Supplementary materials:

A multi-pixel time series analysis method accounting for ground motion, atmospheric noise and orbital errors

R. Jolivet[1] **& M. Simons**[2]

[1]Laboratoire de Géologie, Département de Géosciences, École Normale Supérieure,

CNRS UMR 8538, PSL Research University, 75005 Paris, France

[2]Seismological Laboratory, California Institute of Technology, Pasadena, California 91125, USA

Corresponding author: Romain Jolivet, romain.jolivet@ens.fr

We describe technical details for the implementation of the method presented in the main text, including details on the design matrix of the inverse problem, on the implementation of the conjugate gradient and on the Fourier-domain covariance convolution method. We also describe the set of covariances used for the synthetic test and for the case of northern Chile, as well as performance summaries.

1 Time series analysis design matrix

The design matrix of the time series inversion is given as

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_1 & 0 & \dots & 0 & 0 & \mathbf{O}_1 \\ 0 & \mathbf{G}_2 & 0 & \dots & 0 & \mathbf{O}_2 \\ \vdots & 0 & \ddots & 0 & \vdots & \vdots \\ 0 & \dots & 0 & \mathbf{G}_{n-1} & 0 & \mathbf{O}_{n-1} \\ 0 & 0 & \dots & 0 & \mathbf{G}_n & \mathbf{O}_n \end{pmatrix}, \tag{1}$$

where $\mathbf{G}_i$ is the design matrix for the i-th pixel and $\mathbf{O}_i$ is the corresponding ramp matrix.

In the case of the SBAS method, the matrix $\mathbf{G}_i$ of one pixel i, relating the phase at each acquisition to the interferograms contains 0, 1 and -1 values [called incidence matrix in *Schmidt and Burgmann*, 2003]. In a simple example where we have all 5 interferograms possible between 4 acquisitions, since the model parameters are the phase values at the time of acquisitions, the matrix $\mathbf{G}_i$ is simply given as

$$\mathbf{G}_i = \begin{pmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \tag{2}$$

Such a system is by definition underdetermined as it is impossible to find the absolute value at all acquisitions from differential interferograms, but rather the phase values with respect to a reference in time. We remove the column corresponding to the pre-defined reference in the method implemented in the code. This block matrix is already sparse itself, allowing to build very large problems when combined with other pixels.

In the case of the dictionary approach, the block matrix $\mathbf{G}_i$ is full. Given the complexity of the parameterization, the available computer memory will be a limiting factor. However, for simple cases, this block matrix might be very small. If we consider the func-

tion $f(t)$, so that,

$$f(t) = a_0 g_0(t) + a_1 g_1(t) + \cdots + a_n g_n(t), \tag{3}$$

where a_j is the j-th term of the model vector $\mathbf{m}$ and $g_j(t)$ the corresponding basis function, then, for the same set of interferograms as in the previous example, the unit block matrix $\mathbf{G}_i$ is given as

$$\mathbf{G}_i = \begin{pmatrix} g_0(t_2) - g_0(t_1) & g_1(t_2) - g_1(t_1) & \dots & g_{n-1}(t_2) - g_{n-1}(t_1) & g_n(t_2) - g_n(t_1) \\ g_0(t_3) - g_0(t_1) & g_1(t_3) - g_1(t_1) & \dots & g_{n-1}(t_3) - g_{n-1}(t_1) & g_n(t_3) - g_n(t_1) \\ g_0(t_3) - g_0(t_2) & g_1(t_3) - g_1(t_2) & \dots & g_{n-1}(t_3) - g_{n-1}(t_2) & g_n(t_3) - g_n(t_2) \\ g_0(t_4) - g_0(t_2) & g_1(t_4) - g_1(t_2) & \dots & g_{n-1}(t_4) - g_{n-1}(t_2) & g_n(t_4) - g_n(t_2) \\ g_0(t_4) - g_0(t_3) & g_1(t_4) - g_1(t_3) & \dots & g_{n-1}(t_4) - g_{n-1}(t_3) & g_n(t_4) - g_n(t_3) \end{pmatrix}, \tag{4}$$

where t_1 to t_4 are the time of each of the 4 acquisitions.

