



Efficient multi-GNSS processing based on raw observations from large global station networks

Sebastian Strasser* and Torsten Mayer-Gürr
Graz University of Technology, Austria

EGU2020: Sharing Geoscience Online, 4 May 2020

1 Introduction

The year 2020 is going to mark the first time of four global navigation satellite systems (i.e., GPS, GLONASS, Galileo, and BeiDou) in full operational capability. Utilizing the various available observation types together in global multi-GNSS processing offers new opportunities, but also poses many challenges. The raw observation approach (Strasser et al., 2019) facilitates the incorporation of any undifferenced and uncombined code and phase observation on any frequency into a combined least squares adjustment. Due to the increased number of observation equations and unknown parameters, using raw observations directly is more computationally demanding than using, for example, ionosphere-free double-differenced observations. This is especially relevant for our contribution to the third reprocessing campaign of the International GNSS Service (IGS; Johnston et al., 2017), where we process observations from up to 800 stations per day to three GNSS constellations at a 30-second sampling. For a single day, this results in more than 200 million raw observations, from which we estimate almost 5 million parameters.

Processing such a large number of raw observations together is computationally challenging and requires a highly optimized processing chain. In this contribution, we detail some key steps that make such a processing feasible in the context of a distributed computing environment (i.e., large computer clusters). Section 2 describes the setup of observation equations by eliminating slant total electron content (STEC) parameters to significantly reduce the number of parameters in the normal equation system. A suitable structure of the normal equation system that reduces computational and memory load is illustrated in Section 3.

2 Elimination of STEC at observation level

This is a short description of how to eliminate the STEC parameters at the observation level during GNSS processing with the raw observation approach. The procedure is mathematically identical to just estimating the STEC parameters, but is used to significantly reduce the number of parameters in the normal equation system and therefore reduce computational and memory load.

We split up the design matrix by using the chain rule and by introducing intermediate parameters \mathbf{y} . This simplifies STEC elimination and has benefits in the software implementation. We can

*Author contact: [sebastian.strasser\[at\]tugraz.at](mailto:sebastian.strasser@tugraz.at)

write the partial derivatives of the observations \mathbf{l} with respect to the parameters \mathbf{x} (e.g., orbit parameters) as

$$\frac{\partial \mathbf{l}}{\partial \mathbf{x}} = \frac{\partial \mathbf{l}}{\partial \mathbf{y}} \cdot \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{B} \cdot \mathbf{A} . \quad (1)$$

\mathbf{A} contains the partial derivatives of the parameters in \mathbf{y} with respect to, for example, the orbit parameters. We use variational equations for this, for details see Montenbruck and Gill (2000), among others. The vector \mathbf{y} contains parameters for the position, clock error, the observations themselves, and STEC. This can be done for any number of observation types, but in case of dual-frequency code and phase observations, it reads as

$$\mathbf{y} = (x, y, z, \delta t, \phi_1, \phi_2, R_1, R_2, STEC)^T . \quad (2)$$

\mathbf{y} only contains the position parameters once, since the partial derivatives for transmitter and receiver position only differ by the sign, e.g.

$$\frac{\partial \phi_1}{\partial x} = \frac{\partial}{\partial x} \sqrt{(x_r - x^s)^2 + (y_r - y^s)^2 + (z_r - z^s)^2} = \pm \frac{x_r - x^s}{\sqrt{(x_r - x^s)^2 + (y_r - y^s)^2 + (z_r - z^s)^2}} . \quad (3)$$

So we access the same elements in both cases and adjust the sign. The same is done for the clock errors. Using a simple example for one epoch with code and phase observations on two frequencies, we get

$$\mathbf{l} = \begin{pmatrix} \phi_1 \\ \phi_2 \\ R_1 \\ R_2 \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} \frac{x_r - x^s}{\sqrt{\dots}} & \frac{y_r - y^s}{\sqrt{\dots}} & \frac{z_r - z^s}{\sqrt{\dots}} & 1 & 1 & \frac{-40.3}{f_1^2} \\ \vdots & \vdots & \vdots & 1 & 1 & \frac{-40.3}{f_2^2} \\ \vdots & \vdots & \vdots & 1 & & \frac{40.3}{f_1^2} \\ \vdots & \vdots & \vdots & 1 & & \frac{40.3}{f_2^2} \end{bmatrix} . \quad (4)$$

