Thèse d'habilitation à diriger des recherches
Contribution en Signal, Image et Instrumentation pour l'Astronomie

Annexe F: Auto-calibration de phase en astronomie et calibration de réseaux GNSS pour la géodésie

Habilitation thesis, "Accreditation to supervise research"
Contribution to Signal, Image & Instrumentation in Astronomy

Appendix F: Phase self-calibration in astronomy and calibration of GNSS networks for geodesy

Jean-Louis Prieur

Version May 13, 2014

Mots clés:
Techniques: synthèse d'ouverture, imagerie par auto-calibration de phase, positionnement par satellite (GPS), réseaux GNSS globaux, positionnement ponctuel précis (PPP)
Techniques: moindres carrés entiers (ILS), point du réseau le plus proche (NLP) – positionnement géodésie – GNSS

Keywords:
Techniques: aperture synthesis · phase closure imaging · self-calibration · Global Positioning System (GPS) · geodetic positioning · GNSS · Global and regional networks · Clock biases · Uncalibrated phase delays (UPD) · Fractional clock biases (FCB) · Network real-time kinematics (NRTK) · Precise Point Positioning (PPP) · Closure difference (CD) · Nearest lattice point (NLP) · Integer least squares (ILS)

Contents

Chapter I

Introduction

Recent theoretical developments in astronomical aperture synthesis have revealed the existence of integer-ambiguity problems. Those problems, which appear in the self-calibration procedures of radio imaging, have been shown to be similar to the nearest-lattice point (NLP) problems encountered in high-precision geodetic positioning, and in global navigation satellite systems.

In Chapter II, we analyse the phase-calibration problem encountered in astronomy when mapping incoherent sources with aperture-synthesis devices. More precisely, this analysis concerns the phase-calibration operation involved in the self-calibration procedures of phase-closure imaging. In this chapter, we revisit and complete the analysis presented in Lannes (2005). We also take profit of subsequent developments made for solving similar problems encountered in global navigation satellite systems. In radio-astronomy, the related optimization problems have been stated and solved hitherto at the phasor level. We present here an analysis conducted at the phase level, from which we derive a method for diagnosing and solving the difficulties of the phasor approach. In the most general case, the techniques to be implemented appeal to the algebraic graph theory and the algebraic number theory. The minima of the objective functionals to be minimized are identified by raising phase-closure integer ambiguities. We also show that in some configurations, to benefit from all the available information, closure phases of order greater than three are to be introduced. In summary, this study leads to a better understanding of the difficulties related to the very principle of phase-closure imaging. To circumvent these difficulties, we propose a strategy both simple and robust.

In Chapter III, we analyse the theoretical aspects the NLP problem encountered both in this phase self-calibration problem and in the calibration of global navigation satellite systems, and we propose new resolution methods. The related optimization aspects concern both the preconditioning stage, and the discrete-search stage in which the integer ambiguities are finally fixed. Our algorithms, which are described in an explicit manner, can easily be implemented. They lead to substantial gains in the processing time of both stages. Their efficiency was shown via intensive numerical tests.

In global navigation satellite systems (GNSS), the calibration problem, consisting of retrieving clock-phase biases from network data, has a basic rank defect. In Chapter IV we analyze the different ways of removing this rank defect, and define an efficient strategy for obtaining these phase biases in a standard form. The minimum-constrained problem to be solved in the least-squares (LS) sense depends on some integer vector which can be fixed in

an arbitrary manner. We propose to solve the problem via an undifferenced approach based on the notion of closure ambiguity. We present a theoretical justification of this closure-ambiguity approach (CAA), and the main elements for a practical implementation. The links with other methods are also established. We analyze all those methods in a unified theoretical framework, and derive functional relations between the corresponding solutions and our CAA solution. On the grounds of this analysis, we make some suggestions for improving the calibration procedures currently used in GNSS. A substantial reduction of the processing time could thus be obtained. This would be particularly interesting for many GNSS applications like real-time kinematic precise point positioning for instance. To compare the methods providing LS estimates of clock-phase biases, we define a particular solution playing the role of reference solution. For this solution, when a phase bias is estimated for the first time, its fractional part is confined to the one-cycle width interval centred on zero; the integer-ambiguity function is modified accordingly. Our theoretical study is illustrated with some simple and generic examples; it could have applications in data processing of most GNSS networks, and particularly global networks using GPS, GLONASS, Galileo, IRNSS or BeiDou/Compass satellites.

I.1 Lannes' approach of the problem

When processing times series of global positioning data, one is led to introduce ***local variables*** u_k which depend on the successive epochs t_k of the time series, and a ***global variable*** v which remains the same all over these epochs, with however possible *state transitions* from time to time. For example, state transitions occur when some satellites appear or disappear.

In the period defined by two successive transitions, the problem to be solved in the least-square (LS) sense is governed by a system of linear equations in which the key matrix has an angular block structure. This structure is well suited to ***recursive QR factorisation***, which was implemented for the first time in GNSS by Lannes & Gratton (2008) for solving the case of continuous observations in RTK mode (Real-Time Kinematics) with a local-scale single baseline. This technique was then applied to the more general case of GNSS networks by Lannes & Gratton (2009). In that paper, they introduce the notion of closure delay (CD) which generalizes that of double difference (DD) which is widely used in GNSS. The recursive QR factorisation prove to be well suited for handling the state transitions induced by the variations of the GNSS graph in an optimal manner. Furthermore, Lannes & Gratton (2009) also implemented LLL decorrelation techniques that appeared very efficient for facilitating the integer-ambiguity problem resolution.

Notice that the satellite-clock biases were not estimated in Lannes & Gratton (2009). The estimation of all the clock-biases of a GNSS network, which can be considered as the ***calibration*** of this network, was investigated in Lannes & Prieur (2013).

I.2 Observational equations

Global positioning techniques are based on carrier-phase and code measurements, $\Phi_{\nu,k}(i,j)$ and $P_{\nu,k}(i,j)$, obtained for the receiver-satellite pair $(i,j) \overset{\text{def}}{=} (r_i, s_j)$ (r_i receiver, s_j, satellite), at the time t_k and for the frequency ν. There are two frequencies $\nu = \nu_1, \nu_2$ transmitted by each GPS/NAVSTAR satellite of the former generation, and at least three for the latest generation satellites. To simplify, we use here *satellite* as a synonym of *transmitter*, which is a valid assumption when building the visibility grids and graphs. We will split the indices of the satellites into two (or more) sub-indices of *transmitters* when considering *receiver-transmitter* accurate distances, since the satellite antennas (or feeders) of the different frequencies are not located at the same place.

For each epoch t_k, the carrier-phase and code data are respectively of the form (Teunissen & Kleusberg, 1998; Mercier & Laurichesse, 2008):

$$\left\{\begin{array}{rcl} \Phi_{\nu,k}(i,j) & = & \rho_k(i,j) + T_k(i,j) - \kappa_\nu I_{k;\nu_1}(i,j) + \left[f^{(r)}_{\Phi;k}(i) - f^{(s)}_{\Phi;k}(j)\right] \\ & & -\kappa_\nu \left[g^{(r)}_{\Phi;k}(i) - g^{(s)}_{\Phi;k}(j)\right] + \lambda_\nu\, N_\nu(i,j) + \varepsilon_{\Phi;\nu,k}(i,j) \\ P_{\nu,k}(i,j) & = & \rho_k(i,j) + T_k(i,j) + \kappa_\nu I_{k;\nu_1}(i,j) + \left[f^{(r)}_{P;k}(i) - f^{(s)}_{P;k}(j)\right] \\ & & +\kappa_\nu \left[g^{(r)}_{P;k}(i) - g^{(s)}_{P;k}(j)\right] + \epsilon_{P;\nu,k}(i,j) \end{array}\right.$$

In these equations, that are expressed in length units, $\Phi_{\nu,k}(i,j)$ is carrier-phase measurement, $\rho_k(i,j)$ is the receiver-satellite range (distance between the satellite s_j at the time $t_k - \tau$

when the signal was emitted and the receiver r_i at the time t_k of its reception); $T_k(i,j)$ and $\kappa_\nu\, I_{k;\nu_1}(i,j)$ are the tropospheric and ionospheric delays, respectively. The ionospheric delay is proportional to ν^{-2}, so:

$$\kappa_\nu = \nu_1^2/\nu^2 = \lambda_\nu^2/\lambda_1^2 \quad \text{for} \quad \nu = \nu_1, \nu_2 \tag{I.1}$$

where λ_ν is the wavelength of the carrier wave. Note that $\kappa_{\nu_1} = 1$.

The instrumental delays and clocks errors that depend only on r_i (for a given epoch t_k) are lumped together in the *extended receiver-clock biases* $f^{(r)}_{\Phi;k}(i)$, $f^{(r)}_{P;k}(j)$. Likewise, the instrumental delays and clocks errors that depend only on s_j (for a given epoch t_k) are lumped together in the *extended satellite-clock biases* $f^{(s)}_{\Phi;k}(i)$, $f^{(s)}_{P;k}(j)$.

Similarly, $g^{(s)}_{\Phi;k}(i)$, $g^{(s)}_{P;k}(j)$ denote the biases induced by the time group delays.

$N_\nu(i,j)$ are the integer carrier-phase ambiguities. They are constant when the electronics of the phase loop is locked (Phase-Loop Locked, PLL).

In this model the expectation values of the noise terms $\varepsilon_{\Phi;\nu,k}(i,j)$ and $\varepsilon_{P;\nu,k}(i,j)$ are supposed to be nought. (i.e. $\langle\varepsilon_{\Phi;\nu,k}(i,j)\rangle \approx 0$ and $\langle\varepsilon_{P;\nu,k}(i,j)\rangle \approx 0$).

Chapter II

Self-Calibration in Astronomy

The phase-calibration problems encountered in aperture synthesis may have various forms that are linked to the physical context of the experiments. In astronomy, it mainly concerns radio imaging and optical interferometry. There exists a wide range of situations: various wavelength domains, snapshot or integrations over some time interval, different nature of the calibrator (point-source, extended source), etc. The way of modelling the influence of the atmosphere may also change the nature of the problem. Self-calibration corresponds to the situation where the object source to be imaged plays the role of the calibrator. As shown in the pioneering work of Cornwell & Wilkinson (1981), this problem can often be solved by alternate phase-calibration operations and Fourier-synthesis processes. In most cases, to ensure the reliability and the robustness of those techniques, the phase-calibration operations must then be conducted with much care. The main objective of this chapter is to develop the corresponding analysis, and thereby to derive a method for coping with critical situations.

Related contributions have been made in this field by our group; see, e.g., the study of the phase-calibration problem presented in Lannes (2005). Furthermore, similar calibration problems appear in the processing of network signals in global navigation satellite systems (GNSS); see Lannes & Teunissen (2011). In this chapter the GNSS contributions of Lannes et al. (2010) and Lannes & Gratton (2009) are also taken into account. The original study of the phase-calibration problem presented in Lannes (2005) is thus revisited and completed.

In this chapter, we first recall some preliminary notions of aperture synthesis, and introduce two approaches (Sect. II.1). In the first one, the 'chord approach,' one minimizes the size of the chords defined by the phasor pairs involved in the functional to be minimized. The problem is thus handled at the phasor level in a way similar to that of the traditional approach of Cornwell & Wilkinson (1981). The second approach, the 'arc approach,' corresponds to that of Lannes (2005). The problem is handled at the phase level. The quantity that is then minimized is the size of the arcs associated with the chords of the first approach.

The arc approach gives a better insight into the problem; see Sects. II.4 and II.5. The techniques to be implemented then appeal to the algebraic graph theory; see, e.g., Biggs (1996). The corresponding theoretical framework is presented in Sect. II.2. The notion of 'phase closure' is then introduced in a context more general than that usually defined in radio imaging and in optical interferometry. In particular, closure phases of order greater than three may then be defined. The interest of the related approach is specified in Sect. II.2.6. Particular attention is paid to the related algebraic structures; see Property 1 in Sect. II.2.3 together with the algebraic aspects presented in Sect A. The phase-calibration problem can be stated in a

reduced form in which the notion of phase closure plays an important role. This key point is presented in Sect. II.3.

Section II.4 is devoted to the study of the minima of the 'reduced arc functional' g. Some notions of algebraic number theory are then taken into account (Cohen, 1996). We then show that the minimizers of g can be identified in a systematic manner. The connection with the usual approach via the 'reduced chord functional' f is presented in Sect. II.5. The minimizers of g can be used for initializing the minimization processes of f. As shown in Sect. II.5.3, a minimum of f may be 'tightly linked' to a minimum of g. To illustrate our method in a simple and concrete manner, a few simulations concerning four-element arrays are presented in Sect. II.6.

In Sect. II.7, we first show how our approach can be included in the self-calibration procedures of phase-closure imaging (Sect. II.7.1). On the grounds of our algebraic analysis, we then propose a simple strategy for conducting the phase-calibration operations in a robust manner (Sect. II.7.2). As a general rule, such an operation then reduces to solving a linear equation.

In the concluding comments (Sect. II.8), we will recall the main points of our method (arc and chord approaches, phase-closure aspects, nonlinear situations, conflicting secondary minima, etc, see Sect. II.8.1), and then discuss the impact of our study on the other methods of phase-closure imaging (see also Sect. II.8.2).

II.1 Preliminary notions

Let us consider an interferometric array observing an incoherent source of small angular size. Relative to the tracking centre, the object source is characterized by some two-dimensional angular brightness distribution $s_{\mathrm{o}}(\boldsymbol{\xi})$. Denoting by $\mathcal{F}$ the Fourier-transform operator, we have

$$(\mathcal{F}s_{\mathrm{o}})(\boldsymbol{u}) \stackrel{\mathrm{def}}{=} \int s_{\mathrm{o}}(\boldsymbol{\xi})\, e^{-2\mathrm{i}\pi \boldsymbol{u}\cdot\boldsymbol{\xi}}\, d\boldsymbol{\xi} \tag{II.1}$$

Let $\boldsymbol{r}(i)$ denote, at some epoch, the position vector of the ith pupil element (antenna or telescope) projected onto a plane normal to the tracking axis. Each pair of these pupil elements defines a baseline (i, j). The angular spatial frequency associated with this baseline is given by the formula

$$\boldsymbol{u}(i,j) = [\boldsymbol{r}(i) - \boldsymbol{r}(j)]/\lambda \tag{II.2}$$

where λ is the wavelength of the electromagnetic field under consideration. According to the theorem of Van Cittert-Zernike (Born & Wolf, 1970), we define the 'complex visibility' function of the object $V_{\mathrm{o}}(i,j)$ by the relation

$$V_{\mathrm{o}}(i,j) \stackrel{\mathrm{def}}{=} (\mathcal{F}s_{\mathrm{o}})\{\boldsymbol{u}(i,j)\} \tag{II.3}$$

The baselines of the interferometric array thereby define some discrete sampling of the Fourier transform of s_{o}. The 'complex-visibility data' function $V_{\mathrm{d}}(i,j)$ is related to this discrete sampling by a relation of the form

$$V_{\mathrm{d}}(i,j) = V_{\mathrm{o}}(i,j)\, e^{\mathrm{i}[\alpha_{\mathrm{d}}(i)-\alpha_{\mathrm{d}}(j)]} + \epsilon_{\mathrm{d}}(i,j) \tag{II.4}$$

The $\alpha_{\rm d}(i)$s are unknown pupil-bias phases; $\epsilon_{\rm d}(i,j)$ is an error term. Once these aberration phases have been calibrated (somehow), the inversion of the basic relationship (II.3) yields an estimate of $s_{\rm o}$. Within well defined limits, the corresponding operation is associated with the notion of Fourier synthesis (Lannes et al., 1997). Here, for clarity, we restrict ourselves to the snapshot case: a single epoch is considered; see however Remark II.3.2.1.

Whenever possible, the pupil-bias phases $\alpha_{\rm d}(i)$ are calibrated by using a reference object source: the 'phase calibrator.' This reference is *a priori* known, for example when it is a point source. There however exist circumstances where such a calibration is impossible. A way out may be to use the very object (source) to be imaged as calibrator. One then speaks of 'self-calibration.' More precisely, an estimate $s_{\rm m}$ of $s_{\rm o}$ is then used as calibrator: the 'model.' The complex visibilities of this model are then denoted by $V_{\rm m}(i,j)$ By performing alternate (self-)phase-calibration operations and Fourier-synthesis processes, the model can thus be progressively refined; see, e.g., Cornwell & Wilkinson (1981). In the most general case, the phase-calibration operation is therefore performed on the grounds of a relationship of the form [see Eq. (II.4)]

$$V_{\rm d}(i,j) = V_{\rm m}(i,j)\, e^{{\rm i}[\alpha_{\rm d}(i)-\alpha_{\rm d}(j)]} + \epsilon_{\rm dm}(i,j) \tag{II.5}$$

Here, the error term $\epsilon_{\rm dm}$ also takes into account the fact that $V_{\rm m}$ is an approximation to $V_{\rm o}$. Note that the pupil-bias phases appear in such equations only as differences $\alpha_{\rm d}(i)-\alpha_{\rm d}(j)$. We can therefore take the value at $i=1$ as reference. This amounts to introducing the pupil-bias phases $\alpha_d(i)$s

$$\alpha_d(i) \stackrel{\rm def}{=} \alpha_{\rm d}(i) - \alpha_{\rm d}(1) \tag{II.6}$$

The guiding idea of the phase-calibration operation is to find estimates of the pupil-bias phases $\alpha_d(i)$ modulo 2π; clearly, $\alpha_d(1)=0$. Denoting by $\alpha_{d\star}(i)$ such estimates, the 'calibrated visibility function' $V_{\rm d\star}(i,j)$ is then defined by the relation [see Eqs. (II.4) and (II.5)]

$$V_{\rm d\star}(i,j) \stackrel{\rm def}{=} V_{\rm d}(i,j)\, e^{-{\rm i}[\alpha_{d\star}(i)-\alpha_{d\star}(j)]} \tag{II.7}$$

According to Eqs. (II.1) and (II.2), a (small) angular shift $\boldsymbol{\delta\xi}$ of the source distribution induces at the level of the phase of $V_{\rm o}(i,j)$ a phase variation of the form $\alpha(i)-\alpha(j)$ where

$$\alpha(i) = -\frac{2\pi}{\lambda}\big(\boldsymbol{r}(i)\cdot\boldsymbol{\delta\xi}\big)$$

These terms cannot be distinguished from the unknown pupil-bias phases $\alpha_{\rm d}(i)$. As a result, the image provided by a self-calibration procedure can only be defined up to a translation.

II.1.1 Phase-calibration graph

Let $\mathscr{E}$ be the set of baselines (i,j), with $i<j$, on which the phase of $V_{\rm d}$ is well defined, i.e., the set of baselines on which the amplitude (i.e., the modulus) of $V_{\rm d}$ is above some threshold; see Eqs. (II.4) and (II.3). The set of pupil elements involved in the definition of the baselines of $\mathscr{E}$ is then denoted by $\mathscr{V}$. The 'phase-calibration graph' is thus characterized by the couple $\mathscr{G} \stackrel{\rm def}{=} (\mathscr{V}, \mathscr{E})$; $\mathscr{V}$ is the set of its vertices (its antennas or telescopes), and $\mathscr{E}$ that of its edges (its baselines); see for example Fig. II.1.

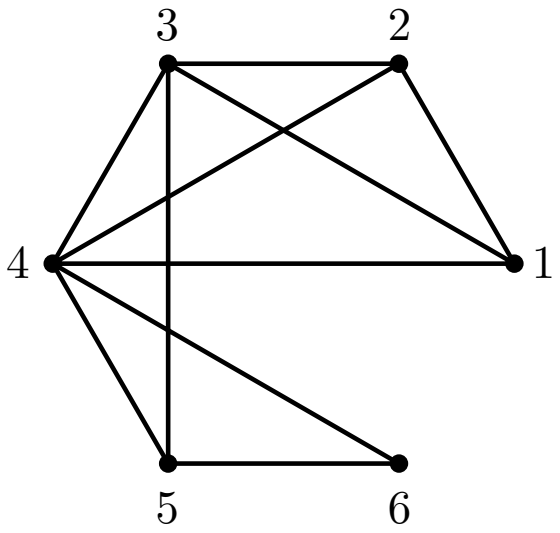

Figure II.1: Notion of phase-calibration graph. On the edges of this graph $\mathscr{G} \overset{\text{def}}{=} (\mathscr{V}, \mathscr{E})$, the modulus of the complex visibility $V_{\rm d}$ is above some threshold. In this example, $\mathscr{V}$ includes 6 vertices (6 antennas or telescopes): $n_{\rm v} = 6$; $\mathscr{E}$ includes 10 edges (10 baselines): $n_{\rm e} = 10$.

For clarity, we now assume that $\mathscr{G}$ is connected (see, e.g., Biggs (1996)): given any two vertices of $\mathscr{V}$, i_1 and i_2, there exists a path of edges of $\mathscr{E}$ connecting i_1 to i_2; see Fig. II.1.

Let us finally note that $\mathscr{G}$ is not necessarily complete: two vertices of $\mathscr{V}$ do not necessarily define an edge of $\mathscr{E}$. Denoting by $n_{\rm v}$ the number vertices of $\mathscr{G}$, and by $n_{\rm e}$ that of its edges, we therefore have $n_{\rm e} \leq n_{\rm v}(n_{\rm v} - 1)/2$.

II.1.2 Phase-calibration functionals

We now present two different ways of stating the phase-calibration problem. The guiding idea is to minimize a functional $f_\circ(\alpha_d)$ of the form [see Eqs. (II.5) and (II.7)]

$$\begin{aligned} \mathfrak{f}_\circ(\alpha_d) \overset{\text{def}}{=} & \sum_{(i,j)\in\mathscr{E}} w_\circ(i,j) \\ & \times \left| V_{\rm d}(i,j)\, e^{-{\rm i}[\alpha_d(i)-\alpha_d(j)]} - V_{\rm m}(i,j) \right|^2 \end{aligned} \tag{II.8}$$

Here, $w_\circ$ is a weight function strictly positive. When $V_{\rm m}(i,j)$ is supposed to be a good approximation to $V_\circ(i,j)$, $w_\circ(i,j)$ is of course set equal to a relatively high value.

Let us define the baseline functions $\varphi_{\rm d}$ and $\varphi_{\rm m}$:

$$V_{\rm d} = \rho_{\rm d}\, e^{{\rm i}\varphi_{\rm d}}, \qquad V_{\rm m} = \rho_{\rm m}\, e^{{\rm i}\varphi_{\rm m}} \tag{II.9}$$

we have, from Eq. (II.8),

$$\begin{aligned} & \mathfrak{f}_\circ(\alpha_d) = \sum_{(i,j)\in\mathscr{E}} w_\circ(i,j) \\ & \times \left| \rho_{\rm d}(i,j)\, e^{{\rm i}\{[\varphi_{\rm d}(i,j)-\varphi_{\rm m}(i,j)]-[\alpha_d(i)-\alpha_d(j)]\}} - \rho_{\rm m}(i,j) \right|^2 \end{aligned}$$

For clarity, let us define the baseline phase function φ as:

$$\varphi(i,j) \overset{\text{def}}{=} \varphi_{\rm d}(i,j) - \varphi_{\rm m}(i,j) \qquad \text{(modulo } 2\pi) \tag{II.10}$$

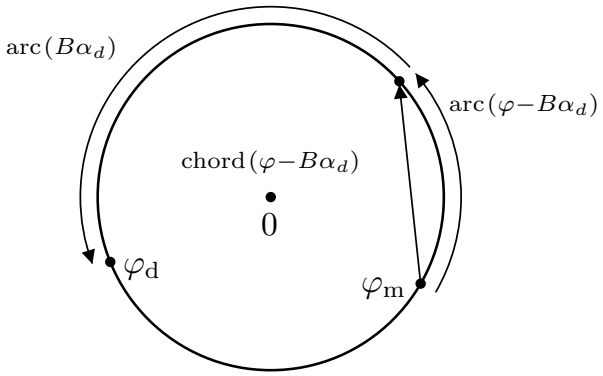

Figure II.2: Illustration of the phase-calibration principle for a generic baseline (i,j) of $\mathscr{E}$. By definition $\varphi(i,j) = \varphi_{\rm d}(i,j) - \varphi_{\rm m}(i,j)$ where $\varphi_{\rm d}$ and $\varphi_{\rm m}$ are the phases of the data and model visibilities, respectively; see Eqs. (II.10) and (II.9). The phase-calibration operation consists in minimizing in α_d the size of $\mathrm{arc}(\varphi - B\alpha_d)$ on $\mathscr{E}$; see Eqs. (II.19), (II.14), (II.15), and Fig. II.1 In the phasor approach, the functional to be minimized is a measure of the size of $\mathrm{chord}(\varphi - B\alpha_d)$ on $\mathscr{E}$; see Eq. (II.18). As illustrated here, on each baseline of $\mathscr{E}$, $\mathrm{chord}(\varphi - B\alpha_d)$ is less than or equal to $|\mathrm{arc}(\varphi - B\alpha_d)|$; see Eq. (II.21).

Let us now define the norm $\|\cdot\|_{w_d}$ as follows:

$$\|\beta\|_{w_\circ}^2 \stackrel{\rm def}{=} \sum_{(i,j)\in\mathscr{E}} w_\circ(i,j)\,|\beta(i,j)|^2$$

In functional terms, we therefore have

$$\mathfrak{f}_\circ(\alpha_d) = \left\|\rho_{\rm d}\, e^{{\rm i}(\varphi - B\alpha_d)} - \rho_{\rm m}\right\|_{w_\circ}^2 \tag{II.11}$$

where B is the baseline-bias phase operator

$$(B\alpha_d)(i,j) \stackrel{\rm def}{=} \alpha_d(i) - \alpha_d(j) \qquad [\alpha_d(1) = 0] \tag{II.12}$$

To analyze the problem on trigonometric circles (see Fig. II.2), we now introduce the variant of $\mathfrak{f}_\circ$,

$$f_\circ(\alpha_d) \stackrel{\rm def}{=} \|e^{{\rm i}(\varphi - B\alpha_d)} - 1_{\mathscr{E}}\|_w^2 \tag{II.13}$$

where $1_{\mathscr{E}}$ is the following function: $1_{\mathscr{E}}(i,j) = 1$ on $\mathscr{E}$. The norm $\|\cdot\|_w$ is defined by the relation

$$\|\beta\|_w^2 \stackrel{\rm def}{=} \sum_{(i,j)\in\mathscr{E}} w(i,j)\,|\beta(i,j)|^2 \tag{II.14}$$

where

$$w(i,j) \stackrel{\rm def}{=} \frac{w_\circ(i,j)\,\sqrt{\rho_{\rm d}(i,j)\rho_{\rm m}(i,j)}}{\displaystyle\sum_{(i,j)\in\mathscr{E}} w_\circ(i,j)\,\sqrt{\rho_{\rm d}(i,j)\rho_{\rm m}(i,j)}} \tag{II.15}$$

Note that, by construction, w is normalized so that

$$\sum_{(i,j)\in\mathscr{E}} w(i,j) = 1 \tag{II.16}$$

Introducing the 'chord function' chord:

$$\operatorname{chord}\theta \stackrel{\text{def}}{=} |e^{\mathrm{i}\theta} - 1| = 2|\sin(\theta/2)| \tag{II.17}$$

we therefore have

$$f_\circ(\alpha_d) = \|\operatorname{chord}(\varphi - B\alpha_d)\|_w^2 \tag{II.18}$$

In radio-astronomy, since the pioneering work of Cornwell & Wilkinson (1981), the phase-calibration problem is generally handled at the chord level by minimizing functionals such as $f_\circ$. Under certain circumstances, $\mathfrak{f}_\circ$ and $f_\circ$ may have secondary minima. As a result, the nonlinear optimization algorithms may fail in finding the global minimum. As shown in Sect. II.5, this difficulty can be overcome by solving the problem at the arc level. The following variant $g_\circ(\alpha_d)$ of $f_\circ$ is then considered (see Fig. II.2):

$$g_\circ(\alpha_d) \stackrel{\text{def}}{=} \|\operatorname{arc}(\varphi - B\alpha_d)\|_w^2 \tag{II.19}$$

Note that for any θ in $\mathbb{R}$, the 'arc function' arc is defined by the relation

$$\operatorname{arc}\theta \stackrel{\text{def}}{=} \theta - 2\pi\left\lfloor\frac{\theta}{2\pi}\right\rceil \tag{II.20}$$

Here, $\lfloor x\rceil$ denotes the integer of $\mathbb{Z}$ closest to x; when $x = k + 1/2$ for some k in $\mathbb{Z}$,$\lfloor x\rceil$ is set equal to k. We thus have $-\pi < \operatorname{arc}\theta \leq \pi$. As $\operatorname{chord}\theta \leq |\operatorname{arc}\theta|$,

$$\sqrt{f_\circ} \leq \sqrt{g_\circ} \leq \pi \tag{II.21}$$

II.2 Theoretical framework

In Sect. II.2.1, we first introduce the notions of spanning tree and loops. We then define related spaces of functions (Sect. II.2.2). The key property on which our analysis is based is presented in Sect. II.2.3. The concept of phase closure is defined in this general framework (Sects. II.2.4 and II.2.5). The connection with the usual notion of phase closure is specified in Sect. II.2.6. Sections II.2.7 and II.2.8 are devoted to some technical aspects concerning related variance-covariance matrices.

II.2.1 Spanning tree and loops

As illustrated in Fig. II.3, a spanning tree of a phase-calibration graph $\mathscr{G} \stackrel{\text{set}}{=} \mathscr{G}(\mathscr{V}, \mathscr{E})$ is a subgraph $\mathscr{G}_{\text{st}} \stackrel{\text{set}}{=} \mathscr{G}(\mathscr{V}, \mathscr{E}_{\text{st}})$ formed by n_{v} vertices and $n_{\text{v}} - 1$ edges, with no 'cycle' in it. Here, 'cycle' is used in the sense defined in algebraic graph theory (see, e.g., Biggs (1996)). To avoid any confusion with the notion of wave cycle, we will substitute the term of 'loop' for that of 'cycle.' In this context, the number of loops defined through a given spanning tree is the number of baselines of $\mathscr{E}$ that do not lie in $\mathscr{E}_{\text{st}}$ These baselines,

$$\mathrm{e}_\ell \stackrel{\text{def}}{=} (i_\ell, j_\ell) \tag{II.22}$$

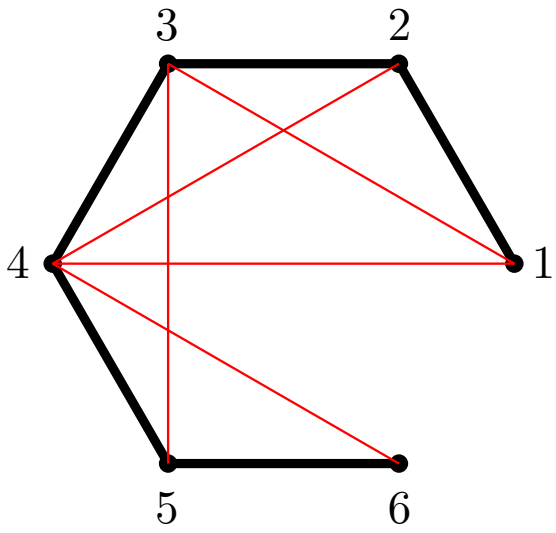

Figure II.3: Spanning tree and loops. The graph $\mathscr{G}$ shown here is the same as that presented in Fig. II.1. It includes six vertices (six pupil elements) and ten edges (ten baselines): $n_{\rm v} = 6$, $n_{\rm e} = 10$. The 5 edges represented with thick lines are those of the selected spanning tree: $n_{\rm st} \overset{\rm def}{=} n_{\rm v} - 1$. The remaining 5 edges define as many loops: $n_{\rm c} = n_{\rm e} - n_{\rm st}$. Here, these loops are the following: $(1, 3, 2)$, $(1, 4, 3, 2)$, $(2, 4, 3)$, $(3, 5, 4)$ and $(4, 6, 5)$. Note that the second loop includes four edges.

which form a set denoted by $\mathscr{E}_{\rm c}$, are said to be loop-entry baselines (see Figs. II.3 and II.4). Their number is $n_{\rm c}$ Denoting by

$$n_{\rm st} \overset{\rm def}{=} n_{\rm v} - 1 \tag{II.23}$$

the number of edges of the spanning tree, we therefore have

$$n_{\rm c} = n_{\rm e} - n_{\rm st} \tag{II.24}$$

As illustrated in Figs. II.3 and II.4, the loops may include more than three edges.

A spanning tree of maximal weight is a spanning tree for which the sum of the weights of its edges is maximal. An example is shown in Fig. II.4. According to the principle of the Kruskal algorithm (1956), such a tree can be obtained as follows:

(1) For $p = 1, \ldots, n_{\rm e}$, sort the edges ${\rm e}(p) \overset{\rm set}{=} (i, j)$ of $\mathscr{E}$ so that their weights $\varpi(p) \overset{\rm set}{=} w(i, j)$ are in non-increasing order: $\varpi(1) \geq \cdots \geq \varpi(n_{\rm e})$.

(2) Set $p \overset{\rm set}{=} 0$, $q \overset{\rm set}{=} 0$, and $\mathscr{E}_{\rm st} \overset{\rm set}{=} \emptyset$ (the empty set).

(3) If $q = n_{\rm st}$, terminate the process; otherwise, set $p \overset{\rm set}{=} p + 1$.

(4) When the vertices of ${\rm e}(p)$ are not connected via edges of $\mathscr{E}_{\rm st}$, set $\mathscr{E}_{\rm st} \overset{\rm set}{=} \mathscr{E}_{\rm st} \cup \{{\rm e}(p)\}$, and go to step (3) with $q \overset{\rm set}{=} q + 1$.

When the weights $w(i, j)$ are all distinct, the spanning tree is unique.

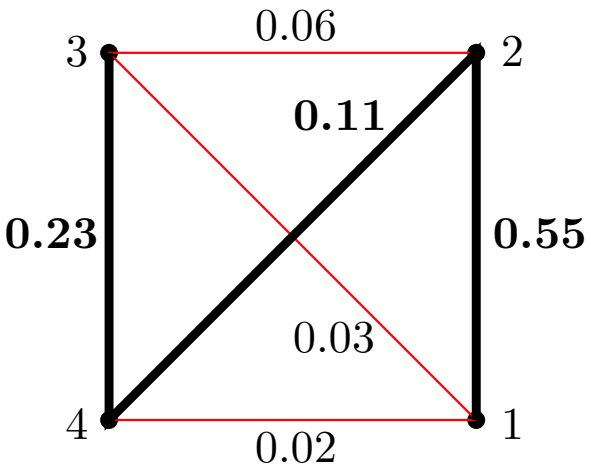

Figure II.4: Spanning tree of maximal weight (example). For the weights $w(i,j)$ displayed on the edges of the graph $\mathscr{G}$ shown here, the spanning tree of maximal weight is formed by the baselines $(1,2)$, $(3,4)$ and $(2,4)$ plotted with thick lines. The remaining baselines are loop-entry baselines.

