GeoStrain: An open source software for calculating crustal strain rates

Mohammad Ali Goudarzi *, Marc Cocard, Rock Santerre

Department of Geomatics Sciences, Louis-Jacques-Casault Building, Laval University, Quebec (QC), G1V 0A6, Canada

ARTICLE INFO

Article history:
Received 28 August 2014
Received in revised form 23 April 2015
Accepted 11 May 2015
Available online 14 May 2015

Keywords:
Strain rate
Least-squares collocation
Adjustment
Interpolation

ABSTRACT

We developed an open source software for crustal strain analysis using the least-squares collocation method based on the spherical model of the earth. The software is able to simultaneously determine the signal and noise of the velocities at the observation points with the best possible removal of the observational errors, or at any other position with no velocity observation. Furthermore, the software can calculate strain and rotation rate tensors at any points of interest, including observation points, grid points or points along a fault line. The advantage of the software is that the sphericity of the earth is considered in all the calculations. Moreover, it can optionally consider the effect of the vertical velocities on the strain rates that is principally important for regions where vertical deformation is the major geophysical signal compared to the horizontal deformation.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Earth deformation concerns changes in earth’s shape, and therefore, it is directly related to geodetic observables. The increasing number and accuracy of satellite based geodetic measurements such as continuously operating GPS (cGPS) stations during the past two decades provide measured values of displacements and velocities of geodetic networks. Crustal strain tensors, as a direct result of the geodetic networks, can numerically describe geodynamic processes. Tensors have an existence independent of any coordinate system or reference frame and retain their properties independently from the reference system. Therefore, tensor analysis reveals different aspects of deformation such as fault strain accumulation, uplift or subsidence, the dilatation strain rate or the maximum shear strain rate including its direction which are important parameters for seismic hazard assessment. These parameters are suitable to characterize the mechanism of an ongoing deformation (Hackl et al., 2009).

As the great importance of the strain rate tensors in tectonic and geodynamic studies, there are a high number of published research papers on the topic, in which, only results are presented without the computer code. To the best of our knowledge, there are only a few open source or scientific software publically available, among them: Cruststrain (Schneider, 1982), Coulomb (Stein, 2003), grid_strain and grid_strain3 (Teza et al., 2008), STRAINGPS (Pietrantonio and Riguzzi, 2004) and SSPX (Cardozo and Allmendinger, 2009; Allmendinger et al., 2012). From the range of the commercial software, one can mention: “Geodetic strain analysis Tool” (Kedar et al., 2011). Despite the variety, however, developing a new software program is still interesting because: (a) the software redundancy can facilitate the validation of model solution by comparing the results obtained from different software (Hill and Blewitt, 2006), and (b) not all the software are publicly available or work on every operating system. Therefore, we developed the GeoStrain software at the Center for Research in Geomatics (CRG), Laval University, Canada, for estimating the strain rate tensors of a deformation area based on the method of least-squares collocation (Section 4). Furthermore, the software can filter the velocities at observation points, and predicts or interpolates the velocities at points of interest other than the observation points.

This paper has the following structure. Strain modeling methods are briefly reviewed. The mathematical definition of the deformation is presented right after and formulated as strain/strain rate and rotation/rotation rate tensors. Then, we elaborate on our selected method, least-squares collocation, and discuss the concept of trend surface and covariance functions in this method. In a while, we show how the strain rate tensor is derived from the deformation tensor, and develop mathematical formula for a Gaussian and a rational covariance function. Finally, the computational performance of the software is analyzed and its main features are introduced.

2. Strain modeling methodology

Crustal strain measurement has been long time an interesting topic in earth sciences. Advances of the navigation satellite systems have made it possible to detect small displacements of the
measuring points on earth’s surface with millimeter precision using geodetic networks. The strain analysis methods can be categorized based on three factors: (a) the adopted geodetic method, (b) the available data set or observables, and (c) the data processing strategy (Pietrantonio and Riguzzi, 2004). The latter factor can be performed in three ways: (a) segmentation, (b) inversion, and (c) gridding or interpolation. Even with the same input data, different approaches of strain calculation may produce different results. Therefore, knowing strengths and weaknesses of each method or the methodology, is very important not only from mathematical formulation and computation point of view, but also from geological and geophysical characteristics of the study area.

2.1. Segmentation approaches

Although the segmentation method is not limited to triangulation only and segments can have any geometrical shape, triangulation of the measuring points, e.g., using the Delaunay method is the most common example for this approach (Frank, 1966; Shen et al., 1996; Cai and Grafarend, 2007; Fernandes et al., 2007; Wdowinski et al., 2007). In this method, the area is segmented into several geometrical units and the local strain and rotation rates are computed for each unit, and finally all the individual results are integrated (Wu et al., 2011).

In this method, when the study area is small, the sphericity of the earth is neglected and earth surface is approximated by a flat surface model. Then, the strain can be calculated for each geometrical unit as formulated by Jaeger (1956, p. 39). As far as the angular aperture (dimensions of the geometrical unit in latitude and longitude expressed in radians) of the unit is small, the flat earth approximation is adequate. However, when the sphericity of the earth cannot be neglected, the formula must be developed in spherical coordinates (e.g., Savage et al., 2001; Reddy, 2013).

This method is more suitable for accounting known discontinuities like faults which practically make the strain field non-uniform, as well as strain analysis over small areas with limited number of measuring points. This method is very similar to the computation of stress and strain for each element in a finite element model, the other method of calculating a continuous strain field from geodetic data (Jiménez-Munt et al., 2003). In contrast to simplicity, this method does not allow a real estimation of the strain field but only a computation because the triangles are almost chosen arbitrarily. No outlier detection and removal is possible because of the zero or very small redundancy. Furthermore, the method produces a continuous displacement field but the obtained strain rate is discontinuous.