Pseudo-parameters for the observations are added to \mathbf{y} so we still have access to them after the STEC parameter is eliminated. In \mathbf{B} this results in an identity matrix part, see above. Eliminating the STEC parameter removes one row from \mathbf{B} and the remaining identity matrix elements then contain the resulting linear combinations of the other rows. This is necessary so we can set up bias and ambiguity parameters for all individual raw observation types later.

We then apply parameter elimination at the observation level to \mathbf{B} and \mathbf{l} . \mathbf{y} and \mathbf{B} can be written as

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \quad \text{and} \quad \mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2] , \quad (5)$$

with \mathbf{y}_2 containing only the STEC parameter and \mathbf{B}_2 being the STEC column of \mathbf{B} .

QR decomposing \mathbf{B}_2 as

$$\mathbf{B}_2 = \mathbf{Q}\mathbf{R} = [\mathbf{Q}_1, \mathbf{Q}_2] \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \quad (6)$$

using very fast LAPACK (Anderson et al., 1999) routines allows us to transform/rotate \mathbf{l} and \mathbf{B}_1 to eliminate the STEC parameter:

$$\mathbf{l}' = \mathbf{Q}_2^T \mathbf{l} \quad (7)$$

$$\mathbf{B}' = \mathbf{Q}_2^T \mathbf{B}_1 \quad (8)$$

\mathbf{l}' and \mathbf{B}' are then used for setting up the normal equations and right hand side. Both of them have one less row than \mathbf{l} and \mathbf{B} , respectively.

3 Structure of the normal equation system

An efficient normal equation structure is integral when dealing with millions of parameters in large-scale GNSS processing. It is important to order the parameters in a way that maximizes the sparsity of the matrix by grouping nonzero elements to highly dense blocks. Since only nonzero blocks have to be kept in memory, this massively reduces the memory load. It also significantly speeds up the processing due to the reduced number of unnecessary operations that involve zeros.

Figure 1 illustrates the normal equation structure currently implemented in our GNSS software in a schematic way. We utilize a kite structure (Boxhammer, 2006) consisting of three major parts: a block-diagonal part containing one block per epoch (green), a part containing nonepoch receiver and satellite parameters (red), and an ambiguity part (blue). Each epoch block contains all receiver and satellite clock parameters for this epoch. The epoch blocks are assumed to be uncorrelated to each other, although it is possible to consider correlations between epochs by adding subdiagonal blocks. The epoch part is correlated to both the nonepoch part and the ambiguity part, shown as chequered areas in Figure 1. These correlation parts constitute the majority of the required memory.

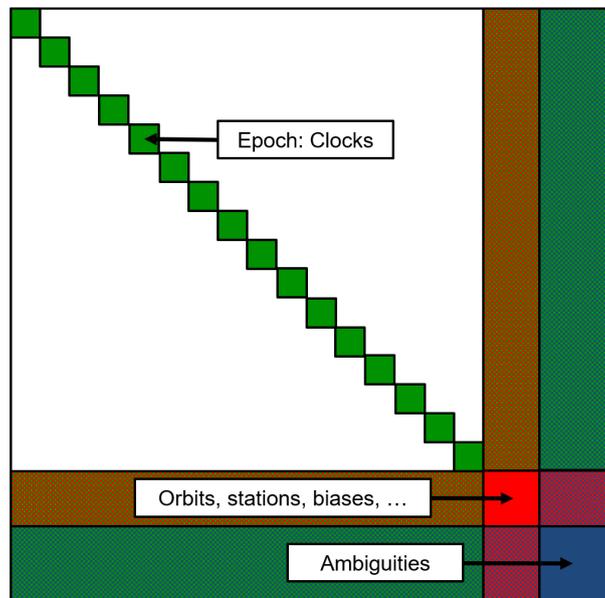


Figure 1: Schematic view of normal equation structure (not to scale)