II.2.2 Reference spaces

In this section, we introduce several Euclidean spaces. As illustrated in Fig. II.5, our analysis can thus be followed in a geometrical manner.

Pupil-phase space. Any real-valued function α taking its values on the pupil elements of $\mathscr{V}$, with $\alpha(1) = 0$, can be regarded of a vector of $\mathbb{R}^{n_\mathrm{v}-1}$; see Eq. (II.6). The pupil-phase space V_b is the space of those functions (or vectors). From Eq. (II.23), this space is isomorphic to $\mathbb{R}^{n_\mathrm{st}}$:

$$V_b \cong \mathbb{R}^{n_\mathrm{st}} \tag{II.25}$$

The 'integer lattice' $V_b(\mathbb{Z})$ is the subset of V_b formed by its integer-valued functions: $V_b(\mathbb{Z}) \cong \mathbb{Z}^{n_\mathrm{st}}$.

Baseline-phase space. Any real-valued function β taking its values on the baselines of $\mathscr{E}$ can be regarded as a vector of $\mathbb{R}^{n_\mathrm{e}}$. The baseline-phase space E is the space of those baseline-phase functions (or vectors):

$$E \cong \mathbb{R}^{n_\mathrm{e}} \tag{II.26}$$

The values of β on $\mathscr{E}$ are then regarded as the components of β in the standard basis of E. The integer lattice $E(\mathbb{Z})$ is the subset of E formed by its integer-valued functions: $E(\mathbb{Z}) \cong \mathbb{Z}^{n_\mathrm{e}}$.

The inner product on E is defined by the relation

$$(\beta_1 \mid \beta_2) \overset{\mathrm{def}}{=} \sum_{(i,j)\in\mathscr{E}} \beta_1(i,j)\,\beta_2(i,j) \tag{II.27}$$

i.e., in terms of matrices,

$$(\beta_1 \mid \beta_2) = \beta_1^\mathrm{t}\beta_2 \tag{II.28}$$

Spanning-tree phase space. The functions of E that vanish on $\mathscr{E}_{\rm c}$ form a subspace of E denoted by $E_{\rm st}$ the spanning-tree phase space. This space is of dimension $n_{\rm st}$ [see Eq. (II.23)]:

$$\dim E_{\rm st} = n_{\rm st} \tag{II.29}$$

Closure-phase space. The orthogonal complement of $E_{\rm st}$ in the Euclidean space E is denoted by $E_{\rm c}$ see Fig. II.5. Its functions vanish on $\mathscr{E}_{\rm st}$. As justified in Sect. II.2.4, $E_{\rm c}$ can be referred to as the closure-phase space. From Eq. (II.24),

$$\dim E_{\rm c} = n_{\rm c} \tag{II.30}$$

Baseline-bias phase space. Consider the operator B from V_b into E defined by Eq. (II.12). Its range, which is denoted by $E_{\rm b}$ $E_{\rm b}$ (see Fig. II.5), is referred to as the baseline-bias phase space. Its functions $\beta_{\rm b}(i,j)$ are of the form $\alpha(i) - \alpha(j)$ with $\alpha(1) = 0$.

The operator from V_b into $E_{\rm st}$ induced by B is denoted by $B_{\rm st}$ Likewise, the operator from V_b into $E_{\rm c}$ induced by B is denoted by $B_{\rm c}$ The matrix of B is generally expressed in the standard bases of V_b and E.

For example, let us sort the baselines of the graph shown in Fig. II.4 in the order defined for the application of the Kruskal algorithm: $\mathbf{(1,2)}$, $\mathbf{(3,4)}$, $\mathbf{(2,4)}$, $(2,3)$, $(1,3)$, $(1,4)$, We then have

$$[B]\alpha = \begin{bmatrix} \mathbf{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{-1} \\ \mathbf{1} & \mathbf{0} & \mathbf{-1} \\ 1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \alpha(2) \\ \alpha(3) \\ \alpha(4) \end{bmatrix} \tag{II.31}$$

The columns of $[B]$ then define the standard basis of $E_{\rm b}$. Clearly,

$$[B_{\rm st}] = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \tag{II.32}$$

and

$$[B_{\rm c}] = \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{II.33}$$

The condition $B_{\rm st}\alpha = 0$, i.e., $B\alpha = 0$ on the baselines of $\mathscr{E}_{\rm st}$, implies that α is constant on $\mathscr{V}$; as $\alpha(1) = 0$, this constant is zero. The null space of $B_{\rm st}$ is therefore reduced to $\{0\}$. As $B\alpha = 0$ implies $B_{\rm st}\alpha = 0$, the null space of B is also reduced to $\{0\}$. We thus have

$$\ker B = \ker B_{\rm st} = \{0\} \tag{II.34}$$

As a result, B is of full rank, hence [from Eq. (II.25)]

$$\dim E_{\rm b} = \dim V_b = n_{\rm st}$$

II.2.3 Key property

As $\ker B_{\rm st} = \{0\}$ [Eq. (II.34)], and $\dim E_{\rm st} = \dim V_b$ [see Eqs. (II.29) and (II.25)], $B_{\rm st}$ maps V_b onto $E_{\rm st}$; $B_{\rm st}$ is therefore invertible. As specified in this section, our analysis derives from this property.

Let us introduce the $Q_{\rm st}$ the orthogonal projection of E onto $E_{\rm st}$ (see Fig. II.5). For any β in E, $Q_{\rm st}\beta$ is the function of $E_{\rm st}$ whose values are those of β on $\mathscr{E}_{\rm st}$. Let us now concentrate on the pupil-phase function

$$\alpha_\beta \stackrel{\rm def}{=} B_{\rm st}^{-1} Q_{\rm st}\beta \tag{II.35}$$

The following process provides the values of α_β in a recursive manner, and thereby (if need be) a way of computing the matrix elements of $[B_{\rm st}]^{-1}$.

Recursive approach. Set $\alpha_\beta(1) \stackrel{\rm set}{=} 0$; then, span the baselines of $\mathscr{E}_{\rm st}$ in a given order. For each baseline (i,j) thus encountered, then proceed as follows:

- when $\alpha_\beta(i)$ has already been set, and $\alpha_\beta(j)$ is not set yet, then $\alpha_\beta(j) \stackrel{\rm set}{=} \alpha_\beta(i) - \beta(i,j)$.
- when $\alpha_\beta(j)$ has already been set, and $\alpha_\beta(i)$ is not set yet, then $\alpha_\beta(i) \stackrel{\rm set}{=} \beta(i,j) + \alpha_\beta(j)$.

To obtain all these biases, $\mathscr{E}_{\rm st}$ is to be spanned in this way as many times as required.

The only operations involved in the process are algebraic sums. As a result, when β is an integer-valued function ν, the function

$$\mu_\nu \stackrel{\rm def}{=} B_{\rm st}^{-1} Q_{\rm st}\nu \tag{II.36}$$

is also integer valued: μ_ν lies in $V_b(\mathbb{Z})$. As a corollary, when β is defined modulo 2π (on each baseline of $\mathscr{E}$), α_β is defined modulo 2π (on each pupil-element of $\mathscr{V}$).

To illustrate this recursive approach, we now follow the action of this process on the spanning tree of Fig. II.4. As $\alpha_\beta(1)$ is zero, we then obtain successively:

$$\begin{aligned}
\alpha_\beta(2) &= \alpha_\beta(1) - \beta(1,2) = -\beta(1,2)\\
\alpha_\beta(4) &= \alpha_\beta(2) - \beta(2,4) = -\beta(1,2) - \beta(2,4)\\
\alpha_\beta(3) &= \beta(3,4) + \alpha_\beta(4) = \beta(3,4) - \beta(1,2) - \beta(2,4)
\end{aligned}$$

We thus have

$$\begin{bmatrix} \alpha_\beta(2) \\ \alpha_\beta(3) \\ \alpha_\beta(4) \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ -1 & 1 & -1 \\ -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \beta(1,2) \\ \beta(3,4) \\ \beta(2,4) \end{bmatrix}$$

hence, here,

$$[B_{\rm st}]^{-1} = \begin{bmatrix} -1 & 0 & 0 \\ -1 & 1 & -1 \\ -1 & 0 & -1 \end{bmatrix} \tag{II.37}$$

Alternative approach. In fact, as specified in A1, $B_{\rm st}$ is a particular unimodular matrix whose inverse can be obtained via another integer-programming technique; see how Eq. (II.37) is

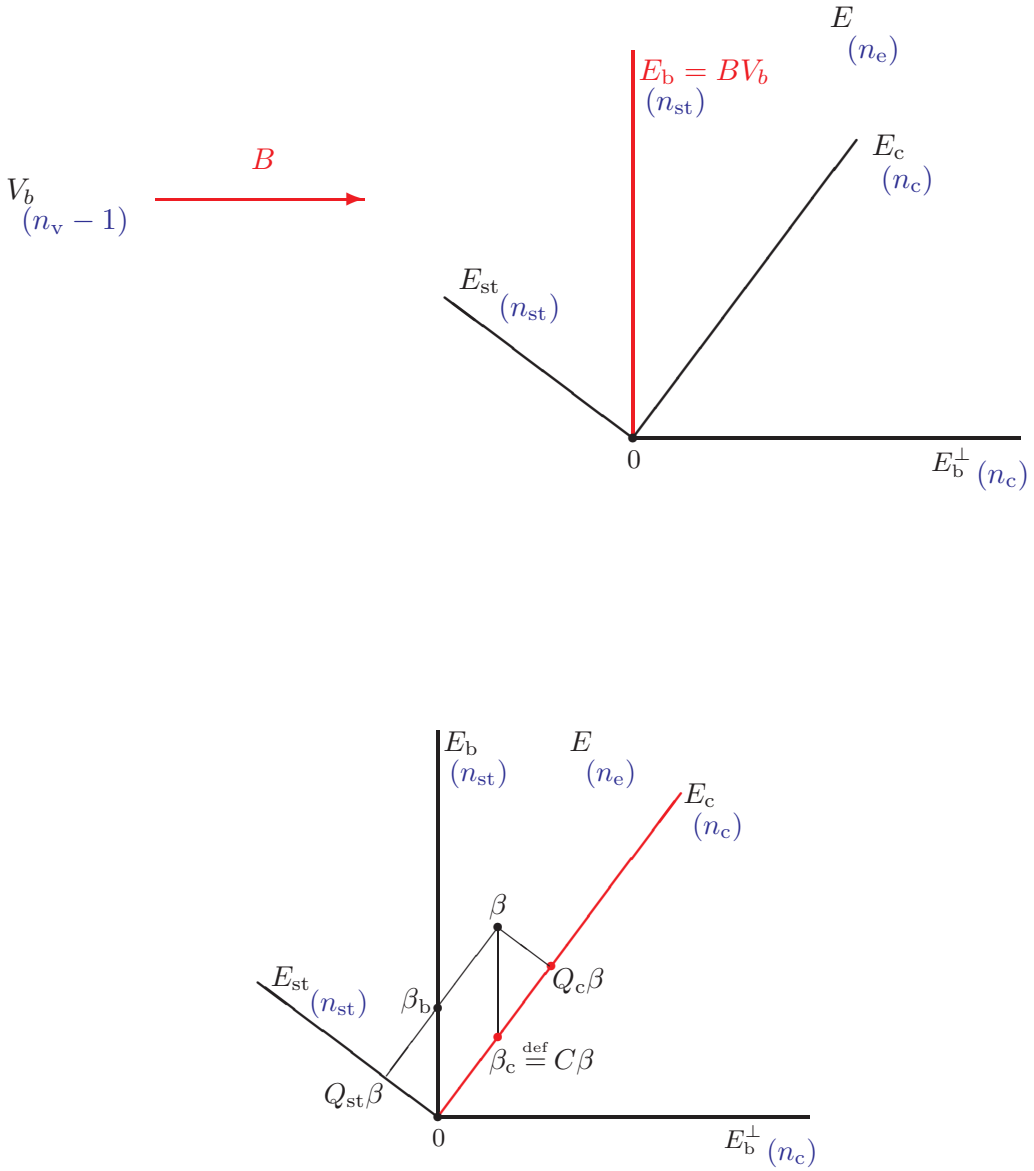

Figure II.5: Canonical decompositions of the baseline-phase space E. In this geometrical representation of E, E_{st} is the spanning-tree phase space. This space is isomorphic to the pupil-phase space V_b. The orthogonal complement of E_{st} in E is the closure-phase space E_c. The baseline-bias phase space (the range of operator B) is a subspace of E denoted by E_b. As B is of full rank [see Eq. (II.34)], this space is isomorphic to V_b and thereby to E_{st}. The dimensions of these main spaces are written within parentheses. As illustrated here, E is the oblique direct sum of E_b and E_c: for any β in E, we have $\beta = \beta_b + \beta_c$. The phase-closure operator C is the oblique projection of E onto E_c along E_b; for further details see Property 1 and Eq. (II.35). The orthogonal complement of E_b in E is the range of the transpose of matrix $[C]$; see Eq. (II.46). As illustrated here, for any β in E, $[C]^t\beta - \beta$ is orthogonal to E_c in E; see text.

obtained in Sect. A1.4 [Eq. (A.8)]. (By definition, a unimodular matrix is a square integer matrix with determinant ± 1.)

Let us now consider the following function of $E_{\rm b}$:

$$\beta_{\rm b} \stackrel{\rm def}{=} B\alpha_\beta \tag{II.38}$$

According to Eq. (II.35), the values of $\beta_{\rm b}$ and β coincide on $\mathscr{E}_{\rm st}$. The function $\beta_{\rm c}$ defined by the relation

$$\beta_{\rm c} \stackrel{\rm def}{=} \beta - \beta_{\rm b} \tag{II.39}$$

therefore lies in $E_{\rm c}$. We thus have the following property (see Fig. II.5):

Property 1. *Any baseline-phase function β of E can be decomposed in the form $\beta = \beta_{\rm b} + \beta_{\rm c}$ with $\beta_{\rm b} \stackrel{\rm def}{=} B\alpha_\beta$, and $\beta_{\rm c}$ in $E_{\rm c}$. For a given spanning tree, this decomposition is unique. As a corollary, E is the direct sum of $E_{\rm b}$ and $E_{\rm c}$: $E = E_{\rm b} + E_{\rm c}$ with $E_{\rm b} \cap E_{\rm c} = \{0\}$.*

In particular, any function ν of $E(\mathbb{Z})$ can be decomposed in the form $\nu = \nu_{\rm b} + \nu_{\rm c}$ with $\nu_{\rm b} \stackrel{\rm def}{=} B\mu_\nu$ and $\nu_{\rm c}$ in $E_{\rm c}$. According to our analysis, the functions μ_ν, $\nu_{\rm b}$ and $\nu_{\rm c}$ are then integer valued. The integer lattice $E(\mathbb{Z})$ is therefore the direct sum of the integer lattices $E_{\rm b}(\mathbb{Z})$ and $E_{\rm c}(\mathbb{Z})$. This point is illustrated in Fig. II.8.

II.2.4 Closure phases

According to Eqs. (II.39) and (II.38), the quantities $\beta_{\rm c}(i_\ell, j_\ell)$, for $\ell = 1, \ldots, n_{\rm c}$, can be computed via the formula

$$\beta_{\rm c}(i_\ell, j_\ell) = \beta(i_\ell, j_\ell) - \big[\alpha_\beta(i_\ell) - \alpha_\beta(j_\ell)\big] \tag{II.40}$$

where α_β is determined via Eq. (II.35). As clarified in this section, these quantities can be referred to as the 'closure phases' of β.

For example, let us consider the second loop of Fig. II.4, i.e., the loop associated with loop-entry baseline $(i_2, j_2) = (1, 3)$. The successive baselines of this loop are the following: $(1, 3)$, $(3, 4)$, $(2, 4)$ and $(1, 2)$. In a telescoping manner, we then have, since $\beta_{\rm b}(i, j) = \alpha_\beta(i) - \alpha_\beta(j)$,

$$\beta_{\rm b}(1,3) + \beta_{\rm b}(3,4) - \beta_{\rm b}(2,4) - \beta_{\rm b}(1,2) = 0$$

Furthermore, as $\beta_{\rm c}$ vanishes on $\mathscr{E}_{\rm st}$,

$$\beta_{\rm c}(1,3) + \beta_{\rm c}(3,4) - \beta_{\rm c}(2,4) - \beta_{\rm c}(1,2) = \beta_{\rm c}(1,3)$$

Since $\beta = \beta_{\rm b} + \beta_{\rm c}$ (from Property 1), it follows that

$$\beta(1,3) + \beta(3,4) - \beta(2,4) - \beta(1,2) = \beta_{\rm c}(1,3)$$

This explicitly shows that $\beta_{\rm c}(i_2, j_2)$ can be regarded as the closure phase of β on the second loop. The generalization is straightforward. In the special case of Fig. II.4, we thus have

$$\left|\begin{array}{l} \beta_{\rm c}(2,3) = \beta(2,3) + \beta(3,4) - \beta(2,4) \\ \beta_{\rm c}(1,3) = \beta(1,3) + \beta(3,4) - \beta(2,4) - \beta(1,2) \\ \beta_{\rm c}(1,4) = \beta(1,4) - \beta(2,4) - \beta(1,2) \end{array}\right. \tag{II.41}$$

The closure phases $\beta_c(i_\ell, j_\ell)$ are associated with loops whose order is greater than or equal to 3. According to Eq. (II.40), these closure phases can however be computed without knowing the baselines of their loop. How to identify these baselines if need be is specified in Sect. II.2.5. Subject to some condition, these closure phases can be expressed as linear combinations of closure phases of order 3. The related matter is presented in Sect. II.2.6.

II.2.5 Phase-closure operator

From Property 1, the closure-phase function β_c is the oblique projection of β on E_c along E_b; see Fig. II.5. The corresponding operator is the 'phase-closure operator' C:

$$\beta_c = C\beta \tag{II.42}$$

Its null space (i.e., its kernel) is the range of B:

$$\ker C = E_b \tag{II.43}$$

Note that β_b is the oblique projection of β on E_b along E_c.

According to the definitions of B_{st} and B_c (see Sect. II.2.2), the vector $\beta_b \stackrel{\text{def}}{=} B\alpha_\beta$ can be orthogonally decomposed in the form [see Eq. (II.35)]

$$\beta_b = B_{st}\alpha_\beta + B_c\alpha_\beta = Q_{st}\beta + B_c\alpha_\beta$$

Likewise,

$$\beta = Q_{st}\beta + Q_c\beta$$

where $Q_c\beta$ is the orthogonal projection of β on E_c. We have thus defined Q_c the orthogonal projection of E onto E_c (see Fig. II.5). For any β in E, it then follows that

$$\beta_c = \beta - \beta_b = Q_c\beta - B_c\alpha_\beta = Q_c\beta - B_cB_{st}^{-1}Q_{st}\beta$$

Denoting by $[C]$ the matrix of C expressed in the standard bases of E and E_c, we therefore have, from Eq. (II.42),

$$[C]\beta = -[B_c][B_{st}]^{-1}[Q_{st}\beta] + [Q_c\beta] \tag{II.44}$$

The column vectors of $[C]$ corresponding to the spanning-tree baselines (on which $Q_c\beta$ vanishes) are therefore those of $-[B_c][B_{st}]^{-1}$. It is also clear that the column vectors of $[C]$ corresponding to the loop-entry baselines (on which $Q_{st}\beta$ vanishes) are those of the identity matrix on E_c.

For example, in the special case of Fig. II.4, with the same baseline ordering, we have, from Eqs. (II.33) and (II.37),

$$[B_c][B_{st}]^{-1} = \begin{bmatrix} 0 & -1 & 1 \\ 1 & -1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

As a result [from Eq. (II.44)],

$$[C] = \begin{bmatrix} 0 & 1 & -1 & 1 & 0 & 0 \\ -1 & 1 & -1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 & 0 & 1 \end{bmatrix} \tag{II.45}$$

Applied to β, this matrix of course yields Eq. (II.41). Clearly, the baselines of each loop can be identified via the nonzero entries of the corresponding line of $[C]$.

For any α in V_b, and any β in E,

$$([C]^{\mathrm{t}}\beta \mid B\alpha) = (\beta \mid CB\alpha) = 0.$$

The range of $[C]^{\mathrm{t}}$ is therefore the orthogonal complement of E_{b} in E:

$$E_{\mathrm{b}}^{\perp} = [C]^{\mathrm{t}}E \tag{II.46}$$

Moreover, $[C]^{\mathrm{t}}\beta - \beta$ is orthogonal to E_{c} in E; see Fig. II.5. Indeed, $([C]^{\mathrm{t}}\beta - \beta \mid \beta_{\mathrm{c}}) = (\beta \mid C\beta_{\mathrm{c}}) - (\beta \mid \beta_{\mathrm{c}}) = 0$.

II.2.6 Connection with the usual notion of phase closure

Until now in astronomy, to the best of our knowledge, the closure phases have been computed on loops of order three only. Given some phase-calibration graph $\mathscr{G}$, the operator that provides all these closure phases is denoted by $\mathcal{C}_3^{\circ}$ The rank of its matrix $[\mathcal{C}_3^{\circ}]$ (the dimension of its range), which is equal to the rank of $[\mathcal{C}_3^{\circ}]^{\mathrm{t}}$, is denoted by n_{c3}

For example, let us consider the phase-calibration graph $\mathscr{G}$ of Fig. II.4. We then have four triangles which can be sorted in the following order:

$(1,2,3)$
$(1,2,4)$
$(1,3,4)$
$(2,3,4)$

Let us then order the edges of $\mathscr{G}$ as follows:

$$(1,2),\ (1,3),\ (1,4),\ (2,3),\ (2,4),\ (3,4)$$

The 'closure matrix' $[\mathcal{C}_3^{\circ}]$ is then the following:

$$[\mathcal{C}_3^{\circ}] = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix} \tag{II.47}$$

As shown in A1 (see Example 3 in that appendix), n_{c3} is then equal to 3. As n_{c3} is the rank of $[\mathcal{C}_3^{\circ}]^{\mathrm{t}}$, the maximum number of independent closure phases of order three is then equal to 3.

We will call 'usual phase-closure operator' $\mathcal{C}_3$ an operator whose matrix is a closure matrix that provides a 'maximum set independent closure phases of order three.' Such a matrix,

denoted by $[\mathcal{C}_3]$, therefore provides n_{c3} independent closure phases of order three. For example, in the case of Fig. II.4, the matrix formed by the first three lines of $[\mathcal{C}_3^\circ]$ is such a matrix:

$$[\mathcal{C}_3] = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 \end{bmatrix} \tag{II.48}$$

Beyond this example, as $\mathrm{Rank}\,[\mathcal{C}_3] = \mathrm{Rank}\,[\mathcal{C}_3]^{\mathrm{t}} = n_{c3}$, we have

$$n_{\mathrm{e}} = \dim(\ker \mathcal{C}_3) + n_{c3} \tag{II.49}$$

Likewise, $n_{\mathrm{e}} = \dim(\ker C) + n_{\mathrm{c}}$. Consequently, since $\ker C = E_{\mathrm{b}}$ [Eq. (II.43)],

$$n_{\mathrm{e}} = \dim E_{\mathrm{b}} + n_{\mathrm{c}} \tag{II.50}$$

As $\mathcal{C}_3$ annihilates the pupil phases, we have $\mathcal{C}_3 E_{\mathrm{b}} = \{0\}$, hence $E_{\mathrm{b}} \subseteq \ker \mathcal{C}_3$. As a result, $\dim E_{\mathrm{b}} \leq \dim(\ker \mathcal{C}_3)$. It then follows from Eqs. (II.49) and (II.50) that

$$n_{c3} \leq n_{\mathrm{c}} \tag{II.51}$$

When $n_{c3} = n_{\mathrm{c}}$, as this is the case in Fig. II.4 with $n_{c3} = 3$ (and in Fig. II.1 with $n_{c3} = 5$), we have [from Eqs. (II.49) and (II.50)]: $\dim(\ker \mathcal{C}_3) = \dim E_{\mathrm{b}}$. As $E_{\mathrm{b}} \subseteq \ker \mathcal{C}_3$, it then follows that

$$\ker \mathcal{C}_3 = \ker C = E_{\mathrm{b}} \qquad (\text{if } n_{c3} = n_{\mathrm{c}}) \tag{II.52}$$

The closure phases $\beta_{\mathrm{c}}(i_\ell, j_\ell)$ can then be expressed as linear combinations of independent closure phases of order three. More precisely, the matrix involved in this relationship is unimodular; see Example 5 in Sect. A1.4.

When n_{c3} is strictly less than n_{c} the closure information provided by any maximum set of independent closure phases of order three is incomplete. An example of such a situation is given in Fig. II.6. In such cases, our new approach is of course more efficient.

II.2.7 Weighted baseline-phase space

The definitions of Functionals (II.18) and (II.19) refer to the norm defined by Eq. (II.14), i.e., the norm induced by the inner product

$$(\beta_1 \mid \beta_2)_w \overset{\text{def}}{=} \sum_{(i,j)\in\mathscr{E}} w(i,j)\, \beta_1(i,j)\, \beta_2(i,j) \tag{II.53}$$

In terms of matrices, we thus have

$$(\beta_1 \mid \beta_2)_w = \beta_1^{\mathrm{t}}\, [\mathcal{V}]^{-1} \beta_2 \tag{II.54}$$

where $[\mathcal{V}]$ is a diagonal matrix whose entries are equal to $1/w(i,j)$; $[\mathcal{V}]$ can therefore be regarded as 'a variance-covariance matrix' of the baseline phase vector φ [see Eqs. (II.10)

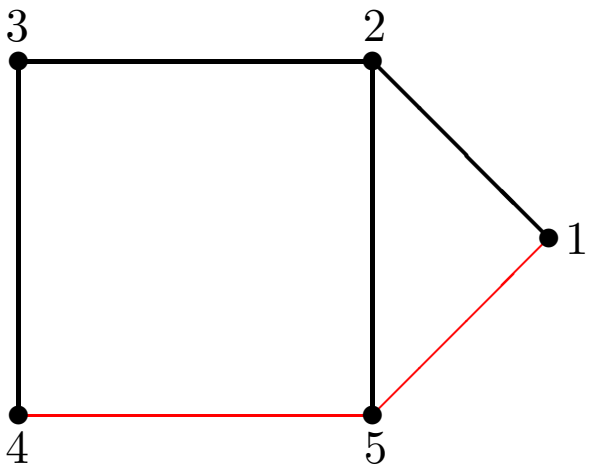

Figure II.6: The closure information provided by a maximum set of independent closure phases of order three may be incomplete. In this example of phase-calibration graph, for the spanning tree formed by the baselines $(1,2)$, $(2,3)$, $(2,5)$ and $(3,4)$, plotted with thick lines, we have one loop of order three, and one loop of order four. The information provided by the closure phase of order three is then incomplete: $n_{\mathrm{c}3} = 1$, while $n_{\mathrm{c}} = 2$; see Eq. (II.51).

and (II.15)]. Endowed with this inner product, E is a real Hilbert space denoted by E_w the 'weighted baseline-phase space.' Clearly, from Eqs. (II.54) and (II.28), we have

$$(\beta_1 \mid \beta_2)_w = (\beta_1 \mid [\mathcal{V}]^{-1}\beta_2) \tag{II.55}$$

For any β_{b} in E_{b}, and any β in $[\mathcal{V}]E_{\mathrm{b}}^{\perp}$, we have $(\beta_{\mathrm{b}} \mid \beta)_w = (\beta_{\mathrm{b}} \mid [\mathcal{V}]^{-1}\beta) = 0$; indeed, $[\mathcal{V}]^{-1}\beta$ then lies in $E_{\mathrm{b}}^{\perp}$. This explicitly shows that $[\mathcal{V}]E_{\mathrm{b}}^{\perp}$ is the orthogonal complement of E_{b} in E_w (see Fig. II.7):

$$E_{\mathrm{b}}^{\perp w} = [\mathcal{V}]E_{\mathrm{b}}^{\perp} \tag{II.56}$$

Let us denote by B^+ the Moore-Penrose pseudo-inverse of B, where B is regarded as an operator from V_b into E_w. As $\ker B = \{0\}$ [Eq. (II.34)], we have

$$B^+\beta = \underset{\alpha \in V_b}{\operatorname{argmin}} \|\beta - B\alpha\|_w^2 \tag{II.57}$$

This pupil phase is therefore the solution of the normal equation

$$B^*B\alpha = B^*\beta \tag{II.58}$$

where B^* is the adjoint of B relative to the inner product w. Note that in terms of matrices, we have [see Eq. (II.54)],

$$\begin{aligned}[B^*]\beta &= [B]^{\mathrm{t}}[\mathcal{V}]^{-1}\beta \\ &= [B]^{\mathrm{t}}[w\beta]\end{aligned} \tag{II.59}$$

According to the definition of B^+ [Eq. (II.57)], BB^+ is the orthogonal projection of E_w onto E_{b}. The orthogonal projection of E_w onto $E_{\mathrm{b}}^{\perp w}$ is therefore explicitly defined by the relation

$$S\beta \overset{\mathrm{def}}{=} \beta - BB^+\beta \tag{II.60}$$

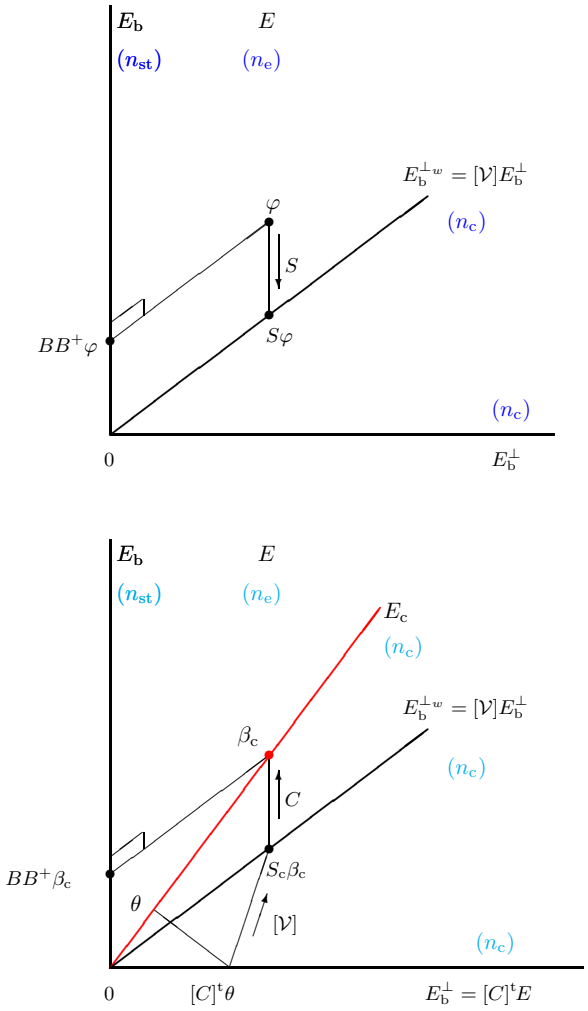

Figure II.7: Geometrical representation of the orthogonal complement of E_b in the weighted baseline-phase space E_w: $E_\mathrm{b}^{\perp w}$. As $E_\mathrm{b}^{\perp}$ is the orthogonal complement of E_b in the baseline-phase space E (see Fig. II.5), $E_\mathrm{b}^{\perp w}$ is equal to $[\mathcal{V}]E_\mathrm{b}^{\perp}$; see text. By definition, S_c is the orthogonal projection of E_c onto $E_\mathrm{b}^{\perp w}$. Note that CS_c is the identity on E_c. As illustrated here, BB^+ is the orthogonal projection of E_w onto E_b.

In what follows, S_c is the operator from E_c into $E_\mathrm{b}^{\perp w}$ induced by S [see Fig. II.7]:

$$S_\mathrm{c}\colon E_\mathrm{c} \to E_\mathrm{b}^{\perp w}, \qquad S_\mathrm{c}\beta_\mathrm{c} = S\beta_\mathrm{c} \tag{II.61}$$

Denoting by $\mathfrak{e}_\ell$ the characteristic function of the loop-entry baseline (i_ℓ, j_ℓ) [see Eq. (II.22)], i.e., the function of E_c that is equal to unity on this baseline, and is zero on the other ones, let us introduce the vectors (of V_b)

$$\zeta_\ell \overset{\text{def}}{=} B^+\mathfrak{e}_\ell \qquad (\ell = 1, \ldots, n_\mathrm{c}) \tag{II.62}$$

The vectors (of $E_\mathrm{b}^{\perp w}$)

$$\begin{aligned}\eta_\ell &\overset{\text{def}}{=} S_\mathrm{c}\mathfrak{e}_\ell \\ &= \mathfrak{e}_\ell - B\zeta_\ell\end{aligned} \qquad (\ell = 1, \ldots, n_\mathrm{c}) \tag{II.63}$$

are the column vectors of the matrix of S_c expressed in the standard bases of E and E_c.

II.2.8 Inverse of the variance-covariance matrix of the closure phases of φ

We now concentrate on the variance-covariance matrix $[\mathcal{V}_c]$ of the closure phases $\varphi_c(i_\ell, j_\ell)$. In matrix terms, we have

$$\varphi_c = [C]\varphi$$

As matrix $[\mathcal{V}]$ can be regarded as the expectation value of $[\delta\varphi][\delta\varphi]^t$ (see Sect. 3.7), the variance-covariance matrix $[\mathcal{V}_c]$ induced by $[\mathcal{V}]$ is therefore given by the formula

$$[\mathcal{V}_c] = [C][\mathcal{V}][C]^t \tag{II.64}$$

As illustrated in Fig. II.7, for any θ in E_c, we have

$$[S_c]^t\,[\mathcal{V}]^{-1}[S_c]\;[C][\mathcal{V}]\,[C]^t\theta = [S_c]^t[C]^t\theta \tag{II.65}$$

But, CS_c is the identity on E_c: $[C][S_c] = [I]_{n_c}$; see Fig. II.7. As a result, $[S_c]^t[C]^t$ is also equal to $[I]_{n_c}$. It then follows from Eqs. (II.65) and (II.64) that

$$[\mathcal{V}_c]^{-1} = [S_c]^t[\mathcal{V}]^{-1}[S_c] \tag{II.66}$$

We therefore have, from Eq. (II.63),

$$\begin{aligned}\big(\mathfrak{e}_\ell \mid [\mathcal{V}_c]^{-1}\mathfrak{e}_{\ell'}\big) &= \big(\mathfrak{e}_\ell \mid [S_c]^t[\mathcal{V}]^{-1}[S_c]\mathfrak{e}_{\ell'}\big)\\ &= (\eta_\ell \mid [\mathcal{V}]^{-1}\eta_{\ell'})\end{aligned} \tag{II.67}$$

In the standard basis of E_c, the matrix elements of the inverse of $[\mathcal{V}_c]$ are therefore the inner products $(\eta_\ell \mid \eta_{\ell'})_w$.