2.2. Inversion approaches

In general, the inversion based techniques are computationally expensive and require assumptions on the constitutive law of earth’s crust to relate the observed deformations to strain rates. For example, Spakman and Nyst (2002) have used seismic tomography for their inversion. They assigned strain rate to a previously discretized area by using different paths of relative displacement between pairs of observation points. In tomographic methods, the location of faults should be imposed to inversion models. Therefore, they can only be used for quantification of the slip deficit on known structures, but not for identification of new faults. Haines et al. (1998), Kreemer et al. (2000), and Beavan and Haines (2001), among others, have used point displacement observations of geodetic networks and strain evaluated by geologic and geophysical information such as earthquake focal mechanisms to invert for the Euler pole that locally minimizes the strain rate and velocity field residuals along a regional curvilinear reference system. Mazzotti et al. (2005) decomposed the velocity field into three components including a uniform horizontal strain rate, a translation (horizontal and vertical) and a horizontal rigid rotation. Then, they adjusted this seven-parameter model to the GPS velocities at the measuring points.

2.3. Gridding or interpolation approaches

In order to estimate the three dimensional strain filed, the continuous displacement field should be known. However, conventional geodetic observations provide discrete information and are limited to points on the two dimensional surface of earth. The continuous strain field cannot be directly calculated even by using the three dimensional displacement vectors of the measuring points. This makes the interpolation necessary. In this method, a functional model is first postulated for displacement fields and points, and then partial derivatives of displacement or velocity are calculated with respect to position at grid points as a direct access to the distribution of strain field. The interpolation methods are based on this assumption that strain field is homogeneous. A homogeneous field means that its mean is constant and its autocovariances depend on the distance between different points. Furthermore, if autocovariances are invariant with respect to direction and depend only on the distances of the points, the field is called isotropic (El-Fiky et al., 1997).

Interpolation techniques can determine displacements not only on earth’s surface but also inside and outside of the crust. For points within the crust, the strains are computed by analytically differentiating the interpolated function. Therefore, the result of the estimated strain depends on the geodetic observations as well as the interpolation technique. If the interpolation technique considers the physical reality of earth’s crust more, the computed strain filed is closer to the real one. There is an additional problem for calculating the strain for the points on the crust surface: mathematically interpolation techniques predict displacements even outside the medium, but in the physical reality, displacements are not defined outside the medium and boundary conditions should be introduced (Dermanis and Livieratos, 1983). If the strain calculation depends on such predictions, it does not reflect the physical reality. In this case, the surface strain only has a descriptive value and should not be used for conclusions on strain-stress relations.

The interpolation methods are the opposite side of the segmentation methods in terms of strengths and weaknesses. The continuous strain rate fields that they produce are suitable for identification of new structures without having to assume anything about the deformation mechanisms, or even if the local geology is not fully known. Therefore, different schemes have been suggested. Wdowinski et al. (2001) interpolated the velocity field along small circles relative to the pole of rotation that minimizes the residual (called pole of deformation). Allmendinger et al. (2007) used different approaches (nearest neighbor and distance weighted) to obtain continuous velocity fields from which a strain rate pattern can be calculated. Kahle et al. (2000) interpolated velocity fields in eastern Mediterranean by using a least-square collocation method based on a covariance function. Hackl et al. (2009) based their work on the concept of the splines in tension algorithm described by Wessel and Bercovici (1998), and performed a separate interpolation of the east and north velocity components on a regular grid for the Southern California GPS network.

Interpolation is necessary in the first approach for the data processing, because the strain tensor is estimated for every triangle. However, this method is not able to define how the strain is changed from a triangle to another. In contrast, interpolation is not necessary in the third strategy, because the strain field is supposed to be homogeneous. Interpolation of geodetic data, as Hackl et al.
(2009) discuss, is not limited to the study of velocity fields and the associated strain rates, but can also be applied to other fields like coseismic displacement data. In contrast, they address a disadvantage of their interpolation method which is unable to calculate absolute values of strain rates. Since this value is related to the gradient of the velocity field, the rate is limited by the grid size that is the minimal distance along which it is possible to observe a velocity change.

3. Mathematical background

The theory of elasticity states that an infinitesimally small change of position in a ductile body, at the first degree of approximation, can be described by a sum of a translation $u^{[r]}$, a dilatation $u^{[d]}$ and a rotation $u^{[R]}$. This can be resulted from the expansion of the displacement field $u(x)$ using the Taylor’s theorem:

$$u(x) = x(t_2) - x(t_1)$$

where $x(t_1)$ is the vector of position at time $t_1$. We can linearize (1) using the Taylor’s theorem to the first degree as:

$$u(x + d x) = u(x) + \frac{\partial u}{\partial x} d x_k$$

Using the Einstein summation convention, we can rewrite (2) as:

$$u_i(x + d x) = u_i(x) + \frac{\partial u_i}{\partial x_k} d x_k$$

in which $d x_k = (\partial u_i/\partial x_k)$ is a second order tensor in the $n$-dimensional space called deformation tensor that can be expanded into its three dimensional components as:

$$\begin{bmatrix}
\frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \frac{\partial u_1}{\partial x_3} \\
\frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} & \frac{\partial u_2}{\partial x_3} \\
\frac{\partial u_3}{\partial x_1} & \frac{\partial u_3}{\partial x_2} & \frac{\partial u_3}{\partial x_3}
\end{bmatrix} = \begin{bmatrix}
\epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\
\epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\
\epsilon_{31} & \epsilon_{32} & \epsilon_{33}
\end{bmatrix}$$

Diagonal elements of (4) show the changes in length or extensions along the three independent orthogonal directions $x_1$, $x_2$ and $x_3$, respectively. The off-diagonal elements show the angular changes in the body. An asymmetric tensor represents both distortion (change in shape) and rotation. However, in the case of rotation only, the strain tensor would be antisymmetric, i.e., $\epsilon_{i j} = -\epsilon_{j i}$ for $i \neq j$.