Figure 2 shows the structure of the design matrix that leads to the normal equation structure illustrated in Figure 1. The high sparsity of the design matrix can again be utilized to reduce

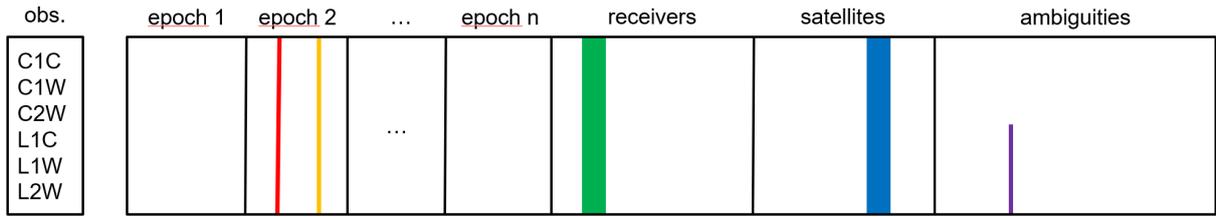


Figure 2: Schematic view of the design matrix structure for a set of observations between one receiver and one satellite at one epoch (not to scale). Parameters in the epoch block are the receiver clock error (red) and satellite clock error (orange). Receiver parameters (green) comprise station position, troposphere, and signal biases. Satellite parameters (blue) include initial state, solar radiation pressure, stochastic pulses, and signal biases. Phase observations additionally have ambiguity parameters (purple).

memory and computational load. This is achieved by grouping nonzero columns into blocks and only storing these blocks in memory. Since the computational efficiency of LAPACK routines suffers when blocks get too small, some blocks might not contain strictly nonzero columns. Therefore, a balance between computational and memory load has to be found, which may vary depending on hardware and software architecture.

The actual normal equation structure when processing a small network of 50 stations together with the GPS constellation at an observation sampling of 3 minutes is visualized in Figure 3. Compared to the schematic view in Figure 1, it becomes obvious that the memory requirements of the epoch blocks are negligible. It is also clearly visible that the correlation part between epochs and ambiguities is not fully populated. Ordering the ambiguity parameters based on their time span results in a roughly banded matrix. Again, this benefits memory load as it allows to store dense blocks of nonzero elements in memory, which constitute only roughly 25% of the overall epoch-ambiguity correlation part.