For any β in E,

$$\begin{aligned}\|S\beta\|_w^2 &= (S\beta \mid [\mathcal{V}]^{-1}S\beta)\\ &= \|T\beta\|^2\end{aligned} \tag{II.68}$$

where

$$[T] \overset{\text{def}}{=} [\mathcal{V}]^{-1/2}[S] \tag{II.69}$$

Setting $[T_c] \overset{\text{def}}{=} [\mathcal{V}]^{-1/2}[S_c]$, we have from Eq. (II.66):

$$[T_c]^t[T_c] = [\mathcal{V}_c]^{-1} \tag{II.70}$$

As S_c and T_c are invertible, $[T_c]^t[T_c]$ is positive definite.

II.3 Phase calibration: survey

On the grounds of the analysis developed in Sect. II.2.3, the phase-calibration problem stated in Sect. II.1 can be reformulated in phase-closure terms (Sect. II.3.1). We then derive a useful expression for the corresponding calibrated visibility function (Sect. II.3.2).

II.3.1 Statement of the problem in phase-closure terms

Let us consider the phase discrepancy involved in the definitions of the functionals $f_\circ$ and $g_\circ$ to be minimized [see Eqs. (II.18), (II.19), (II.10) and Fig. II.2]:

$$\delta \stackrel{\text{def}}{=} \varphi - B\alpha_d \qquad \text{(modulo } 2\pi\text{)} \tag{II.71}$$

With regard to the spanning tree of maximal weight (see for example Fig. II.4), φ can be expanded in the form (see Property 1)

$$\varphi = B\alpha_\varphi + \varphi_{\text{c}} \tag{II.72}$$

where [see Eq. (II.35)]

$$\alpha_\varphi \stackrel{\text{def}}{=} B_{\text{st}}^{-1} Q_{\text{st}} \varphi \tag{II.73}$$

According to our analysis, the pupil phase function α_φ can be determined via the recursive process described in Sect II.2.3, or via the integer-programming technique presented in A1.4; see Example 4 in that appendix. As φ is defined modulo 2π, α_φ and φ_{c} are also defined modulo 2π.

From Eqs. (II.71) and (II.72), we have, modulo 2π (on each baseline of $\mathscr{E}$),

$$\delta = \varphi_{\text{c}} - B\alpha$$

where the pupil phase function α is defined as:

$$\alpha \stackrel{\text{def}}{=} \alpha_d - \alpha_\varphi \tag{II.74}$$

Introducing the 'reduced closure-phase function' [see Eq. (II.20)]

$$\widehat{\varphi}_{\text{c}} \stackrel{\text{def}}{=} \text{arc}(\varphi_{\text{c}}) \tag{II.75}$$

we therefore have

$$\delta = \widehat{\varphi}_{\text{c}} - B\alpha \qquad \text{(modulo } 2\pi\text{)} \tag{II.76}$$

It then follows that minimizing $f_\circ(\alpha_d)$ [Eq. (II.18)] amounts to minimizing the functional $f(\alpha)$:

$$f(\alpha) \stackrel{\text{def}}{=} \|\text{chord}(\widehat{\varphi}_{\text{c}} - B\alpha)\|_w^2 \tag{II.77}$$

Likewise, minimizing $g_\circ(\alpha_d)$ [Eq. (II.19)] amounts to minimizing the functional $g(\alpha)$:

$$g(\alpha) \stackrel{\text{def}}{=} \|\text{arc}(\widehat{\varphi}_{\text{c}} - B\alpha)\|_w^2 \tag{II.78}$$

If α_f is a minimizer of $f(\alpha)$, the corresponding minimizer of $f_\circ(\alpha_d)$ is given by the formula [see Eq. (II.74)]

$$\alpha_{f_\circ} = \alpha_f + \alpha_\varphi \qquad \text{(modulo } 2\pi\text{)} \tag{II.79}$$

Likewise, for $g_\circ(\alpha_d)$,

$$\alpha_{g_\circ} = \alpha_g + \alpha_\varphi \qquad \text{(modulo } 2\pi\text{)} \tag{II.80}$$

II.3.2 Derivation of the calibrated visibility function

For clarity, let us introduce the phase-calibration residual ε_α:

$$\varepsilon_\alpha \overset{\text{def}}{=} \arc(\widehat{\varphi}_\text{c} - B\alpha) \tag{II.81}$$

Let α_g now be some minimizer of g, for example a global minimizer. Denoting by ε_{α_g} the corresponding 'phase-calibration residual,' we have [see Eq. (II.78)],

$$g(\alpha_g) = \|\varepsilon_{\alpha_g}\|_w^2 \tag{II.82}$$

From Eqs. (II.75), (II.80) and (II.72),

$$\varepsilon_{\alpha_g} = \arc(\varphi - B\alpha_{g_\circ})$$

As a result [see Eq. (II.10)],

$$\varepsilon_{\alpha_g} = (\varphi_\text{d} - B\alpha_{g_\circ}) - \varphi_\text{m} \qquad \text{(modulo } 2\pi\text{)} \tag{II.83}$$

By taking into account Eqs. (II.9) and (II.12), the calibration formula (II.7) yields, for $\alpha_{d\star} \overset{\text{set}}{=} \alpha_{g_\circ}$ [see Eq. (II.80)],

$$V_{\text{d}\star} = \rho_\text{d}\, e^{\text{i}(\varphi_\text{d} - B\alpha_{g_\circ})} \qquad (\text{on } \mathscr{E}) \tag{II.84}$$

On the baselines of $\mathscr{E}$, we thus have, from Eq. (II.83),

$$V_{\text{d}\star} = \rho_\text{d}\, e^{\text{i}(\varphi_\text{m} + \varepsilon_{\alpha_g})} \qquad (\text{on } \mathscr{E}) \tag{II.85}$$

As the phase calibration residual ε_{α_g} is a result of the minimization process (see Sect. II.4), this formula is often preferable; see Fig. II.10.

Remark II.3.2.1. When the signal-to-noise ratio (SNR) of the instantaneous baseline-phase data $\varphi_\text{d}^{(t)}$ is low, Property 1 suggests that our approach should be modified as follows. Let $z^{(t)}$ be the closure phasor of $\varphi_\text{d}^{(t)}$: $z^{(t)} \overset{\text{def}}{=} e^{\text{i}C\varphi_\text{d}^{(t)}}$. On each baseline of $\mathscr{E}$, φ_d is then defined via the relation $e^{\text{i}\varphi_\text{d}} \overset{\text{def}}{=} \langle z^{(t)}\rangle_t / |\langle z^{(t)}\rangle_t|$; φ_d is then a virtual instantaneous baseline-phase function of E_c whose closure components $\varphi_\text{d}(i_\ell, j_\ell)$ are the phases (modulo 2π) of the time-averaged closure phasors $\langle z^{(t)}(i_\ell, j_\ell)\rangle_t / |\langle z^{(t)}(i_\ell, j_\ell)\rangle_t|$. In our approach, phase-closure averaging is simply a way of improving the SNR of the pseudo baseline-phase data to be processed.

Remark II.3.2.2. The baselines of the interferometric device defined by the pupil elements of $\mathscr{V}$ form a set denoted by $\mathscr{E}_\text{d}$. By construction, we have $\mathscr{E} \subseteq \mathscr{E}_\text{d}$; see Sect. II.1.1. When the SNR has not been enhanced by phase-closure averaging, the calibration formula (II.84) can be extended to all the baselines of $\mathscr{E}_\text{d}$.

Remark II.3.2.3. In the limiting case where $\widehat{\varphi}_\text{c} = 0$, we have from Eq. (II.78): $\alpha_g = 0$ (modulo 2π in V_b); indeed, $\ker B = \{0\}$ [Eq. (II.34)]. Then, $\alpha_{g_\circ} = \alpha_\varphi$ (modulo 2π in V_b), and $\varepsilon_{\alpha_g} = 0$; see Eqs. (II.80) and (II.81) successively. The phase-calibration residual has then reached its lowest possible value, which is a clear indication of the convergence of the 'self-cal' image-restoration procedure; see Sects. II.1 and II.7.

II.4 The arc approach

We first characterize the minimizers of the arc functional g (Sect. II.4.1), and then show how to find them explicitly (Sect. II.4.2). We finally consider the special case where all the reduced closure phases are less than $\pi/2$ in absolute value (Sect. II.4.3).

II.4.1 Constrained minimizers

Let us denote by τ is a positive function strictly less than $1/2$ all over $\mathscr{E}$:

$$\forall (i,j) \in \mathscr{E}, \qquad 0 < \tau(i,j) < 1/2$$

Our objective is to identify the minimizers α_g of $g(\alpha)$ for which [see Eq. (II.81)]

$$|\varepsilon_{\alpha_g}| \leq 2\pi\tau \qquad \text{(all over } \mathscr{E}) \tag{II.86}$$

The minima for which this condition is satisfied are said to be the 'τ-constrained minima' of g. By setting τ equal to a constant function arbitrarily close to $1/2$, our analysis can be extended to all the minima of g; see Sect. 5.2, and the simulations presented in Sect. II.6.

Let v be a vector of E_{c}:

$$v = \sum_{\ell=1}^{n_{\mathrm{c}}} v^{(\ell)} \mathfrak{e}_\ell \qquad\qquad v^{(\ell)} \stackrel{\text{def}}{=} v(i_\ell, j_\ell) \tag{II.87}$$

For simplicity, we will denote by the same symbol v the vector of $\mathbb{R}^{n_{\mathrm{c}}}$ with components $v^{(\ell)}$ for $\ell = 1, \ldots, n_{\mathrm{c}}$. A notation such as B^+v or Sv can then still be used: B^+v is a function lying in V_b, and Sv is a function lying in $E_{\mathrm{b}}^{\perp_w}$; see Sect. II.2.7. To make the link with similar problems encountered in GNSS (see Lannes & Teunissen (2011)), we denote by $\hat{v}$ the vector of $\mathbb{R}^{n_{\mathrm{c}}}$ with components

$$\hat{v}^{(\ell)} \stackrel{\text{def}}{=} \frac{1}{2\pi} \widehat{\varphi}_{\mathrm{c}}(i_\ell, j_\ell) \quad (\text{for } \ell = 1, \ldots, n_{\mathrm{c}}) \tag{II.88}$$

For each point $\dot{v}$ of the integer lattice $\mathbb{Z}^{n_{\mathrm{c}}}$, we now introduce the pupil-phase function

$$\dot{\alpha} \stackrel{\text{def}}{=} 2\pi B^+(\hat{v} - \dot{v}) \tag{II.89}$$

The subset of points $\dot{v}$ of $\mathbb{Z}^{n_{\mathrm{c}}}$ such that $|S(\dot{v} - \hat{v})|$ is less than (or equal to) τ all over $\mathscr{E}$ is denoted by Γ_τ:

$$\Gamma_\tau \stackrel{\text{def}}{=} \{\dot{v} \in \mathbb{Z}^{n_{\mathrm{c}}} : |S(\dot{v} - \hat{v})| \leq \tau\} \tag{II.90}$$

Property 2. *The τ-constrained minimizers of g are associated with the points $\dot{v}$ of Γ_τ. More precisely, the τ-constrained minimizers α_g associated with a point $\dot{v}$ of Γ_τ are equal to $\dot{\alpha}$ modulo 2π (on each pupil element of $\mathscr{V}$). As a corollary, $e^{\mathrm{i}\alpha_g} = e^{\mathrm{i}\dot{\alpha}}$, $g(\alpha_g) = g(\dot{\alpha})$, and $\varepsilon_{\alpha_g} = \varepsilon_{\dot{\alpha}} = 2\pi S(\hat{v} - \dot{v})$ with $|\varepsilon_{\dot{\alpha}}| \leq 2\pi\tau$.*

The proof of this property is to be found in Appendix 2.

II.4.2 Discrete search of all the minima

A systematic search of all the minima is often useful. According to Property 2, this can be done by listing the points of Γ_τ for τ arbitrarily close to $1/2$. In what follows, a notation such as $\tau = 1/2$ or $\Gamma_{1/2}$ is to be understood in that sense.

From Eqs. (II.90) and Eq. (II.14), Γ_τ is contained in the set of points v of $\mathbb{R}^{n_c}$ such that

$$\|S(v - \hat{v})\|_w^2 \leq \chi_\tau \tag{II.91}$$

where

$$\chi_\tau \stackrel{\text{def}}{=} \sum_{(i,j)\in\mathscr{E}} w(i,j)\, \tau^2(i,j) \tag{II.92}$$

The weights $w(i,j)$ being normalized according to Eq. (II.16), we thus have

$$\chi_{1/2} = \frac{1}{4}$$

Let us now consider the ellipsoid, centred on $\hat{v}$, with size parameter $\mathfrak{c}$,

$$\mathfrak{E}(\mathfrak{c}) \stackrel{\text{def}}{=} \{v \in \mathbb{R}^{n_c} : \|T_c(v - \hat{v})\|^2 \leq \mathfrak{c}\} \tag{II.93}$$

As $\|S(v - \hat{v})\|_w^2 = \|T_c(v - \hat{v})\|^2$ [see Eq. (II.68)], $\Gamma_{1/2}$ is contained in $\mathfrak{E}(1/4)$.

The points $\dot{v}$ of lattice $\mathbb{Z}^{n_c}$ lying in $\mathfrak{E}(1/4)$ can be listed by using the discrete-search algorithms described in A3 (Appendix 3). For each of these points, we compute [see Eqs. (II.87) and (II.63)]

$$S(\dot{v} - \hat{v}) = \sum_{\ell=1}^{n_c} (\dot{v}^{(\ell)} - \hat{v}^{(\ell)})\, \eta_\ell \tag{II.94}$$

When $|S(\dot{v} - \hat{v})|$ is less than $1/2$ all over $\mathscr{E}$, $\dot{v}$ lies in $\Gamma_{1/2}$ [see Eq. (II.90)]. According to Property 2, $\dot{\alpha}$ is then a minimizer of g. If need be, $\dot{\alpha}$ can be determined via the formula [see Eqs. (II.89) and (II.62)]

$$\dot{\alpha} = 2\pi \sum_{\ell=1}^{n_c} (\hat{v}^{(\ell)} - \dot{v}^{(\ell)})\, \zeta_\ell \tag{II.95}$$

We then compute [see Property 2 and Eqs. (II.82) & (II.68)]

$$g(\dot{\alpha}) = 4\pi^2 \|T_c(\dot{v} - \hat{v})\|^2 \tag{II.96}$$

This is done via Eq. (C.23) [in which ω is defined in Eq. (C.2)]. The points of $\Gamma_{1/2}$ are finally ordered so that

$$g(\dot{\alpha}_1) < g(\dot{\alpha}_2) < \cdots$$

From Eq. (II.96), the global minimizer of g, $\check{\alpha} \stackrel{\text{set}}{=} \dot{\alpha}_1$, corresponds to the point $\check{v} \stackrel{\text{set}}{=} \dot{v}_1$ of lattice $\mathbb{Z}^{n_c}$ for which $\|T_c(\dot{v} - \hat{v})\|^2$ is minimum. Clearly,

$$\|T_c(\dot{v} - \hat{v})\|^2 = \big(\dot{v} - \hat{v} \mid [T_c]^t [T_c](\dot{v} - \hat{v})\big)$$

where $[T_c]^t[T_c] = [\mathcal{V}_c]^{-1}$ [see Eq. (II.70)]; $\check{v}$ is therefore the point of $\mathbb{Z}^{n_c}$ closest to the 'float solution' $\hat{v}$, the distance being that induced by the quadratic form whose matrix $[T_c]^t[T_c]$ is the inverse of $[\mathcal{V}_c]$. The notation '$\hat{v}$, $\check{v}$,' and the terminology adopted here are those used in GNSS (see, e.g., Lannes & Teunissen (2011)).

II.4.3 Special case of small closure phases

We now consider the special case where all the reduced closure phases are less than $\pi/2$ in absolute value: $|\widehat{\varphi}_{\mathrm{c}}|$ is less than $\pi/2$ all over $\mathscr{E}_{\mathrm{c}}$; from Eq. (II.88), $|\hat{v}|$ is then less than $1/4$. Clearly, this case corresponds to the situations where "the model is not too far from the object," e.g., the example of Sect. II.6.1. As a general rule, the global minimum is then associated with the point zero of $\mathbb{Z}^{n_{\mathrm{c}}}$:

$$\dot{v}_0 = 0 \qquad (\dot{v}_0^{(\ell)} = 0 \text{ for } \ell = 1, \ldots, n_{\mathrm{c}}) \tag{II.97}$$

In other words, we then have $\dot{v}_1 = \dot{v}_0$, hence $\dot{\alpha}_1 = \dot{\alpha}_0$. From Property 2 and Eqs. (II.89) and (II.88), $\dot{\alpha}_0$ is equal to $B^+\widehat{\varphi}_{\mathrm{c}}$:

$$\dot{\alpha}_0 = B^+\widehat{\varphi}_{\mathrm{c}} \tag{II.98}$$

From Eqs. (II.57) and (II.58), $\dot{\alpha}_0$ is therefore the solution of the equation

$$B^*B\alpha = B^*\widehat{\varphi}_{\mathrm{c}} \tag{II.99}$$

The phase-calibration operation is then linear. Furthermore, from Eq. (II.96), we have

$$g(\dot{\alpha}_0) = 4\pi^2\|T_{\mathrm{c}}\hat{v}\|^2 \tag{II.100}$$

Check point. We can verify, *a posteriori*, in a very fast manner, whether the surmise $\dot{v}_1 = \dot{v}_0$ holds or not. This must be done (in particular) when some reduced closure phases are of the order of $\pi/2$ in absolute value. Let us consider the ellipsoid $\mathfrak{E}(\mathfrak{c}_0)$ with $\mathfrak{c}_0 \overset{\text{def}}{=} \|T_{\mathrm{c}}\hat{v}\|^2$; see Eq. (II.93). In this ellipsoid, we search for the point $\check{v}$ for which the global minimum of $g(\dot{\alpha})$ is attained. If $\check{v} = \dot{v}_0$, the surmise holds.

Compared to the complete discrete search described in Sect. II.4.2, the process is here much faster for two reasons. Firstly, we simply want to identify the global minimum (see Remark A3.2.1 in A3), and secondly, the size of the ellipsoid $\mathfrak{E}(\mathfrak{c}_0)$ is much smaller than that of $\mathfrak{E}(1/4)$. Indeed, we have $\mathfrak{c}_0 = \|T_{\mathrm{c}}\hat{v}\|^2 = \|S\hat{v}\|_w^2$ with $\|S\hat{v}\|_w^2 \leq \|\hat{v}\|_w^2$ (since S is an orthogonal projection). Furthermore, $|\hat{v}|$ is here less $1/4$, hence $\|\hat{v}\|_w^2 < 1/16$ from Eqs. (II.14) and (II.16). As a result, $\mathfrak{c}_0$ is less than $1/16$.

II.5 Connection with the chord approach

The chord functional (II.77) is explicitly defined by the relation [see Eq. (II.17)]

$$f(\alpha) = \left\| 2\sin\left(\frac{\widehat{\varphi}_{\mathrm{c}} - B\alpha}{2}\right)\right\|_w^2 \tag{II.101}$$

Note that f is a functional from V_b into $\mathbb{R}$. To search for the minimizers of f, iterative methods such as the trust-region method can be implemented; see A4 (Appendix 4). These techniques are based on the Taylor quadratic approximation to $f(\alpha + h)$ at the current iterate α:

$$f(\alpha + h) \simeq q(h) \tag{II.102}$$

where

$$q(h) \stackrel{\text{def}}{=} f(\alpha) + \big(f'(\alpha) \mid h\big)_{V_b} + \frac{1}{2}\big(h \mid [f''(\alpha)]\, h\big)_{V_b} \tag{II.103}$$

Here, $(\cdot \mid \cdot)_{V_b}$ is the inner product on the Euclidean space V_b; see Eq. (II.25). The iteration of the trust-region method is of the traditional form $\alpha \stackrel{\text{set}}{=} \alpha + h$. The originality of this method is to choose h (at each iteration) so that the smallest value of q is attained in the largest possible ball, the degree of confidence in the quadratic approximation (II.102) being controlled throughout the process by means of the Armijo test. One thus goes down 'continuously' to a local minimum; for further details, see A4.

We first show how to find the minimizers of f from those of g (Sect. II.5.1), and the converse (Sect. II.5.2). We then introduce the notion of 'linked minima' (Sect. II.5.3). The chord approach of the special case of small closure phases is presented in Sect. II.5.4.

II.5.1 Search for the minimizers of f from those of g

The minima of the chord functional may be obtained from those of the arc functional. More precisely, the minimizers of g can be used as starting points of the trust-region method. We now specify the corresponding initialization process.

Let $\dot{v}$ be a point of $\Gamma_{1/2}$; see Eq. (II.90). According to Property 2, the minimizer of g associated with $\dot{v}$ is given by the formula $\dot{\alpha} = 2\pi B^+(\hat{v} - \dot{v})$ [Eq. (II.89)], and we then have

$$\varepsilon_{\dot{\alpha}} = 2\pi S(\hat{v} - \dot{v}) \quad \text{with} \quad |\varepsilon_{\dot{\alpha}}| < \pi$$

When using the pupil phase $\dot{\alpha}$ as starting point, the trust-region method yields a minimizer of f denoted by $[\dot{\alpha}]_f$. As already emphasized, this minimizer is obtained by diminishing the value of f at $\dot{\alpha}$ in a continuous manner; see A4.

As shown in Appendix 5, the gradient and the Hessian of f at α are respectively given by the formulae

$$\left|\begin{array}{l} f'(\alpha) \;= -2B^* \sin(\widehat{\varphi}_{\mathrm{c}} - B\alpha) \\ f''(\alpha) \;= \;\; 2B^* [\cos(\widehat{\varphi}_{\mathrm{c}} - B\alpha)] B \end{array}\right. \tag{II.104}$$

We thus have [see Definition (II.81)]

$$\left|\begin{array}{l} f'(\alpha) \;= -2B^* \sin \varepsilon_\alpha \\ f''(\alpha) \;= \;\; 2B^* [\cos \varepsilon_\alpha] B \end{array}\right. \tag{II.105}$$

As shown in A5 [see Eq. (E.7)], the condition

$$|\varepsilon_\alpha| < \pi/2 \qquad (\text{all over } \mathscr{E}) \tag{II.106}$$

is a sufficient condition for $f''(\alpha)$ to be positive definite.

When, for some iterate α, Condition (II.106) is satisfied, the step h of the iteration of the trust-region method is then a simple Newton-Raphson step: h is the solution of Eq. (D.10), i.e.,

$$[f''(\alpha)]h = -f'(\alpha) \tag{II.107}$$

When $|\varepsilon_\alpha|$ is greater than (or equal to) $\pi/2$ on some baseline(s), $f''(\alpha)$ may not be positive definite. Then, to determine h, the trust-region method takes into the account the fact that the smallest eigenvalue of $f''(\alpha)$ may be less than (or equal to) zero; see A4.

II.5.2 Search for the minimizers of g from those of f

Let us assume that a minimizer of f has been obtained somehow, for example by using the trust-region method from a given starting point; see A4. Denoting by α_f this minimizer, let us consider the pupil phase

$$\alpha_{fg} \overset{\text{def}}{=} \alpha_f + B^+ \varepsilon_{\alpha_f} \tag{II.108}$$

where

$$\varepsilon_{\alpha_f} \overset{\text{def}}{=} \operatorname{arc}(\widehat{\varphi}_{\text{c}} - B\alpha_f) \tag{II.109}$$

Note that from the definition of B^+ [Eq. (II.57)], we have

$$B^+ \varepsilon_{\alpha_f} = \underset{\alpha \in V_b}{\operatorname{argmin}} \, \|\varepsilon_{\alpha_f} - B\alpha\|_w^2$$

We now show that when

$$|S\varepsilon_{\alpha_f}| \leq \pi \qquad \text{(all over } \mathscr{E}\text{)} \tag{II.110}$$

then, α_{fg} is a minimizer of g lying in a 'small neighbourhood' of α_f. Moreover, we then have $\varepsilon_{\alpha_{fg}} = S\varepsilon_{\alpha_f}$.

To conduct the related analysis, we first introduce the following definition. Let $\nu(\alpha)$ be the function from V_b into $E(\mathbb{Z})$ defined by the relation [see Eq. (II.20)]

$$\nu(\alpha) \overset{\text{def}}{=} \left\lfloor \frac{\widehat{\varphi}_{\text{c}} - B\alpha}{2\pi} \right\rceil \tag{II.111}$$

From Eq. (II.109), we thus have

$$\varepsilon_{\alpha_f} = (\widehat{\varphi}_{\text{c}} - B\alpha_f) - 2\pi\nu(\alpha_f) \tag{II.112}$$

Likewise,

$$\varepsilon_{\alpha_{fg}} = (\widehat{\varphi}_{\text{c}} - B\alpha_{fg}) - 2\pi\nu(\alpha_{fg}) \tag{II.113}$$

According to Property 1, $\nu(\alpha)$ can be decomposed in the form

$$\nu(\alpha) = \nu_{\text{b}}(\alpha) + \nu_{\text{c}}(\alpha) \;\text{ with }\; \nu_{\text{b}}(\alpha) = B[\mu_\nu(\alpha)] \tag{II.114}$$

where $\mu_\nu(\alpha) \overset{\text{def}}{=} B_{\text{st}}^{-1} Q_{\text{st}}[\nu(\alpha)]$; see Eq. (II.36); $\nu_{\text{c}}(\alpha)$ is a vector of E_{c} with integer components. In what follows, for clarity, we use the same notation $\nu_{\text{c}}(\alpha)$ for the vector of $\mathbb{Z}^{n_{\text{c}}}$ whose components are those of $\nu_{\text{c}}(\alpha)$ in the standard basis of E_{c}. For $\ell = 1, \ldots, n_{\text{c}}$, these components are computed via the formula [see Eq. (II.114)]

$$\big[\nu_{\text{c}}(\alpha)\big]^{(\ell)} = \big[\nu(\alpha)\big](i_\ell, j_\ell) - \big\{\big[\mu_\nu(\alpha)\big](i_\ell) - \big[\mu_\nu(\alpha)\big](j_\ell)\big\}$$

Let us now consider the discrepancy vector [see Fig. II.8]

$$\Delta \overset{\text{def}}{=} (\widehat{\varphi}_{\text{c}} - B\alpha_{fg}) - 2\pi\nu(\alpha_f) \tag{II.115}$$

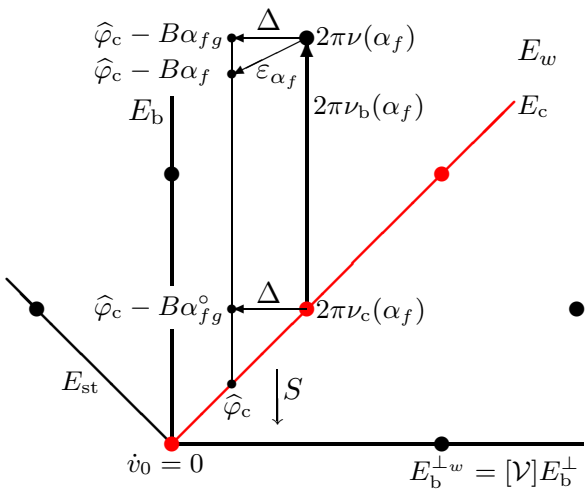

Figure II.8: Main elements involved in the analysis developed in Sect. II.5.2. In this geometrical representation of the weighted baseline-phase space E_w, the horizontal axis corresponds to the orthogonal complement of E_b in E_w: $E_b^{\perp_w}$; compare with Fig. II.7. The big dots correspond to points of the integer lattice $E(\mathbb{Z}) \cong \mathbb{Z}^{n_e}$. Here, α_f is a minimizer of f; ε_{α_f} is the arc of the discrepancy vector $\widehat{\varphi}_c - B\alpha_f$; see Eqs. (II.109) and (II.111); Δ is the projection of ε_{α_f} on $E_b^{\perp_w}$: $\Delta = S\varepsilon_{\alpha_f}$ [Eq. (II.116)]; see text.

From Eq. (II.108), we have

$$\Delta = \widehat{\varphi}_c - B\alpha_f - BB^+\varepsilon_{\alpha_f} - 2\pi\nu(\alpha_f)$$

Taking into account Eq. (II.112), we therefore have [see Eq. (II.60) and Fig. II.8]

$$\Delta = \varepsilon_{\alpha_f} - BB^+\varepsilon_{\alpha_f} = S\varepsilon_{\alpha_f} \tag{II.116}$$

We have thus shown that Δ is the projection of ε_{α_f} on the orthogonal complement of E_b in E_w: $E_b^{\perp_w}$; see Fig. II.8 and the context of Eq. (II.60). We therefore have [from Eq. (II.57)]

$$B^+\Delta = 0 \tag{II.117}$$

It then follows from Eqs. (II.115) that α_{fg} can also be obtained via the formula

$$\alpha_{fg} = B^+[\widehat{\varphi}_c - 2\pi\nu(\alpha_f)] \tag{II.118}$$

Furthermore, taking into account Eq. (II.114), we also have, from Eq. (II.115),

$$\begin{aligned}\Delta &= \widehat{\varphi}_c - B\alpha_{fg} - 2\pi[B\mu_\nu(\alpha_f) + \nu_c(\alpha_f)] \\ &= [\widehat{\varphi}_c - 2\pi\nu_c(\alpha_f)] - B\alpha^\circ_{fg}\end{aligned} \tag{II.119}$$

where

$$\alpha^\circ_{fg} \stackrel{\text{def}}{=} \alpha_{fg} + 2\pi\mu_\nu(\alpha_f) \tag{II.120}$$

From Eqs. (II.117) and (II.119), we thus have

$$\begin{aligned}\alpha^\circ_{fg} &= B^+[\widehat{\varphi}_\mathrm{c} - 2\pi\nu_\mathrm{c}(\alpha_f)] \\ &= 2\pi B^+\big[\hat{v} - \nu_\mathrm{c}(\alpha_f)\big]\end{aligned} \tag{II.121}$$

and

$$\begin{aligned}\Delta &= 2\pi[\hat{v} - \nu_\mathrm{c}(\alpha_f)] - 2\pi BB^+[\hat{v} - \nu_\mathrm{c}(\alpha_f)] \\ &= 2\pi S[\hat{v} - \nu_\mathrm{c}(\alpha_f)]\end{aligned} \tag{II.122}$$

It is therefore clear from Eq. (II.116) that Condition (II.110) is equivalent to the condition

$$\big|S[\nu_\mathrm{c}(\alpha_f) - \hat{v}]\big| \leq 1/2 \qquad \text{(all over } \mathscr{E}\text{)} \tag{II.123}$$

i.e., $\nu_\mathrm{c}(\alpha_f)$ lies in $\Gamma_{1/2}$; see Eq. (II.90). From Property 2, when this condition is satisfied, α°_{fg} is the minimizer of g associated with $\nu_\mathrm{c}(\alpha_f)$ [see Eq. (II.121)], and we have $\varepsilon_{\alpha^\circ_{fg}} = 2\pi S[\hat{v} - \nu_\mathrm{c}(\alpha_f)]$. Hence $\varepsilon_{\alpha^\circ_{fg}} = \Delta$ from Eq. (II.122). As α_{fg} is equal to α°_{fg} modulo 2π [see Eq. (II.120)], α_{fg} is then a minimizer of g too, and we have $\varepsilon_{\alpha_{fg}} = \varepsilon_{\alpha^\circ_{fg}} = \Delta$. It then follows from Eqs. (II.113) and (II.115) that

$$\nu(\alpha_{fg}) = \nu(\alpha_f) \tag{II.124}$$

This means that ν is stable under the pupil-phase shift $B^+\varepsilon_{\alpha_f}$; see Eqs. (II.108), (II.111) and Fig. II.8. As a result, α_{fg} then lies in a small neighbourhood of α_f.

In summary, given some minimizer α_f of f, α_{fg} can be obtained from Eq. (II.118), in which $\nu(\alpha_f)$ is defined via Eq. (II.111). When Condition (II.123) is satisfied, then α_{fg} is a minimizer of g lying in a small neighbourhood of α_f. Otherwise, α_{fg} is not a minimizer of g.

When α_{fg} is a minimizer of g, the minima of g less than or equal to $g(\alpha_{fg})$, if any, can be identified by listing the points of $\dot{v}$ of $\mathbb{Z}^{n_\mathrm{c}}$ contained in the ellipsoid (II.93) for $\mathfrak{c} \overset{\text{set}}{=} \|T(\nu_\mathrm{c}(\alpha_f) - \hat{v}\|^2$; see Eq. (II.96) and the procedure described in Sect. II.4.2.

II.5.3 Linked minima

Given some point $\dot{v}$ in $\Gamma_{1/2}$, the corresponding minimizer of g, $\dot{\alpha}$, is given by Eq. (II.89). With the aid of the trust-region method, we are then able to find, in a certain neighbourhood of $\dot{\alpha}$, a minimizer of f denoted by $[\dot{\alpha}]_f$; see Sect. II.5.1. When the minimizer of g obtained from $[\dot{\alpha}]_f$ coincides with $\dot{\alpha}$, we say that $f([\dot{\alpha}]_f)$ and $g(\dot{\alpha})$ are 'linked minima.' This is the case when the following property is satisfied:

$$\nu([\dot{\alpha}]_f) = \dot{v} \tag{II.125}$$

Indeed, as [from Eq. (II.118)]

$$\begin{aligned}[\dot{\alpha}]_{fg} &= B^+\big[\widehat{\varphi}_\mathrm{c} - 2\pi\nu([\dot{\alpha}]_f)\big] \\ &= 2\pi B^+\big[\hat{v} - \nu([\dot{\alpha}]_f)\big]\end{aligned} \tag{II.126}$$

we then have [from Eqs. (II.89) and (II.125)],

$$[\dot{\alpha}]_{fg} = \dot{\alpha} \tag{II.127}$$

As clarified in Sect. II.6, the situations *a priori* reliable are those for which there exists only one pair of linked minima.

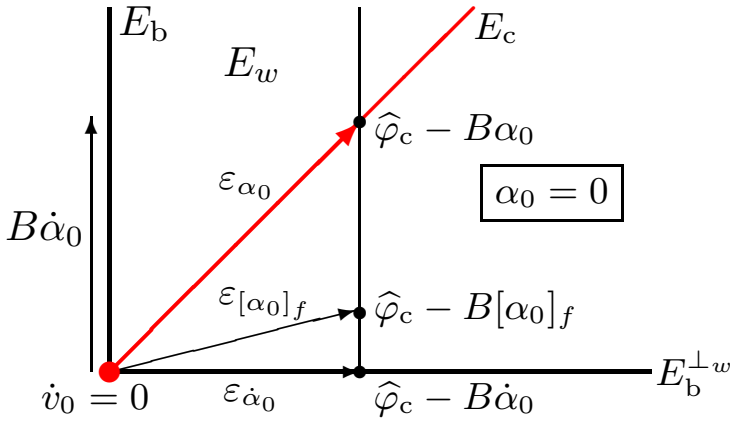

Figure II.9: Geometrical illustration of the special case where the reduced closure phases are small in absolute value. By definition, $\dot{\alpha}_0 = B^+\widehat{\varphi}_{\rm c}$; see Sect. II.4.3. Here, $[\alpha_0]_f$ is the minimizer of f obtained via the Newton-Raphson method when using $\alpha_0 = 0$ as starting point; see Sect. II.5.4. Note that $\nu(\alpha_0) = \nu([\alpha_0]_f) = \dot{v}_0$. In this process, the size of ε_{α_0} is reduced to that of $\varepsilon_{[\alpha_0]_f}$. The minimizer of g obtained from $[\alpha_0]_f$ then coincides with $\dot{\alpha}_0$; see Eq. (II.130). As illustrated here, $\varepsilon_{\dot{\alpha}_0}$ is the projection of $\varepsilon_{\alpha_0} = \widehat{\varphi}_{\rm c}$ onto $E_{\rm b}^{\perp w}$.