We know from the tensor theory that any tensor of the second order can be written as sum of a symmetric and an antisymmetric tensor (Wald, 1984, p. 26). Applying this concept on (3) yields (Peter, 2000):

$$u_i(x + d x) = u_i(x) + \frac{d u_i}{d x_j} d x_j + \frac{d u_j}{d x_i} d x_i$$

The terms to the right side of (5) represent a translation $T$, a dilatation $D$ and a rotation $R$, respectively. The strain tensor is defined by the symmetric part of the deformation tensor shown by $\epsilon_{i j}$ as

$$d u_{i j}^{[d]}(x) = \frac{d u_i}{d x_j} d x_j = \epsilon_{i j} d x_k$$

The expansion of the strain tensor definition for a three-dimensional body subject to an infinitesimally small amount of displacement is given by

$$\epsilon = \begin{bmatrix}
\frac{\partial u_1}{\partial x_1} & \frac{1}{2} (\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}) & \frac{1}{2} (\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}) \\
\frac{1}{2} (\frac{\partial u_2}{\partial x_1} + \frac{\partial u_1}{\partial x_2}) & \frac{\partial u_2}{\partial x_2} & \frac{1}{2} (\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}) \\
\frac{1}{2} (\frac{\partial u_3}{\partial x_1} + \frac{\partial u_1}{\partial x_3}) & \frac{1}{2} (\frac{\partial u_3}{\partial x_2} + \frac{\partial u_2}{\partial x_3}) & \frac{\partial u_3}{\partial x_3}
\end{bmatrix}$$

$$= \begin{bmatrix}
\epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\
\epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\
\epsilon_{31} & \epsilon_{32} & \epsilon_{33}
\end{bmatrix}$$

(7)

where $u_i(x_1, x_2, x_3)$ and $\epsilon_{i j}(x_1, x_2, x_3)$ are the same as in (1), and $\epsilon$ is the corresponding strain element of the strain tensor $\epsilon$ along the corresponding axes. Displacements and distortions are infinitesimally small at the limit of an elastic deformation, and therefore, $\epsilon_{i j} = \epsilon_{j i}$ for $i \neq j$. This implies that a symmetric strain tensor in a three dimensional space can be defined by six elements in total, namely, three diagonal and three upper or lower triangle elements. While off-diagonal elements of the strain tensor represent shear strain, its diagonal elements show the longitudinal strain equals to extension if they have positive signs or contraction if they have negative signs.

The rotation tensor is defined by the antisymmetric part of the deformation tensor shown by $\epsilon_{i j}$. It is usually written as a vector product of an infinitesimally small rotation matrix $\text{rot } u$ and an infinitesimally small displacement vector $d x_k$ as

$$d u^{[R]}(x) = \frac{d u_k}{d x_l} \frac{d x_l}{d x_k} = \frac{\text{rot } u \times d x_k}{2}$$

(8)

that is known to be equal to an infinitesimally small rotation $\omega_k$ of the vector $d x_k$ around the axis $\text{rot } u$ with an angle equals to $\text{rot } u/2$ (Peter, 2000):

$$\Omega_k = \frac{1}{2} \begin{bmatrix}
0 & -\text{rot } u_3 & \text{rot } u_2 \\
\text{rot } u_3 & 0 & -\text{rot } u_1 \\
-\text{rot } u_2 & -\text{rot } u_1 & 0
\end{bmatrix}$$

(9)

With the same condition for (7), the rotation tensor can be expressed as (Hackl et al., 2009):

$$\Omega = \begin{bmatrix}
0 & \frac{1}{2} (\frac{\partial u_1}{\partial x_2} - \frac{\partial u_2}{\partial x_1}) & \frac{1}{2} (\frac{\partial u_1}{\partial x_3} - \frac{\partial u_3}{\partial x_1}) \\
\frac{1}{2} (\frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}) & 0 & \frac{1}{2} (\frac{\partial u_2}{\partial x_3} - \frac{\partial u_3}{\partial x_2}) \\
\frac{1}{2} (\frac{\partial u_3}{\partial x_1} - \frac{\partial u_1}{\partial x_3}) & \frac{1}{2} (\frac{\partial u_3}{\partial x_2} - \frac{\partial u_2}{\partial x_3}) & 0
\end{bmatrix}$$

(10)

where $\omega$ is the corresponding strain rotation element of the strain rotation tensor $\Omega$. This geometrical interpretation conforms to the theorem postulated at the beginning of this section.

The concept of the strain tensors can yet be developed for the velocity gradients instead of the spatial gradients. Assuming infinitesimally small displacement of a position over time, the strain rate tensor $\dot{\epsilon}$ and the rotation rate tensor $\dot{\Omega}$ are defined as the partial derivatives of the strain tensor $\epsilon$ in (7) and the rotation tensor $\Omega$ in (10) with respect to time, respectively. They describe the rate of deformation (velocity gradient) and the change in rotation, respectively.
3.1. Spherical model of the strain rate tensor

The strain tensor in a three dimensional orthogonal coordinate system was introduced in (7). The strain rate tensor in a spherical coordinate system, e.g. on earth’s surface, can directly be expressed in terms of spherical displacements as (Love, 1944, p. 56; Savage et al., 2001; Wu et al., 2011):

\[
\begin{bmatrix}
\varepsilon_{\phi} & 1 & v_{\phi} \\
1 & \varepsilon_{\phi} & v_{\phi} \\
v_{\phi} & v_{\phi} & \varepsilon_{\phi} \\
\end{bmatrix}
\]

where \( \varphi \) and \( \lambda \) are the spherical coordinates and \( v_{\phi} \) and \( v_{\lambda} \) are spherical velocities of the point of interest in the latitude and longitude directions, respectively. \( \varepsilon_{\phi} \), \( \varepsilon_{\lambda} \) and \( \varepsilon_{\phi\lambda} \) represent the three independent components of the strain rate tensor.