In the case of NSBAS, the block matrix $\mathbf{G}_i$ is separated into 2 blocks. The first block concerns the reconstruction of the phase while the second block provides the functional description of the phase. Therefore the block matrix $\mathbf{G}_i$ is written as

$$\mathbf{G}_i = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & 0 & 0 & g_0(t_1) & g_1(t_1) & \dots & g_{n-1}(t_1) & g_n(t_1) \\ 0 & 1 & 0 & 0 & g_0(t_2) & g_1(t_2) & \dots & g_{n-1}(t_2) & g_n(t_2) \\ 0 & 0 & 1 & 0 & g_0(t_3) & g_1(t_3) & \dots & g_{n-1}(t_3) & g_n(t_3) \\ 0 & 0 & 0 & 1 & g_0(t_4) & g_1(t_4) & \dots & g_{n-1}(t_4) & g_n(t_4) \end{pmatrix}. \tag{5}$$

Ramp parameters are adjusted to interferograms in all three formulations.

2 Conjugate direction method

In the case of a linear forward problem, at each iteration n, the *steepest ascent* vector, γ_n, is obtained from the model vector $\mathbf{m}_n$ given that

$$\gamma_n = \mathbf{C}_\mathrm{m}\mathbf{G}^T\mathbf{C}_\mathrm{d}^{-1}(\mathbf{G}\mathbf{m}_n - \mathbf{d}) + (\mathbf{m}_n - \mathbf{m}_\mathrm{prior}). \tag{6}$$

This vector is then combined with all previous *steepest ascent* vectors to form ϕ_n, defined as

$$\phi_n = \gamma_n + \alpha_n \phi_{n-1}, \tag{7}$$

where α_n is obtained through the Polak-Ribière formula [*Tarantola*, 2005; *Polack and Ribière*, 1969] such as

$$\alpha_n = \frac{\gamma_n^T \mathbf{C}_\mathrm{m}^{-1} \gamma_n - \gamma_{n-1}^T \mathbf{C}_\mathrm{m}^{-1} \gamma_n}{\gamma_{n-1}^T \mathbf{C}_\mathrm{m}^{-1} \gamma_{n-1}}, \tag{8}$$

where γ_n is the *steepest ascent* vector at iteration n.We do not employ any preconditioning.

The model vector at step $n+1$, $\mathbf{m}_{n+1}$, is obtained from $\mathbf{m}_n$ by computing

$$\mathbf{m}_{n+1} = \mathbf{m}_n - \mu_n \phi_n, \tag{9}$$

where μ_n is a scalar found by linear search.

3 Covariance convolution

3.1 Covariance kernel convolution

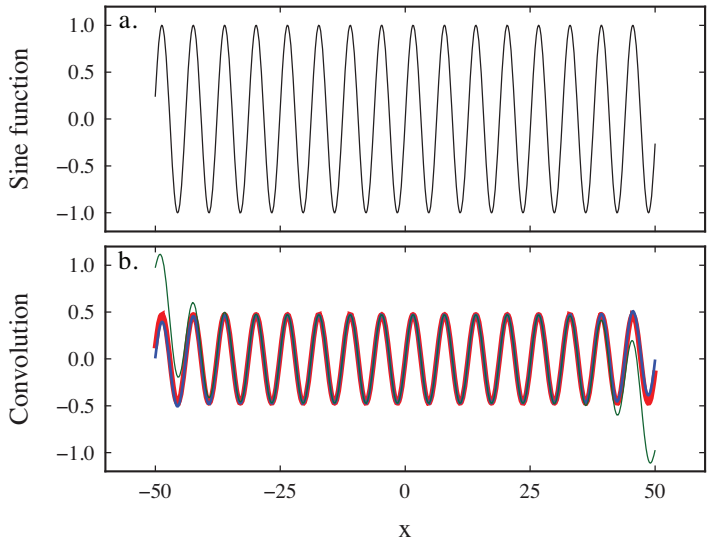

Figure 1. A simple 1D convolution – a. Sinusoidal function g to be convolved. **b.** Red line is the analytical result from equation 14. Blue line is the result of the numerical convolution described in the text. Green line is the result of the covariance matrix multiplication. Here, $\lambda = 4$ and $\sigma = 1$ on equation 10.