II.5.4 Reference special case of small closure phases

We now consider the special case where $|\widehat{\varphi}_{\rm c}|$ is less than $\pi/2$ all over $\mathscr{E}_{\rm c}$. This case was already studied in the arc approach; see Sect. II.4.3.

Let us then concentrate on Eq. (II.104) for $\alpha \overset{\text{set}}{=} \alpha_0 \overset{\text{def}}{=} 0$:

$$\left| \begin{array}{l} f'(0) \;= -2B^* \sin\widehat{\varphi}_{\rm c} \\ f''(0) = \;\; 2B^* \cos\widehat{\varphi}_{\rm c}\, B \end{array} \right. \tag{II.128}$$

As $\cos\widehat{\varphi}_{\rm c}$ is positive all over $\mathscr{E}$ the Hessian of f at α_0 is positive definite; see Condition (II.106). The point α_0 is then a good starting point for the trust-region method which then reduces to the Newton-Raphson iteration; see Eq. (II.107). The minimizer of f thus obtained is denoted by $[\alpha_0]_f$.

As $[\alpha_0]_f$ is a minimizer of f, $f''([\alpha_0]_f)$ is positive definite. We may therefore surmise from Eq. (II.104) that $|\widehat{\varphi}_{\rm c} - B[\alpha_0]_f|$ is less than $\pi/2$ all over $\mathscr{E}$ From Eq. (II.111), $\nu([\alpha_0]_f)$ is then the zero vector of $\mathbb{Z}^{n_{\rm c}}$ (see Fig. II.9):

$$\nu([\alpha_0]_f) = \dot{v}_0 \tag{II.129}$$

According to Eqs. (II.118) and (II.98), we then have

$$[\alpha_0]_{fg} = B^+\widehat{\varphi}_{\rm c} = \dot{\alpha}_0 \tag{II.130}$$

From the analysis developed in Sect. II.5.2, $\dot{\alpha}_0$ therefore lies in a small neighbourhood of $[\alpha_0]_f$; see Fig. II.9. As $f''([\alpha_0]_f)$ is positive definite, $[\dot{\alpha}_0]_f$ is thus expected to be equal to $[\alpha_0]_f$. As a result [from Eq. (II.129)]

$$\nu([\dot{\alpha}_0]_f) = \dot{v}_0 \tag{II.131}$$

From Eq. (II.125), the minima $f([\alpha_0]_f)$ and $g(\dot{\alpha}_0)$ are therefore linked minima. Moreover, as a general rule, $g(\dot{\alpha}_0)$ is then the global minimum of g; see Sect. II.4.3. In this case, as

$f([\alpha_0]_f)$ and $g(\dot{\alpha}_0)$ are linked minima, $f([\alpha_0]_f)$ is likely to be the global minimum of f. The pertinence of all these surmises can be easily verified *a posteriori*; see Sect. II.6.1 & the *check point* of Sect. II.4.3.

In this case, the phase-calibration operation is robust. In particular, it can be done, indifferently, with f or g. This pointed out, it is important to emphasize that $\dot{\alpha}_0$ is obtained by solving a single equation [Eq. (II.99)], while the determination of $[\alpha_0]_f$ requires solving a few equations of the same type. For example, from Eq. (II.107), the first iterate of the Newton-Raphson algorithm is the solution of the equation $[f''(\alpha_0)]\, h = -f'(\alpha_0)$. This iterate (practically) coincides with the solution of the 'arc equation' (II.99) only when $\cos\widehat{\varphi}_{\rm c} \simeq 1$; see Eq. (II.128).

II.6 Simulations

The simulations presented in this section correspond to the weighted calibration graph presented in Fig. II.4. We then have (see Sect. II.2.2)

$$\widehat{\varphi}_{\rm c}^{(1)} \stackrel{\rm def}{=} \mathrm{arc}\{\varphi_{\rm c}(2,3)\},\quad \widehat{\varphi}_{\rm c}^{(2)} \stackrel{\rm def}{=} \mathrm{arc}\{\varphi_{\rm c}(1,3)\}$$
$$\widehat{\varphi}_{\rm c}^{(3)} \stackrel{\rm def}{=} \mathrm{arc}\{\varphi_{\rm c}(1,4)\}$$

These 'reduced closure phases' correspond to the directed loops $(2,3,4)$, $(1,3,4,2)$ and $(1,4,2)$, respectively; see Sect. II.2.4.

Two sets of such closure phases are examined. For clarity, all the angles (such as $\widehat{\varphi}_{\rm c}^{(\ell)}$ and $\varepsilon_{\check{\alpha}}$) are expressed in degrees, as well as the values of the minima of the functionals $\sqrt{f}$ and $\sqrt{g}$.

II.6.1 Set 1

In this first example, the reduced closure phases are less than $90°$ in absolute value:

$$\widehat{\varphi}_{\rm c}^{(1)} = -15° \qquad \widehat{\varphi}_{\rm c}^{(2)} = -70° \qquad \widehat{\varphi}_{\rm c}^{(3)} = -40°$$

The discrete search of all the minima then yields (see Sect. II.4.2):

$$\sqrt{g(\dot{\alpha}_1)} \simeq 10.62°$$
$$\sqrt{g(\dot{\alpha}_2)} \simeq 69.79°$$
$$\sqrt{g(\dot{\alpha}_3)} \simeq 71.04°$$

The corresponding minimizers are associated with the following 'closure-ambiguity points' of $\mathbb{Z}^3$:

$$\dot{v}_1 = \check{v} = (0,0,0)$$
$$\dot{v}_2 = (-1,-1,0)$$
$$\dot{v}_3 = (-1,-1,-1)$$

The corresponding values of $\sqrt{f}$ are then

$$\sqrt{f(\dot{\alpha}_1)} \simeq 10.43°$$
$$\sqrt{f(\dot{\alpha}_2)} \simeq 55.14°$$
$$\sqrt{f(\dot{\alpha}_3)} \simeq 55.76°$$

By using the trust-region method, the values of the chord functional f can be reduced, continuously, from its value at $\dot{\alpha}_1$; see Sect. II.5.1. One thus obtains a minimizer $[\dot{\alpha}_1]_f$ of f. Here,

$$\sqrt{f([\dot{\alpha}_1]_f)} \simeq 10.42^\circ \qquad \sqrt{g([\dot{\alpha}_1]_f)} \simeq 10.63^\circ$$

As expected from our analysis when $|\widehat{\varphi}_c|$ is less than $\pi/2$ (see Sect. II.4.3), we then have [see Eq. (II.111)]

$$\begin{aligned}\nu([\dot{\alpha}_1]_f) &\stackrel{\text{def}}{=} \left\lfloor \frac{\widehat{\varphi}_c - B[\dot{\alpha}_1]_f}{2\pi} \right\rceil = \dot{\imath}_0 \\ &= \dot{\imath}_1\end{aligned}$$

Equation (II.125) is thus satisfied for $\dot{\alpha} \stackrel{\text{set}}{=} \dot{\alpha}_1$. As a result $f([\dot{\alpha}_1]_f)$ and $g(\dot{\alpha}_1)$ are linked minima: $[\dot{\alpha}_1]_{fg} = \dot{\alpha}_1$; see Fig. II.9 in Sect. II.5.3.

In the special case under consideration, we found that $[\dot{\alpha}_2]_f = [\dot{\alpha}_1]_f$, and modulo 2π: $[\dot{\alpha}_3]_f = [\dot{\alpha}_1]_f$. This explicitly shows that $f([\dot{\alpha}_1]_f)$ is then the single minimum of f.

Here, $\{f([\dot{\alpha}_1]_f), g(\dot{\alpha}_1)\}$ is therefore the only pair of linked minima. This is related to the fact that $f''(\dot{\alpha}_1)$ is positive definite, while this is not the case for $f''(\dot{\alpha}_2)$ and $f''(\dot{\alpha}_3)$.

Let us finally note that all the surmises of Sects. II.4.3 and II.5.4 are then satisfied. As expected, all the components of the phase-calibration residual $\varepsilon_{\check{\alpha}}$ are then less than 90° in absolute value:

$$\begin{aligned}&\varepsilon_{\check{\alpha}}(1,2) \simeq 3^\circ \qquad \varepsilon_{\check{\alpha}}(3,4) \simeq -5^\circ \qquad \varepsilon_{\check{\alpha}}(2,4) \simeq 15^\circ \\ &\varepsilon_{\check{\alpha}}(2,3) \simeq 5^\circ \qquad \varepsilon_{\check{\alpha}}(1,3) \simeq -47^\circ \qquad \varepsilon_{\check{\alpha}}(1,4) \simeq -22^\circ\end{aligned}$$

With regard to the selected model phase φ_m, the calibrated visibility function is then defined by Eq. (II.85) for $\alpha_g \stackrel{\text{set}}{=} \check{\alpha} = \dot{\alpha}_1$ with here $\dot{\alpha}_1 = \dot{\alpha}_0$. In summary, for this set of reduced closure phases, the phase-calibration operation is both simple and reliable.

II.6.2 Set 2

We now examine a situation *a priori* critical with

$$\widehat{\varphi}_c^{(1)} = -177^\circ \qquad \widehat{\varphi}_c^{(2)} = -171^\circ \qquad \widehat{\varphi}_c^{(3)} = 176^\circ$$

The arc functional has then three distinct minima:

$$\begin{aligned}\sqrt{g(\dot{\alpha}_1)} &\simeq 38.39^\circ \\ \sqrt{g(\dot{\alpha}_2)} &\simeq 39.63^\circ \\ \sqrt{g(\dot{\alpha}_3)} &\simeq 57.47^\circ\end{aligned}$$

The corresponding minimizers are associated with the following closure-ambiguity points

$$\begin{aligned}\dot{\imath}_1 = \check{\imath} &= (0,0,1) \\ \dot{\imath}_2 &= (-1,-1,0) \\ \dot{\imath}_3 &= (-1,0,1)\end{aligned}$$

(Note that the point zero of $\mathbb{Z}^3$, $\dot{v}_0$, is no longer associated with a minimizer of g.) The corresponding values of $\sqrt{f}$ are then

$$\sqrt{f(\dot{\alpha}_1)} \simeq 35.82^\circ$$
$$\sqrt{f(\dot{\alpha}_2)} \simeq 36.99^\circ$$
$$\sqrt{f(\dot{\alpha}_3)} \simeq 40.46^\circ$$

Here, $f''(\dot{\alpha}_1)$ and $f''(\dot{\alpha}_2)$ are positive definite. We then found that $\nu([\dot{\alpha}_1]_f) = \dot{v}_1$ and $\nu([\dot{\alpha}_2]_f) = \dot{v}_2$, and thereby $[\dot{\alpha}_1]_{fg} = \dot{\alpha}_1$ and $[\dot{\alpha}_2]_{fg} = \dot{\alpha}_2$. We therefore have two pairs of linked minima: $\{f([\dot{\alpha}_1]_f), g(\dot{\alpha}_1)\}$ and $\{f([\dot{\alpha}_2]_f), g(\dot{\alpha}_2)\}$. Furthermore,

$$\sqrt{f([\dot{\alpha}_1]_f)} \simeq 35.69^\circ \qquad \sqrt{g([\dot{\alpha}_1]_f)} \simeq 38.55^\circ$$
$$\sqrt{f([\dot{\alpha}_2]_f)} \simeq 36.82^\circ \qquad \sqrt{g([\dot{\alpha}_2]_f)} \simeq 39.83^\circ$$

and $[\dot{\alpha}_3]_f = [\dot{\alpha}_1]_f$.

Concerning the phase-calibration residuals, we have

$$\varepsilon_{\dot{\alpha}_1}(1,2) \simeq 7^\circ \qquad \varepsilon_{\dot{\alpha}_1}(3,4) \simeq -27^\circ \qquad \varepsilon_{\dot{\alpha}_1}(2,4) \simeq 76^\circ$$
$$\varepsilon_{\dot{\alpha}_1}(2,3) \simeq -74^\circ \qquad \varepsilon_{\dot{\alpha}_1}(1,3) \simeq -61^\circ \qquad \varepsilon_{\dot{\alpha}_1}(1,4) \simeq -101^\circ$$

and

$$\varepsilon_{\dot{\alpha}_2}(1,2) \simeq -7^\circ \qquad \varepsilon_{\dot{\alpha}_2}(3,4) \simeq 29^\circ \qquad \varepsilon_{\dot{\alpha}_2}(2,4) \simeq -78^\circ$$
$$\varepsilon_{\dot{\alpha}_2}(2,3) \simeq 76^\circ \qquad \varepsilon_{\dot{\alpha}_2}(1,3) \simeq 74^\circ \qquad \varepsilon_{\dot{\alpha}_2}(1,4) \simeq 91^\circ$$

As expected, the phase-calibration operation is then very ill conditioned: $g(\dot{\alpha}_2) \simeq g(\dot{\alpha}_1)$ with $\varepsilon_{\dot{\alpha}_2} \simeq -\varepsilon_{\dot{\alpha}_1}$.

As the selected spanning tree is of maximal weight, the weight of a loop-entry baseline is less than (or equal to) that of any other baseline of its loop; see Sect. II.2.1 and Fig. II.4. A reduced closure phase of high value, say greater than 120° in absolute value, therefore reveals that the error ϵ_{dm} is too large on the corresponding loop-entry baseline, in particular; see Eq. (II.5). Here, to regularize the problem, the baselines $(2,3)$, $(1,3)$ and $(1,4)$ must therefore be removed from the phase-calibration graph: "the model is too far from the object." We are then left with the baselines of the spanning tree only. Without any information on the closure phases of φ, the calibration formula (II.85) then reduces to $V_{\mathrm{d}\star} = \rho_{\mathrm{d}}\, e^{\mathrm{i}\varphi_{\mathrm{m}}}$.

II.7 Recommended strategy for phase-closure imaging

In this chapter we assume that the phase data were calibrated epoch by epoch, in an independent manner. For simplicity, we therefore restrict ourselves to the snapshot case: the Fourier data involved in the imaging process are the complex visibilities $V_{\mathrm{d}}(i,j)$ on the baselines of the interferometric device at some given epoch; see the introduction of Sect. II.1. The problem is then to calibrate the pupil-bias phases involved in Eq. (II.4).

We now propose a simple and robust strategy for conducting the phase-calibration operations of the self-calibration procedures. We first recall the principle of these procedures (Sect. II.7.1), and then describe our phase-calibration method step by step (Sect. II.7.2).

II.7.1 Self-calibration principle

In the self-calibration procedures of phase-closure imaging, a model of the object-source distribution is used as calibrator. By performing alternate phase-calibration operations and Fourier-synthesis processes, this model is iteratively refined; see Fig. II.10.

At qth iteration of the self-calibration procedure, the phase-calibration operation uses as input [see Eq. (II.9)]

1) the experimental complex visibility function:
$V_{\mathrm{d}} = \rho_{\mathrm{d}}\, e^{\mathrm{i}\varphi_{\mathrm{d}}}$;

2) the complex visibility function of the current model:
$V_{\mathrm{m}}^{(q)} = \rho_{\mathrm{m}}\, e^{\mathrm{i}\varphi_{\mathrm{m}}^{(q)}}$.

We then set [see Eq. (II.10)]

$$\varphi^{(q)} \overset{\text{def}}{=} \varphi_{\mathrm{d}} - \varphi_{\mathrm{m}}^{(q)} \qquad (\text{modulo } 2\pi) \tag{II.132}$$

and search for the global minimizer $\check{\alpha}^{(q)}$ of the functional [see Eq. (II.78)]

$$g^{(q)}(\alpha) \overset{\text{def}}{=} \|\arc(\widehat{\varphi}_{\mathrm{c}}^{(q)} - B\alpha)\|_w^2 \tag{II.133}$$

where [see Eq. (II.75)]

$$\widehat{\varphi}_{\mathrm{c}}^{(q)} \overset{\text{def}}{=} \arc(\varphi_{\mathrm{c}}^{(q)}) \tag{II.134}$$

We thus obtain as output the calibrated visibility function [see Eqs. (II.84) and (II.85)]

$$V_{\mathrm{d}\star}^{(q)} = \rho_{\mathrm{d}}\, e^{\mathrm{i}\varphi_{\mathrm{d}\star}^{(q)}} \tag{II.135}$$

where

$$\begin{aligned} \varphi_{\mathrm{d}\star}^{(q)} &= \varphi_{\mathrm{d}} - B\alpha_{d\star}^{(q)} \\ &= \varphi_{\mathrm{m}} + \varepsilon_{\star}^{(q)} \end{aligned} \qquad (\text{modulo } 2\pi) \tag{II.136}$$

in which [see Eqs. (II.80) and (II.73)]

$$\alpha_{d\star}^{(q)} = \check{\alpha}^{(q)} + \alpha_{\varphi^{(q)}} \qquad (\text{modulo } 2\pi) \tag{II.137}$$

and [see Eq. (II.81)]

$$\varepsilon_{\star}^{(q)} \overset{\text{def}}{=} \arc(\widehat{\varphi}_{\mathrm{c}}^{(q)} - B\check{\alpha}^{(q)}) \tag{II.138}$$

Clearly, $\alpha_{d\star}^{(q)}$ is the estimate of the pupil-bias phase at iteration q, and $\varepsilon_{\star}^{(q)}$ is the corresponding phase-calibration residual; see Fig. II.10.

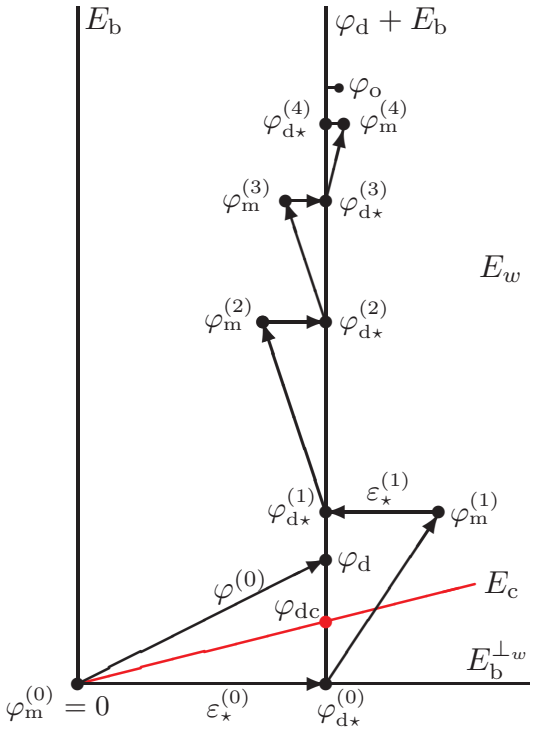

Figure II.10: Geometrical illustration of the self-calibration principle. Here, for clarity, we assume that the weighted baseline-phase space E_w does not change throughout the iterative self-calibration procedure; see Remark II.7.2.1. Furthermore, the initial model is assumed to be a point source. In this schematic representation, we consider the case where the reduced closure phases of $\varphi^{(q)} \stackrel{\rm def}{=} \varphi_{\rm d} - \varphi_{\rm m}^{(q)}$ are small in absolute value for all q; see Sects. II.5.4, II.7.2 and Fig. II.9. The phase-calibration operations correspond to the arrows parallel to $E_{\rm b}^{\perp w}$. The arrows from $\varphi_{\rm d\star}^{(q)}$ to $\varphi_{\rm m}^{(q+1)}$ correspond to the Fourier-synthesis processes. The procedure has practically converged when the closure component of $\varphi_{\rm m}^{(q)}$ is almost equal to that of $\varphi_{\rm d}$: $\varphi_{\rm mc}^{(q)} \simeq \varphi_{\rm dc}$ modulo 2π. As illustrated here, the calibrated phases $\varphi_{\rm d\star}^{(q)}$ have then converged to a phase point close to the projection of the object-visibility phase $\varphi_{\rm o}$ on $\varphi_{\rm d} + E_{\rm b}$; for further details see Sect. II.7.1.

The calibrated visibility function $V_{\rm d\star}^{(q)}$ is then used as data in the Fourier-synthesis process. A priori information on the object and related constraints are then introduced: object support, positivity, resolution level, etc. The model visibilities for the next iteration are then computed from the image thus obtained; see Fig. II.10.

As pointed out in Sect. II.1, the image provided by a self-calibration procedure can only be defined up to a translation. Throughout the procedure, to stabilize the solution, the model $s_{\rm m}^{(q)}(\boldsymbol{\xi})$ is usually centred on the origin. For example, in the initialization stage, in the absence of any *a priori* information on the object source, $s_{\rm m}^{(0)}(\boldsymbol{\xi})$ is defined as the distribution of a point source centred on the origin; $\varphi_{\rm m}^{(0)}$ is then zero all over $\mathscr{E}$.

As illustrated in Fig. II.10, the successive calibrated phases $\varphi_{\rm d\star}^{(q)}$ lie in the affine space $\varphi_{\rm d} + E_{\rm b}$: the vertical line passing through the phase-data point $\varphi_{\rm d}$. According to Eqs. (II.136) and (II.138), $\varphi_{\rm d\star}^{(q)}$ is the projection of $\varphi_{\rm m}^{(q)}$ on $\varphi_{\rm d} + E_{\rm b}$ in E_w (modulo 2π). Indeed, from Property 2, $\varepsilon_\star^{(q)}$ lies in $E_{\rm b}^{\perp w}$. As a result, $\varphi_{\rm d\star}^{(q)}$ can be regarded as the phase of a pseudo-model visibility function whose closure component would be equal to that of $\varphi_{\rm d}$ (modulo 2π). In general, in the first iterations of a self-calibration procedure, $\varphi_{\rm d\star}^{(q)}$ does not correspond to a

realistic object distribution satisfying, for example, the positivity constraint. The Fourier processes progressively correct for this anomaly, while thereby refining the $E_{\rm b}$-component of the phase of the visibility function of the model. The calibrated phases $\varphi_{{\rm d}\star}^{(q)}$ then converge to a phase point close to the projection of the object-visibility phase $\varphi_{\rm o}$ on $\varphi_{\rm d} + E_{\rm b}$; see Fig. II.10.

II.7.2 Selected method for phase calibration

The recommended approach for the phase-calibration operation derives from the special case examined in Sects. II.4.3 and II.5.4. At each iteration q of the self-calibration procedure, we then proceed as follows.

Stage 1. We first define the phase-calibration graph $\mathscr{G} \stackrel{\rm def}{=} (\mathscr{V}, \mathscr{E})$; see Fig. II.1 in Sect. II.1.1. On each baseline (i, j) of $\mathscr{G}$, we then consider, modulo 2π, the difference between the phases of $V_{\rm d}$ and $V_{\rm m}^{(q)}$: $\varphi \stackrel{\rm set}{=} \varphi^{(q)} \stackrel{\rm def}{=} \varphi_{\rm d} - \varphi_{\rm m}^{(q)}$; see Eqs. (II.132) and Fig. II.10.

Stage 2. As specified in Sect II.1.2, some weight $w(i, j)$ is allocated to each baseline of $\mathscr{G}$: Eq. (II.15). The spanning tree of maximal weight, $\mathscr{G}_{\rm st}$, is then built; see Sect. II.2.1 and Fig. II.4.

Stage 3. We then determine the pupil-phase function α_φ such that $B\alpha_\varphi$ coincides with φ on the baselines of $\mathscr{G}_{\rm st}$; see Eq. (II.73) and Sect. II.2.3. By construction, the function $\varphi_{\rm c} \stackrel{\rm def}{=} \varphi - B\alpha_\varphi$ vanishes on those baselines. As justified in Sect. II.2.4, the values of $\varphi_{\rm c}$ on the other baselines are the closure phases $\varphi_{\rm c}(i_\ell, j_\ell)$. We then consider the reduced closure-phase function $\widehat{\varphi}_{\rm c} \stackrel{\rm def}{=} {\rm arc}(\varphi_{\rm c})$. When $\widehat{\varphi}_{\rm c}$ is negligible (see Remark II.3.2.3), the self-calibration procedure has converged: the current image is the reconstructed image. The phase-calibration operation is then interrupted; see Fig. II.10. Otherwise :

Stage 4. For the robustness of the phase-calibration operation, the loop-entry baselines (i_ℓ, j_ℓ) on which $\widehat{\varphi}_{\rm c}$ is greater than or equal to $\pi/2$ in absolute value are discarded from the phase-calibration graph. Operator B is modified consequently. This idea, which considerably simplifies the problem, results from the analysis presented in Sects II.4.3 and II.5.4; see also Sect II.6.1. Clearly, we thus get rid of possible 'conflicting secondary minima;' see Sect II.6.2.

Stage 5. We then determine the pupil-phase function $\dot{\alpha}_0$ solution of Eq. (II.99):

$$B^* B\dot{\alpha}_0 = B^* \widehat{\varphi}_{\rm c}$$

Stage 6. Setting $\hat{v} \stackrel{\rm def}{=} \widehat{\varphi}_{\rm c}/(2\pi)$ [see Eq. (II.88)], we consider the ellipsoid $\mathfrak{E}(\mathfrak{c}_0)$ where $\mathfrak{c}_0 \stackrel{\rm def}{=} \|T_{\rm c}\hat{v}\|^2$. According to the check process described at the end of Sect. II.4.3, we then search in $\mathfrak{E}(\mathfrak{c}_0)$ for the point $\check{v}$ for which the minimum of $g(\dot{\alpha})$ is attained. We then check whether $\check{v} = \dot{v}_0$ or not; $\check{\alpha}$ is defined consequently; see Sect. II.4.2. The calibrated visibilities are then computed; see Sect. II.7.1 together with Remarks II.3.2.1 and II.3.2.2.

Remark II.7.2.1. The parameters of the phase-calibration operation strongly depend on the reliability of the current model. This is why, in the first iterations of the self-calibration procedure, when the model is far from the object, $\mathscr{G}$, w and thereby E_w may greatly change.

II.8 Concluding comments

In this chapter we have presented an analysis of the phase-calibration problem encountered in astronomy when mapping incoherent sources with aperture-synthesis devices. More precisely, we concentrated on the phase-calibration operation involved in the self-calibration procedures of phase-closure imaging. In radio-astronomy, the related optimization problems have been stated and solved hitherto at the phasor level. By conducting our analysis at the phase level, we derived a new method for diagnosing and solving the difficulties of the phasor approach. In the most general case, the techniques to be implemented appeal to the algebraic graph theory and the algebraic number theory.

II.8.1 Main points of the analysis

The key graph of our analysis is the phase-calibration graph $\mathscr{G} \stackrel{\text{def}}{=} (\mathscr{V}, \mathscr{E})$; see for example Fig. II.1; $\mathscr{V}$ is the set of pupil elements of $\mathscr{G}$, i.e., the set of its vertices; $\mathscr{E}$ is the set of baselines of $\mathscr{G}$, i.e., the set of its edges. As specified in Sect II.1.2, some weight can be allocated to each baseline of $\mathscr{G}$. The spanning tree (of $\mathscr{G}$) of maximal weight, $\mathscr{G}_{\text{st}} \stackrel{\text{def}}{=} (\mathscr{V}, \mathscr{E}_{\text{st}})$, is then built; see Sect. II.2.1 and Fig. II.4. The baselines of $\mathscr{E}$ that do not lie in $\mathscr{E}_{\text{st}}$ define as many loops. The notion of closure is basically associated with these loops. In some configurations, to benefit from all the available information, closure phases of order greater than three are thus introduced; see Figs. II.4 and II.6. These closure phases can be computed without having to identify the baselines of their loops; see Property 1 and Eq. (II.40) in Sect. II.2.3.

We have shown in Sect. II.3.1 that the phase-calibration problem can be reformulated in phase-closure terms. The main entries of the problem are then the reduced closure phases $\widehat{\varphi}_{\text{c}}(i_\ell, j_\ell)$ for $\ell = 1, \ldots, n_{\text{c}}$; see Eq. (II.75), and Sects. II.2.3 to II.2.5. To make the link with similar problems encountered in GNSS (see Lannes & Teunissen (2011)), the vector of $\mathbb{R}^{n_{\text{c}}}$ with components $\widehat{\varphi}_{\text{c}}(i_\ell, j_\ell)/(2\pi)$ is denoted by $\hat{v}$ [Eq. (II.88)].

The functionals to be minimized may have several minima. The best way to overcome this difficulty is to solve the problem at the arc level via the arc functional g [Eq. (II.78)]. Indeed, according to Property 2, the minimizers $\dot{\alpha}$ of $g(\alpha)$ are associated with some particular points $\dot{v}$ of lattice $\mathbb{Z}^{n_{\text{c}}}$. These points, which lie in some ellipsoid centred on $\hat{v}$ in $\mathbb{R}^{n_{\text{c}}}$, can be easily listed; see Sect. II.4.2. The minima of g can therefore be identified in a systematic manner. We thus have $g(\dot{\alpha}_1) < g(\dot{\alpha}_2) < g(\dot{\alpha}_3)$, and so on.

The global minimizer of g, $\check{\alpha} \stackrel{\text{set}}{=} \dot{\alpha}_1$, is associated with the point $\check{v} \stackrel{\text{set}}{=} \dot{v}_1$ of $\mathbb{Z}^{n_{\text{c}}}$ closest to $\hat{v}$, the distance being that induced by a quadratic form whose matrix is the inverse of the variance-covariance matrix of the closure phases.

As a general rule, when all the reduced closure phases are less than $\pi/2$ (in absolute value), $\check{\alpha}$ is associated with the point zero of $\mathbb{Z}^{n_{\text{c}}}$: $\dot{v}_0$; see however the *check point* of Sect. II.4.3. The calibration pupil phase $\check{\alpha}$ is then obtained as the solution of the normal equation (II.99). Furthermore, the global minimum and the nearest secondary minimum (if any) are then often well distinct; see Sect. II.6.1. In this case, the phase-calibration operation is reliable; it is also very simple. The strategy proposed in Sect. II.7 derives from that analysis.

The minima of the chord functional f [Eq. (II.77)] can be obtained from those of g. As clarified through Sects. II.5 and II.6, all the minima of f can thus be exhibited too. A minimum of f may be tightly linked to a minimum of g; see Sects. II.5.3 and II.6. These

pairs of linked minima therefore play an important part in the understanding of the matter.

As specified in Sect. II.5, in the general case, the search for the global minimum of the chord functional f can be performed via the trust-region method by using $\check{\alpha}$ as starting point. In the important special case where all the reduced closure phases are less than $\pi/2$ (in absolute value), the pupil-phase function $\alpha_0 \stackrel{\text{def}}{=} 0$ proves to be a good starting point for the trust-region method which then reduces to the Newton-Raphson algorithm. This algorithm then yields, in a few iterations, a solution $[\alpha_0]_f$ very close to $\check{\alpha} = \dot{\alpha}_0$; see Fig. II.9 and Sect. II.6.1. The arc approach is however still preferable. Indeed, $\dot{\alpha}_0$ is then obtained by solving a single equation [Eq. (II.99)], while the determination of $[\alpha_0]_f$ requires solving a few equations of the same type.

When some reduced closure phases are greater than or equal to $\pi/2$ in absolute value, the phase-calibration operation must be conducted with much care. As illustrated in Sect. II.6.2, the point zero of $\mathbb{Z}^{n_c}$ may not even be associated with a minimizer of g. In particular, the situations where the secondary minimum $g(\dot{\alpha}_2)$ is of the order of $g(\dot{\alpha}_1)$ with $\varepsilon_{\dot{\alpha}_2}$ very different from $\varepsilon_{\dot{\alpha}_1}$ are to be discarded; see Sect. II.6.2. With the aid of the techniques presented in this chapter, these delicate situations can be diagnosed and dealt with. To regularize these situations, the phase-calibration graph can be truncated by removing the loop-entry baselines for which the reduced closure phases are too large (in absolute value). It is thus possible to find a good compromise between the coverage of the phase-calibration graph, which must be as complete as possible, and the quality of the solution which must of course be reliable.

II.8.2 Discussion on phase-closure imaging

According to Property 1, with regard to a given spanning tree of $\mathscr{G}$, any baseline-phase function β can be decomposed in the form $\beta = \beta_b + \beta_c$; see Fig. II.5. Here, $\beta_b(i,j) = \alpha_\beta(i) - \alpha_\beta(j)$, where α_β is the pupil-phase function defined from β by Eq. (II.35). The other component is the closure-phase function β_c. This function is characterized by n_c components, the closure phases $\beta_c(i_\ell, j_\ell)$. The latter are associated with loops whose order may be greater than three; see Sects. II.2.4 through II.2.6.

Our algebraic approach thus allowed us to exhibit, in any baseline-phase function β, a component that cannot be perturbated by pupil-bias phases: the closure component β_c. We have thus been able to formulate the phase-calibration problem at the baseline-phase level, while benefiting from the property of insensibility of the closure phases to the pupil-bias phases; see Sect. II.3.1, and in particular, Eq. (II.78) and Remark II.3.2.1.

This approach is particularly well suited to the image restoration methods that handle, separately, the phase-calibration and Fourier-synthesis problems; see, e.g., Cornwell & Wilkinson (1981). The model can thus be progressively refined; see Fig. II.10. As specified in Sect. II.7.1, the phase-calibration operations try imposing the appropriate E_c-constraint on the model-visibility phase, whereas the Fourier-synthesis processes adjust its E_b-component. In our formulation of the phase-calibration problem at the phase level, those two aspects are well disentangled. Indeed, our formulation is based on the representation of the baseline-phase space E as the direct sum of E_b and E_c; see Fig. II.5.

Different approaches can be envisaged. For example, consider the image-reconstruction problem in which the following information is to be processed:

– a maximum set of independent closure-phase data of order three;

– the moduli of the visibility data;

– some object model defined *a priori*.

By using appropriate optimization techniques, one could try solving this problem without separating phase calibration and Fourier synthesis. The drawback of such a global approach is that the complexity of phase-closure imaging is then hidden. The potential instabilities of the phase problem remain, as well as those of the Fourier-synthesis process; see, e.g., Lannes et al. (1997). In this context, our algebraic approach could be used as a tool for analyzing the difficulties related to the phase, and providing a diagnostic on the stability of the problem. For example, the closure information provided by a maximum set of independent closure phases of order three may be incomplete; see Sect. II.2.6.

In summary, our approach leads to a better understanding of phase-closure imaging. In the case of the self-calibration approach, we have proposed a method for solving the related problems. The usual applicability domain of the corresponding image-reconstruction methods can thereby be extended. This should therefore be particularly useful for ground-based interferometric arrays working with severe turbulence problems encountered for instance with sub-millimeter or optical wavelengths.