3.2. Representation of the strain rate tensor

Since the strain tensor is symmetric, it is possible to find a reference frame where the shear strain is zero and the two other components are maximal. The principal deformations \( \lambda_1 \) are obtained by diagonalizing the strain rate tensor. This is accomplished by solving the classical eigenvalue problem for the strain rate tensor as:

\[
\det(\dot{\varepsilon} - \lambda I) = 0
\]

Solving this problem is equivalent to transforming the strain rate tensor to a new coordinates system called major axis system, in which, the tensor is expressed by a set of independent vectors. While eigenvectors show the principal axes of the strain rate tensor, eigenvalues show the principal strain rates. In this way, we get a diagonal tensor \( \dot{\varepsilon}' \) whose elements are the eigenvalues of the original strain rate tensor. In general, if \( \dot{\varepsilon} \) is an \( n \times n \) matrix and has \( n \) independent eigenvectors, then \( \dot{\varepsilon} \) can be diagonalized as

\[
\dot{\varepsilon}' = \mathbf{R}^{-1} \dot{\varepsilon} \mathbf{R}
\]

where \( \mathbf{R} \) is eigenvector matrix whose columns are eigenvectors of \( \dot{\varepsilon} \) (Strang and Borre, 1997, p. 222) which is invertible as its columns are linearly independent. Because \( \dot{\varepsilon} \) is symmetric, these eigenvectors make an orthogonal reference system whose axes pointing to the direction of the maximal deformation. For a two dimensional symmetric random strain rate tensor, the eigenvalues \( \lambda_{1,2} \) or maximum and the minimum principal strains as well as the orientation parameter \( \alpha \) known as the principal angle are obtained as (Cai and Grafarend, 2007)

\[
\lambda_{1,2} = \frac{1}{2} \left( \lambda_1 + \lambda_2 \pm \sqrt{\left(\lambda_1 - \lambda_2 \right)^2 + 4\dot{\varepsilon}_{12}^2} \right), \quad \lambda \in \mathbb{R}
\]

\[
\alpha = \frac{1}{2} \tan^{-1} \left( \frac{\lambda_1 - \lambda_2}{2\dot{\varepsilon}_{12}} \right), \quad -\pi/2 < \alpha \leq \pi/2
\]

The principal angle is obtained by rotating the strain tensor as \( \alpha \) and set the shear component to zero. Using (14), it is possible to visualize a two dimensional symmetric strain rate tensor. Fig. 1A shows a strain ellipse and strain crosses with axes co-linear with the eigenvectors of the strain tensor, i.e., the semi-major axis is coincides with the maximum principal strain and the semi-minor axis with the minimum principal strain. Directions of the strain crosses are defined by the signs of the eigenvalues: when \( \text{sgn}(\lambda_1) = \text{sgn}(\lambda_2) = +1 \), there is an extension on the point, and when \( \text{sgn}(\lambda_1) = \text{sgn}(\lambda_2) = -1 \), it is a contraction. Fig. 1B illustrates the strain hyperbola and strain crosses in which its axes are directed along the eigenvectors of the strain tensor. In this case, \( \lambda_{1,2} \) Show the real and the imaginary axes of the strain hyperbola, respectively.

Fig. 1. A typical (A) strain ellipse and (B) strain hyperbola and their strain crosses. Axes are directed along the eigenvectors of the strain tensor.
3.3. Specifications of the strain rate tensor

The fractional change in volume of a body in the elastic limit when its surface area approaches to zero, is called dilatation $\Theta$ (Lowrie, 2007). The dilatation is defined as the trace of the strain tensor, which is equal to the divergence of the deformation field and is invariant to coordinate transformation (Section 3.2):

$$\Theta = \sum_{i=1}^{3} \epsilon_{ii} = \sum_{i=1}^{3} \frac{\partial \bar{u}_i}{\partial x_i} - \text{div} \bar{u}$$  \hspace{1cm} (15)

In two dimensions, dilatation is corresponding to the relative variation of a surface area, and therefore, can indicate regions with thrusting or normal faulting systems. This is called area strain and is shown as

$$\epsilon_{\text{area}} = \epsilon_{x1} + \epsilon_{x2}$$  \hspace{1cm} (16)

However, when dealing with strain rate tensors, the term is changed to the dilatation rate and is denoted by $\Theta$.

The maximum shear strain is defined as

$$\epsilon_{\text{max shear}} = \frac{1}{2} \left( \lambda_1 - \lambda_2 \right)$$  \hspace{1cm} (17)

where $\lambda_{1,2}$ are the same in (14). Its direction angle $\beta$ is oriented $45^\circ$ from $\sigma$ as calculated by (14). Therefore, $\beta = \theta \pm 45^\circ$, and additional constrain is necessary to distinguish the exact direction of the maximum shear. Since motion along faults relates shear on the structure, the maximum shear strain and its direction can be used for identification of active faults. Faults having the same orientation with this direction are most likely ruptured in a seismic event.

3.4. Effect of the vertical velocity on the strain rate tensor

In the tectonically active regions, horizontal strain rates can be derived directly from the horizontal velocity field. This simplification is justified by the very low vertical velocity gradients compared to the horizontal velocities. However, this is not the case for the intraplate regions such as the Saint Lawrence valley, eastern Canada, in which the vertical velocities can be 5–10 times larger than the differential horizontal velocities (Mazzotti et al., 2005). In such a case, Malvern (1969) considers the vertical velocities and proposes a complete formulation to calculate the horizontal strain rate tensors as

$$\dot{e} = \begin{bmatrix} \frac{\partial \bar{u}_1}{\partial x_1} + \frac{\bar{v}_1}{r} & 1 \frac{\partial \bar{u}_1}{\partial x_1} + \frac{\partial \bar{u}_1}{\partial x_2} \\ 1 \frac{\partial \bar{u}_1}{\partial x_1} + \frac{\partial \bar{u}_1}{\partial x_2} & \frac{\partial \bar{u}_1}{\partial x_2} + \frac{\bar{v}_2}{r} \end{bmatrix}$$  \hspace{1cm} (18)

where $\bar{u}(\bar{x}, \bar{y}, \bar{z})$ is the velocity vector of the deformation in north, east and up direction, $\bar{x}(\bar{u}_1, \bar{u}_2, \bar{u}_3)$ is the position vector of the measuring stations in a local geodetic system, and $\mathbb{r}$ is earth’s radius, $\bar{y}$ is the vertical velocity that adds an isotropic extension/shortening component to the horizontal strain rate tensor. This is equivalent to add the term $\frac{\bar{v}_2}{r} \mathbb{I}$ to $\bar{e}$ in (11). This method can optionally be used in GeoStrain to calculate the strain rate tensors.