We show examples of the equivalence of the covariance kernel convolution with the matrix vector multiplication involved in the conjugate gradient solver. Given a function g of a single variable x and a covariance kernel K so that

$$\forall x \in \mathbb{R} \quad g(x) = \sin x \quad \text{and} \quad K(x) = \sigma^2 e^{-x/\lambda}, \tag{10}$$

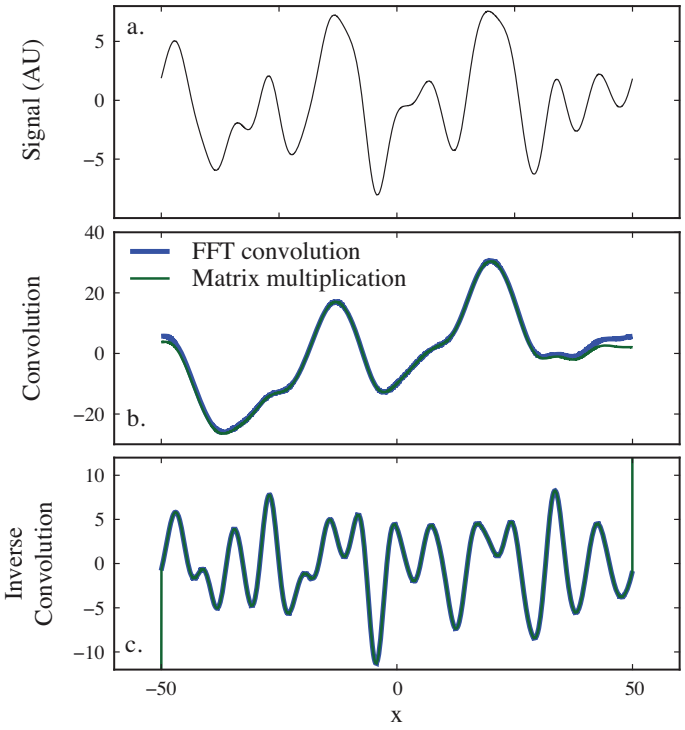

Figure 2. Convolution of a random function – **a.** Function to be convolved. This randomly generated signal is the convolution of a white noise with a Gaussian function. **b.** Results of the convolution approach (blue) and the matrix multiplication approach (green). **c.** Results of the inverse convolution approach (blue) and the multiplication with the inverse of the covariance matrix (green). Here, $\lambda = 10$ and $\sigma = 1$

where σ and λ are two positive real numbers, the Fourier transform of K and g are then given by

$$\forall u \in \mathbb{R} \quad \hat{K}(u) = \frac{2\lambda\sigma^2}{1 + \lambda^2 u^2} \tag{11}$$

and

$$\hat{g}(u) = (i\sqrt{\frac{\pi}{2}}\delta(u-1) - i\sqrt{\frac{\pi}{2}}\delta(u+1)), \tag{12}$$

where δ is the Dirac delta function. Then, the Fourier transform of the inverse function G is simply

$$\forall u \in \mathbb{R} \quad \hat{G}(u) = \frac{1}{\hat{K}}(u) = \frac{1 + \lambda^2 u^2}{2\lambda\sigma^2}. \tag{13}$$

After inverse Fourier transform of $\hat{K}\hat{g}$ and $\hat{G}\hat{g}$, one gets

$$\forall x \in \mathbb{R} \quad (K * g)(x) = \frac{2\lambda\sigma^2}{1 + \lambda^2} \sin x \quad \text{and} \quad (G * g)(x) = \frac{1 + \lambda^2}{2\lambda\sigma^2} \sin x. \tag{14}$$

We compare on Figure 1 the difference between the analytical result presented in equation 14 for the case of the convolution by K, the equivalent convolution realized numerically

and the matrix multiplication for the corresponding covariance matrix. The numerical convolution is obtained by a numerical Fourier transform of $g(x)$, the multiplication by $\hat{K}(u)$ and an inverse Fourier transform. Figure 2 shows the comparison of the convolution approach and the matrix multiplication approach for a randomly generated signal. In both cases, results are indiscernible except at the edges where the Fourier approach assumes periodicity of the signal.