Chapter III

Integer-ambiguity resolution in astronomy and geodesy

Astronomical images obtained from ground-based observatories are degraded by atmospheric turbulence. In particular, the phase of the Fourier transform of the object-source distribution is severely perturbed which leads to a significant loss of angular resolution in the resulting images. Thanks to the theoretical and technical developments of the last half century, large interferometric arrays circumvent this difficulty in radio-astronomy, and now routinely provide sharp-edged images with a very high angular resolution.

One of the methods used for obtaining those nice results is 'self-calibration.' In the most general case, the vectorial nature of the electromagnetic field must be taken into account in the very formulation of the problem; see Hamaker (2000) and references therein. In this chapter, we however restrict ourselves to scalar self-calibration (i.e. we do not take polarimetry into account). We thus refer to the same framework as that defined in Sect. II. In particular, we show that in the scalar case, the phase-calibration problem has a close similarity with the calibration problems encountered in high-precision geodetic positioning and in global navigation satellite systems (GNSS).

In fact, the approach we propose for solving the phase-calibration problem in the scalar case is a good starting point for tackling the more complex problem of full polarimetric phase calibration. This possible extension however deserves a particular analysis which goes beyond the scope of our present work. Some guiding ideas for the corresponding 'matrix self-calibration' approach are to be found in Hamaker (2000) and Yatawatta (2012). The scalar case presented in Sect. III.1.1 has already its own complexity. Any vectorial analysis should therefore start from a good understanding of that analysis.

In Sect. II, we have analysed the self-calibration procedure in the scalar case. In that special case, we have proposed a new approach to the problem: the 'arc-approach.' The final step of that approach consists in solving a nearest-lattice-point (NLP) problem; for a precise definition of this problem, see Sect. III.2.1.

In fact, NLP problems appear in many fields of applied mathematics. In particular, as already mentioned, they play a central role in high-precision geodetic positioning and in GNSS (see Sect. IV). In this chapter, we present new methods for solving those NLP problems. These methods can therefore be applied both in astronomy and geodesy.

The standard way of solving an NLP problem includes two stages: a preconditioning stage, and a discrete-search stage in which the integer ambiguities are finally fixed. The prob-

lem is usually preconditioned by implementing the algorithm introduced by Lenstra, Lenstra & Lovász (1982): the LLL algorithm. The LAMBDA decorrelation method of Teunissen (1995) can also be used for this purpose; for the theoretical link between LLL-reduction and LAMBDA-decorrelation, see Lannes (2013). The NLP problem is then solved in the reduced basis thus obtained. This is done via appropriate discrete-search techniques. In this general context, we present our implementation of the LLL algorithm, as well as our discrete-search techniques. In this chapter we thus revisit and complete the analysis of Sect. C, that was presented in Lannes & Prieur (2011). With regard to the current state of the art (see, e.g., Agrell et al. (2002); Jazaeri et al. (2012)), the methods described in this chapter lead to a speed-up of the order of two.

In Sect. III.1, we show how those problems appear in astronomy and geodesy. The main guidelines of our study are presented in Sect. III.2. Some basic notions are then defined among which that of LLL-reduced basis. In Sect. III.3, we then describe an LLL-type algorithm allowing an LLL-reduced basis to be built. Section III.4 is devoted to the main contribution presented here: the discrete-search techniques to be implemented for finding the nearest lattice point in the selected reduced basis. We also describe the techniques to be used for identifying the points lying in some neighbourhood of the nearest lattice point. Indeed, these points are also useful for the analysis of the related problems. The computational issues of our contribution and its main results are summarized in Sects. III.5 and V.

III.1 NLP problems in astronomy and geodesy

We here present some NLP problems encountered in astronomy (Sect. III.1.1) and geodesy (Sect. III.1.2). The similarities between the scalar case presented in Sect. III.1.1, and the global positioning problems of Sect. III.1.2 are thus explicitly exhibited.

III.1.1 Self-calibration procedures in phase-closure imaging

When mapping incoherent sources with aperture-synthesis devices, the pupil-phase perturbations (hereafter pupil-phase biases) caused by the atmospheric turbulence degrade the angular resolution of the restored image. A standard way for obtaining high angular-resolution images is to estimate those pupil-phase biases from observations of a calibrator (usually a reference star). However when the turbulence is strong and quickly varies with time, this procedure is not possible. A way out is to use 'self-calibration' which corresponds to the situation where the object source to be imaged plays the role of the calibrator. Following the pioneering work of Cornwell & Wilkinson (1981) in the scalar case, this problem can then be solved by alternate phase-calibration operations and Fourier-synthesis processes. However, this procedure is generally rather unstable. To ensure the reliability and the robustness of those techniques, the phase-calibration operations must then be conducted with much care.

The model of the object source is refined throughout the iterative self-calibration procedure. At each iteration, the phase-calibration operation consists in estimating virtual pupil-phase biases $\alpha_{\rm d}(i)$ so that the following equation is satisfied in a least-squares sense to be defined:

$$\exp \mathrm{i}\varphi_{\rm d}(i,j)\ \exp -\mathrm{i}[\alpha_{\rm d}(i)-\alpha_{\rm d}(j)] = \exp \mathrm{i}\varphi_{\rm m}(i,j) \tag{III.1}$$

Here, $\exp \mathrm{i}\varphi_\mathrm{d}$ and $\exp \mathrm{i}\varphi_\mathrm{m}$ are the 'phasors' of the (complex) 'visibility functions' of the data and the model, respectively. The pairs (i, j), which define the edges of the 'phase-calibration graph' $\mathscr{G}$, correspond to the baselines of the interferometric device; for further details, see Sect. II.1. The self-calibration procedure aims at reducing the phase discrepancy

$$\varphi \stackrel{\text{def}}{=} \varphi_\mathrm{d} - \varphi_\mathrm{m} \tag{III.2}$$

From Eq. (III.1), we infer that the phase-calibration operation consists in finding a function α_d such that the following relationship be valid up to error terms:

$$\varphi(i, j) - [\alpha_\mathrm{d}(i) - \alpha_\mathrm{d}(j)] = 2\pi N(i, j) \tag{III.3}$$

with $N(i, j)$ in $\mathbb{Z}$. In radio-astronomy, the related optimization problems are generally solved at the phasor level: one minimizes the size of the chords associated with the phasors

$$\exp \mathrm{i}\{\varphi(i, j) - [\alpha_\mathrm{d}(i) - \alpha_\mathrm{d}(j)]\}$$

In some critical situations, the 'chord functional' may have several minima. As shown in Lannes (2005), and Lannes & Prieur (2011) (see in Sect. II), the analysis of the problem must then be conducted at the phase level. We then consider the size of the quantities

$$\mathrm{arc}\{\varphi(i, j) - [\alpha_\mathrm{d}(i) - \alpha_\mathrm{d}(j)]\}$$

where function arc is defined as follows:

$$\mathrm{arc}(\theta) \stackrel{\text{def}}{=} \theta - 2\pi \left\lfloor \frac{\theta}{2\pi} \right\rceil$$

Here, $\lfloor x \rceil$ denotes the integer of $\mathbb{Z}$ closest to x; when $x = k + 1/2$ for some k in $\mathbb{Z}$, $\lfloor x \rceil$ is set equal to k. The functional to be minimized is then of the form

$$g_\mathrm{d}(\alpha_\mathrm{d}) \stackrel{\text{def}}{=} \| \mathrm{arc}(\varphi - B\alpha_\mathrm{d}) \|_w \tag{III.4}$$

where

$$(B\alpha_\mathrm{d})(i, j) \stackrel{\text{def}}{=} \alpha_\mathrm{d}(i) - \alpha_\mathrm{d}(j) \tag{III.5}$$

with $\alpha_\mathrm{d}(1) = 0$ for instance; the norm $\| \cdot \|_w$ is defined as specified in Sect. II.1.1.

As explicitly shown in Sects. II.4, II.5 and II.6, the arc approach gives a better insight into the problem. The corresponding theoretical framework appeals both to algebraic graph theory (Biggs, 1996) and algebraic number theory (Cohen, 1996). We now give a survey of the matter which shows how those two main features are tightly imbricated.

The notion of 'phase closure,' which underlies the concept of 'phase-closure imaging' (PCI), is introduced in a context more general than that usually defined in radio imaging and optical interferometry. In particular, closure phases of order larger than three may then be defined. According to our algebraic-graph analysis, the data-model discrepancy can be decomposed in the form (see Sect. II.2.3):

$$\varphi = \varphi_\mathrm{b} + \varphi_\mathrm{c}$$

Here, the baseline-bias function φ_{b} is equal to $B\alpha^{(\varphi)}$ for some $\alpha^{(\varphi)}$ (depending on φ). The function φ_{c} is the 'closure function' of φ; it takes its values on the n_{c} 'closure edges' of $\mathscr{G}$, the 'loop-entry baselines' of the problem; see for example Figs. II.3 and II.4.

Clearly, $\varphi - B\alpha_{\mathrm{d}} = \varphi_{\mathrm{c}} - B(\alpha_{\mathrm{d}} - \alpha^{(\varphi)})$. It then follows from Eq. (III.4) that $g_{\mathrm{d}}(\alpha_{\mathrm{d}})$ is equal to $g(\alpha)$ where

$$g(\alpha) \stackrel{\text{def}}{=} ||\operatorname{arc}(\hat{\varphi}_{\mathrm{c}} - B\alpha)||_w \tag{III.6}$$

with

$$\alpha \stackrel{\text{def}}{=} \alpha_{\mathrm{d}} - \alpha^{(\varphi)} \quad \text{and} \quad \hat{\varphi}_{\mathrm{c}} \stackrel{\text{def}}{=} \operatorname{arc}(\varphi_{\mathrm{c}}) \tag{III.7}$$

The minimizers of g_{d} can therefore be easily deduced from those of g.

Divided by 2π, $\hat{\varphi}_{\mathrm{c}}$ defines some point $\hat{\mathbf{v}}$ of $\mathbb{R}^{n_{\mathrm{c}}}$. We have shown that the minima of the arc functional g are determined via particular 'integer sets' associated with $\hat{\mathbf{v}}$. Those integer sets correspond to some particular points $\dot{\mathbf{v}}$ of lattice $\mathbb{Z}^{n_{\mathrm{c}}}$; see Property II.4.1. In that algebraic-number framework, finding the global minimizer of g (and thereby that of g_{d}) amounts to finding the point $\check{\mathbf{v}}$ of $\mathbb{Z}^{n_{\mathrm{c}}}$ closest to $\hat{\mathbf{v}}$ with regard to some distance; that distance is defined via some quadratic form whose matrix $\mathbf{Q}$ is the inverse of the variance-covariance matrix $\mathbf{V}$ of $\hat{\mathbf{v}}$. As explicitly clarified in Sect. III.2.1, finding the global minimum therefore amounts to solving a NLP problem in which $\hat{\mathbf{v}}$ can be regarded as its 'float solution.' The main secondary minima of g, if any, correspond to $\mathbb{Z}^{n_{\mathrm{c}}}$-lattice points in some neighbourhood of $\check{\mathbf{v}}$. Like for $\hat{\mathbf{v}}$, those points can be identified, in a systematic manner, via the integer-programming techniques presented here.

III.1.2 High-precision geodetic positioning

The techniques involved in high-precision geodetic positioning and global navigation satellite systems (GNSS) are based on two types of data: the (carrier-)phase and code (or pseudo-range) observations; see, e.g., Lannes & Gratton (2009), Lannes & Teunissen (2011). The phase observational equations of GNSS networks are of the form

$$\left|\begin{array}{l} b_\kappa(i,j) - [\beta_{\mathrm{r}\kappa}(i) - \beta_{\mathrm{s}\kappa}(j)] = N(i,j) \\ \text{for } \kappa = 1, \ldots, k \end{array}\right. \tag{III.8}$$

In those problems, κ is the epoch index; k is the index of the current epoch; $\beta_{\mathrm{r}\kappa}(i)$ and $\beta_{\mathrm{s}\kappa}(j)$ are clock-phase biases. Those biases, which are expressed in cycles, depend on the frequency of the transmitted carrier wave; subscripts r and s stand for receiver and satellite,[1] respectively; i is the index of the receiver, and j that of the satellite; $N(i,j)$ is the integer ambiguity of the corresponding carrier-phase measurement. The terms $b_\kappa(i,j)$ include the corresponding phase data and the contributions associated with the real variables of the problem other than the clock-phase biases: position and atmospheric parameters, for instance; see, e.g., de Jonge (1998); Lannes & Teunissen (2011). The set of receiver-satellite pairs (i,j) involved in Eq. (III.8) forms the observational graph $\mathscr{H}_\kappa$ of the GNSS scenario of epoch κ. Owing to the particular structure of the phase equations (III.8), the problem has a basic rank

[1] Here, satellite should be understood as satellite transmitter.

defect. As outlied below, the latter can be eliminated by an appropriate redefinition of its variables.

In the system of Eqs. (III.8), the GNSS functional N takes its values on $\mathscr{G}$, the union of the graphs $\mathscr{H}_\kappa$ until the current epoch k. The similarity of Eqs. (III.3) and (III.8) was first pointed out by Lannes & Teunissen (2011). In Lannes & Prieur (2013), we were therefore led to propose for N a decomposition quite similar to that of φ in Sect. III.1.1: $N = N_\mathrm{b} + N_\mathrm{c}$ with $N_\mathrm{b} = B\mu^{(N)}$. Here, $\mu^{(N)}$ is an integer-valued function depending on N; $\mu^{(N)}$ takes its values on the vertices of $\mathscr{G}$ other than the reference receiver r_1 (for example). The corresponding 'integer variable' $\mathbf{v} := N_\mathrm{c}$ lies in $\mathbb{Z}^{n_\mathrm{c}}$ where n_c is the number of closure edges of $\mathscr{G}$. The redefined clock-phase biases are then of the form $\beta_{\mathrm{r}\kappa}(i) + \mu_\mathrm{r}^{(N)}(i)$ (for $i \neq 1$) and $\beta_{\mathrm{s}\kappa}(j) + \mu_\mathrm{s}^{(N)}(j)$.

In a first stage, at each epoch k, the problem is solved in the LS sense by considering $\mathbf{v}$ as a 'float variable.' A float solution $\hat{\mathbf{v}}$ is thus obtained and updated progressively. In practice, this is done via recursive QR-factorization (see Sect. I). The ambiguity solution $\check{\mathbf{v}}$ is then the point of $\mathbb{Z}^{n_\mathrm{c}}$ closest to $\hat{\mathbf{v}}$ with regard to some distance. Like in PCI, that distance is defined via some quadratic form whose matrix $\mathbf{Q}$ is the inverse of the variance-covariance matrix $\mathbf{V}$ of the float solution $\hat{\mathbf{v}}$. In that case, the points of $\mathbb{Z}^{n_\mathrm{c}}$ lying in some neighbourhood of $\check{\mathbf{v}}$ are involved in the corresponding validation techniques; see Verhagen & Teunissen (2006). Again, like in PCI, those points can be identified, in a systematic manner, via the integer-programming techniques presented here. Once $\check{\mathbf{v}}$ has been fixed and validated, the real variables, among which the redefined clock-phase biases, are then estimated accordingly.

III.2 Guidelines

This chapter is essentially devoted to the methods to be used for solving the NLP problems encountered in astronomy and geodesy. Setting $n := n_\mathrm{c}$, we first define these problems as follows.

III.2.1 NLP problems

Given some vector $\hat{\mathbf{v}}$ of $\mathbb{R}^n$, consider the (or a) vector $\check{\mathbf{v}}$ of $\mathbb{Z}^n$ such that

$$\check{\mathbf{v}} = \underset{\mathbf{v}\in\mathbb{Z}^n}{\operatorname{argmin}} \|\mathbf{v} - \hat{\mathbf{v}}\|^2_{\mathbf{Q}} \tag{III.9}$$

The norm introduced here is that of $(\mathbb{R}^n, \mathbf{Q})$: the space $\mathbb{R}^n$ endowed with the inner product

$$(\mathbf{v} \mid \mathbf{v}')_{\mathbf{Q}} \overset{\text{def}}{=} (\mathbf{v} \cdot \mathbf{Q}\mathbf{v}') \tag{III.10}$$

where $\mathbf{Q}$ is the inverse of the variance-covariance matrix of the 'float solution' $\hat{\mathbf{v}}$: $\mathbf{Q} \overset{\text{def}}{=} \mathbf{V}^{-1}$. Clearly, $(\,\cdot\,)$ is the Euclidean inner product of $\mathbb{R}^n$. In matrix terms, we therefore have

$$(\mathbf{v} \mid \mathbf{v}')_{\mathbf{Q}} = \mathbf{v}^{\mathrm{T}}\mathbf{Q}\,\mathbf{v}' \tag{III.11}$$

All the quantities appearing in these equations are expressed in the standard basis

$$\{\mathfrak{e}_1, \mathfrak{e}_2, \cdots, \mathfrak{e}_n\}$$

of $\mathbb{R}^n$ and $\mathbb{Z}^n$. Note that this basis can be represented by the row matrix

$$\mathbf{B} \stackrel{\text{def}}{=} [\mathfrak{e}_1 \quad \mathfrak{e}_2 \quad \cdots \quad \mathfrak{e}_n] \tag{III.12}$$

whose entries are the vectors $\mathfrak{e}_j$ for $j = 1, \ldots, n$.

The integer lattice $\mathbb{Z}^n$ regarded as a subset of $(\mathbb{R}^n, \mathbf{Q})$ is denoted by $(\mathbb{Z}^n, \mathbf{Q})$; $\check{\mathbf{v}}$ is therefore a nearest lattice point to $\hat{\mathbf{v}}$ in $(\mathbb{Z}^n, \mathbf{Q})$. Equation (III.9) therefore defines an NLP problem.

III.2.2 Factorizations of Q

In this chapter, we write the Cholesky factorization of $\mathbf{Q}$ in the form

$$\mathbf{Q} = \mathbf{R}^{\mathrm{T}}\mathbf{R} \tag{III.13}$$

where $\mathbf{R}$ is an upper-triangular matrix. Denoting by $\|\cdot\|$ the Euclidean norm of $\mathbb{R}^n$, we therefore have, from Eqs. (III.9) and (III.11),

$$\check{\mathbf{v}} = \underset{\mathbf{v}\in\mathbb{Z}^n}{\operatorname{argmin}} \|\mathbf{R}(\mathbf{v} - \hat{\mathbf{v}})\|^2 \tag{III.14}$$

Let $\mathbf{D}$ be the diagonal matrix defined via the relation

$$\mathbf{R} = \mathbf{D}^{1/2}\mathbf{U} \tag{III.15}$$

where $\mathbf{U}$ is an upper-triangular matrix whose diagonal elements $\mathrm{u}_{j,j}$ are equal to unity. For clarity, the diagonal entries of $\mathbf{D}$ will be denoted by $\mathfrak{d}_j$. From Eq. (III.13), we have

$$\mathbf{Q} = \mathbf{U}^{\mathrm{T}}\mathbf{D}\,\mathbf{U} \tag{III.16}$$

III.2.3 Q-Orthogonality defect

Any basis of $\mathbb{Z}^n$ is characterized by a row matrix of the form

$$\boldsymbol{B} \stackrel{\text{def}}{=} [\boldsymbol{e}_1 \quad \boldsymbol{e}_2 \quad \cdots \quad \boldsymbol{e}_n] \tag{III.17}$$

In general, such a basis is far from being $\mathbf{Q}$-orthogonal; see Eq. (III.10). To provide a measure of this defect, we introduce the following notion.

DEFINITION 3.1. The parameter

$$\delta_{\mathbf{Q}}(\boldsymbol{B}) \stackrel{\text{def}}{=} \left(\frac{\prod_{j=1}^n \boldsymbol{e}_j^{\mathrm{T}}\mathbf{Q}\boldsymbol{e}_j}{\det \mathbf{Q}} \right)^{1/(2n)} \tag{III.18}$$

is the 'dilute $\mathbf{Q}$-orthogonality defect' of $\boldsymbol{B}$ $\boxdot$

In the notation adopted in Eq. (III.18), $\boldsymbol{e}_j$ denotes the column matrix whose entries are the components of the corresponding vector in the standard basis. Those entries therefore lie in $\mathbb{Z}$. Clearly, $\det \mathbf{Q}$ is the determinant of $\mathbf{Q}$.

According to Eqs. (III.18) and (III.13),

$$\delta_{\mathbf{Q}}(\boldsymbol{B}) = \left(\frac{\prod_{j=1}^n \|\boldsymbol{b}_j\|}{\det \mathbf{R}} \right)^{1/n} \qquad \boldsymbol{b}_j \stackrel{\text{def}}{=} \mathbf{R}\boldsymbol{e}_j \tag{III.19}$$

This relation shows that $\delta_{\mathbf{Q}}(\boldsymbol{B})$ is the 'dilute Euclidean-orthogonality defect' of the basis $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n\}$. It can be shown that $\det \mathbf{R}$ is the volume of the n-dimensional parallelepiped defined by these vectors. Clearly, $\delta_{\mathbf{Q}}(\boldsymbol{B})$ is greater than or equal to 1, the zero defect corresponding to the case where $\delta_{\mathbf{Q}}(\boldsymbol{B}) = 1$.

The matrix $\boldsymbol{M}$ whose columns are the column matrices $\boldsymbol{e}_j$ of Eq. (III.18) is unimodular: $\boldsymbol{M}$ is an integer n-by-n matrix whose determinant is equal to ± 1. The matrix relation

$$\boldsymbol{B} = \mathbf{B}\boldsymbol{M} \tag{III.20}$$

gathers the vector relations

$$\boldsymbol{e}_j = \sum_{i=1}^{n} m_{ij}\, \mathfrak{e}_i \qquad (\text{for } j = 1, \ldots, n)$$

Clearly, the integers $m_{i,j}$ are the entries of $\boldsymbol{M}$. In the same way as $\boldsymbol{M}$ is associated with $\boldsymbol{B}$, the identity matrix $\mathbf{I}_n$ is associated with $\mathbf{B}$. In terms of matrices, we have $\boldsymbol{e}_j = \boldsymbol{M}\mathfrak{e}_j$, hence (from Eq. (III.18))

$$\delta_{\mathbf{Q}}(\boldsymbol{B}) = \left(\frac{\prod_{j=1}^{n} \mathfrak{e}_j^{\mathrm{T}} \boldsymbol{Q}\, \mathfrak{e}_j}{\det \mathbf{Q}} \right)^{1/(2n)} \tag{III.21}$$

where

$$\boldsymbol{Q} \stackrel{\text{def}}{=} \boldsymbol{M}^{\mathrm{T}} \mathbf{Q} \boldsymbol{M} \tag{III.22}$$

Note that $\det \boldsymbol{Q} = \det \mathbf{Q} = (\det \mathbf{R})^2$. To compute $\delta_{\mathbf{Q}}(\boldsymbol{B})$, one is led to consider the factorization

$$\boldsymbol{Q} = \boldsymbol{U}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{U} \tag{III.23}$$

where $\boldsymbol{U}$ is an upper-triangular matrix whose diagonal elements $u_{j,j}$ are equal to unity; $\delta_{\mathbf{Q}}(\boldsymbol{B})$ is then obtained via the logarithmic formula

$$\ln\big(\delta_{\mathbf{Q}}(\boldsymbol{B})\big) = \frac{1}{2n} \sum_{j=2}^{n} \ln\Big(1 + \sum_{i=1}^{j-1} \frac{d_i}{d_j} u_{i,j}^2\Big) \tag{III.24}$$

where the d_j's are the diagonal entries of $\boldsymbol{D}$. Note that

$$\ln\big(\delta_{\mathbf{Q}}(\mathbf{B})\big) = \frac{1}{2n} \sum_{j=2}^{n} \ln\Big(1 + \sum_{i=1}^{j-1} \frac{\mathfrak{d}_i}{\mathfrak{d}_j} \mathrm{u}_{i,j}^2\Big) \tag{III.25}$$

As $\boldsymbol{Q} \stackrel{\text{def}}{=} \boldsymbol{M}^{\mathrm{T}} \mathbf{Q} \boldsymbol{M}$ (Eq. (III.22)), $\delta_{\mathbf{Q}}(\boldsymbol{B})$ can also be regarded as the 'reduction defect' of $\mathbf{Q}$ in basis $\boldsymbol{B}$, or in a more concise manner, as the reduction defect of $\boldsymbol{Q}$.

In what follows, the guiding idea is to choose $\boldsymbol{M}$ so that $\delta_{\mathbf{Q}}(\mathbf{B})$ be reduced somehow: $\delta_{\mathbf{Q}}(\boldsymbol{B}) < \delta_{\mathbf{Q}}(\mathbf{B})$. The notion of reduced basis introduced by Lenstra, Lenstra & Lovász (1982) was a key step in that direction.

III.2.4 LLL-reduced basis

DEFINITION 3.2. The column vectors $\boldsymbol{e}_j$ of $\boldsymbol{M}$ define an LLL-reduced basis of $(\mathbb{Z}^n, \mathbf{Q})$ if the matrix elements of $\boldsymbol{U}$ and $\boldsymbol{D}$ in factorization (III.23) satisfy the conditions

$$|u_{i,j}| \leq \frac{1}{2} \quad \text{for} \quad 1 \leq i < j \leq n \tag{III.26}$$

and

$$d_j \geq (\omega - u_{j-1,j}^2) d_{j-1} \quad \text{for} \quad 2 \leq j \leq n \tag{III.27}$$

with $1/4 < \omega < 1$ ⊡

Condition (III.26) reduces $\delta_{\mathbf{Q}}(\mathbf{B})$ by reducing the size of the matrix elements $\mathrm{u}_{i,j}$; see Eqs. (III.25) and (III.24). Condition (III.27) requires the d_j's be loosely sorted in increasing order with no distinctive discontinuity; the ratios d_i/d_j (for $i < j$) are then made as small as 'LLL$_\omega$-possible.'

III.2.5 Statement of the NLP problem in the reduced basis

To complete Sect. III.2.1, we now state the NLP problem (III.9) in the selected reduced basis $\boldsymbol{B}$; see the context of Eq (III.20). Clearly, $\|\mathrm{v} - \hat{\mathrm{v}}\|_{\mathbf{Q}}^2 = \left\|\boldsymbol{M}\left[\boldsymbol{M}^{-1}(\mathrm{v} - \hat{\mathrm{v}})\right]\right\|_{\mathbf{Q}}^2$. Setting

$$\boldsymbol{v} \stackrel{\text{def}}{=} \boldsymbol{M}^{-1}\mathrm{v} \qquad \hat{\boldsymbol{v}} \stackrel{\text{def}}{=} \boldsymbol{M}^{-1}\hat{\mathrm{v}} \tag{III.28}$$

we therefore have

$$\begin{aligned} \|\mathrm{v} - \hat{\mathrm{v}}\|_{\mathbf{Q}}^2 &= \left\|\boldsymbol{M}(\boldsymbol{v} - \hat{\boldsymbol{v}})\right\|_{\mathbf{Q}}^2 \\ &= [\boldsymbol{v} - \hat{\boldsymbol{v}}]^{\mathrm{T}} \boldsymbol{M}^{\mathrm{T}} \mathbf{Q} \boldsymbol{M} [\boldsymbol{v} - \hat{\boldsymbol{v}}] \end{aligned}$$

It then follows that

$$\|\mathrm{v} - \hat{\mathrm{v}}\|_{\mathbf{Q}}^2 = q(\boldsymbol{v}) \tag{III.29}$$

where, from Eq. (III.23),

$$q(\boldsymbol{v}) \stackrel{\text{def}}{=} \left\|\boldsymbol{D}^{1/2}\boldsymbol{U}\left(\boldsymbol{v} - \hat{\boldsymbol{v}}\right)\right\|^2 \tag{III.30}$$

Let $\check{\boldsymbol{v}}$ now be a vector of $\mathbb{Z}^n$ minimizing $q(\boldsymbol{v})$:

$$\check{\boldsymbol{v}} = \underset{\boldsymbol{v} \in \mathbb{Z}^n}{\operatorname{argmin}}\, q(\boldsymbol{v}) \tag{III.31}$$

In the standard basis $\mathbf{B}$, the corresponding nearest lattice point is then obtained via the relation (see Eq. (III.28))

$$\check{\mathrm{v}} = \boldsymbol{M}\check{\boldsymbol{v}} \tag{III.32}$$

To tackle the optimization problem (III.31), it is convenient to introduce the vector $\tilde{\boldsymbol{v}}$ defined via the relation

$$\boldsymbol{v} - \tilde{\boldsymbol{v}} \stackrel{\text{def}}{=} \boldsymbol{U}(\boldsymbol{v} - \hat{\boldsymbol{v}}) \tag{III.33}$$

As the diagonal elements of $\boldsymbol{U}$ are equal to unity, the components of $\tilde{\boldsymbol{v}}$, the 'float conditioned ambiguities' $\tilde{v}_j$, are explicitly defined by the formula

$$\tilde{v}_j \overset{\text{def}}{=} \left| \begin{array}{ll} \hat{v}_n & \text{if } j = n \\ \hat{v}_j - \sum_{k=j+1}^{n} u_{j,k}(v_k - \hat{v}_k) & \text{if } 1 \leq j < n \end{array} \right. \tag{III.34}$$

From Eqs. (III.30) and (III.33), we have

$$q(\boldsymbol{v}) = \sum_{j=1}^{n} d_j (v_j - \tilde{v}_j)^2 \tag{III.35}$$

The discrete-search methods presented in Sect. III.4 derive from this equation.

III.3 LLL reduction

In Sects. III.3.1 and III.3.2, we introduce the reduction procedures that allow an LLL-reduced basis to be built; see Sect. III.2.4. These procedures are basically involved in the LLL algorithm which provides all the related results. Our version of this algorithm, which derives from that of Luo & Qiao (2011), is presented in Sect. III.3.3.

Throughout this section, $\boldsymbol{D}$ and $\boldsymbol{U}$ are the matrices of the factorization (III.23): $\boldsymbol{Q} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{U}$ for $\boldsymbol{Q} \overset{\text{def}}{=} \boldsymbol{M}^{\mathrm{T}}\mathbf{Q}\boldsymbol{M}$; $\boldsymbol{M}$ is some unimodular matrix.

III.3.1 Procedure Reduce

If $|u_{i,j}| > 1/2$ for some $i < j$, a procedure can be applied to ensure Condition (III.26). This procedure is referred to as REDUCE(i, j).

Procedure R: REDUCE(i, j)

Consider the n-by-n unimodular matrix

$$\boldsymbol{M}_{i,j} \overset{\text{def}}{=} \mathbf{I}_n - \lfloor u_{i,j} \rceil \, \boldsymbol{\mathfrak{e}}_i \boldsymbol{\mathfrak{e}}_j^{\mathrm{T}} \qquad (i < j)$$

(Here, $\boldsymbol{\mathfrak{e}}_i$ is the column matrix associated with the ith unit vector of $\mathbf{B}$.) Then, apply $\boldsymbol{M}_{i,j}$ to $\boldsymbol{U}$ and $\boldsymbol{M}$ from the right-hand side:

$$\boldsymbol{U} := \boldsymbol{U}\boldsymbol{M}_{i,j} \qquad \boldsymbol{M} := \boldsymbol{M}\boldsymbol{M}_{i,j} \quad \square$$

Only the elements of the jth columns of $\boldsymbol{U}$ and $\boldsymbol{M}$ can be affected by the action of $\boldsymbol{M}_{i,j}$: $u_{i',j} := u_{i',j} - u_{i',i}\lfloor u_{i,j} \rceil$ for all i', and likewise $m_{i',j} := m_{i',j} - m_{i',i}\lfloor u_{i,j} \rceil$. Concerning $\boldsymbol{U}$, as $u_{i',j} = 0$ for $i' > i$, only the elements $u_{i',j}$ for $i' \leq i$ are affected. In particular, $u_{i,j} := u_{i,j} - \lfloor u_{i,j} \rceil$. In the updated version of $\boldsymbol{U}$, we thus have $|u_{i,j}| \leq 1/2$.

III.3.2 Swap procedures

To ensure Condition (III.27), which is more subtle, some particular procedure is to be implemented. The core of the problem is then governed by the 2-by-2 matrices

$$D_j \overset{\text{def}}{=} \begin{bmatrix} d_{j-1} & 0 \\ 0 & d_j \end{bmatrix} \tag{III.36}$$

and

$$U_j \overset{\text{def}}{=} \begin{bmatrix} 1 & u \\ 0 & 1 \end{bmatrix} \qquad u \overset{\text{def}}{=} u_{j-1,j} \tag{III.37}$$

Setting (see procedure R)

$$M_j^{\text{r}} \overset{\text{def}}{=} \begin{bmatrix} 1 & -\lfloor u \rceil \\ 0 & 1 \end{bmatrix} \tag{III.38}$$

we have

$$U_j M_j^{\text{r}} = \begin{bmatrix} 1 & \breve{u} \\ 0 & 1 \end{bmatrix} \qquad \breve{u} \overset{\text{def}}{=} u - \lfloor u \rceil \tag{III.39}$$

Clearly, $|\breve{u}|$ is less than or equal to $1/2$.

Now, consider Condition (III.27) with $u_{j-1,j} := \breve{u}$:

$$d_j \geq (\omega - \breve{u}^2) d_{j-1}$$

When this condition is not satisfied, one is led to change the order of the corresponding ambiguity variables. We then say that

$$M_j \overset{\text{def}}{=} M_j^{\text{r}} S \qquad \text{where} \qquad S \overset{\text{def}}{=} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{III.40}$$

is a reduce-swap operator. From Eqs. (III.38) and (III.39), it follows that

$$M_j = \begin{bmatrix} -\lfloor u \rceil & 1 \\ 1 & 0 \end{bmatrix} \qquad U_j M_j = \begin{bmatrix} \breve{u} & 1 \\ 1 & 0 \end{bmatrix} \tag{III.41}$$

Clearly, $U_j M_j$ is not an upper-triangular matrix. Its original structure can be restored as specified in the following property. (The proof of this property is given in Sect. F.)

Property RSR: REDUCESWAPRESTORE

Matrix $(U_j M_j)^{\text{T}} D_j \, (U_j M_j)$ can be factorized in the form

$$\bar{U}_j^{\text{T}} \bar{D}_j \bar{U}_j$$

where

$$\bar{D}_j \overset{\text{def}}{=} \begin{bmatrix} \bar{d}_{j-1} & 0 \\ 0 & \bar{d}_j \end{bmatrix} \qquad \bar{U}_j \overset{\text{def}}{=} \begin{bmatrix} 1 & \bar{u} \\ 0 & 1 \end{bmatrix}$$

in which

$$\bar{d}_{j-1} \overset{\text{def}}{=} d_j + \breve{u}^2 d_{j-1} \qquad \bar{d}_j \overset{\text{def}}{=} d_j \, \frac{d_{j-1}}{\bar{d}_{j-1}} \qquad \bar{u} \overset{\text{def}}{=} \breve{u} \, \frac{d_{j-1}}{\bar{d}_{j-1}}$$

As a corollary,

$$G_j U_j M_j = \bar{U}_j \quad \text{where} \quad G_j \stackrel{\text{def}}{=} \begin{bmatrix} \bar{u} & 1 - \breve{u}\bar{u} \\ 1 & -\breve{u} \end{bmatrix}$$

Moreover, $[G_j^{-1}]^{\mathrm{T}} D_j G_j^{-1} = \bar{D}_j$ ⊡

The following procedure in which $u \stackrel{\text{def}}{=} u_{j-1,j}$ results from this property.