4. Least-squares collocation

The major difficulty in the computation of the strain rate tensors is to find a mathematical definition of the continuous deformation field. However, the geodetic observables are discrete quantities at the measuring points, and therefore can only provide a discontinuous deformation field. The Least-Squares Collocation (LSC) is a superior method to find this deformation field. Despite the mathematical strength of the LSC method, it was rarely applied in the crustal deformation studies, mainly because estimating an empirical covariance function for the crustal deformation data had long been considered to be difficult (Fujii and Xia, 1993). Wu et al. (2011) compared different methods of Delaunay triangulation, multi-surface functions, spherical harmonics, and LSC for computing GPS strain rates using simulated and real data, and concluded that the latter method is superior in terms of precision of the estimated deformation signal and robustness to noise in the data. Specially, sparseness of the data has less influence on the results compared with other methods.

Moritz (1980) provides a detailed discussion on the subject, however, we give a more general and simpler description of the method from El-Fiky et al. (1997) and Mikhail and Ackermann (1976) and some more details about implementation of the method in the form of a computer software. The LSC method can be considered as an extension for the ordinary least-squares adjustment where stochastic correlated signal $\mathbf{s}$ is added to the deterministic model of the unknown parameters $\mathbf{A}x$ and to the stochastic noise $\mathbf{n}$. Therefore, the linear form of LSC is formulated as

$$\mathbf{l} = \mathbf{Ax} + \mathbf{s} + \mathbf{n}$$  \hspace{1cm} (19)

where $\mathbf{l}$ is the observation vector, $\mathbf{A}$ is the design matrix of the linearized trend function $f(x)$, $\mathbf{x}$ is the vector of unknown parameters, $\mathbf{s}$ is the vector of correlated signal at observation points, and $\mathbf{n}$ represents the vector of the stochastic noise in the system. In this form of equation, $\mathbf{l}$ composed of a deterministic part $\mathbf{Ax}$, and two stochastic parts of $\mathbf{s}$ and $\mathbf{n}$. If we obtain the parameter $\mathbf{x}$ by adjustment, calculate the signal $\mathbf{s}$ at any arbitrary point (even without any observed value) by interpolation, and remove the noise $\mathbf{n}$ by filtering, then we can say LSC is a combination or a collocation of adjustment, interpolation and filtering. In the concept of this text, $\mathbf{s}$ and $\mathbf{n}$ are respectively correspond to the crustal deformation and the measuring errors at the observation points.

The deterministic part $\mathbf{Ax}$ in (19) can be eliminated using free network least-squares adjustment with an a-priori constraint, such as the zero-sum condition. Then, only the stochastic part is remained as

$$\mathbf{z} = \mathbf{s} + \mathbf{n}$$  \hspace{1cm} (20)

$s$ and $\mathbf{n}$ have random values and are assumed to be unbiased. Therefore, their statistical expectations are both zero, i.e., $E(\mathbf{s}) = 0$ and $E(\mathbf{n}) = 0$, then:

$$E(\mathbf{z}) = E(\mathbf{s}) + E(\mathbf{n}) = 0$$  \hspace{1cm} (21)

where $E(\mathbf{z})$ is the statistical expectation operator. Using this operator, we can calculate the covariance matrices of the signal $\mathbf{C}_s$ and noise $\mathbf{C}_n$ as

$$\mathbf{C}_s = E(\mathbf{s}\mathbf{s}^T) \quad \text{and} \quad \mathbf{C}_n = E(\mathbf{n}\mathbf{n}^T)$$  \hspace{1cm} (22)

where $\mathbb{T}$ denotes the matrix transpose operator. Assuming the signal and the noise have no correlation, the cross-covariance matrix of the signal and noise $\mathbf{C}_{sn}$ is given by

$$\mathbf{C}_{sn} = E(\mathbf{s}\mathbf{n}^T)$$  \hspace{1cm} (23)

The signal $\mathbf{\hat{s}}$ can be interpolated at any arbitrary point, including the observation point $\mathbf{s}$ and any other point $\mathbf{t}$, as
\[ \hat{s} = \left[s^T t^T \right]^T \]

A solution for (24) can be found by applying the least-squares minimum criterion as

\[ s_i C_i^{-1} s + n_i C_n^{-1} n \rightarrow \min \]

This minimum condition problem can be solved using the Lagrange multipliers method (Cross, 1990). Thus, the unknown parameters are obtained as (Moritz, 1972):

\[ \hat{s} = C_s k \]

\[ \hat{n} = C_n k \]

where \( k = C_i^{-1} z \) is called the correlation vector, and \( C_s \) is the covariance matrix between signal \( s \) at the observation point and the signal \( t \) at any arbitrary point. The total covariance of observations is:

\[ C = C_s + C_n \]

The correlation vector \( k \) depends only on the observations, and therefore contains full information about them and their spatial distribution. Nevertheless, the interpolated signal in (26) is a function of the site-dependent covariance matrix \( C_s \) and the site-independent correlation vector \( k \).

### 4.1. Trend surface

Stationary random functions have constant mean with a correlation function approaching to the mean when the distance approaches to the infinity. We consider the mean as zero to simplify the presented formulae for the LSC method. Therefore, for such a function \( C(d) \to 0 \) as \( d \to \infty \). This property will hold for a set of observational data, when the trend is removed. Trend is generally defined as the component of a random phenomenon with a period larger than the recorded data sample (Mikhail and Ackermann, 1976), and it is often represented by a linear or very low-order polynomial terms that depends to the nature of observations (Fig. 2). In this figure, both the observation and interpolation points are randomly distributed around the trend line in a way that their means are zero. This characteristic is necessary to remove the trend.

When a proper expression for the trend surface was selected, the original observation points are transformed from the original datum to the trend surface using the least-squares method. Once the trend is removed, the signal surface is referred to as the trend surface. The covariance function relates the auto-covariances of two points. The covariance function is deﬁned only to the length of the vector separation, and usually the function would take the general form shown in Fig. 3. This covariance function is called empirical covariance function. When the covariance function is deﬁned from a set of data, it is possible to evaluate the covariance for any two points with an arbitrary distance. In addition, we can commonly apply the covariance function to another data set with different time interval in the same area or another area with similar tectonic settings (El-Fiky et al., 1997).