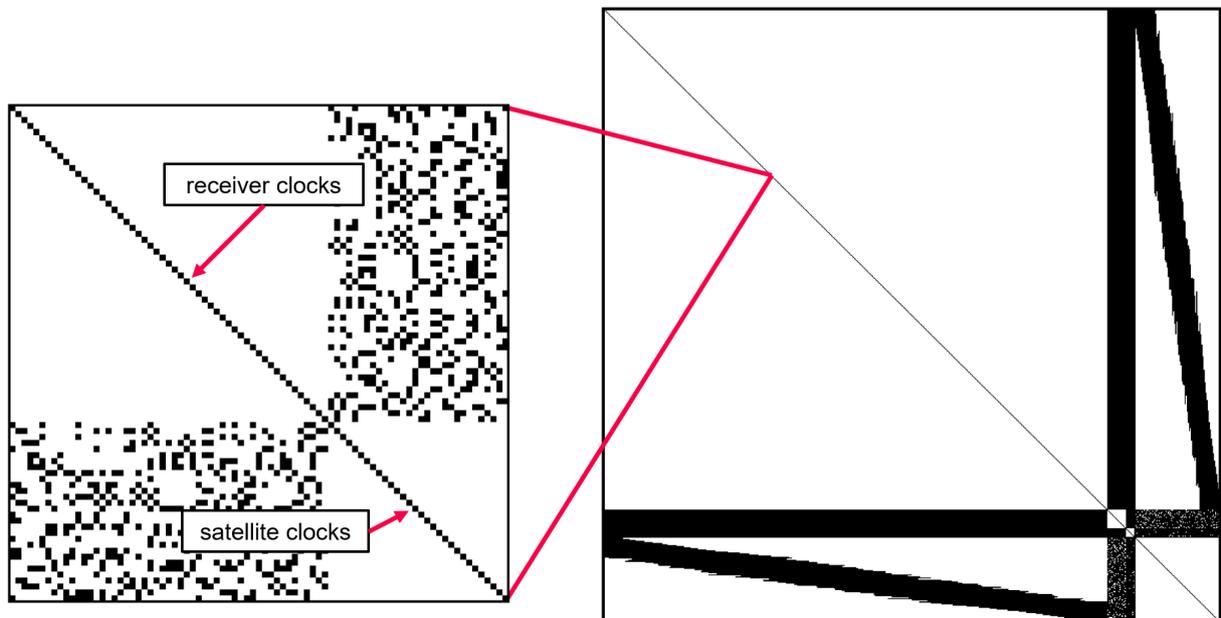


Figure 3: Actual (to scale) structure of the normal equation system for a small network of 50 stations observing the GPS constellation (right). Detailed view of a single epoch block (left). White parts are zero/unfilled, black parts are nonzero

Figure 3 further shows a close-up of one of the epoch blocks, revealing their actual structure.

Ordering the clock parameters by receivers and then satellites results in two diagonal matrices and the respective correlation blocks. Depending on the number of receivers and satellites, it can be beneficial to further split each epoch block into subblocks. While this may not save much (or any) memory in the epoch block itself, it becomes very relevant in the epoch-nonepoch correlation part. Contrary to how it appears in Figure 3 (due to limited image resolution), this part is not fully populated. As this part constitutes the majority of all required memory, splitting it into small subblocks that can be discarded in case they are not populated has a large impact on overall memory load. Reducing memory load (by splitting blocks into subblocks) without inflating the computational burden (due to too many small blocks) is a balancing act.

Another benefit of this normal equation structure is that only the nonepoch and ambiguity parts become fully populated when the normal equations are solved using the Cholesky decomposition.

While the memory load increases linearly with the observation/clock sampling, it increases exponentially with the number of receivers and satellites. Our normal equation structure enables us to process observations from up to 800 stations per day to three GNSS constellations at a 30-second sampling for our contribution to the third IGS reprocessing campaign, which is currently ongoing. The resulting normal equations contain up to 5 million parameters and require up to several hundreds of gigabytes of memory. Without the optimizations mentioned above, GNSS processing at such a large scale would be completely infeasible with our current hardware infrastructure.

Strasser et al. (2019) provide further details on the raw observation approach and on a processing strategy feasible for large station networks.

References

- Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., & Sorensen, D. (1999). *LAPACK Users' Guide* (Third). Philadelphia, PA, Society for Industrial; Applied Mathematics.
- Boxhammer, C. (2006). *Effiziente numerische Verfahren zur sphärischen harmonischen Analyse von Satellitendaten* (Doctoral dissertation). Rheinische Friedrich-Wilhelms-Universität Bonn. <http://nbn-resolving.de/urn:nbn:de:hbz:5N-07990>
- Johnston, G., Riddell, A., & Hausler, G. (2017). The International GNSS Service (P. J. Teunissen & O. Montenbruck, Eds.). In P. J. Teunissen & O. Montenbruck (Eds.), *Springer handbook of global navigation satellite systems*. Cham, Springer International Publishing. https://doi.org/10.1007/978-3-319-42928-1_{_}33
- Montenbruck, O., & Gill, E. (2000). *Satellite Orbits*. Berlin, Heidelberg, Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-58351-3>
- Strasser, S., Mayer-Gürr, T., & Zehentner, N. (2019). Processing of GNSS constellations and ground station networks using the raw observation approach. *Journal of Geodesy*, 93(7), 1045–1057. <https://doi.org/10.1007/s00190-018-1223-2>