Procedure RSR: REDUCESWAPRESTORE(j)

Compute $\breve{u} = u - \lfloor u \rceil$,

$$\bar{d}_{j-1} = d_j + \breve{u}^2 d_{j-1} \qquad \bar{d}_j = d_j \frac{d_{j-1}}{\bar{d}_{j-1}} \qquad \bar{u} = \breve{u} \frac{d_{j-1}}{\bar{d}_{j-1}}$$

To update $\boldsymbol{D}$, set $d_{j-1} := \bar{d}_{j-1}$ and $d_j := \bar{d}_j$.

Then, for $j \geq 2$, let $\boldsymbol{M}_j \stackrel{\text{def}}{=} \mathbf{diag}([\mathbf{I}_{j-2} \;\; M_j \;\; \mathbf{I}_{n-j}])$ be the matrix obtained from the identity matrix $\mathbf{I}_n$ by substituting

$$M_j = \begin{bmatrix} -\lfloor u \rceil & 1 \\ 1 & 0 \end{bmatrix}$$

for its 2-by-2 block with largest diagonal index j; see Eq. (III.41). Likewise, define $\boldsymbol{G}_j \stackrel{\text{def}}{=} \mathbf{diag}([\mathbf{I}_{j-2} \;\; G_j \;\; \mathbf{I}_{n-j}])$ where

$$G_j = \begin{bmatrix} \bar{u} & 1 - \breve{u}\bar{u} \\ 1 & -\breve{u} \end{bmatrix}$$

Matrices $\boldsymbol{U}$ and $\boldsymbol{M}$ are then updated as follows:

$$\boldsymbol{U} := \boldsymbol{G}_j \boldsymbol{U} \boldsymbol{M}_j \qquad \boldsymbol{M} := \boldsymbol{M}\boldsymbol{M}_j \quad ⊡$$

When implementing the operation $\boldsymbol{G}_j \boldsymbol{U} \boldsymbol{M}_j$, the diagonal 2-by-2 block of $\boldsymbol{U}$ with largest diagonal index j is updated separately. Indeed, according to the corollary of Property III.3.2, it is equal to $\bar{U}_j$.

In the case where $\lfloor u \rceil = 0$, this procedure reduces to Procedure SR: SWAPRESTORE(j).

III.3.3 LLL-type algorithms

The original LLL algorithm provides the matrices $\boldsymbol{U}$ and $\boldsymbol{D}$ involved in the LLL-reduced version of $\mathbf{Q}$ (see Eqs. (III.23) and (III.22)):

$$\boldsymbol{Q} = \boldsymbol{U}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{U} \quad \text{for} \quad \boldsymbol{Q} \stackrel{\text{def}}{=} \boldsymbol{M}^{\mathrm{T}} \mathbf{Q} \boldsymbol{M}$$

It also yields the LLL-reduced basis $\boldsymbol{B} \stackrel{\text{def}}{=} \mathbf{B}\boldsymbol{M}$; see Sects. III.2.3 and III.2.4. Its main instructions are the following (see Eq. (III.16) for its initialization).

Original LLL algorithm

```
1   U := U; D := D; M := I_n
2   j := 2
3   while j ≤ n
4     if |u_{j-1,j}| > 1/2, REDUCE(j − 1, j)
5     if d_j < (ω − u²_{j-1,j}) d_{j-1}
6       SWAPRESTORE(j)
7       j := max(j − 1, 2)
8     else
9       for i := j − 2 down to 1
10        if |u_{i,j}| > 1/2, REDUCE(i, j)
11      endfor 9
12      j := j + 1
13    endif 5
14  endwhile 3
```

Recently, Luo & Qiao (2011) proposed a modified LLL algorithm which can save a significant amount of operations, and also provides a basis for a parallel implementation. In that approach, which is justified via an example presented in Sect. 3 of their paper, the procedures imposing condition (III.26) are implemented at the end of this algorithm, once the LLL condition (III.27) has been imposed.

LLL algorithm with delayed size-reduction

```
1   U := U; D := D; M := I_n
2   j := 2
3   while j ≤ n          [to impose Condition (III.27)]
4     u := u_{j-1,j}
5     if |u| > 1/2
6       ReduceOption := true
7       ŭ := u − ⌊u⌉
8     else
9       ReduceOption := false
10      ŭ := u
11    endif 5
12    if d_j < (ω − ŭ²) d_{j-1}
13      if ReduceOption = true
14        REDUCESWAPRESTORE(j)
15      else
16        SWAPRESTORE(j)
17      endif 13
18      j := max(j − 1, 2)
19    else
20      j := j + 1
21    endif 12
22  endwhile 3
23  for j := 2 : n       [to impose Condition (III.26)]
24    for i := j − 1 down to 1
25      if |u_{i,j}| > 1/2
26        REDUCE(i, j)
27      endif
28    endfor 24
29  endfor 23
```

Typically, this LLL algorithm with 'delayed size-reduction' runs twice as fast as the original LLL algorithm. Compared to the algorithm of Luo & Qiao (2011), we made here the distinction between the procedures RSR and SR. Some CPU time can thus still be saved. Those changes concern the instruction blocks 5-11 and 13-17.

The procedures described in Sects. III.3.1 and III.3.2 can be completed so that this algorithm also provides the float solution in the LLL-reduced basis: $\hat{\boldsymbol{v}} = \boldsymbol{M}^{-1}\hat{\mathbf{v}}$; see Eq. (III.28). This can be done without forming $\boldsymbol{M}^{-1}$ explicitly.

According to Property III.3.2, we have

$$\bar{d}_{j-1} = d_j + \breve{u}^2 d_{j-1}$$

Instruction 12 can therefore be equally well written in the form

12 if $\bar{d}_{j-1} < \omega d_{j-1}$

At level j, the procedures RSR and SR modify, in particular, the matrix element $u_{j-2,j-1}$. As a result, this algorithm has a 'one-step up-and-down structure;' see instructions 18 and 20. Lenstra, Lenstra and Lovász have shown that for any ω in the open interval $]1/4\ 1[$, the algorithm terminates: the number of times that the algorithm encounters the case where $\bar{d}_{j-1} < \omega d_{j-1}$ is bounded. In the limit case where $\omega = 1$, the convergence can also be guaranteed; for further details, see Akhavi (2003); Nguyen & Stehlé (2009).

The convergence of the LLL algorithm is faster when reducing the value of the relaxation parameter ω, but below some value (for example $\omega = 0.70$), the dilute$\mathbf{Q}$-orthogonality defect of the LLL-reduced basis $\boldsymbol{B}$ thus obtained begins to increase. The choice of ω therefore depends on the context.

For example, in GNSS, when handling a regional network in real-time with $n = 168$ and $\delta_{\mathbf{Q}}(\mathbf{B}) \simeq 6.62$, ω may reasonably be set equal to 0.9; $\delta_{\mathbf{Q}}(\boldsymbol{B})$ can then be reduced to 1.19 for example. One then has a good compromise between the CPU time required for finding the reduced basis, and that used for the discrete search; see Sect. III.4. On our old computers, the CPU time used for that LLL-reduction was 0.075 second with our LLL-type algorithm, against 0.141 second with the original LLL algorithm. The LLL algorithm with delayed size-reduction effectively leads to a gain of the order of two.

For the statistical developments involved in the GNSS validation procedures, such as those of Verhagen & Teunissen (2006), the choice $\omega = 1$ is preferable. Indeed, as the discrete search is performed many times in the same reduced basis, the latter must be as $\mathbf{Q}$-orthogonal as possible.

III.4 Discrete search

This section is essentially devoted to the solution of the NLP problem in the selected reduced basis; see Sects. III.2.1, III.2.4, III.2.5, and III.3.3. The problem is therefore to minimize $q(\boldsymbol{v})$ for $\boldsymbol{v}$ lying in $\mathbb{Z}^n$; see Eqs. (III.31) and (III.35).

Once the integer ambiguities $v_n,\ v_{n-1}, \ldots,\ v_{i+1}$ have been conditioned somehow (see the example given below), Eq. (III.34) provides the float conditioned ambiguity $\tilde{v}_j$.

Example: Babai point. Let us concentrate on Eq. (III.35) where the d_j's are loosely sorted in increasing order with no distinctive discontinuity. To find a point $\boldsymbol{v}$ for which $q(\boldsymbol{v})$ is a priori small, one is led to perform the 'bootstrapping' recursive process described below. The point thus formed is the Babai point $\boldsymbol{v}^{\mathrm{B}}$ (Babai, 1986):

$$\begin{aligned}
&\text{level } n: && v_n^{\mathrm{B}} = \lfloor \tilde{v}_n \rceil && \text{where } \tilde{v}_n = \hat{v} \\
&\text{level } n-1: && v_{n-1}^{\mathrm{B}} = \lfloor \tilde{v}_{n-1} \rceil && \text{where } \tilde{v}_{n-1} = \hat{v}_{n-1} - u_{n-1,n}(v_n^{\mathrm{B}} - \hat{v}_n) \\
&\qquad\vdots \\
&\text{level } 1: && v_1^{\mathrm{B}} = \lfloor \tilde{v}_1 \rceil && \text{where } \tilde{v}_1 = \hat{v}_1 - \sum_{k=2}^{n} u_{1,k}(v_k^{\mathrm{B}} - \hat{v}_k)
\end{aligned}$$

The Babai point is often the solution of the NLP problem, but not necessarily. In any case however (as explicitly shown in this section), it is the 'natural starting point' for searching this solution □

III.4.1 Ambiguity conditioning at level *j*

In the general case, in the process of conditioning ambiguity v_j, we will use the following notation (see Eq. (III.35))

$$s_j \overset{\text{def}}{=} \sum_{i=j}^{n} d_i (v_i - \tilde{v}_i)^2 \tag{III.42}$$

where $\tilde{v}_i$ is given by (see Eq. (III.34))

$$\tilde{v}_i = \left| \begin{array}{ll} \hat{v}_n & \text{if } i = n; \\ \hat{v}_i - \sum_{k=i+1}^{n} u_{i,k}(v_k - \hat{v}_k) & \text{if } 1 \leq i < n \end{array} \right.$$

Note that $s_j = t_j + d_j (v_j - \tilde{v}_j)^2$ where

$$t_j \overset{\text{def}}{=} \left| \begin{array}{ll} 0 & \text{if } j = n; \\ s_{j+1} & \text{if } j < n \end{array} \right. \tag{III.43}$$

Let us now assume that the ambiguities $v_n, v_{n-1}, \ldots, v_{i+1}$ have already been conditioned. Denoting by ℓ an integer candidate for v_j, we then set

$$s \equiv s_j^{(\ell)} \overset{\text{def}}{=} t_j + d_j (\ell - \tilde{v}_j)^2 \tag{III.44}$$

The first ambiguity value ℓ to be considered at level j is then

$$m = \lfloor \tilde{v}_j \rceil \tag{III.45}$$

Indeed, $|\ell - \tilde{v}_j|$ and thereby s are then as small as possible. In the process of minimizing $q(\boldsymbol{v})$, one is led to consider values of ℓ other than m. These integers, $\ell_1, \ell_2, \ldots, \ell_p, \ldots$, where $\ell_1 = m$, are then sorted so that the discrepancies $|\ell_p - \tilde{v}_j|$ form an increasing sequence. The second integer to be considered is therefore $m+1$ or $m-1$. Two cases are thus distinguished (see Schnorr & Euchner (1994)):

Schnorr$^{(+)}$: $m < \tilde{v}_j$. Ambiguity v_j may then be conditioned at the successive terms of the Schnorr list$^{(+)}$

$$m,\ m+1,\ m-1,\ m+2,\ m-2,\ m+3,\ \ldots$$

Schnorr$^{(-)}$: $m \geq \tilde{v}_j$. Ambiguity v_j may then be conditioned at the successive terms of the Schnorr list$^{(-)}$

$$m,\ m-1,\ m+1,\ m-2,\ m+2,\ m-3,\ \ldots$$

In our implementation of the related approach, we save CPU time in the computation of the successive values of $(\ell_p - \tilde{v}_j)^2$. When handling the ambiguities ℓ, and $\ell + 1$ or $\ell - 1$, the following 'perturbation formulas' are then used:

$$\left| \begin{array}{l} [(\ell+1) - \tilde{v}_j]^2 = w^2 + (1 + 2w) \\ [(\ell-1) - \tilde{v}_j]^2 = w^2 + (1 - 2w) \end{array} \right. \qquad w \stackrel{\text{def}}{=} \ell - \tilde{v}_j \tag{III.46}$$

The multiplication $w^2 := w \times w$ is then performed only for $\ell := m$; see Sect III.4.2. Many multiplications can thus be avoided. Note that the calculation of $2w$ is then to be made in an optimal manner ($2w$ is not necessarily computed as the sum $w + w$).

In the implementation of our approach, we used object-orientated programming (OOP), and introduced a specific object referred to as SL (for Schnorr list). More precisely, at the beginning of our program, we instantiated an array of n such objects, one at each level j. We then added two 'methods' linked to this object: INIT and NEXT. The latter are described in the following section.

III.4.2 Methods INIT and NEXT

The actions of INIT and NEXT consist in initializing and updating a two-element FIFO vectorial queue $(\ell_a rm, \ell_\mathrm{b})$, $(s_a rm, s_\mathrm{b})$ associated with the two-component vector (ℓ, s). The table below shows the structure of queue $(\ell_a rm, \ell_\mathrm{b})$ in the case of the Schnorr list$^{(+)}$:

	sg	$\ell_a rm$	ℓ_b
After INIT:	$+1$	m	m
After NEXT:	-1	m	$m+1$
After NEXT:	$+1$	$m+1$	$m-1$
After NEXT:	-1	$m-1$	$m+2$

Just before the call to INIT, $\tilde{v}_j$ is computed on the grounds of Eq. (III.34); see Remark4.1 further on.

Method INIT: instruction $(\ell, s) :=$ SL$_j$–INIT$(\tilde{v}_j, t_j)$

Set

$\ell := \lfloor \tilde{v}_j \rceil$
$w := \ell - \tilde{v}_j$
$s := t_j + d_j w^2$
$\ell_a rm := \ell_\mathrm{b} := \ell$
$s_a rm := s_\mathrm{b} := s$

if $w < 0$
set sg $:= (+1)$
else
set sg $:= (-1)$

Method NEXT: instruction $(\ell, s) :=$ SL$_j$–NEXT

Set

$w := \ell_a rm - \tilde{v}_j$

$\ell := \ell_a rm + \mathrm{sg}$

if $\mathrm{sg} = 1$

$s := s_a rm + d_j(1 + 2w)$

else

$s := s_a rm + d_j(1 - 2w)$

Set

$\ell_a rm := \ell_\mathrm{b}$; $\ell_\mathrm{b} := \ell$

$s_a rm := s_\mathrm{b}$; $s_\mathrm{b} := s$

$\mathrm{sg} := (-\mathrm{sg})$

Remark.4.1: According to Eq. (III.34), the float conditioned ambiguity $\tilde{v}_j$ is given by the formula

$$\tilde{v}_j = \left| \begin{array}{ll} \hat{v}_n & \text{if } j = n \\ \tilde{u}_{j,j+1} & \text{if } 1 \leq j < n \end{array} \right. \tag{III.47}$$

where

$$\tilde{u}_{j,k} \overset{\text{def}}{=} \hat{v}_j - \sum_{\kappa=k}^{n} u_{j,\kappa}(v_\kappa - \hat{v}_\kappa) \tag{III.48}$$

Now, consider the general case when $\tilde{v}_j$ is to be computed, when it has already been computed, and when in the meanwhile, for some $j_\mathrm{r} > j$, the integer ambiguities $v_{j_\mathrm{r}+1}$, $v_{j_\mathrm{r}+2}$, $\ldots$, v_{n-1}, v_n have not changed. In our conditioning process, to reduce the corresponding CPU cost, $\tilde{v}_j$ is then computed as follows (see Eqs. (III.47) and (III.48)):

If $j_\mathrm{r} = n$ (even if $\tilde{v}_j$ has not been computed yet)

$\mathfrak{u} := \hat{v}_j$

else

$\mathfrak{u} := \tilde{u}_{j,j_\mathrm{r}+1}$

for $k := j_\mathrm{r}$ down to $k := j + 1$

$\mathfrak{u} := \mathfrak{u} - u_{j,k}(v_k - \hat{v}_k)$

$\tilde{u}_{j,k} := \mathfrak{u}$

endfor

$\tilde{v}_j := \mathfrak{u}$

An auxiliary upper-triangular matrix $\tilde{\boldsymbol{U}}$ is thus built and updated through the process. For further details, see Sect. III.4.3 and Remark4.3 □

III.4.3 Discrete-search algorithms

On the grounds of the notions introduced in Sects. III.4.1 and III.4.2, we have designed three discrete-search algorithms referred to as DS, DNS and DSC:

1) algorithm DS yields a nearest lattice point $\check{\boldsymbol{v}}$ and $\check{q} \overset{\text{def}}{=} q(\check{\boldsymbol{v}})$;

2) algorithm DNS provides the first ns NLP solutions $\check{\boldsymbol{v}}_1 \equiv \check{\boldsymbol{v}}, \check{\boldsymbol{v}}_2, \ldots, \check{\boldsymbol{v}}_{\text{ns}}$ with $\check{q} \equiv \check{q}_1 \leq \check{q}_2 \leq \cdots \leq \check{q}_{\text{ns}}$;

3) given some parameter $c > 0$, algorithm DSC identifies all the points $\boldsymbol{v}$ of $\mathbb{Z}^n$ contained in the ellipsoid

$$\mathcal{E}(c) \overset{\text{def}}{=} \{\boldsymbol{v} \in \mathbb{R}^n : q(\boldsymbol{v}) \leq c\} \tag{III.49}$$

Clearly, $\mathcal{E}(c)$ is centred on the float solution $\hat{\boldsymbol{v}}$; c defines the size of this ellipsoid.

Algorithm DS. The objective is to condition the integer ambiguities v_j so that $q(\boldsymbol{v})$ is minimum. We first note that from Eqs. (III.35) and (III.42),

$$q(\boldsymbol{v}) = s_1 = r_j + s_j \tag{III.50}$$

where

$$r_j \overset{\text{def}}{=} \sum_{i=1}^{j-1} d_i (v_i - \tilde{v}_i)^2 \tag{III.51}$$

As r_j is non-negative, we therefore have:

Property 5.1 *If $s_j \geq a$ for some $a > 0$, then $q(\boldsymbol{v}) = s_1 \geq a$.*

We first form the Babai point, here $\boldsymbol{v} := \boldsymbol{v}^{\text{B}}$; see the bootstrapping stage 2-8 of the algorithm displayed in the next page. All the Schnorr lists from $j := n$ down to $j := 1$, as well as $\tilde{\boldsymbol{U}}$, are thus initialized; see Remark4.1 with $j_{\text{r}} = n$. As the Babai point is the first NLP candidate, we then set

$$\check{\boldsymbol{v}} := \boldsymbol{v}, \quad q(\check{\boldsymbol{v}}) \equiv \check{q} := s_1$$

The NLP search starts from the Babai point, but in the opposite sense, with a Boolean variable Forwards equal to true. We therefore move to level $j = 2$. Indeed, if v_1 was set equal to the next integer of SL_1, $q(\boldsymbol{v})$ would then be greater than $\check{q}$.

To understand the principle of the algorithm in the general case, let us assume that we are at some level $j \geq 2$ with $\text{Forwards} = \text{true}$. We then consider the integer ℓ provided by SL_j–NEXT; this method also yields s: the new value of s_j that would be obtained if v_j was set equal to ℓ. Clearly, s is greater than the current value of s_j (and this would be worse with the remaining terms of the Schnorr list at this level). Two cases are then to be considered.

Case 1: $s \geq \check{q}$. If we then set $v_j := \ell$, whatever the conditioning of the integer ambiguities $v_{j-1}, \ldots, v_1$, we would then have $s_1 \geq \check{q}$ from Property III.4.3. Furthermore, another NEXT-type instruction would increase s_j. In this case, we are therefore left to move forwards to level $j := j + 1$.

Case 2: $s < \check{q}$. As there is still a hope of reducing s_1 by conditioning $v_{j-1}, \ldots, v_1$ in an appropriate manner, we then set

$$(v_j, s_j) := (\ell, s), \quad t_{j-1} := s_j, \quad \text{Forwards} := \text{false}$$

and move backwards to level $j := j - 1$; $\tilde{v}_j$ is then updated; note that $(\lfloor \tilde{v}_j \rceil - \tilde{v}_j)^2$ may then be smaller than previously at that level.

When the algorithm moves forwards to level $j := j + 1$, SL_j–NEXT is then called. When it moves backwards to level $j := j - 1$, a new Schnorr list is initialized via SL_j–INIT. In both cases, the situation is then analysed to define what is to be done; see Cases 1 and 2.

Via Case 2, the algorithm may progressively reach level $j = 1$ (several times). If s is less than $\check{q}$, $\check{\boldsymbol{v}}$ and $\check{q}$ are then updated; see instructions 32 to 36.

Via Case 1, the algorithm reaches level n, at least once. When SL_n–NEXT yields an s greater than or equal to $\check{q}$, the algorithm then stops; see instructions 14 to 25. We then have the following property (see Eq. (III.24)):

Property 5.2 *At the end of the algorithm, no point of $\mathbb{Z}^n$ lies in the interior of ellipsoid $\mathcal{E}(\check{q})$; $\check{\boldsymbol{v}}$ is on its boundary.*

Proof. Let us assume that there exists some $\boldsymbol{v}^\circ$ in $\mathbb{Z}^n$ such that $q^\circ \overset{\text{def}}{=} q(\boldsymbol{v}^\circ) < \check{q}$. From Eqs. (III.35) and (III.34),

$$q^\circ = d_n(v_n^\circ - \hat{v}_n)^2 + d_{n-1}(v_{n-1}^\circ - \tilde{v}_{n-1}^\circ)^2 + \cdots + d_1(v_1^\circ - \tilde{v}_1^\circ)^2$$

The quantities

$$s_n^\circ \overset{\text{def}}{=} d_n(v_n^\circ - \hat{v}_n)^2$$

and

$$s_j^\circ \overset{\text{def}}{=} s_n^\circ + d_{n-1}(v_{n-1}^\circ - \tilde{v}_{n-1}^\circ)^2 + \cdots + d_j(v_j^\circ - \tilde{v}_j^\circ)^2 \quad (1 \le j < n)$$

are then less than $\check{q}$. The algorithm starts by setting v_n equal to the first term of the Schnorr list SL_n. In the NLP search, it then comes back to level n via instruction 18, at least once, until v_n is conditioned at v_n°; indeed, s_n° is less than $\check{q}$ (see instructions 19 to 21). The algorithm then starts moving backwards (via instruction 22), and reaches instruction 31 with $j = n - 1$. The Schnorr list SL_{n-1} is then systematically explored, with possible excursions at levels $j < n-1$, and this until v_{n-1} is set equal to v_{n-1}°, since $s_{n-1}^\circ < \check{q}$. The algorithm then proceeds to level $n - 2$. It then behaves similarly for that level, and so on, until level $j = 1$ where v_1 is set equal to the first term of SL_1. The condition $s < \check{q}$ of instruction 33 then holds, hence via instructions 34 and 35, $\check{q} := s \le q^\circ$, which contradicts the initial assumption. □

Remark.4.2: The pathological situations where $\check{\boldsymbol{v}}$ is not the only point of $\mathbb{Z}^n$ on the boundary of $\mathcal{E}(\check{q})$ can be detected at level of instruction 33. The integer-ambiguity solution $\check{\boldsymbol{v}}$ cannot then be validated. As it is presented, the algorithm selects as solution the first $\boldsymbol{v}$ for which $q(\boldsymbol{v}) = \check{q}$; the other ones (if any) are discarded. A subsequent statistical analysis can be used to diagnose such pathological cases. In practice, as expected, such situations never occur ⊡

Remark.4.3: In the NLP search (instructions 13 to 48), the integers j_1 and j_2 keep track of the successive levels j at which the value of the Boolean variable Forwards changes. Note that $j_2^\star$ is the current largest index j_2 at which the algorithm started moving backwards. According to instructions 11 and 38, whenever $j = 1$, j_1 and $j_2^\star$ are set equal to 1. By computing the float conditioned ambiguities in the framework of Remark4.1 in which j_r is defined (from j_1, j_2 and $j_2^\star$) via instructions 28-29, the global CPU time of algorithm DS can be reduced

Table III.1: Algorithm DS

```
1   t_n := 0; j_r := n
2   for j := n down to j := 1      [Babai loop]
3     Compute ṽ_j
4     (ℓ, s) := SL_j–INIT (ṽ_j, t_j)
5     (v_j, s_j) := (ℓ, s)
6     if j > 1 set t_{j−1} := s_j
7   endfor 2
8   (v̆, q̆) := (v, s_1)             [Babai point]
9   NLPfound := false
10  Forwards := true
11  j_1 := 1; j_2^⋆ := 1
12  j := 1
13  while NLPfound = false         [NLP search]
14    if Forwards = true           [move forwards]
15      if j = n
16        NLPfound := true
17      else
18        j := j + 1
19        (ℓ, s) := SL_j–NEXT
20        if s < q̆
21          (v_j, s_j) := (ℓ, s); t_{j−1} := s_j
22          Forwards := false
23          j_2 := j; j_2^⋆ := max(j_2, j_2^⋆)
24        endif 20
25      endif 15
26    else                         [move backwards]
27      j := j − 1
28      if j < j_1 set j_r := j_2^⋆
29      else       set j_r := j_2
30      Compute ṽ_j
31      (ℓ, s) := SL_j–INIT (ṽ_j, t_j)
32      if j = 1  [case j = 1]
33        if s < q̆
34          (v_1, s_1) := (ℓ, s)
35          (v̆, q̆) := (v, s_1)          [new v̆]
36        endif 33
37        Forwards := true
38        j_1 := 1; j_2^⋆ := 1
39      else       [case j > 1]
40        if s < q̆
41          (v_j, s_j) := (ℓ, s); t_{j−1} := s_j
42        else
43          Forwards := true
44          j_1 := j
45        endif 40
46      endif 32
47    endif 14
48  endwhile 13
```

by a factor of the order of two. In this context, the following technical point is also to be mentioned.

First of all, at the beginning of algorithm DS, the values of $\hat{v}_j$ are placed on the diagonal of $\tilde{\boldsymbol{U}}$:

$\tilde{u}_{j,j} := \hat{v}_j \quad$ (for $j := 1, \ldots, n$)

Instructions 5, 21 and 41 are then completed by setting

$v_j^* := \ell - \hat{v}_j$

The instructions $\mathfrak{u} := \hat{v}_j$ and $\mathfrak{u} := \mathfrak{u} - u_{j,k}(v_k - \hat{v}_k)$ of Remark4.1 are then changed into $\mathfrak{u} := \tilde{u}_{j,j}$ and $\mathfrak{u} := \mathfrak{u} - u_{j,k} v_k^*$, respectively. The input variables of the function that computes $\tilde{v}_j$ are then j, $j_{\rm r}$, n, $\boldsymbol{U}$, $\tilde{\boldsymbol{U}}$ and $\boldsymbol{v}^*$ ⊡

Remark.4.4: At the beginning of the NLP search, the size parameter c of the search ellipsoid (III.49) is defined by the value of $q(\boldsymbol{v})$ for the Babai point. When the latter is not the NLP solution, $c \equiv \check{q}$ is reduced via instruction 35 ⊡

Algorithm DNS. The process is similar to that of algorithm DS; but, once the Babai point has been formed, instead of moving forwards to level $j = 2$, SL_1–NEXT is set in motion $\mathrm{ns} - 1$ times. We thus get a 'working set' including ns 'candidate optimal lattice points' $\check{\boldsymbol{v}}[\mathrm{ns}]$ together with their q-values $\check{q}[\mathrm{ns}]$. The last q-value thus obtained, which (by construction) is larger than the previous ones, is denoted by $\check{q}_{\rm ns}$. In algorithm DNS, $\check{q}_{\rm ns}$ is going to play the same role as $\check{q}$ in algorithm DS.

We then move forwards to level $j = 2$; SL_2–NEXT then provides the next term ℓ of the Schnorr list at level 2 together with the value of s for that ℓ. If s is less than $\check{q}_{\rm ns}$, we then set $(v_2, s_2) := (\ell, s)$, $t_1 := s_2$, and move backwards to level 1; SL_1–INIT then defines (via ℓ) some lattice point $\boldsymbol{v}$ with its q-value $q(\boldsymbol{v}) := s_1 := s$. If s is less than $\check{q}_{\rm ns}$, as $\boldsymbol{v}$ does not lie in the current set $\check{\boldsymbol{v}}[\mathrm{ns}]$, s and $\boldsymbol{v}$ have to be inserted at their right places in the sets $\check{q}[\mathrm{ns}]$ and $\check{\boldsymbol{v}}[\mathrm{ns}]$; the previous $\check{q}_{\rm ns}$ and $\check{\boldsymbol{v}}_{\mathbf{ns}}$ are then removed. Instruction SL_j–NEXT is then performed until s is larger than the current value of $\check{q}_{\rm ns}$. After each of these instructions, $\check{q}[\mathrm{ns}]$ and $\check{\boldsymbol{v}}[\mathrm{ns}]$ are of course updated and sorted. In any case, we then finally move forwards to level $j = 2$; SL_2–NEXT is then performed, and so on. Clearly, the principle is the same.

Algorithm DSC. The process is again similar to that of algorithm DS. As all the points of ellipsoid $\mathcal{E}(c)$ are to be identified, the tests $s < \check{q}$ (the instructions 20, 33 and 40 of algorithm DS) are replaced by $s < c$. When SL_j–NEXT is called, we move forwards to level $j + 1$, only when the value of s thus obtained is larger than (or equal to) c; see Eq. (III.49) and Property III.4.3. Otherwise, we set $(v_j, s_j) := (\ell, s)$, $t_{j-1} := s_j$ and move backwards: $j := j - 1$; then SL_j–INIT, and so on. Instruction 35 of algorithm DS is replaced by other instructions which depend on what is to be done with the vector $\boldsymbol{v}$ thus identified; see, e.g., Verhagen & Teunissen (2006); Lannes & Prieur (2011).

III.5 On some computational issues

The serial algorithms presented in Sects. III.3.3 and III.4.3 were implemented in C++ programs, and tested on old PC's working with Windows XP and Linux operating systems.

Intensive testing was performed with real data on a regional GNSS network. As already mentioned at the end of Sect. III.3.3, for $n = 168$, the CPU time for the execution of our LLL-type algorithm with $\omega = 0.9$ was negligible: about 0.075 second. Compared to the original LLL algorithm, as implemented for instance by Agrell et al. (2002) or Jazaeri et al. (2012), the gain was of the order of two. In fact, the parallel approach begins to be of interest for n larger than (say) 200; see the reduction-list implementation of Luo & Qiao (2011).

Concerning the discrete-search algorithms presented here, our method was compared to that of Jazaeri et al. (2012) which corresponds to the present state of the art for the discrete search. Our statistical study on 3×10^5 Gaussian $\hat{\boldsymbol{v}}$-samples was conducted for $n = 168$ in the LLL-reduced basis obtained as already specified. The Gaussian $\hat{\boldsymbol{v}}$-samples were of mean $\boldsymbol{0}$ and variance-covariance matrix $\boldsymbol{V} = \boldsymbol{Q}^{-1}$in that basis. For each sample, $\check{\boldsymbol{v}}_1 \equiv \check{\boldsymbol{v}}$ and $\check{\boldsymbol{v}}_2$ were determined via our DNS algorithm; see Sect. III.4.3. The CPU times for those discrete searches were 236 seconds with the algorithm of Jazaeri et al. (2012), and 129 seconds with our DNS algorithm. This gain, which is of the order of two, is essentially due to the way of computing the float conditioned ambiguities; see Remarks 4.1 and 4.3.

With regard to the self-calibration problems presented in Sect. III.1.1, the previous statistical study gives and idea of the efficiency of our methods for finding the global and secondary minima of the arc functional g; see Sect III.1.1.

For handling the Schnorr lists at best, some object-oriented programming tools have been introduced; see Sect. III.4.2. Our discrete-search algorithms were thereby written in an 'almost-electronic form.' Shortly, they were designed for DSP (digital signal processor) implementation at the 'speed of light.' In GNSS, for example, the integer ambiguities of regional networks can thus be fixed in real time. Let us finally note that for large n, the only discrete-search operations that can be performed in a parallel manner are those associated with the successive terms of the Schnorr lists at levels n and 1.

Chapter IV

Calibration of the clock-phase biases of GNSS networks

IV.1 Determination of clock-phase biases: introduction

In Global Navigation Satellite Systems (GNSS), the calibration of the clock-phase biases of global networks is a challenging problem. In particular, the knowledge of the satellite clock-phase biases is needed for Precise Point Positioning (PPP); see, e.g., Zumberge et al. (1997); Ge et al. (2008); Bertiger et al. (2010); Geng et al. (2010); Zou et al. (2012); Ge et al. (2012). In the general context defined below, the equations governing this GNSS calibration problem have a basic rank defect. In this chapter we analyze the different ways of removing this rank defect, and define an efficient strategy for obtaining the clock-phase biases in a standard form. The link with other related approaches, such as those proposed by Blewitt (1989); de Jonge (1998); Ge et al. (2005); Loyer et al. (2012), is established in that framework.

When modelling the multi-frequency (code and phase) observations of GNSS networks, the system to be considered include phase structures of the form

$$\left| \begin{array}{l} [\varphi_{\mathrm{r}\kappa}(i) - \varphi_{\mathrm{s}\kappa}(j)] + N(i,j) = b_\kappa(i,j) \\ \text{for } \kappa = 1, \ldots, k \end{array} \right. \tag{IV.1}$$

Here, κ is the epoch index; k is the index of the current epoch; $\varphi_{\mathrm{r}\kappa}(i)$ and $\varphi_{\mathrm{s}\kappa}(j)$ are clock (or pseudo-clock) phase biases. (The notion of pseudo-clock phase bias is explicitly defined in Lannes & Teunissen (2011).) These terms are also called 'uncalibrated phase delays' (UPD). They are expressed in cycles, and depend on the frequency of the transmitted carrier wave; subscripts r and s stand for receiver and satellite,[1] respectively; i is the index of the receiver, and j that of the satellite; $N(i,j)$ is the integer ambiguity of the corresponding carrier-phase measurement. The terms $b_\kappa(i,j)$ include the corresponding phase data and all the other contributions of such equations; see, e.g., Lannes & Teunissen (2011); Loyer et al. (2012). The set of receiver-satellite pairs (i,j) involved in Eq. (IV.1) forms the observational graph $\mathscr{H}_\kappa$ of the GNSS scenario of epoch κ. This graph is assumed to be connected; see G. Note that the wide-lane (WL) equation of the ionosphere-free mode is typically of form (IV.1); N is then a wide-lane integer ambiguity; see, e.g., Eq. (4) of Loyer et al. (2012).

