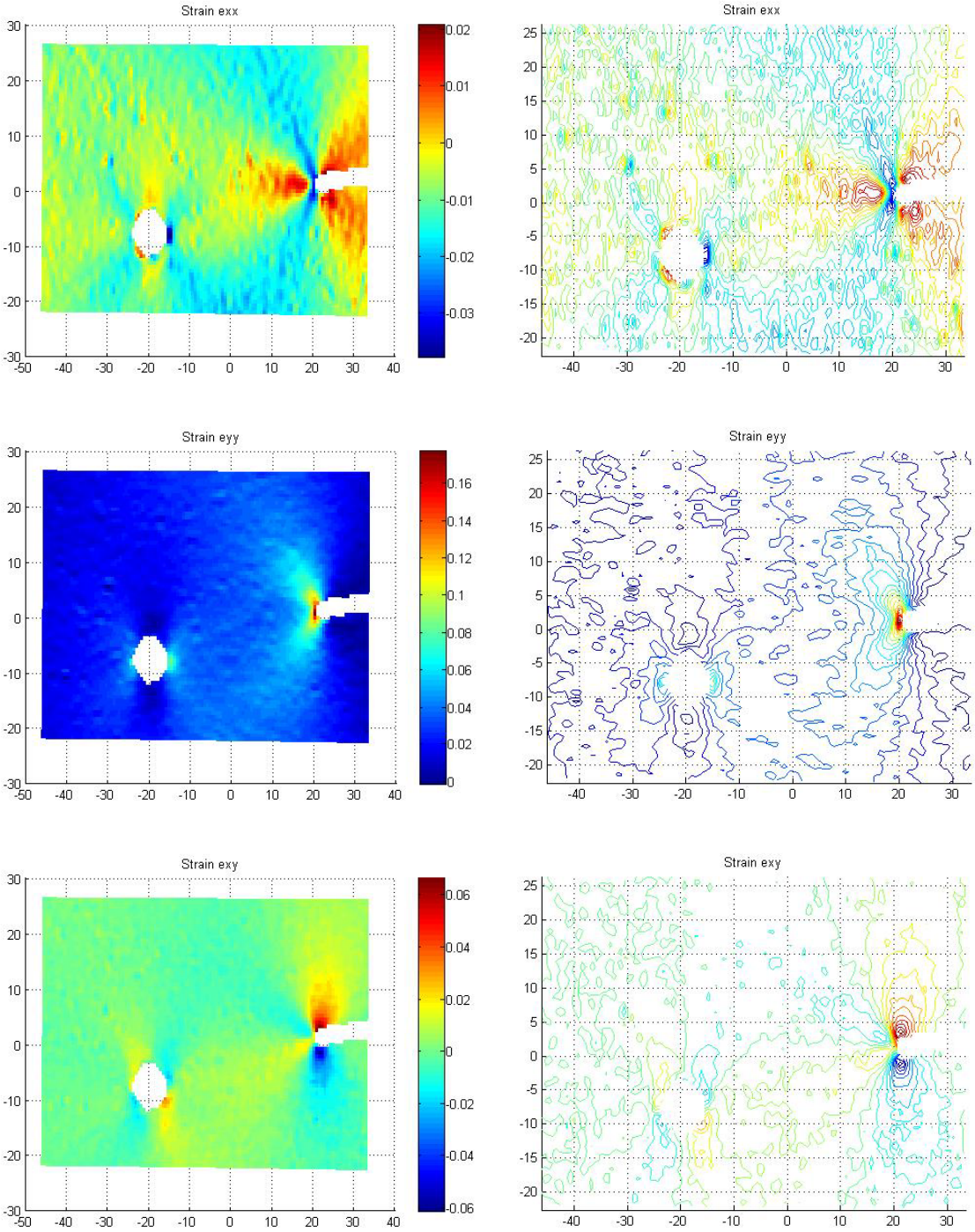

Fig.5 Strains obtained by extrapolation method

From rough results obtained in local estimation it indicates that there exist jumps in the approximated functions between different points. So we need to do a final process as smooth data points by moving average method. The smoothing results are presented in Fig.6.