3.2 Application to interferograms and to model fields

We use the covariance kernel described in equation 9 of the main text and its Fourier transform given in equation 10 of the main text. We perform the convolution using the following steps:

1. Holes in the phase or model fields, due to decorrelation of the interferometric phase, are filled using the inpainting algorithm from John D'Errico freely available on the Matlab Central File Exchange[1] implemented in GIAnT [*Hetland et al.*, 2012; *Agram et al.*, 2013]. In this case, inpainting leads to a full image, allowing us to use Fast Fourier Transform algorithms.
2. Interferograms are zero-padded to avoid edge effects with gaussian smoothing of the edges in the case of the inverse convolution operation.
3. The resulting image is transformed to the Fourier domain using FFT.
4. The result is multiplied by the appropriate covariance kernel.
5. The result is transformed back to the spatial domain.
6. Inpainted pixels are discarded.

Figure 3 illustrates this method by comparing the convolutions with the corresponding covariance matrix multiplications applied to a 70×100 pixel subset of an interferogram. Results from the convolution approach are once again comparable with those from the matrix multiplication approach, although some high-frequency issues appear in the inverse convolution approach. These are related to the proximity of the decorrelation regions and the inpainting strategy used here.

[1] http://www.mathworks.com/matlabcentral/fileexchange/4551-inpaintnans

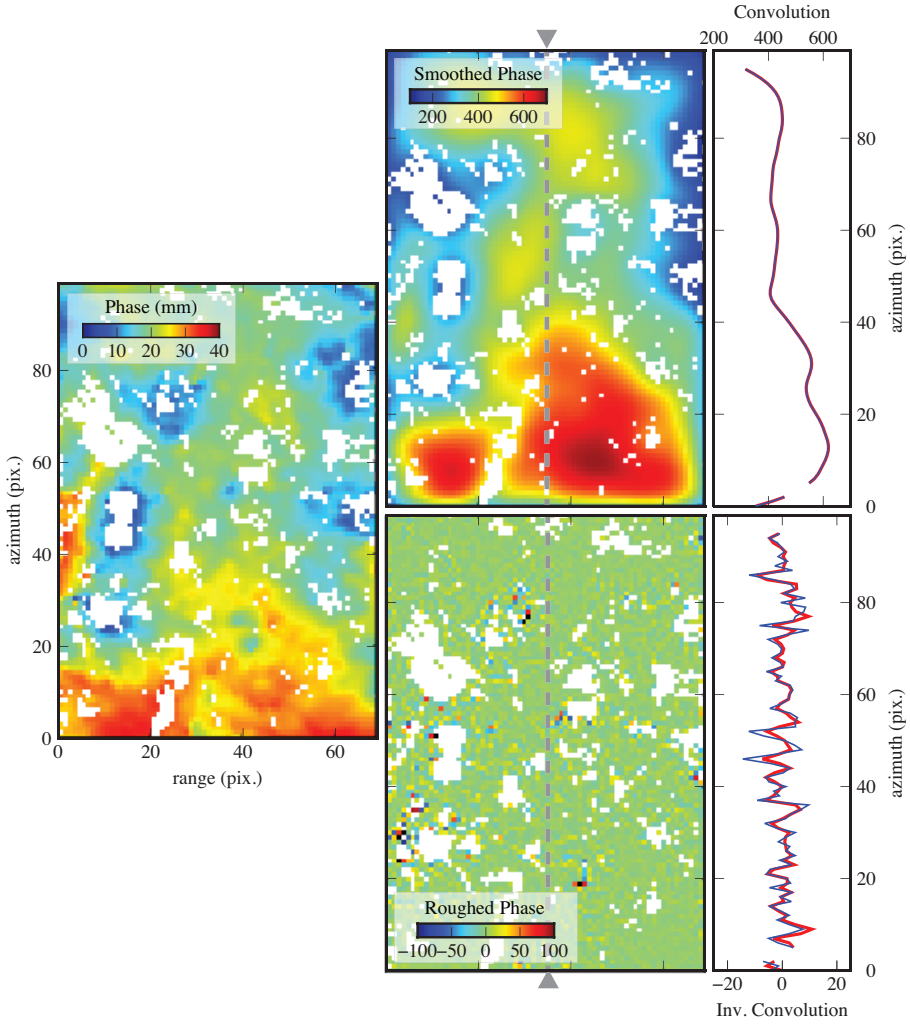

Figure 3. Application to a real interferogram – Comparison of the convolution approach and the matrix multiplication approach to a real interferogram for the case of the convolution (top right) and the inverse convolution (bottom right). Profiles shown on the right are taken along the gray dashed line and show the difference between the convolution approach (blue line) and the matrix multiplication approach (red line). Here, $\lambda = 2$ km, $\sigma = 1$ mm and the pixel size is 650 m.