[1] In this chapter, satellite should be understood as satellite transmitter.

As explicitly clarified further on, whenever phase structures such as (IV.1) appear in GNSS-network problems, a related rank defect is to be removed. In this chapter, we restrict ourselves to the case where the rank defects induced by the phase structures of type (IV.1) can be dealt with in an independent manner. This does not mean of course that those basic rank defects are the only ones to be handled in practice; see, in particular, Teunissen & Odijk (2003). A standard approach for tackling the rank defects is known as the S-system approach; see, e.g., Baarda (1973); Teunissen (1984); de Jonge (1998). Examples of such S-system solutions are to be found in de Jonge (1998); Teunissen et al. (2010); Zhang et al. (2011); Odijk et al. (2012).

In the geodetic and GNSS literature, there exist several ways of removing this basic rank defect. The most general approach is based on the S-system theory already mentioned. Other strategies derive from the pioneering contribution of Blewitt (1989): the relationship between the undifferenced (UD) ambiguities and the double differenced (DD) ambiguities is completed so that the operator D thus defined is invertible. Let us also mention the approach of Collins et al. (2010) which is based on the concept of 'ambiguity datum fixing.' The important developments of those approaches, both at a conceptual and technical level, were often conducted with different physical objectives. They have thus progressively and insidiously masked the fundamental links between the related methods.

Briefly, the Blewitt procedure can be divided in three steps. In the first step, with regard to Eq. (IV.1) for example, the UD data are processed by considering the term on the left-hand side of that equation as a 'constant functional variable;' a float estimate of this 'biased ambiguity function' is thus obtained. In the second step, the values of the corresponding DD ambiguity function are computed, and then fixed. In the third step, the clock-phase biases φ_κ are estimated by using as data the UD ambiguities provided by the action of D^{-1} on those fixed ambiguities. The theoretical analysis developed in the present paper provides in particular an answer to the following question that has never been answered yet: what is the link between the UD ambiguities thus fixed and the fixed 'closure-delay' or 'closure-difference' (CD) ambiguities of the UD approach of Lannes & Teunissen (2011) ? A similar question arises for the UD approach of Collins et al. (2010); an answer is also provided.

In this general GNSS context, the main objective of the paper is to propose a unified theoretical framework in which the various contributions in the related fields of research can be understood and compared more easily. This can lead to some improvement of the implementation of the related methods. For example, we show that removing the rank defect via the D-matrix of Blewitt (1989) can be analyzed in a theoretical framework tightly linked to the S-system approach of Teunissen (1984). We thus show that the intermediate differencing stage of the Blewitt approach can be avoided, without any counterpart, via the approach of Teunissen (1984) as it is formulated for example in Lannes & Teunissen (2011): the closure ambiguities to be fixed then appear, from the outset, in the very formulation of the UD problem to be solved; compare with what is done in Sect. 4 of Ge et al. (2005) for instance.

The theoretical guidelines of this chapter are presented in Sect. IV.2. We first identify the rank defect in question. The minimum-constrained problem to be solved in the least-squares (LS) sense depends on some integer vector which can be fixed in an arbitrary manner. To compare the methods providing LS estimates of the clock-phase biases, we then introduce a particular solution playing the role of reference solution. For this solution, when a clock-phase bias is estimated for the first time, its fractional part is confined to the one-cycle width interval centred on zero; the integer-ambiguity function is modified accordingly. Section IV.3

is devoted to the algebraic framework of our analysis. This framework mainly derives from the original contributions of Lannes & Gratton (2009), and Lannes & Teunissen (2011). As a similar problem arises in phase-closure imaging in astronomy, we also took profit of the analysis presented in Sect. II. An efficient way for finding the reference solution is to adopt an approach based on the notion of closure ambiguity. The principle of the corresponding 'closure-ambiguity approach' (CAA) is defined in that framework (Sect. IV.4). The bulk of our contribution follows the main theoretical guidelines presented in Sect. IV.2. In a related option which is presented in Sect. IV.5, the CAA principle is directly introduced via the S-system approach of Baarda (1973); Teunissen (1984); de Jonge (1998). The corresponding development is performed in the S-system framework defined in H. The study developed in Sects. 3.3, 3.4 and 3.15 of de Jonge (1998) is thus extended to the cases where the union of the graphs $\mathscr{H}_\kappa$ is taken into account progressively. Section IV.6 is devoted to the QR implementation of the CAA principle; related information is to be found in I. In many methods, the rank defect in question is removed in an implicit manner or intuitively. In Sect. IV.7, on the grounds of some results established in Sects. IV.3.5 and IV.3.6, we identify the related constraints explicitly, and thus establish the link between the solutions provided by those methods and the CAA-(S-system) solutions; see Fig. IV.6 in particular. Our analysis is illustrated with some simple and generic examples. It could have applications in data processing of most GNSS networks, and particularly global networks using GPS, GLONASS, Galileo, IRNSS or BeiDou/Compass satellites. The new results provided by this study are commented upon in Sect. V; some conclusions are also presented with possible applications to software packages used for processing GNSS networks.

IV.2 Theoretical guidelines

The problem is formulated in Sect. IV.2.1; the related rank defect is identified in Sect. IV.2.2. This rank defect can be removed by imposing some constraints without affecting the GNSS results such as the estimates of the station-position parameters, for example. The particular LS solutions thus obtained are defined in Sect. IV.2.3. We then define the family of those solutions (Sect. IV.2.4). To compare the particular solutions given by the various GNSS methods providing LS estimates of clock-phase biases, we then introduce a particular solution playing the role of reference solution (Sect. IV.2.5).

IV.2.1 Formulation of the problem

In our formulation of the problem, the ambiguity function N 'includes' all the integer ambiguities $N(i,j)$ involved in the phase measurements until the current epoch. This function therefore takes its values on the edges of

$$\mathscr{G}_k \stackrel{\text{def}}{=} \bigcup_{\kappa=1}^{k} \mathscr{H}_\kappa \tag{IV.2}$$

where $\mathscr{H}_\kappa$ is the observational graph of epoch κ. In what follows, $\mathcal{H}_\kappa$ denotes the 'characteristic function' of $\mathscr{H}_k$ with regard to $\mathscr{G}_k$:

$$\text{for all } (i,j) \in \mathscr{G}_k, \quad \mathcal{H}_k(i,j) \stackrel{\text{def}}{=} \begin{vmatrix} 1 & \text{if } (i,j) \in \mathscr{H}_k; \\ 0 & \text{otherwise.} \end{vmatrix} \tag{IV.3}$$

The number of edges $(\mathrm{r}_i, \mathrm{s}_j)$ of $\mathscr{H}_\kappa$ is denoted by $n_{\mathrm{e}\kappa}$; $n_{\mathrm{e}\kappa}$ is less than or equal to the number of edges of $\mathscr{G}_k$.

To illustrate our analysis, we consider a 'simulated network' including four receivers and five to eight satellites; see Fig. IV.1. The scenarios of the first three epochs are defined by the characteristic functions $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$ displayed in that figure. While looking simple at first sight, this example is rather elaborate. Indeed, it includes the case of the appearance of new satellites in the field of view of the network (s_6 and s_7 at epoch 2, s_8 at epoch 3), and also the case of the disappearance of one satellite (s_3 at epoch 3).

Remark.2.5: When a satellite comes back in the field of view of the network, it is dealt with as a new satellite. In the case of global networks, if need be, the successive passes are thus dealt with in a simple and efficient manner □

At epoch $\kappa \leq k$, only some components of N may be active; see Fig. IV.1. To formalize this point, we introduce the operator $\mathcal{R}^{\mathrm{e}}_\kappa$ that restricts N (which is defined on the edges of $\mathscr{G}_k$) to the edges of $\mathscr{H}_\kappa$:

$$\text{for all } (i,j) \in \mathscr{H}_\kappa, \quad (\mathcal{R}^{\mathrm{e}}_\kappa N)(i,j) \stackrel{\mathrm{def}}{=} N(i,j) \tag{IV.4}$$

Equation (IV.1) can then be written in the form

$$\left| \begin{array}{l} B_\kappa \varphi_\kappa + \mathcal{R}^{\mathrm{e}}_\kappa N = b_\kappa \\ \text{for } \kappa = 1, \ldots, k \end{array} \right. \tag{IV.5}$$

where B_κ is the following bias operator:

$$(B_\kappa \varphi_\kappa)(i,j) \stackrel{\mathrm{def}}{=} \varphi_{\mathrm{r}\kappa}(i) - \varphi_{\mathrm{s}\kappa}(j) \quad (\text{for all } (i,j) \in \mathscr{H}_\kappa) \tag{IV.6}$$

In what follows, we will assume that Receiver 1 defines the reference for the receiver and satellite biases:

$$\varphi_{\mathrm{r}\kappa}(1) = 0 \qquad (\kappa = 1, \ldots, k) \tag{IV.7}$$

This is commonly used by the GNSS investigators for removing the rank defect of operators such as B_κ. The number $n_{\mathrm{b}\kappa}$ of phase biases of epoch κ to be estimated is therefore equal to $n_{\mathrm{v}\kappa} - 1$ where $n_{\mathrm{v}\kappa}$ is the number of vertices of $\mathscr{H}_\kappa$:

$$n_{\mathrm{b}\kappa} = n_{\mathrm{v}\kappa} - 1 \qquad (n_{\mathrm{v}\kappa} = n_{\mathrm{r}\kappa} + n_{\mathrm{s}\kappa}) \tag{IV.8}$$

With regard to its functional variables $\varphi_1, \ldots, \varphi_k$ and N, Eq. (IV.5) proves to have a basic rank defect. We now specify this point.

IV.2.2 Identification of the rank defect

For clarity, let us set

$$\mathscr{G} \stackrel{\mathrm{def}}{=} \mathscr{G}_k \tag{IV.9}$$

At epoch k, the number of ambiguities $N(i,j)$ involved in the problem is equal to the number of edges of $\mathscr{G}$ (for example twenty in Fig. IV.1 for $k = 3$). Again, for clarity, this number is simply denoted by n_{e}. We then set

$$n_{\mathrm{st}} = n_{\mathrm{v}} - 1 \qquad (n_{\mathrm{v}} = n_{\mathrm{r}} + n_{\mathrm{s}}) \tag{IV.10}$$

where n_{v} is the number of vertices of $\mathscr{G}$; n_{r} and n_{s} are the number of receivers and satellites (respectively) involved in that graph (four and eight in Fig. IV.1 for $k = 3$). As specified in Sect. G.2, n_{st} is the number of edges of any spanning tree $\mathscr{G}_{\mathrm{st}}$ of $\mathscr{G}$. The total number of phase biases to be estimated at epoch k, $\sum_{\kappa=1}^{k} n_{\mathrm{b}\kappa}$, is generally much larger than n_{st}; see Eqs. (IV.8) and (IV.10). The part played by the vertices of $\mathscr{G}$ is not obvious. We now show that n_{st} defines the 'size' of the rank defect in question.

Let us denote by B the operator from $\mathbb{R}^{n_{\mathrm{st}}}$ into $\mathbb{R}^{n_{\mathrm{e}}}$ defined by the relation

$$(B\phi)(i,j) \overset{\text{def}}{=} \phi_{\mathrm{r}}(i) - \phi_{\mathrm{s}}(j) \qquad (\text{for all } (i,j) \in \mathscr{G}) \tag{IV.11}$$

Denoting by μ any integer-valued function taking its values on the vertices of $\mathscr{G}$ other than the reference receiver, we have

$$\mathcal{R}^{\mathrm{e}}_{\kappa} B\mu = B_{\kappa} \mathcal{R}^{\mathrm{v}}_{\kappa} \mu \tag{IV.12}$$

where $\mathcal{R}^{\mathrm{v}}_{\kappa}\mu$ is the restriction of μ to the vertices of $\mathscr{H}_{\kappa}$ (other than the reference receiver). Note that μ can be regarded as a vector of $\mathbb{Z}^{n_{\mathrm{st}}}$. It then follows from Eq. (IV.5) that for any μ in $\mathbb{Z}^{n_{\mathrm{st}}}$,

$$\left| \begin{array}{l} B_{\kappa}(\varphi_{\kappa} + \mathcal{R}^{\mathrm{v}}_{\kappa}\mu) + \mathcal{R}^{\mathrm{e}}_{\kappa}(N - B\mu) = b_{\kappa} \\ \text{for } \kappa = 1, \ldots, k \end{array} \right. \tag{IV.13}$$

Via the operators B_{κ}, $\mathcal{R}^{\mathrm{v}}_{\kappa}$ and $\mathcal{R}^{\mathrm{e}}_{\kappa}$, any variation of the 'vertex-ambiguity' vector μ can thus be compensated by a variation of the 'edge-ambiguity' function N. As a result, with regard to the bias and ambiguity variables, Eq. (IV.5) is not of full rank. The dimension of the rank defect is equal to that of vector μ, i.e., n_{st}.

IV.2.3 Particular LS solutions

In GNSS, for the reasons specified in 2.6 (at the end of this section), each clock-phase bias is to be estimated up to a constant integer. As a result, the choice of μ in $\mathbb{Z}^{n_{\mathrm{st}}}$ does not affect the significant part of the values of the bias functions

$$w_{\kappa} \overset{\text{def}}{=} \varphi_{\kappa} + \mathcal{R}^{\mathrm{v}}_{\kappa}\mu \qquad (\kappa = 1, \ldots, k) \tag{IV.14}$$

to be estimated; see Eq. (IV.13). The ambiguity function to be retrieved

$$v \overset{\text{def}}{=} N - B\mu \tag{IV.15}$$

is of course affected by this choice, but this has no actual GNSS impact. As a result, the GNSS methods providing estimates of the clock-phase biases must remove the rank defect of Eq. (IV.5) by choosing μ in $\mathbb{Z}^{n_{\mathrm{st}}}$ somehow, implicitly or explicitly.

In practice, as clarified in the remainder of the paper, removing this rank defect amounts to imposing n_{st} constraints on some values of the biases or ambiguities to be retrieved. In other words μ is defined via these constraints. The minimum-constrained problem to be solved in the LS sense is therefore of the form

$$\left| \begin{array}{l} B_{\kappa} w_{\kappa} + \mathcal{R}^{\mathrm{e}}_{\kappa} v = b_{\kappa} \qquad (\kappa = 1, \ldots, k) \\ \text{subject to } n_{\mathrm{st}} \text{ constraints on } w_{\kappa} \text{ or } v \end{array} \right. \tag{IV.16}$$

With regard to a particular set of such constraints, where v is an integer-valued function from Eq. (IV.15), the LS solution of Eq. (IV.16),

$$(\check{w}_\kappa,\ \check{v}) \qquad (\kappa = 1, \ldots, k) \tag{IV.17}$$

is then unique. For example, the solution provided by the CAA method defined in Sect. IV.4 is the particular LS solution obtained by imposing the *a priori* constraint $v = 0$ on a spanning tree of $\mathscr{G}$ (chosen arbitrarily). The particular LS solution introduced in Sect. IV.2.5 is defined by imposing, *a posteriori*, $n_{\rm st}$ constraints on some bias values. In our analysis, this particular solution plays the role of reference solution; it is denoted by $(\bar{w}_\kappa,\ \bar{v})$.

Remark.2.6: The satellite components of the biases thus obtained (for example those of the reference solution) can be broadcasted to the network users for PPP applications. The fact that $\check{w}_{{\rm s}\kappa}(j)$ is an LS estimate of $\varphi_{{\rm s}\kappa}(j)$ up to some unknown constant integer does not raise any difficulty. One is then simply led to redefine the integer ambiguities involved in the PPP problem to be solved; see, e.g., Sect. 9 in Lannes & Teunissen (2011) □

IV.2.4 Equivalent LS solutions

Given some particular LS solution such as (IV.17), we have

$$B_\kappa \check{w}_\kappa + \mathcal{R}^{\rm e}_\kappa \check{v} \overset{\rm LS}{=} b_\kappa$$

Like for Eq. (IV.13), it then follows from Eq. (IV.12) that

$$B_\kappa(\check{w}_\kappa + \mathcal{R}^{\rm v}_\kappa \mu) + \mathcal{R}^{\rm e}_\kappa(\check{v} - B\mu) \overset{\rm LS}{=} b_\kappa$$

The LS solutions of Eq. (IV.5) are therefore of the form

$$(\check{w}^{(\mu)}_\kappa,\ \check{v}^{(\mu)}) \qquad (\kappa = 1, \ldots, k) \tag{IV.18}$$

with

$$\check{w}^{(\mu)}_\kappa \overset{\rm def}{=} \check{w}_\kappa + \mathcal{R}^{\rm v}_\kappa \mu, \qquad \check{v}^{(\mu)} \overset{\rm def}{=} \check{v} - B\mu \tag{IV.19}$$

where μ is any vector of $\mathbb{Z}^{n_{\rm st}}$.

The methods providing LS estimates of the phase biases generally differ by the choice of the imposed constraints. To compare their results, it is convenient to represent the equivalent solutions (IV.18)-(IV.19) by a reference particular solution. This is done in Sect. IV.2.5.

Remark.2.7: For any fixed μ in $\mathbb{Z}^{n_{\rm st}}$, the temporal variations of the estimated phase biases make sense. For example, if satellite ${\rm s}_j$ remains in the field of view of the network from epoch 1 to κ, we have

$$(\mathcal{R}^{\rm v}_\kappa \mu)_{\rm s}(j) = (\mathcal{R}^{\rm v}_1 \mu)_{\rm s}(j) = \mu_{\rm s}(j)$$

hence from Eqs. (IV.19) and (IV.14) ,

$$\check{w}^{(\mu)}_{{\rm s}\kappa}(j) - \check{w}^{(\mu)}_{\rm s1}(j) = \check{w}_{{\rm s}\kappa}(j) - \check{w}_{\rm s1}(j) \simeq \varphi_{{\rm s}\kappa}(j) - \varphi_{\rm s1}(j)$$

A similar result of course holds for the receiver clock-phase biases □

IV.2.5 Reference solution

We here concentrate on the family of equivalent LS solutions (IV.18)-(IV.19) generated by a particular solution $(\check{w}_\kappa,\, \check{v})$ such as (IV.17). In our analysis, the reference solution of this family is the particular solution

$$(\bar{w}_\kappa,\, \bar{v}) \qquad (\kappa = 1, \ldots, k) \tag{IV.20}$$

defined as follows: $\bar{w}_\kappa$ and $\bar{v}$ are of the form (IV.19)

$$\bar{w}_\kappa \overset{\text{def}}{=} \check{w}_\kappa + \mathcal{R}^{\text{v}}_\kappa \check{\mu}, \qquad \bar{v} \overset{\text{def}}{=} \check{v} - B\check{\mu} \tag{IV.21}$$

in which $\check{\mu}$ is defined by imposing specific constraints on n_{st} bias values; note that here, these constraints are imposed *a posteriori* on the solution $(\check{w}_\kappa,\, \check{v})$ found by any method. We first require the phase bias $\bar{w}$ to be small at epoch 1. More precisely, we impose the condition $|\bar{w}_1| \leq 1/2$, i.e. explicitly,

$$\left| \begin{array}{ll} |\bar{w}_{\text{r}1}(i)| \leq 1/2 & \text{for } i = 2, \ldots, n_{\text{r}1} \\ |\bar{w}_{\text{s}1}(j)| \leq 1/2 & \text{for } j = 1, \ldots, n_{\text{s}1} \end{array} \right. \tag{IV.22}$$

The following values of $\check{\mu}$ are defined accordingly:

$$\left| \begin{array}{ll} \check{\mu}_{\text{r}1}(i) := -\lfloor \check{w}_{\text{r}1}(i) \rceil & \text{for } i = 2, \ldots, n_{\text{r}1} \\ \check{\mu}_{\text{s}1}(j) := -\lfloor \check{w}_{\text{s}1}(j) \rceil & \text{for } j = 1, \ldots, n_{\text{s}1} \end{array} \right. \tag{IV.23}$$

Here, $\lfloor x \rceil$ denotes the integer closest to x. Likewise, at each epoch κ when some satellite(s) s_j appear(s) in the field of view of the network (see Fig. IV.1), we then impose the condition(s)

$$|\bar{w}_{\text{s}\kappa}(j)| \leq 1/2 \tag{IV.24}$$

by setting

$$\check{\mu}_{\text{s}\kappa}(j) := -\lfloor \check{w}_{\text{s}\kappa}(j) \rceil \tag{IV.25}$$

(In the case where new receivers would be activated, similar conditions would be imposed.) At epoch k, we have thus completely defined some vector $\check{\mu}$ of $\mathbb{Z}^{n_{\text{st}}}$; $\bar{v}$ is then obtained via the relation $\bar{v} := \check{v} - B\check{\mu}$; see Eq. (IV.21).

Remark.2.8: When some LS solution $(\check{w}_1, \ldots, \check{w}_k,\, \check{v})$ has been found, for instance that provided by the CAA method defined in Sects. IV.3 to IV.6, the reference solution of its equivalent solutions is obtained as described in this section. Clearly, this can also be done for the LS solution of any method providing estimates of the phase biases; see, e.g., Blewitt (1989); Ge et al. (2005); Laurichesse & Mercier (2007); Collins et al. (2010); Loyer et al. (2012). To compare and validate the results provided by all these methods (and many others), one may inspect the ambiguity functions of their reference solutions. These functions should be identical on all the edges of $\mathscr{G}$ for all methods; otherwise, this would be an indication that the methods are in disagreement, and that some of those results are wrong. The comparison of the reference solutions is therefore a good diagnosis for testing the compatibility of these methods ⊡

Remark.2.9: From a technical point of view, one might try to solve Eq. (IV.16) in the LS sense by imposing the nonlinear bias constraints (IV.22) and (IV.24) on w_1 and some w_κ, from the outset. It is not easy at all to solve the problem that way. Moreover, the number of edge ambiguities to be fixed would then be equal to n_{e}, whereas the number of ambiguities to be fixed in the CAA approach (for example) is equal to $n_{\text{e}} - n_{\text{st}}$; see the next section ⊡

IV.3 Algebraic framework

The preliminary analysis developed in Sect. IV.2 shows that $\mathbb{Z}^{n_{\rm st}}$, $\mathbb{Z}^{n_{\rm e}}$, graph $\mathscr{G}$ and operator B play a key role in the formulation of the problem and the definition of its solutions; see, in particular, Eqs. (IV.13) and (IV.18)-(IV.19). The aim of this section is to define the corresponding algebraic framework.

We first define related spaces of functions (Sect. IV.3.1). The key property on which our analysis is based is presented in Sect. IV.3.2. The related notions of closure difference, CD ambiguity (also called closure ambiguity), and closure matrix are specified in Sects. IV.3.3 and IV.3.4. Sections IV.3.5 and IV.3.6 are devoted to some generalized inverses of the UD-CD and UD-DD relationships. The analysis concerning the operator D introduced in the appendix B of Blewitt (1989) is thus completed. We now draw freely from the elementary notions introduced in G.

IV.3.1 Reference spaces

Given some graph $\mathscr{G} \equiv \mathscr{G}(\mathscr{V}, \mathscr{E})$, with vertex set $\mathscr{V}$ and edge set $\mathscr{E}$ (see Sect. G.1), we introduce some functionals spaces which play a key role in the algebraic analysis of the problem. In what follows, the GNSS grid associated with $\mathscr{G}$ is denoted by G; see Fig. G.1.

Vertex-bias space

Let V_b be the space of real-valued functions

$$\phi \stackrel{\rm def}{=} (\phi_{\rm r}, \phi_{\rm s}) \tag{IV.26}$$

taking their values on the vertices of $\mathscr{G}$ with $\phi_{\rm r}(1) = 0$. This space, which is referred to as the vertex-bias space, is associated with the definition of (virtual) phase biases ϕ on the vertices of $\mathscr{G}$ (other than the reference receiver). From Eq. (IV.10),

$$V_b \cong \mathbb{R}^{n_{\rm st}} \tag{IV.27}$$

Here, the symbol $\cong$ means 'isomorphic to.' Note that $\mathbb{Z}^{n_{\rm st}}$ is the 'integer lattice' of V_b: $V_b(\mathbb{Z}) \cong \mathbb{Z}^{n_{\rm st}}$. The integer vector $\mu \stackrel{\rm def}{=} (\mu_{\rm r}, \mu_{\rm s})$ is a point of this lattice.

Edge-delay space

A real-valued function ϑ taking its values on G, and thereby on $\mathscr{E}$, can be regarded as a vector of

$$E \cong \mathbb{R}^{n_{\rm e}} \tag{IV.28}$$

The values of ϑ on G are then regarded as the components of ϑ in the standard basis of E; $\mathbb{Z}^{n_{\rm e}}$ is the 'integer lattice' of E: $E(\mathbb{Z}) \cong \mathbb{Z}^{n_{\rm e}}$. The integer-ambiguity function N is a point of this lattice.

Spanning-tree delay space. Closure-delay space

Given some spanning tree $\mathscr{G}_{\rm st}$ of $\mathscr{G}$, grid G can be decomposed into two subgrids: $\mathrm{G}_{\rm st}$ and $\mathrm{G}_{\rm c}$; see Sect. G.2. These grids include $n_{\rm st}$ and $n_{\rm c}$ points, respectively (see Fig. G.2):

$$n_{\rm c} = n_{\rm e} - n_{\rm st} \tag{IV.29}$$

The functions of E that vanish on $\mathrm{G}_{\rm c}$ form a subspace of E denoted by $E_{\rm st}$: the spanning-tree delay space. Likewise, the functions of E that vanish on $\mathrm{G}_{\rm st}$ form a subspace of E denoted by $E_{\rm c}$: the closure-delay space; this terminology is justified in Sect. IV.3.3. The corresponding integer lattices are denoted by $E_{\rm st}(\mathbb{Z})$ and $E_{\rm c}(\mathbb{Z})$, respectively. As illustrated in Fig. IV.2, E is the orthogonal sum of $E_{\rm st}$ and $E_{\rm c}$, and we thus have

$$\dim E_{\rm st} = n_{\rm st}, \qquad \dim E_{\rm c} = n_{\rm c} \tag{IV.30}$$

The projections of ϑ on $E_{\rm st}$ and $E_{\rm c}$ are respectively denoted by $Q_{\rm st}\vartheta$ and $Q_{\rm c}\vartheta$.

Edge-bias space

By definition, the bias operator is the operator from V_b into E defined by Eq. (IV.11). The range of B, which is denoted by $E_{\rm b}$ (see Fig. IV.2), can be referred to as the edge-bias space. Its functions are of the form $\phi_{\rm r}(i) - \phi_{\rm s}(j)$.

The operator from V_b into $E_{\rm st}$ induced by B is denoted by $B_{\rm st}$. Likewise, the operator from V_b into $E_{\rm c}$ induced by B is denoted by $B_{\rm c}$.

The matrix of B is generally expressed in the standard bases of V_b and E. For example, let us sort the edges of the graph shown in Fig. G.1 in the order obtained via the application of the Kruskal algorithm; see Sect. G.2. The points of G are then ordered as follows:

$$\mathbf{(1,1)},\ \mathbf{(1,3)},\ \mathbf{(1,4)},\ \mathbf{(2,1)},\ \mathbf{(2,2)},\ \mathbf{(3,2)},\ (2,4),\ (3,3),\ (3,4)$$

We then have

$$[B][\phi] = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \phi_{\rm r}(2) \\ \phi_{\rm r}(3) \\ \phi_{\rm s}(1) \\ \phi_{\rm s}(2) \\ \phi_{\rm s}(3) \\ \phi_{\rm s}(4) \end{bmatrix}$$

The columns of $[B]$ then define the standard basis of $E_{\rm b}$. Clearly,

$$[B_{\rm st}] = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{IV.31}$$

and

$$[B_{\mathrm{c}}] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{IV.32}$$

The condition $B_{\mathrm{st}}\phi = 0$, i.e., $B\phi = 0$ on the edges of $\mathscr{G}_{\mathrm{st}}$, implies that ϕ is constant on $\mathscr{V}$; as $\phi_{\mathrm{r}}(1) = 0$, this constant is zero. The null space of B_{st} is therefore reduced to $\{0\}$. As $B\phi = 0$ implies $B_{\mathrm{st}}\phi = 0$, the null space of B is also reduced to $\{0\}$. We thus have

$$\ker B = \ker B_{\mathrm{st}} = \{0\} \tag{IV.33}$$

As a result, B is of full rank, hence from Eq. (IV.27),

$$\dim E_{\mathrm{b}} = n_{\mathrm{st}} \tag{IV.34}$$

The edge-bias space E_{b} and its ambiguity lattice $E_{\mathrm{b}}(\mathbb{Z})$ are isomorphic to the (vertex-bias space) V_b and (its integer lattice) $V_b(\mathbb{Z})$, respectively; see Sect. IV.3.1.

IV.3.2 Key property

As $\ker B_{\mathrm{st}} = \{0\}$ (Eq. (IV.33)), and $\dim E_{\mathrm{st}} = \dim V_b$ (see Eqs. (IV.30) and (IV.27)), B_{st} maps V_b onto E_{st}; B_{st} is therefore invertible. As specified in this section, our analysis derives from this property.

Let us concentrate on the vertex-bias function

$$\phi_{\mathrm{st}}^{(\vartheta)} \stackrel{\mathrm{def}}{=} B_{\mathrm{st}}^{-1} Q_{\mathrm{st}} \vartheta \qquad (\phi^{(\vartheta)} \equiv \phi_{\mathrm{st}}^{(\vartheta)}) \tag{IV.35}$$

When no confusion may arise, subscript st is omitted. According to its definition (which is illustrated in Fig. IV.2), $Q_{\mathrm{st}}\vartheta$ is the function of E_{st} whose values are those of ϑ on subgrid G_{st}.

The values of $\phi^{(\vartheta)}$ can be obtained from those of $Q_{\mathrm{st}}\vartheta$ in a very simple manner; the corresponding recursive process is described in Sect. 5 of Lannes & Teunissen (2011). The column vectors of $[B_{\mathrm{st}}]^{-1}$ can thus be easily obtained. In fact, $[B_{\mathrm{st}}]$ is a particular unimodular matrix[2] whose inverse can be obtained via another integer-programming technique; see Sect. A1.4 in Lannes & Teunissen (2011). For example, the inverse of matrix (IV.31) is

$$[B_{\mathrm{st}}]^{-1} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \tag{IV.36}$$

Let us now consider the following function of E_{b}:

$$\vartheta_{\mathrm{b}} \stackrel{\mathrm{def}}{=} B\phi^{(\vartheta)} \tag{IV.37}$$

[2] By definition, a unimodular matrix is a square integer matrix with determinant ± 1.

According to Eq. (IV.35), the values of $\vartheta_{\rm b}$ and ϑ coincide on $\mathrm{G}_{\rm st}$. The function $\vartheta_{\rm c}$ defined by the relation

$$\vartheta_{\rm c} \stackrel{\rm def}{=} \vartheta - \vartheta_{\rm b} \tag{IV.38}$$

therefore lies in $E_{\rm c}$. We thus have the following property (see Fig. IV.2):

Property P1. *Any edge function ϑ of E can be decomposed in the form $\vartheta = \vartheta_{\rm b} + \vartheta_{\rm c}$ with $\vartheta_{\rm b} \stackrel{\rm def}{=} B\phi^{(\vartheta)}$ and $\vartheta_{\rm c}$ in $E_{\rm c}$. For a given spanning tree, this decomposition is unique. As a corollary, E is the oblique direct sum of $E_{\rm b}$ and $E_{\rm c}$: $E = E_{\rm b} + E_{\rm c}$ with $E_{\rm b} \cap E_{\rm c} = \{0\}$.*

As illustrated in Fig. IV.2, $\vartheta_{\rm c}$ is the oblique projection of ϑ on $E_{\rm c}$ along $E_{\rm b}$. The corresponding operator is the 'closure operator' C:

$$\vartheta_{\rm c} = C\vartheta \tag{IV.39}$$

Its null space (i.e., its kernel) is the range of B:

$$\ker C = E_{\rm b} \tag{IV.40}$$

with $\dim E_{\rm b} = n_{\rm st}$ (Eq. (IV.34)).

According to Property P1, any function N of the 'ambiguity lattice' $E(\mathbb{Z}) \cong \mathbb{Z}^{n_{\rm e}}$ can be decomposed in the form

$$N = N_{\rm b} + N_{\rm c} \tag{IV.41}$$

with $N_{\rm b} \stackrel{\rm def}{=} B\mu_{\rm st}^{(N)}$ where (from Eq. (IV.35))

$$\mu_{\rm st}^{(N)} \stackrel{\rm def}{=} B_{\rm st}^{-1} Q_{\rm st} N \qquad (\mu^{(N)} \equiv \mu_{\rm st}^{(N)}) \tag{IV.42}$$

As $B_{\rm st}$ is unimodular, $\mu^{(N)}$ is an integer-valued function; $N_{\rm b} \stackrel{\rm def}{=} B\mu^{(N)}$ and $N_{\rm c} \stackrel{\rm def}{=} CN$ are therefore points of the integer lattices $E_{\rm b}(\mathbb{Z}) \cong \mathbb{Z}^{n_{\rm st}}$ and $E_{\rm c}(\mathbb{Z}) \cong \mathbb{Z}^{n_{\rm c}}$, respectively. As a result, the integer lattice $E(\mathbb{Z})$ is the oblique direct sum of the integer lattices $E_{\rm b}(\mathbb{Z})$ and $E_{\rm c}(\mathbb{Z})$:

$$E(\mathbb{Z}) = E_{\rm b}(\mathbb{Z}) + E_{\rm c}(\mathbb{Z}) \qquad E_{\rm b}(\mathbb{Z}) \cap E_{\rm c}(\mathbb{Z}) = \{0\} \tag{IV.43}$$

IV.3.3 Closure delays (closure differences) and closure ambiguities

According to Eqs. (IV.38) and (IV.37), the quantities $\vartheta_{\rm c}(i_\ell, j_\ell)$, for $\ell = 1, \ldots, n_{\rm c}$, can be computed via the formula

$$\vartheta_{\rm c}(i_\ell, j_\ell) = \vartheta(i_\ell, j_\ell) - \left[\phi^{(\vartheta)}(i_\ell) - \phi^{(\vartheta)}(j_\ell)\right] \tag{IV.44}$$

where $\phi^{(\vartheta)}$ is determined via Eq. (IV.35). As clarified in this section, these quantities can be referred to as the 'closure delays' or the 'closure differences' of ϑ; the $N_{\rm c}(i_\ell, j_\ell)$'s are therefore 'CD ambiguities,' also simply called 'closure ambiguities.'