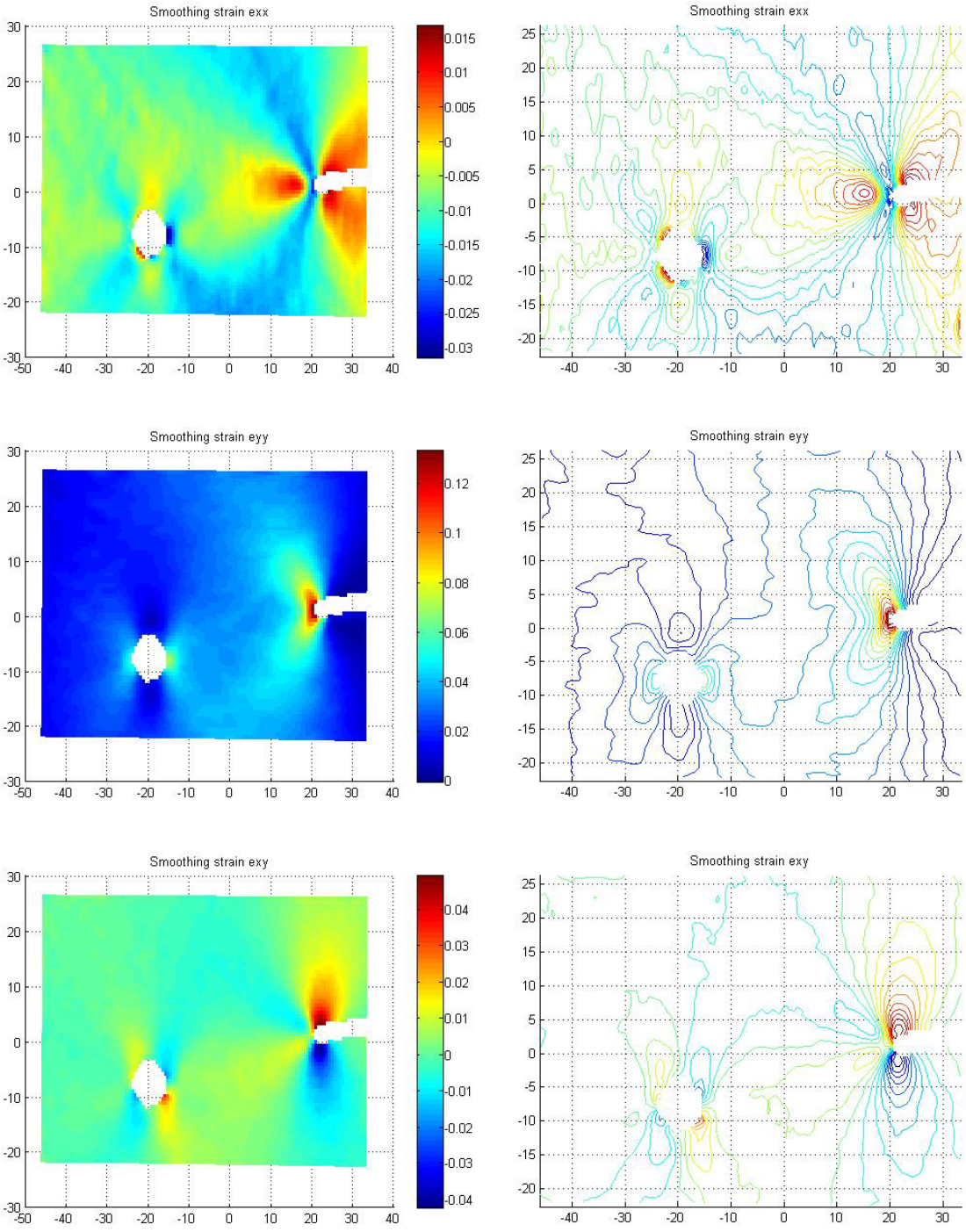

Fig.6 Smoothing strains by moving average

The obtained results by the moving average method is satisfactory and smoother, the distribution of strain components stand out each region manifestly in accordance with an observation from the experiment.

Illustration of estimation errors and the max and/or min jumps of strains by using different element models and polynomials are summarized and listed in Tables 1, 2. The apparent quantities show that an increasing order of approximated functions reduces the jumps, and an increasing element size attaches to cut down the magnitude of concentration strains.