3.3 Implementation in the conjugate gradient solver

In our conjugate gradient solver, we repeat the same operation each time a covariance is applied to a vector. As our vectors are distributed among multiple processes, we first gather the fields to be convolved (i.e. the phase map at each acquisition in the case of the SBAS and NSBAS approaches and the map of each model parameter in the case of the parameterized and NSBAS approaches). These fields are then sent by batches to differ-

ent processes where the convolution is performed. If the choice is to use an exponential-based covariance kernel, then the above described steps are performed. If the choice is to use a diagonal covariance for the concerned field, then the field is simply multiplied or divided by the terms of the diagonal. Convolved fields are then sent back to the appropriate processes for the following calculations.

4 Additional details about the synthetic dataset used for method validation

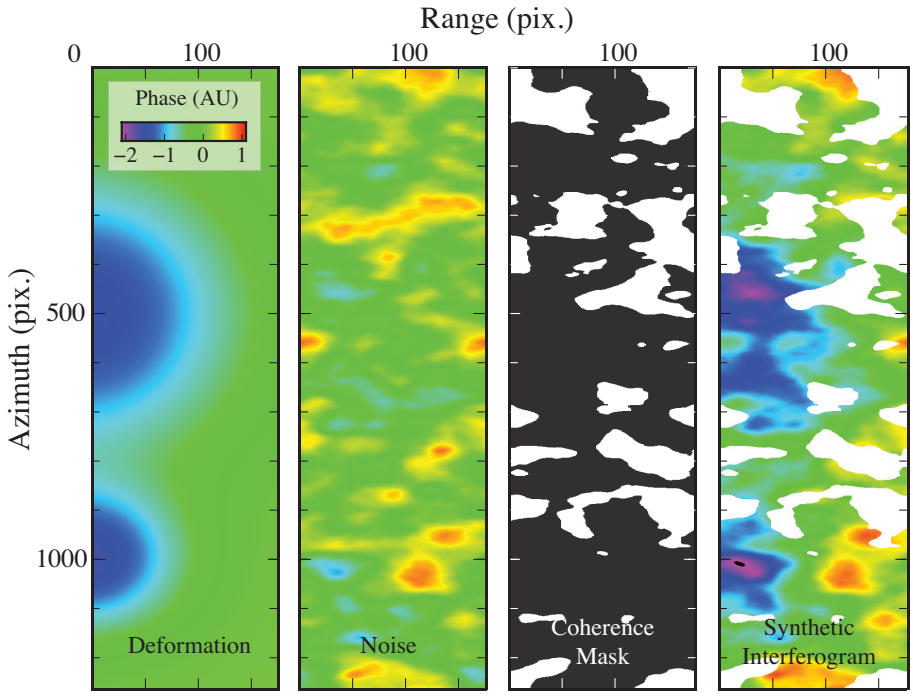

Figure 4. Example of an interferogram of the synthetic dataset used for method validation – Left panel shows the deformation signal resulting from the combination of a linear trend, a step function and a logarithmic decay. This synthetic interferogram has a temporal baseline of 280 days. The "noise" panel shows the correlated noise added to the deformation signal. This noise results from the combination of two random realizations of noise at the time of the acquisitions used to build the deformation signal. The "coherence" panel shows the mask generated randomly to simulate the effect of variable spatial coherence on the unwrapped interferometric phase. The rightmost panel shows the interferogram as used as input in the time series analysis.