We can ﬁnd the covariance of the observation \( l \) by applying the error propagation law to (19) considering the point that \( Ax \) is deterministic:

\[ G(d) = C_s (d) + C_n (d) \]

This equation shows that the covariance of the observations is the sum of the covariances for the signal and the noise when they are uncorrelated. Fig. 3A shows the graphical representation of this equation. The figure shows that the covariance of the noise at any distance \( d \) is equal to the vertical distance between the covariances of the observations and the covariance of the signal. In practice, however, we can assume that there is no spatial correlation in the noise vector. Thus, \( C_s (d) \) would be a single point and the three covariance functions would be as shown in Fig. 3B. Therefore, (29) holds only when \( d \to 0 \). We have:

\[ G(0) = C_s (0) + C_n (0) \text{ when } d \to 0 \]

\[ G(d) = C_s (d) \text{ when } d \neq 0 \]

Even though we can deﬁne a theoretical covariance function in some special cases, it is usually derived from a local data by calculating the variance and covariance of sample points from the observations \( l_i (1 \leq i \leq n) \) using:

\[ G(0) = \frac{1}{n} \sum_{i=1}^{n} l_i^2 \text{ (variance)} \]

\[ G(d_p) = \frac{1}{n_p} \sum_{i=1}^{n_p} l_{ij} \text{ (covariance)} \]

where distances between two data points are divided into finite discrete intervals \( P \), in which \( n_p (1 \leq p \leq P) \) pairs of data whose distance \( d_p \) is dropped in interval \((2p - 3)\delta < d_p \leq (2p - 1)\delta\) for \( p \in \{2, 3, 4, \ldots, P\} \), and \( 0 < d_p \leq \delta \) for \( p = 1 \). \( \delta \) is an arbitrary increment and its value is usually chosen on the basis of the knowledge of the observations (Fig. 3B). Number of sample points in each interval is very important to make the estimated covariances statistically meaningful. Intervals with small number of

\[ \text{Fig. 2. Concept of the collocation method in a one dimensional longitudinal case.} \]

The trend and the signal in data are shown by \( Ax \) and \( s \) in (19), respectively. The noise \( n \) is the difference between the signal and the observation points.
sample points might lead to a covariance point that stands out significantly from others. Such a point must be discarded before fitting the covariance function.

We can then plot the covariance histogram using (31) and fit a function to the covariance points. Function of the form constant, sinusoidal, Gaussian, rational, exponential, exponential cosine, and exponential sine and cosine, are possible based on the problem condition. To obtain a real solution for the minimum problem of the collocation stated in [25], however, all modeled correlation matrices must be positive definite (Peter, 2000). In general, we can consider the following conditions for the covariance function: (a) it is only a function of the distance between points that makes the covariance matrix positive definite, (b) it is a decreasing function of distance approaching zero when the distance approaches the infinity, (c) it has a definite slope or derivative at distance zero. The last condition has a special importance because the strain is defined as the first derivative of the signal relative to the distance.

Two covariance functions are often used in geodynamic applications. The first one is a Gaussian function defined as (Mikhail and Ackermann, 1976)

\[ f(d) = \exp(-b^2d^2) \]  

and the second one is a rational function defined as (Kahle et al., 2000; Peter, 2000)

\[ f(d) = \frac{a}{1 + b^2d^2} \]  

where \( a \) and \( b \) are constant parameters that should be estimated, e.g., by the ordinary least-squares adjustment. \( d \) shows the distance among points. These functions are used to compute the covariance matrix \( C \) in (26) for either observation or arbitrary points where the signal is to be estimated. In both equations, the parameter \( a \) has the same unit of covariance and is considered as \( C(0) \), \( C(0) \), and parameter \( b \) is the so-called “correlation length” or “characteristic wavelength” and has the same unit of distance \( d \). The second parameter reflects the tectonic settings of the study area and is used as a tuning parameter for the dependency among points. Both covariance functions are available in GeoStrain.

There are some issues regarding the estimation of parameters of the covariance. (a) Linearization of (32) and (33) for least-squares adjustment requires good approximated initial points, otherwise convergence problem might occur. (b) When the filtering is applied, i.e., the observations are considered erroneous, the first point of the variance–covariance histogram \( C(0) \) should not be taken into account. However, when the interpolation is done without filtering, this point is used and the covariance function should pass through it. This is normally done by introducing a constraint equation. (c) Alternatively, we can construct a common auto-covariance matrix \( C \) in (28) for both the interpolation and filtering. However, in the case of interpolation, we have to substitute the diagonal elements of \( C \) with \( C(0) \) or \( C(0) \) (which are actually equal), and in the case of filtering, we should change the diagonal elements of \( C \) with \( C(0) \) but not \( C(0) \). The later alternative was implemented in GeoStrain.

4.3. Multidimensional covariance functions

So far, we have discussed the covariance functions of one-dimensional random functions. This concept, however, can be extended to a homogeneous and isotropic random field. In this case, auto-covariance and cross-covariance functions should be evaluated respectively for each element and each pair of elements of the random vector with the same method presented in Section 4.2. The general form of the covariance function for a two-dimensional random vector with components \( x_i(u) \) and \( x_j(u) \) is formulated as

\[ C_{ij}(d_p) = \frac{1}{n} \sum_{i<j} x_i X_j \]  

Equation (34) defines the auto-covariance and cross-covariance functions when \( k, l \in \{1, 1\}, (2, 2) \) and \( k, l \in \{1, 2\}, (2, 1) \), respectively. The covariance matrices can be constructed after calculating the auto-covariance and cross-covariance functions. However, if all the estimated cross-covariances are small enough and insignificant, we can consider a \( n \)-dimensional random vector as \( n \) one-dimensional random functions, and then, interpolate and filter such a random vector with the same method of interpolation and filtering of the \( n \) separate one-dimensional random functions.