Table 1 – Estimation errors when approximated bilinear polynomial

N	$\sum \Delta x^2$	$\sum \Delta y^2$	$\varepsilon_{x\,min/max}$ [*10^{-2}]	$\varepsilon_{y\,min/max}$ [*10^{-2}]	$\varepsilon_{xy\,min/max}$ [*10^{-2}]	$\lvert\Delta\varepsilon_x\rvert_{max}$	$\lvert\Delta\varepsilon_y\rvert_{max}$	$\lvert\Delta\varepsilon_{xy}\rvert_{max}$
9	0,0313	0,0367	-5,08/2,78	-0,34/23,92	-7,20/8,38	0,0296	0,0748	0,1024
16	0,0441	0,0516	-4,70/1,77	-0,06/22,88	-5,90/7,35	0,0380	0,1313	0,0477
25	0,0590	0,0763	-3,44/1,84	0/25,44	-5,00/6,25	0,0296	0,1799	0,1053

Table 2 – Estimation errors when approximated biquadratic polynomial

N	$\sum \Delta x^2$	$\sum \Delta y^2$	$\varepsilon_{x\,min/max}$ [*10^{-2}]	$\varepsilon_{y\,min/max}$ [*10^{-2}]	$\varepsilon_{xy\,min/max}$ [*10^{-2}]	$\lvert\Delta\varepsilon_x\rvert_{max}$	$\lvert\Delta\varepsilon_y\rvert_{max}$	$\lvert\Delta\varepsilon_{xy}\rvert_{max}$
9	0,0028	0,0035	-3,78/2,07	-0,14/17,69	-6,14/6,62	0,0222	0,0627	0,0358
16	0,0177	0,0211	-5,07/3,56	-0,30/29,63	-5,95/6,44	0,0218	0,1384	0,0424
25	0,0289	0,0428	-5,27/2,80	-0,1/30,42	-6,09/7,71	0,0301	0,2194	0,0532

Finally a comparison the different strain components before and after applying a smoothing by moving average method is also shown in Table 3. The increasing smooth simultaneously goes with the decreasing maximal and minimal strains it means that the concentration of strains is disappeared gradually.

Table 3 – The maximal and minimal values of strains and their maximum local differences after smoothing with spans [3,3], [5,5], [7,7] and [9,9]

Spans	$\varepsilon_{x\,min/max}$ [*10^{-2}]	$\varepsilon_{y\,min/max}$ [*10^{-2}]	$\varepsilon_{xy\,min/max}$ [*10^{-2}]	$\lvert\Delta\varepsilon_x\rvert_{max}$	$\lvert\Delta\varepsilon_y\rvert_{max}$	$\lvert\Delta\varepsilon_{xy}\rvert_{max}$
[3,3]	-3.48/1.72	-0.05/13.86	-4.88/5.53	0.0348	0.1386	0.0467
[5,5]	-3.15/1,67	-0.05/13.29	-4.23/4.92	0.0315	0.1328	0.0467
[7,7]	-2.78/1.67	-0.05/12.72	-3.75/4.92	0.0278	0.1225	0.0467
[9,9]	-2.51/1.67	-0.05/12.35	-3.75/4.92	0.0251	0.1178	0.0467

4. Conclusion

The estimation of component strains from measured coordinates in an arbitrary 2-D domain is presented we find that rough estimates in local method work acceptably well. The distribution of the strains is smoother by applying moving average. However it is necessary to use the smooth method carefully since it can increase errors and cut down the extreme local values. The obtained results are satisfactory and in accordance with an observation from the experiment.

Acknowledgement

This work was supported by the subvention from Ministry of Education of the Czech Republic under Contract Code MSM 4674788501

Reference

[1] - Holzapfel, G.A. *Nonlinear Solid Mechanics. A Continuum Approach for Engineering*, John Wiley & Son, Chichester, 2000.

[2] - Karsten Ahnert, Markus Abel. *Numerical differentiation of experimental data: local versus global methods*. Computer Physics Communications, Feb 2008.

[3] - S. Diop, J. W. Grizzle, F. Chaplais. *On numerical differentiation algorithms for nonlinear estimation*. The Proceedings of the IEEE Conference on Decision and Control, Sydney, Australia, 2000.