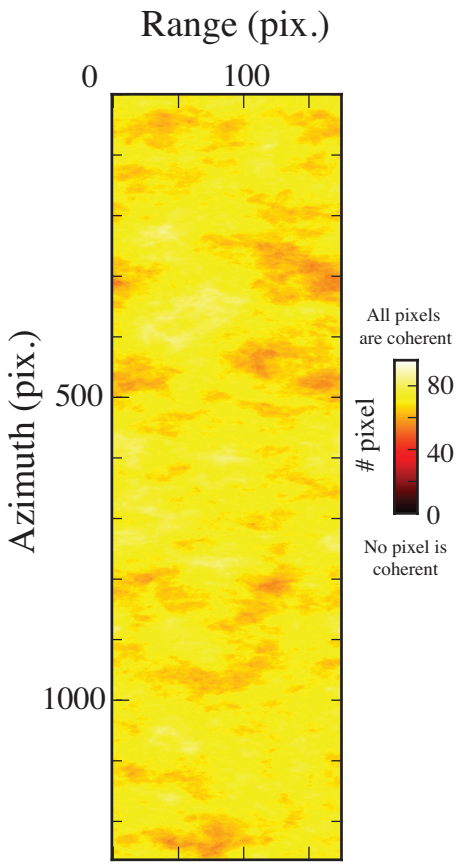

Figure 5. Number of coherent pixels in the synthetic stack of interferograms used as input for the method validation – We build 96 interferograms in total. The minimum number of available phase values for a pixel is 52.

5 Summary of the covariance chosen for the synthetic test

The conjugate gradient solver for the dictionary approach (step 1 in table 1) proceeds in 26 iterations, for a 90% reduction of the least squares norm and a 75% reduction of the L2-norm of the residuals (the difference between data and prediction). Moving to the NSBAS hybrid problem (step 2), the solver proceeds in 25 iterations to further reduce the L2-norm of the residuals by 41% for a >99% reduction of the least squares norm (i.e. 4 orders of magnitude). The complete run takes about 3 hours on 256 processes distributed over 16 nodes.

Table 1. Summary of covariances used for the synthetic test

	Data Covariance		**Model Covariance**		**Ramp Covariance**
	Phase	Function	Phase	Function	
Step 1 Dictionary	N.A.	Exponential (σ = 0.3, λ = 10)	N.A.	Exponential (σ = 1000, λ = 10)	1e-3
Step 2 NSBAS	1e-20	Exponential (σ = 0.3, λ = 10)	1e8	Exponential (σ = 10, λ = 10)	1e-3

6 Summary of the covariance chosen for northern Chile and run performances

For the dictionary approach (step 1 in table 2), the conjugate gradient solver converges in 20 iterations for a least squares norm reduction of 57% and a corresponding reduction of the L2-norm of residual of 30%. Solving the SBAS problem (step 2) reduces the least square norm by 70% in 7 iterations. Solving the NSBAS problem (step 3) further reduces the L2-norm of the residuals by 70% for a drop of 2 orders of magnitude of the least squares norm in 30 iterations. The complete run takes about 3½ hours on 256 processes distributed over 16 nodes.

Table 2. Summary of the covariance chosen for northern Chile

	Data Covariance		**Model Covariance**		**Ramp Covariance**
	Phase	Function	Phase	Function	
Step 1 Dictionary	N.A.	Exponential measured from data	N.A.	Exponential (σ = 10 m, λ = 15 km)	1e-3
Step 2 SBAS	1e-20	N.A.	Exponential (σ = 0.1 m, λ = 15 km)	N.A.	1e-3
Step 3 NSBAS	1e1	Exponential measured from data	Exponential (σ = 10 m, λ = 15 km)	Exponential (σ = 10 m, λ = 15 km)	1e-3

References

Agram, P. S., R. Jolivet, B. Riel, Y. N. Lin, M. Simons, E. Hetland, M. P. Doin, and C. Lasserre (2013), New Radar interferometric time series analysis toolbox released, *Eos*, *94*(7), 69–76, doi:10.1029/2011JB008731.

Hetland, E. A., P. Musé, M. Simons, Y. N. Lin, P. S. Agram, and C. J. DiCaprio (2012), Multiscale InSAR Time Series (MInTS) analysis of surface deformation, *J. Geophys. Res.*, *117*(B2), doi:10.1029/2011JB008731.

Polack, E., and G. Ribière (1969), Note sur la convergence de méthodes de directions conjuguées, *Revue Fr. Inf. Rech. Oper.*, *16?R1*, 35–43.

Schmidt, D. A., and R. Burgmann (2003), Time-dependent land uplift and subsidence in the Santa Clara valley, California, from a large Interferometric Synthetic Aperture Radar data set, *J. Geophys. Res.*, *108*(B9), doi:10.1029/2002JB002267.

Tarantola, A. (2005), *Inverse problem theory and methods for model parameter estimation*, SIAM.