5. GeoStrain software description

We developed the GeoStrain software based on mathematical models introduced in Sections 3 and 4 and the appendix using MATLAB with a Graphical User Interface (GUI). This choice makes the software cross-platform that runs under the majority of operating systems (OS). GeoStrain is available as an open source software under the BSD license (available at: www.opensource.org/licenses/bsd-license.php), and can be downloaded from http://sourceforge.net/projects/geostrain/. The source code includes a set of MATLAB functions, GUIs, map data files and user’s manual as well as some sample input files. An X-Window environment is
necessary to be installed in UNIX like operating systems before running the software and MATLAB should also be started with the Java Virtual Machine (JVM) enabled. The Mapping toolbox of MATLAB is also required. Fig. 4 shows the main window of the software.

GeoStrain accepts three different types of ASCII input files, namely, velocity file, grid file and fault file. In the former file, all the velocities and their associated errors defined in the local geodetic Cartesian coordinate system are stored in ten columns. In grid files, geographical coordinates (latitude and longitude) of grid points in decimal degrees are stored in two columns. Fault files have the same format of the grid files, except that the first line of every break line representing a fault is a single ‘4’ sign.

Estimation of the empirical covariance function is done separately for the horizontal and vertical velocities. All spherical distances among stations are calculated using (37), and the software plots the histogram of the distances and reports the number of points per each bin. The sample variance points are calculated and plotted for each bins separately for the north, east, and covariance points for north-east using (34). The user has the option to select a covariance function of either Gaussian form in (32) or a rational form in (33). The software reports the correlation length equals to \( C_0 / 2 \). The correlation length is a distance that there is no or negligible correlation among the measuring points after that. Geometrically, the correlation length is the distance that the covariance function becomes asymptotic to the distance axis. Due to the mathematical form of the covariance functions, however, this distance tends towards infinity. Therefore, to get an estimation of the correlation length that is also in accordance with the definition, this distance should be manually multiplied by two.

Filtering or interpolating the observations, or estimating the strain and rotation tensors becomes possible when the coefficients of the covariance functions were estimated or loaded. Filtering and interpolation are done separately for horizontal and vertical velocities. The input velocities are filtered at the observations points using (26) or (27). However, velocities are interpolated at grid points using (26), which are not necessary to be regular. Estimation of the strain rate and rotation tensors is possible after estimating both the horizontal and vertical covariance functions (Eqs. (32) and (33)) at either of observation points, grid points or along faults. The effect of the vertical velocities can optionally be considered as stated in (18).

6. Computational performance analysis

In order to test the computational performance of the software, a velocity field was generated for \( 2^\circ \times 2^\circ \) grid points in an area bounded to 60°W–90°W and 40°N–60°N using the following model (Wu et al., 2011):

\[
\begin{align*}
\psi_v &= 10 \sin(5\lambda) \cos(3\phi) + 9 \sin(3\phi^2) \cos(3\phi^2) + 1.5\phi^2 - 4\phi^3 + 5\lambda - 8\phi + \text{randn}() \\
\psi_e &= 12 \sin(3\phi) \cos(2\phi) + 11 \sin(4\phi^2) \cos(4\phi^2) + 3.5\phi^2 - 6\phi^3 + 1 - 2\phi + \text{randn}()
\end{align*}
\]

where \( \phi \) and \( \lambda \) are the latitude and longitude of the grid points in radians, and \( \psi_v \) and \( \psi_e \) are their north and east velocities in mm/yr. The \text{randn}() function, in this model, produces Gaussian random numbers and simulates the noise associated with the GPS velocities.
The idea to add this function is to test the effect of noise level on the estimated results. Therefore, we produced three velocity fields with the noise level of 0, \( \pm 0.5 \text{ mm/y} \) and \( \pm 1.0 \text{ mm/y} \) named model A, B and C, respectively. The performance analysis was done in two steps for these three models. In the first step, velocities were estimated at the grid points and compared with the theoretical velocities computed from (35). In the next step, strain rate tensors were estimated at the grid points and compared with the theoretical strain rate tensors computed from (11). Before presenting results, however, the parameter selection method is introduced. Although both correlation functions in (32) and (33) are available in GeoStrain, the presented results are based on the former one whose parameters can be estimated by a linear regression after logarithmic transformation to 

\[
\ln(f(d)) = \ln(a) - b \cdot d^2.
\]

For this specific data set, we found \( a = 75 \text{ (mm/yr)}^2 \) and \( b^2 = 2.3 \times \left(10^3 \text{ km}\right)^{-2} \) which is equal to the correlation length of 550 km.

The results of interpolating signal at the grid points are presented in Fig. 5A–C. The figures show that, when there is no noise in the data, the estimated velocities are identical with the predicted signal at observation points with the correlation coefficient equals to 1, shown in Fig. 5D. When more noise is added, the difference between the theoretical velocities and the predicted signal increased while their corresponding correlation is decreased. Furthermore, the latter figure shows that the correlation coefficient of the north velocities decreases faster than the east velocities due primarily to the higher signal to noise ratio in longitudes compared with latitudes.

The results of interpolating signal at the grid points are presented in Fig. 5A–C. The figures show that, when there is no noise in the data, the estimated velocities are identical with the predicted signal at observation points with the correlation coefficient equals to 1, shown in Fig. 5D. When more noise is added, the difference between the theoretical velocities and the predicted signal increased while their corresponding correlation is decreased. Furthermore, the latter figure shows that the correlation coefficient of the north velocities decreases faster than the east velocities due primarily to the higher signal to noise ratio in longitudes compared with latitudes.

Fig. 6 displays the results of theoretical and estimated strain rate tensors, rotation rate tensors, and maximum shear strain rates in the test area using the model A velocity field. Comparison of the results shows that the theoretical and estimated values are almost identical. The correlation coefficients between rotation rates and maximum shear strain rates are 0.99 and 0.97, respectively. However, dissimilarity emerges with introducing noise to the velocities (model B and C), but the correlation coefficients for rotation rates and maximum shear strain rates stays above 0.9 same as Fig. 5D (this was tested but not shown here). This shows basically the strength of the LSC method to deal with noise in the data and its superiority over other strain calculation methods as compared by Wu et al. (2011) and also implies that the software works correctly.

7. Conclusion

GeoStrain is MATLAB program that makes it almost an OS independent software. The main application of the software is to estimate the strain and rotation rate tensors from 2+1D velocity observations using the LSC method. Even though this is a common method in the field of geodetic strain analysis, the fact that the effect of the vertical velocity is considered in the structure of the strain rate tensors is important especially when the vertical velocities are significant compared to the horizontal velocities. GeoStrain has the following advantages: (a) no limitation on the number of input stations, (b) existing of two different types of covariance functions
namely Gaussian and rational functions for fitting to the variance–covariance points, (c) using two methods for making the histogram of the covariance sample points and interactive editing of the points, (d) possibility to save or load coefficients of covariance functions and covariance sample points, (e) several options to estimate the strain rates at points of interest, (f) filtering velocity data at the observation points and interpolating them at points of interest other than the measuring points, (g) calculating the normal and shear strain rates on the fault lines, (h) visualization of the input data and output results, and (i) exporting the stations and strain or rotation rates data as text, Google Earth, and ESRI shape files.

**Acknowledgment**

This project has been partly funded by the grants of the Canadian NSERC and Forestry, Geography and Geomatics Faculty (FFGG) of the Laval University grants. We included two functions of the geodetic toolbox developed by Mike Craymer in our software in order to reduce the number of the required MATLAB toolboxes to run the software. Furthermore, the software relies on the “arrow” function developed by E. A. Johnson (www-bcf.usc.edu/~johnsone/) to plot strain rate tensors, and the “findjobj” function developed by Y. Altman (www.undocumentedmatlab.
Appendix: calculating the strain tensor

The LSC method is used to estimate the signal at points of interest. The tectonic signal contains the dilatation and the main part of the rotation. The first spatial derivative of the signal gives the deformation tensor \( \mathbf{D} \) which can be decomposed into the strain tensor and the rotation tensor (Peter, 2000). Therefore, we can calculate the deformation tensor using (26) as

\[
\mathbf{D} = \frac{\partial \mathbf{C}_{\text{st}}}{\partial \mathbf{x}} = \mathbf{C}_{\text{tt}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \mathbf{C}_{\text{tt}} - \mathbf{k}
\]

The second term in (36) is zero because the correlation vector \( \mathbf{k} \) is independent of the position \( \mathbf{x} \). Thus, in order to calculate the deformation tensor, it is necessary to calculate the spatial derivative of the covariance function \( \mathbf{C}_{\text{tt}} \).

We use spherical distances among points to estimate the covariances in (32) and (33), however, the vector of displacements is defined in a local topocentric reference systems, centered on every point. The spherical distance between points \( i \) and \( j \) is calculated as

\[
t_i = \xi, \quad \theta_i = \xi \cos^{-1} (\sin \varphi_i \cos \varphi_j - \sin \varphi_j \cos \varphi_i \cos \Delta \tau_i)
\]

where \( \xi \) is the spherical radius of the earth, \( \theta_i \) is the radial distance, \( \varphi_i \) and \( \varphi_j \) are the latitudes, and \( \Delta \tau_i \) is the difference of the longitudes between points \( i \) and \( j \). We can now substitute (37) in (32) and (33), and calculate the spatial derivative of \( \mathbf{C}_{\text{tt}} \) as

\[
\frac{\partial \mathbf{C}_{\text{tt}}}{\partial \mathbf{x}} = \frac{\partial \mathbf{C}_{\text{tt}}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = -2ab^2 \exp(-b^2r^2) \frac{\partial \mathbf{r}}{\partial \mathbf{x}}
\]

and

\[
\frac{\partial \mathbf{C}_{\text{tt}}}{\partial \mathbf{x}} = \frac{\partial \mathbf{C}_{\text{tt}}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{x}} = (1 + b^2r^2) \frac{\partial \mathbf{r}}{\partial \mathbf{x}}
\]

We can write the spatial derivative of \( \mathbf{r} \) using (37) as

\[
\frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \mathbf{e} \frac{\partial \theta}{\partial \mathbf{x}}
\]

The derivative of \( \theta \) relative to \( \mathbf{x} \) has different forms of:

\[
\frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \mathbf{e} \sin \varphi_i \cos \varphi_j \partial \theta_i
\]

and

\[
\frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \mathbf{e} \cos \varphi_i \partial \varphi_i
\]

We substitute (41) and (42) in (40):

\[
\frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \frac{\partial \mathbf{r}}{\partial \theta_i} = -\frac{1}{\sin \theta} \left( \cos \varphi_i \sin \varphi_j - \sin \varphi_i \cos \varphi_j \cos \Delta \tau_i \right)
\]

and

\[
\frac{\partial \mathbf{r}}{\partial \mathbf{x}} = \frac{\partial \mathbf{r}}{\partial \varphi_i} = -\frac{\cos \varphi_j \sin \Delta \tau_i}{\sin \theta}
\]

Finally, we can substitute (43) and (44) in both (38) and (39), and get the deformation tensor for each of the covariance functions as

\[
\mathbf{D} = 2ab^2 \exp(-b^2r^2) \left( \frac{1}{\sin \theta} \left( \cos \varphi_i \sin \varphi_j - \sin \varphi_i \cos \varphi_j \cos \Delta \tau_i \right) \mathbf{e} \right)
\]

for the Gaussian covariance function, and:

\[
\mathbf{D} = 2ab^2 \exp(-b^2r^2) \left( \frac{1}{\sin \theta} \left( \cos \varphi_i \sin \Delta \tau_i \right) \mathbf{e} \right)
\]

for the rational covariance function. Then, we can calculate the strain and the rotation tensor by decomposing \( \mathbf{D} \) into a symmetric and an antisymmetric part as

\[
\mathbf{D} = \frac{1}{2} (\mathbf{D} + \mathbf{D}^T) + \frac{1}{2} (\mathbf{D} - \mathbf{D}^T) = \varepsilon + \Omega
\]


Moritz, H., 1972. Advanced Least-squares Methods. Ohio State University, Columbus, USA.