In the example of Fig. G.2, let us consider the second loop, i.e., the loop associated with the closure point $(i_2, j_2) = (3, 3)$. In G, the successive points of this loop are the following:

$(3,3)$, $(3,2)$, $(2,2)$, $(2,1)$, $(1,1)$, and $(1,3)$. Since $\vartheta_{\mathrm{b}}(i,j) = \phi_{\mathrm{r}}^{(\vartheta)}(i) - \phi_{\mathrm{r}}^{(\vartheta)}(j)$, we then have, in a telescoping manner,

$$\vartheta_{\mathrm{b}}(3,3) - \vartheta_{\mathrm{b}}(3,2) + \vartheta_{\mathrm{b}}(2,2) - \vartheta_{\mathrm{b}}(2,1) + \vartheta_{\mathrm{b}}(1,1) - \vartheta_{\mathrm{b}}(1,3) = 0.$$

Furthermore, as ϑ_{c} vanishes on G_{st},

$$\vartheta_{\mathrm{c}}(3,3) - \vartheta_{\mathrm{c}}(3,2) + \vartheta_{\mathrm{c}}(2,2) - \vartheta_{\mathrm{c}}(2,1) + \vartheta_{\mathrm{c}}(1,1) - \vartheta_{\mathrm{c}}(1,3) = \vartheta_{\mathrm{c}}(3,3)$$

Since $\vartheta = \vartheta_{\mathrm{b}} + \vartheta_{\mathrm{c}}$ from Property P1, it follows that

$$\vartheta(3,3) - \vartheta(3,2) + \vartheta(2,2) - \vartheta(2,1) + \vartheta(1,1) - \vartheta(1,3) = \vartheta_{\mathrm{c}}(3,3)$$

This explicitly shows that $\vartheta_{\mathrm{c}}(i_2,j_2)$ can be regarded as the closure difference of ϑ on the second loop. The generalization is straightforward. In the example of Fig. G.2, we thus have

$$\left|\begin{aligned}
\vartheta_{\mathrm{c}}(2,4) &= \vartheta(2,4) - \vartheta(2,1) + \vartheta(1,1) - \vartheta(1,4) \\
\vartheta_{\mathrm{c}}(3,3) &= \vartheta(3,3) - \vartheta(3,2) + \vartheta(2,2) - \vartheta(2,1) \\
&\qquad + \vartheta(1,1) - \vartheta(1,3) \\
\vartheta_{\mathrm{c}}(3,4) &= \vartheta(3,4) - \vartheta(3,2) + \vartheta(2,2) - \vartheta(2,1) \\
&\qquad + \vartheta(1,1) - \vartheta(1,4)
\end{aligned}\right. \tag{IV.45}$$

More generally, owing to the telescoping structure of their construction, the closure differences $\vartheta_{\mathrm{c}}(i_\ell,j_\ell)$ are associated with loops whose order is even, and greater than or equal to 4. In this limit case, the notion of closure difference (CD) reduces to that of double difference (DD). According to Eq. (IV.44), the $\vartheta_{\mathrm{c}}(i_\ell,j_\ell)$'s can however be computed without knowing the edges of their loop. How to identify these edges, if need be, is specified in Sect. IV.3.4. Subject to some condition, these CD's can be expressed as linear combinations of DD's. The related matter is analyzed in Sect. 10 of Lannes & Teunissen (2011).

IV.3.4 Closure matrix

According to the definitions of B_{st} and B_{c} (introduced in Sect. IV.3.1), the vector $\vartheta_{\mathrm{b}} \overset{\mathrm{def}}{=} B\phi^{(\vartheta)}$ can be orthogonally decomposed in the form

$$\vartheta_{\mathrm{b}} = B_{\mathrm{st}}\phi^{(\vartheta)} + B_{\mathrm{c}}\phi^{(\vartheta)} = Q_{\mathrm{st}}\vartheta + B_{\mathrm{c}}\phi^{(\vartheta)}$$

Likewise,

$$\vartheta = Q_{\mathrm{st}}\vartheta + Q_{\mathrm{c}}\vartheta$$

where $Q_{\mathrm{c}}\vartheta$ is the orthogonal projection of ϑ on E_{c}; see Fig. IV.2. It then follows from Eq. (IV.35) that

$$\vartheta_c = \vartheta - \vartheta_{\mathrm{b}} = Q_{\mathrm{c}}\vartheta - B_{\mathrm{c}}\phi^{(\vartheta)} = Q_{\mathrm{c}}\vartheta - B_{\mathrm{c}}B_{\mathrm{st}}^{-1}Q_{\mathrm{st}}\vartheta$$

Denoting by $[C]$ the matrix of C expressed in the standard bases of E and E_c, we thus have, from Eq. (IV.39),

$$[C][\vartheta] = -[B_c][B_{st}]^{-1}[Q_{st}\vartheta] + [Q_c\vartheta].$$

The column vectors of $[C]$ corresponding to the spanning-tree edges (on which $Q_c\vartheta$ vanishes) are therefore those of $-[B_c][B_{st}]^{-1}$. It is also clear that the column vectors of $[C]$ corresponding to the closure edges (on which $Q_{st}\vartheta$ vanishes) are those of the identity matrix on E_c. Consequently, with regard to the orthogonal direct sum $E_{st} \oplus E_c$,

$$[C] = \begin{bmatrix} -[B_c][B_{st}]^{-1} & [I_{c,c}] \end{bmatrix} \tag{IV.46}$$

In the example of Fig. G.2, we thus have, from Eqs. (IV.31), (IV.36), and (IV.32), with the same edge ordering,

$$[B_c][B_{st}]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

As a result,

$$[C] = \begin{bmatrix} 1 & 0 & -1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 1 & -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & -1 & 1 & -1 & 0 & 0 & 1 \end{bmatrix}$$

Applied to $[\vartheta]$, this matrix of course yields Eq. (IV.45). More generally, the edges of a 'closure loop' are identified via the nonzero entries of the corresponding row of $[C]$. In fact, this is the most efficient way of identifying the loops in question. Note however that in the CAA method presented through Sects. IV.4 to IV.6, the action of this matrix is never explicitly performed.

IV.3.5 On some generalized inverse of the UD-CD relationship

The closure operator C, which is an oblique projection, is not of full rank. The simplest way of removing its rank defect is to introduce the operator $C_\diamond$ from E into $E_{st} \times E_c$

$$C_\diamond\vartheta \stackrel{\text{def}}{=} (Q_{st}\vartheta,\, C\vartheta) \tag{IV.47}$$

According to Property P1, $C_\diamond$ is invertible; this can be immediately understood from Fig. IV.2 for example; $C_\diamond^{-1}$ can then be regarded as some generalized inverse of C. We now specify this point, explicitly, in matrix terms. The corresponding development is aimed at analysing the approaches of Blewitt (1989) and Collins et al. (2010) in our algebraic framework; see Sects. IV.3.6 and IV.7 further on.

In the standard bases of $E = E_{st} \oplus E_c$ and $E_{st} \times E_c$, the matrix of $C_\diamond$ can be written in the form (see Eq. (IV.46))

$$[C_\diamond] \stackrel{\text{def}}{=} \begin{bmatrix} [Q_{st}] \\ [C] \end{bmatrix} = \begin{bmatrix} [I_{st,st}] & [0_{st,c}] \\ -[B_c][B_{st}]^{-1} & [I_{c,c}] \end{bmatrix} \tag{IV.48}$$

It is readily verified that

$$[C_\diamond]^{-1} = \begin{bmatrix} [I_{\mathrm{st}}] & [0_{\mathrm{st,c}}] \\ [B_{\mathrm{c}}][B_{\mathrm{st}}]^{-1} & [I_{\mathrm{c}}] \end{bmatrix} \tag{IV.49}$$

Given some point $\breve{N}_{\mathrm{st}}$ *arbitrarily fixed* in $E_{\mathrm{st}}(\mathbb{Z})$, let us now consider the ambiguity point $\breve{N}$ of $E(\mathbb{Z})$ defined by the relation

$$[\breve{N}] \overset{\mathrm{def}}{=} [C_\diamond]^{-1} \begin{bmatrix} [\breve{N}_{\mathrm{st}}] \\ [N_{\mathrm{c}}] \end{bmatrix} \qquad (N_{\mathrm{c}} \overset{\mathrm{def}}{=} CN) \tag{IV.50}$$

In the following property,

$$E_{\mathrm{b}}^{(N)}(\mathbb{Z}) \overset{\mathrm{def}}{=} N + E_{\mathrm{b}}(\mathbb{Z}) \tag{IV.51}$$

is the 'affine lattice' passing through N and parallel to the integer lattice $E_{\mathrm{b}}(\mathbb{Z})$ of the edge-bias space E_{b}; see Sect. IV.3.1 and Fig. IV.3.

Property P2. *The ambiguity point $\breve{N}$ is the point of the affine lattice $E_{\mathrm{b}}^{(N)}(\mathbb{Z})$ whose projection on E_{st} is equal to $\breve{N}_{\mathrm{st}}$. More precisely, $\breve{N} = N_{\mathrm{c}} + B\mu^{(\breve{N}_{\mathrm{st}})}$. As a corollary, in the special case where $\breve{N}_{\mathrm{st}}$ is set equal to 0, $\breve{N}$ is nothing else than N_{c}.*

Proof. From Eqs. (IV.50) and (IV.49), we have

$$[\breve{N}] = \begin{bmatrix} [\breve{N}_{\mathrm{st}}] \\ [B_{\mathrm{c}}][B_{\mathrm{st}}]^{-1}[\breve{N}_{\mathrm{st}}] + [N_{\mathrm{c}}] \end{bmatrix}$$

hence, from Eq. (IV.46),

$$[\breve{N}] = \begin{bmatrix} [\breve{N}_{\mathrm{st}}] \\ -[C\breve{N}_{\mathrm{st}}] + [N_{\mathrm{c}}] \end{bmatrix}$$

Clearly, $Q_{\mathrm{st}}\breve{N} = \breve{N}_{\mathrm{st}}$. As from Property P1,

$$\breve{N}_{\mathrm{st}} = B\mu^{(\breve{N}_{\mathrm{st}})} + C\breve{N}_{\mathrm{st}}$$

where $\mu^{(\breve{N}_{\mathrm{st}})} = B_{\mathrm{st}}^{-1}\breve{N}_{\mathrm{st}}$ from Eq. (IV.42), we have

$$\begin{bmatrix} [\breve{N}_{\mathrm{st}}] \\ -[C\breve{N}_{\mathrm{st}}] \end{bmatrix} = [B\mu^{(\breve{N}_{\mathrm{st}})}]$$

As a result,

$$\breve{N} = B\mu^{(\breve{N}_{\mathrm{st}})} + N_{\mathrm{c}}$$

When $\breve{N}_{\mathrm{st}} := 0$, $\breve{N}$ therefore reduces to N_{c}. □

IV.3.6 On the Blewitt generalized inverse of the UD-DD relationship

We now apply the results of the previous section to the UD-DD relationship, and thus make the link with the approach of Blewitt (1989).

According to Eq. (68) of Lannes & Teunissen (2011), the maximum number of independent DD's is less than or equal to n_c: $n_d^m \leq n_c$. For clarity, let us now set $n_d := n_d^m$. In the important special case where

$$n_c = n_d \tag{IV.52}$$

the information contained in the DD data is equivalent to that contained in the closure data. Let us then denote by $D_{d,e}$ the operator providing a maximum set of n_d DD's. By definition, $D_{d,e}$ is an operator from E into $\mathbb{R}^{n_d}$, i.e. then, $\mathbb{R}^{n_c}$. By sorting the edges of $\mathscr{G}$ as specified in Sect. IV.3.1, the matrix of $D_{d,e}$ has then the following block structure:

$$[D_{d,e}] = \begin{bmatrix} [D_{d,st}] & [D_{d,c}] \end{bmatrix} \tag{IV.53}$$

Here, matrix $[D_{d,e}]$ is expressed in the standard bases of $E = E_{st} \oplus E_c$ and $\mathbb{R}^{n_d}$. The columns of $[D_{d,st}]$ and $[D_{d,c}]$ therefore correspond to the edges of $\mathscr{G}_{st}$ and to the closure edges, respectively. Provided that Condition (IV.52) is satisfied, $[D_{d,c}]$ is invertible; moreover, the entries of $[D_{d,c}]^{-1}$ are then equal to ± 1 or 0; see Lannes & Teunissen (2011).

Like for C (see Eq. (IV.48)), we then introduce the operator

$$[D_\diamond] \overset{\text{def}}{=} \begin{bmatrix} [Q_{st}] \\ [D_{d,e}] \end{bmatrix} \tag{IV.54}$$

As $N = N_b + CN$ from Property P1, and $[D_{d,e}][N_b] = 0$, we have, from Eq. (IV.53),

$$[N_d] \overset{\text{def}}{=} [D_{d,e}][N] = [D_{d,c}][CN] \tag{IV.55}$$

It then follows from Eq. (IV.48) that

$$[D_\diamond] = \begin{bmatrix} [I_{st}] & [0_{st,c}] \\ [0_{c,st}] & [D_{d,c}] \end{bmatrix} [C_\diamond] \tag{IV.56}$$

We then have

$$\begin{aligned} [D_\diamond]^{-1} \begin{bmatrix} [\breve{N}_{st}] \\ [N_d] \end{bmatrix} &= [C_\diamond]^{-1} \begin{bmatrix} [I_{st}] & [0_{st,c}] \\ [0_{c,st}] & [D_{d,c}]^{-1} \end{bmatrix} \begin{bmatrix} [\breve{N}_{st}] \\ [N_d] \end{bmatrix} \\ &= [C_\diamond]^{-1} \begin{bmatrix} [\breve{N}_{st}] \\ [D_{d,c}]^{-1}[N_d] \end{bmatrix} \end{aligned}$$

where $[D_{d,c}]^{-1}[N_d] = [CN]$ from Eq. (IV.55). It then follows from Eq. (IV.50) that

$$[D_\diamond]^{-1} \begin{bmatrix} [\breve{N}_{st}] \\ [N_d] \end{bmatrix} = [\breve{N}] \tag{IV.57}$$

Given some DD ambiguity set (IV.55), and some point $\breve{N}_{\rm st}$ arbitrarily fixed in $E_{\rm st}(\mathbb{Z})$, the UD ambiguity thus obtained is equal to $N_{\rm c} + B\mu^{(\breve{N}_{\rm st})}$; see Property P2 and Fig. IV.3. Via the action of $[D_\diamond]^{-1}$, we thus retrieve the closure-ambiguity point $N_{\rm c}$ up to the vector $B\mu^{(\breve{N}_{\rm st})}$ of $E_{\rm b}(\mathbb{Z})$.

In fact, $[D_\diamond]$ is a version of the D-matrix of Blewitt (1989). In the appendix B of that paper, the spanning tree $\mathscr{G}_{\rm st}$ is implicitly defined by arbitrarily selecting a set of '$n - m$ undifferenced biases which pass the Gram-Schmidt test' of that appendix, i.e., in our algebraic framework, by arbitrarily selecting a set of '$n_{\rm st}$ edges whose characteristic functions pass the Gram-Schmidt test.' Property P2 therefore completes the analysis of the UD-DD relationship of Blewitt (1989) by specifying how the ambiguity solution is determined by the choice of $\breve{N}_{\rm st}$.

IV.4 CAA principle

As already emphasized, with regard to the variables introduced in Eqs. (IV.14) and (IV.15), the choice of μ in $\mathbb{Z}^{n_{\rm st}}$ is defined via appropriate constraints. As a matter of fact, the most natural way of removing the rank defect is to select μ via *a priori* constraints on the ambiguity variable v. Indeed, these constraints can then be integrated in the very definition of v. The CAA principle presented in this section results from the following preliminary analysis. (A possible introduction of this principle via the S-system approach is presented in Sect. IV.5; it should however be noted that the other classes of possible choices for μ do not then appear.)

Let $\mathscr{G}_{\rm st}$ be a spanning tree of $\mathscr{G}$ chosen arbitrarily; see Sect. G.2 and Fig. IV.1. In the algebraic framework defined in Sect. IV.3, the ambiguity function N can be decomposed in the form $N = N_{\rm b} + N_{\rm c}$ where

$$N_{\rm b} \overset{\rm def}{=} B\mu^{(N)}, \quad N_{\rm c} = N - B\mu^{(N)} \tag{IV.58}$$

in which $\mu^{(N)} \overset{\rm def}{=} B_{\rm st}^{-1} Q_{\rm st} N$; see Eq. (IV.42). In these equations, $N_{\rm b}$ and $N_{\rm c}$ have an implicit subscript: st for $\mathscr{G}_{\rm st}$. Note that $\mu^{(N)}$ is defined from the values of N on the edges of $\mathscr{G}_{\rm st}$ in a unique manner. (As N is unknown, this definition is virtual.) By construction, $\mu^{(N)}$ lies in $V_b(\mathbb{Z}) \cong \mathbb{Z}^{n_{\rm st}}$, and $N_{\rm c}$ vanishes on the edges of $\mathscr{G}_{\rm st}$. As specified in Sect. IV.3.3, $N_{\rm c} \overset{\rm def}{=} CN$ is the closure ambiguity of N with regard to $\mathscr{G}_{\rm st}$.

The spanning tree $\mathscr{G}_{\rm st}$ is built progressively from epoch 1 to k. More precisely, when new satellites are to be taken into account, it is completed as specified in Sect. IV.6.3 (see Figs. IV.1 and IV.5); $n_{\rm st}$ is thus equal to $n_{\underline{1}}$ plus the number of satellites having appeared in the field of view of the network from epoch $\bar{2}$ until k included. The number of values of $N_{\rm c}$ that are not equal to zero by definition is equal to $n_{\rm c} = n_{\rm e} - n_{\rm st}$ (Eq. (IV.29)). For example, in Fig. IV.1, for $k = 3$, we have $n_{\rm e} = 20$, $n_{\rm st} = n_{\underline{1}} + 2 + 1$ with $n_{\underline{1}} = 8$, hence $n_{\rm c} = 9$.

With regard to Eqs. (IV.14) to (IV.16), the CAA solution corresponds to the following choice of μ:

$$\mu := \mu^{(N)} \tag{IV.59}$$

From Eq. (IV.58), we then have $v = N - B\mu = N_{\rm c}$. This solution thus corresponds to the case where the ambiguities to be fixed are null on the edges of the spanning tree. According

to Eqs. (IV.14) and (IV.15), the functional variable of the corresponding LS approach is then $(w_1, \ldots, w_k,\ v)$ where

$$\left| \begin{array}{l} w_\kappa \stackrel{\text{def}}{=} \varphi_\kappa + \mathcal{R}^{\text{v}}_\kappa \mu, \qquad v \stackrel{\text{def}}{=} N_{\text{c}} \\ \text{with } \mu := \mu^{(N)} \end{array} \right. \tag{IV.60}$$

The choice of μ defined in Eq. (IV.59) therefore defines the closure-ambiguity approach (CAA). Clearly, this choice depends on the selected spanning tree $\mathscr{G}_{\text{st}}$; the definition of the integer-valued function v therefore also depends on this choice. Note that all these spanning trees have the same number of edges: $n_{\text{st}} = n_{\text{v}} - 1$; see Sect. G.2. As a result, whatever the choice of $\mathscr{G}_{\text{st}}$, the number of ambiguities to be fixed is the same: $n_{\text{c}} = n_{\text{e}} - n_{\text{st}}$ (Eq. (IV.29)).

Let us now denote by $(\check{w}_1, \ldots, \check{w}_k,\ \check{v})$ the LS solution resulting from such a choice. The reference solution of its equivalent solutions, which can be obtained as specified in Sect. IV.2.5, does not depend on the choice of $\mathscr{G}_{\text{st}}$. Indeed, a change in $\mathscr{G}_{\text{st}}$ amounts to selecting another μ in $V_b(\mathbb{Z})$; see Eq. (IV.42) and Fig. IV.6 further on.

Remark.4.1: Although the ambiguity variables of the closure ambiguity approach are closure ambiguities, it is an UD approach. Indeed, the action of the closure operator C (induced by the selected spanning tree $\mathscr{G}_{\text{st}}$) is never performed explicitly ⊡

IV.5 Derivation of the CAA principle via the S-system approach

The S-system approach (Baarda, 1973) was used by some investigators to remove various GNSS rank defects; see, e.g., Teunissen (1984); de Jonge (1998); Teunissen & Odijk (2003). In the previous section, the CAA principle was introduced in a concise manner in the theoretical framework defined through Sects. IV.2 and IV.3; the aim of the present section is to show that this principle can be introduced via the S-system approach directly.

In the general framework of the S-system approach presented in H, the Euclidean space $\mathfrak{E}$ to be considered is then the space of the functional variable

$$\xi \stackrel{\text{def}}{=} (\varphi_1, \ldots, \varphi_k;\ \vartheta) \tag{IV.61}$$

of Eq. (IV.5); ϑ is then the float version of the integer-valued functional variable N. Denoting by

$$V_\varphi \stackrel{\text{def}}{=} V_{\varphi_1} \oplus V_{\varphi_2} \oplus \cdots \oplus V_{\varphi_k} \tag{IV.62}$$

the direct sum of the vertex-bias spaces V_{φ_κ} with generic vector φ_κ, we have

$$\mathfrak{E} = V_\varphi \oplus E \tag{IV.63}$$

where E is the edge-delay space with generic vector ϑ; see Sect. IV.3.1. As illustrated in Figs. IV.2 and IV.4, given some spanning tree $\mathscr{G}_{\text{st}}$ of $\mathscr{G}$, E can be decomposed in the Euclidean orthogonal form

$$E = E_{\text{st}} \oplus E_{\text{c}} \tag{IV.64}$$

From Eqs. (IV.8) and (IV.30), we have

$$\left|\begin{array}{ll} \dim V_\varphi = \sum_{\kappa=1}^{k} n_{\mathrm{b}\kappa} & \\ \dim E_{\mathrm{st}} = n_{\mathrm{st}} & \dim E_{\mathrm{c}} = n_{\mathrm{c}} \end{array}\right. \tag{IV.65}$$

The function b_κ of Eq. (IV.5) lies in some space isomorphic to $\mathbb{R}^{n_{\mathrm{e}\kappa}}$; see the context of Eq. (IV.3), and Fig. IV.1. Setting

$$\mathrm{m} := \sum_{\kappa=1}^{k} n_{\mathrm{e}\kappa} \tag{IV.66}$$

we then define the operator $\mathcal{A}$ of the S-system approach as the operator from $\mathfrak{E}$ into $\mathbb{R}^{\mathrm{m}}$:

$$\mathcal{A}\xi \overset{\mathrm{def}}{=} \begin{pmatrix} B_1\varphi_1 + \mathcal{R}_1^{\mathrm{e}}\vartheta \\ \vdots \\ B_k\varphi_k + \mathcal{R}_k^{\mathrm{e}}\vartheta \end{pmatrix} \tag{IV.67}$$

Clearly, Eq. (IV.5) can then be explicitly written in the form

$$\mathcal{A}\begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_k \\ N \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_k \end{pmatrix} \tag{IV.68}$$

Let $\mathcal{K}$ now be the operator from $V_b \cong \mathbb{R}^{n_{\mathrm{st}}}$ into $\mathfrak{E}$ (see Sect. IV.3.1):

$$\mathcal{K}\phi \overset{\mathrm{def}}{=} \begin{pmatrix} -\mathcal{R}_1^{\mathrm{v}}\phi \\ \vdots \\ -\mathcal{R}_k^{\mathrm{v}}\phi \\ B\phi \end{pmatrix} \tag{IV.69}$$

The float version of Eq. (IV.12) yields

$$-B_\kappa\mathcal{R}_\kappa^{\mathrm{v}}\phi + \mathcal{R}_\kappa^{\mathrm{e}}B\phi = 0 \qquad (\text{for } \kappa = 1, \ldots, k) \tag{IV.70}$$

The null space of $\mathcal{A}$ is therefore the range of $\mathcal{K}$; see Eqs. (IV.67) and (IV.69). We thus have

$$\ker \mathcal{A} = \mathrm{Ran}\, \mathcal{K} \tag{IV.71}$$

with

$$\dim(\ker \mathcal{A}) = n_{\mathrm{st}} \tag{IV.72}$$

The dimension of the null space of $\mathcal{A}$ is therefore equal to that of E_{st}; see Eq. (IV.65) and Fig. IV.4.

In the framework of the S-system approach, we now remove the rank defect of Eq. (IV.5) by imposing the following constraint: *the functional variable to be estimated lies in the orthogonal complement of* $E_{\rm st}$ *in* $\mathfrak{E}$; see H. We thus define $\mathfrak{F}$ via the relation $\mathfrak{F} := E_{\rm st}^{\perp}$. Clearly, that constraint is the same as that introduced in Sect. IV.4; see Fig. IV.2. The corresponding estimable functional variable is then obtained as follows.

From Eqs. (IV.63) and (IV.64), $\mathfrak{F}$ is the direct sum of V_φ and $E_{\rm c}$:

$$\mathfrak{F} = V_\varphi \oplus E_{\rm c} \tag{IV.73}$$

We now show that $\mathfrak{E}$ is the direct sum of $\mathfrak{F}$ and $\ker \mathcal{A}$ (see Fig. IV.4):

$$\mathfrak{E} = \mathfrak{F} \oplus \ker \mathcal{A} \tag{IV.74}$$

Proof. Let ξ be a function lying in $\mathfrak{F} \cap \ker \mathcal{A}$. As ξ lies in $\mathfrak{F}$, its component ϑ vanishes on $\mathscr{G}_{\rm st}$. As ξ also lies in $\mathrm{Ran}\, \mathcal{K}$ from Eq. (IV.71), ϑ is in the range of B from Eq. (IV.69). We then have $\vartheta = B_{\rm st}\phi = 0$, hence $\phi = 0$ from Eq. (IV.33), and therefore $\xi = 0$ from Eq. (IV.69). As a result, $\mathfrak{F} \cap \ker \mathcal{A} = \{0\}$. As $\dim(\ker \mathcal{A}) = \dim E_{\rm st}$ and $\dim \mathfrak{E} = \dim \mathfrak{F} + \dim(\ker \mathcal{A})$, Eq. (IV.74) is thus established □

The oblique projection § of $\mathfrak{E}$ onto $\mathfrak{F}$ along $\ker \mathcal{A}$ plays a key role in the S-system approach (see H). It is defined by the relation $\S\xi = \xi - \eta$ with $\eta \in \ker \mathcal{A}$ and $\xi - \eta \in \mathfrak{F}$; see Eq. (H.4) and Fig. H.1. We now specify its definition explicitly.

Let $\mathcal{Q}_{\rm st}$ be the orthogonal projection of $\mathfrak{E}$ onto $E_{\rm st}$. We then have (see Fig. IV.4):

$$\mathcal{Q}_{\rm st}\eta = \mathcal{Q}_{\rm st}\xi = Q_{\rm st}\vartheta \tag{IV.75}$$

From Eqs. (IV.71) and (IV.69),

$$\eta = \mathcal{K}\phi$$

for some ϕ in $V_b \cong \mathbb{R}^{n_{\rm st}}$. From Eq. (IV.75), that bias function ϕ satisfies the condition

$$\mathcal{Q}_{\rm st}\mathcal{K}\phi = Q_{\rm st}\vartheta$$

According to Eq. (IV.69), we have $\mathcal{Q}_{\rm st}\mathcal{K}\phi = Q_{\rm st}B\phi$, i.e., $\mathcal{Q}_{\rm st}\mathcal{K}\phi = B_{\rm st}\phi$, where $B_{\rm st}$ is the operator defined in Sect. IV.3.1. We thus have $\phi = \phi^{(\vartheta)}$ where $\phi^{(\vartheta)}$ is defined by Eq. (IV.35):

$$\phi^{(\vartheta)} \stackrel{\rm def}{=} B_{\rm st}^{-1} Q_{\rm st}\vartheta$$

It then follows from Eq. (H.4) that

$$\S\xi = \xi - \eta \quad \text{where} \quad \eta = \mathcal{K}\phi^{(\vartheta)} \tag{IV.76}$$

Here, the matrices $[S_\perp]$ and $[W]$ of Eq. (H.7) have been implicitly defined by the standard bases of $E_{\rm st}$ and $\mathrm{Ran}\, \mathcal{K}$, respectively. From Eq. (IV.69), we thus have

$$\S\xi = \begin{pmatrix} \varphi_1 + \mathcal{R}_1^{\rm v}\phi^{(\vartheta)} \\ \vdots \\ \varphi_k + \mathcal{R}_k^{\rm v}\phi^{(\vartheta)} \\ \vartheta - B\phi^{(\vartheta)} \end{pmatrix} \tag{IV.77}$$

hence, in particular (when ϑ is set equal to N),

$$\S \begin{pmatrix} \varphi_1 \\ \vdots \\ \varphi_k \\ N \end{pmatrix} = \begin{pmatrix} \varphi_1 + \mathcal{R}_1^{\mathrm{v}} \mu^{(N)} \\ \vdots \\ \varphi_k + \mathcal{R}_k^{\mathrm{v}} \mu^{(N)} \\ N - B \mu^{(N)} \end{pmatrix} \tag{IV.78}$$

where $\mu^{(N)} = B_{\mathrm{st}}^{-1} Q_{\mathrm{st}} N$ (Eq. (IV.42)). We thus retrieve the 'estimable functional variables' of Eq. (IV.60):

$$w_\kappa \overset{\text{def}}{=} \varphi_\kappa + \mathcal{R}_\kappa^{\mathrm{v}} \mu^{(N)} \qquad v = N_{\mathrm{c}} \tag{IV.79}$$

The estimable functional variable of the CAA method is therefore the oblique projection of $(\varphi_1, \ldots, \varphi_k;\ N)$ on $\mathfrak{F}$ along $\ker \mathcal{A}$; see Fig. IV.4.

This introduction of the CAA principle gives another insight into the analysis presented in Sect. IV.3. Furthermore, it completes the study developed in Sects. 3.3, 3.4 and 3.15 of de Jonge (1998) on two points:

(i) operator $\mathcal{A}$ is defined with regard to the union of the observational graphs until the current epoch; see Sect. IV.2.1, and Eqs. (IV.67) and (IV.68);

(ii) a related geometrical representation is proposed: Fig. IV.4; in that representation, $C\vartheta$ is the CD point of ϑ; when $\vartheta := N$, CN is therefore the closure-ambiguity point of N.

IV.6 CAA implementation

In the closure-ambiguity approach, the rank defect of Eq. (IV.5) is removed via the choice of the variables w_κ and v defined in Eq. (IV.60). The equation to be solved in the LS sense is then Eq. (IV.16) with the constraint $v = 0$ on the edges of $\mathscr{G}_{\mathrm{st}}$. The CAA problem is therefore: solve in the LS sense the equation

$$\left| \begin{array}{l} B_\kappa w_\kappa + \mathcal{R}_\kappa^{\mathrm{e}} v = b_\kappa \qquad (\kappa = 1, \ldots, k) \\ \text{subject to the constraint '} v = 0 \text{ on } \mathscr{G}_{\mathrm{st}} \text{'} \end{array} \right. \tag{IV.80}$$

In what follows, the noise is taken into account by using the variance-covariance matrix $\mathcal{V}_\kappa$ of b_κ in the definition of the 'Euclidean forms' $\boldsymbol{B}_\kappa$, $\boldsymbol{A}_\kappa$ and $\boldsymbol{b}_\kappa$ of B_κ, $\mathcal{R}_\kappa^{\mathrm{e}}$ and b_κ, respectively; see the context of Eq. (I.12) in Sect. I.3. The system (IV.80) to be solved in the Euclidean LS sense is then the following:

$$\boldsymbol{B}_\kappa \boldsymbol{w}_\kappa + \boldsymbol{A}_\kappa \boldsymbol{v} = \boldsymbol{b}_\kappa \qquad (\kappa = 1, \ldots, k) \tag{IV.81}$$

i.e.,

$$\begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_1 \\ \boldsymbol{0} & \boldsymbol{B}_2 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_2 \\ \boldsymbol{0} & \boldsymbol{0} & \ddots & \boldsymbol{0} & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_k & \boldsymbol{A}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_k \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_k \end{bmatrix} \tag{IV.82}$$

The angular block structure of matrix $[\boldsymbol{B}\ \boldsymbol{A}]$ is well suited to recursive QR factorization. (A complete implementation of the corresponding LS procedure is presented in I.) The advantage of this procedure is to provide some gain in numerical accuracy when dealing with large-scale problems; see Björck (1996). More interestingly, the corresponding techniques prove to be very efficient for GNSS quality control; see, e.g., Blewitt (1989); Tiberius (1998); Loehnert et al. (2000) Sect. 9 in Lannes & Gratton (2009).

In this section, we first give a survey of the QR implementation of the CAA principle (Sect. IV.6.1). We then specify the definitions of the column matrices $\boldsymbol{w}_k$ and $\boldsymbol{v}$ of Eq. (IV.82); this is done in Sects. IV.6.2 and IV.6.3, respectively. More information about the technical aspects of this implementation, like for instance the construction of matrices $\boldsymbol{A}_k$, $\boldsymbol{B}_k$ and $\boldsymbol{b}_k$, is to be found in I.

Remark.6.1: In most situations encountered in practice, the second member of equations such as (IV.80) includes a large number of additional variables; see, e.g., Ge et al. (2005); Loyer et al. (2012). In the remainder of the paper, we assume that the related models have been well selected so that we can concentrate on the first members of these equations only ⊡

Remark.6.2: The method presented in this section can be applied as such for solving the WL equation (4) of Loyer et al. (2012); N_{WL} and τ_{WL} are then to be substituted for N and φ, respectively; b is then the term on the left-hand side of that equation. In our approach, the WL ambiguity point $\check{v}_{\mathrm{WL}}$ would then be the closure-ambiguity point of N_{WL}: $\check{v}_{\mathrm{WL}} = CN_{\mathrm{WL}}$; N_{WL} would thus be retrieved up to a vector of $B\mathbb{Z}^{n_{\mathrm{st}}}$ ⊡

IV.6.1 Survey

In a first stage, at each epoch k, the 'float solution' $\hat{\boldsymbol{v}}$ is computed or updated. This is done by solving the float version of Eq. (IV.82) in the Euclidean LS sense via recursive QR factorization; see Sect. I.2 and Eq. (I.7). The integer least-squares (ILS) solution $\check{\boldsymbol{v}}$, and thereby $\check{v}$, is then defined as the solution of the nearest-lattice-point (NLP) problem (I.9). This solution is obtained in a second stage via appropriate integer-programming techniques; see Lannes & Prieur (2013) and Lannes (2013) for the parallelization of the related LLL[3]/LAMBDA reduction/decorrelation algorithms. The ambiguities are thus fixed. Once at some epoch k_{val}, these ambiguities can be validated (see Verhagen & Teunissen (2006), the bias matrices $\check{\boldsymbol{w}}_k$, and thereby the phase biases $\check{w}_k$, are obtained via the relations (I.10) for $k \leq k_{\mathrm{val}}$. The variance-covariance matrix of $\check{w}_\kappa$, which is required for implementing the PPP mode properly (see Sect. 9 in Lannes & Teunissen (2011), is obtained in that framework; see the end of Sect. I.2.

Remark.6.3: Once at some epoch k_{val}, all the closure ambiguities have been validated, the QR recursive process only needs to focus on the few closure ambiguities associated with the new closure edges. At the epochs $k > k_{\mathrm{val}}$, the new closure ambiguities can therefore be determined very quickly. Indeed, the dimension of the matrix $\boldsymbol{v}$ handled by the QR process is then much smaller than previously. This shows that this approach is well suited to integer-ambiguity resolution in real time. For instance, the satellite-clock biases could then be broadcasted to the network users in real time; see Sect. IV.2.3 ⊡

[3]Here LLL stands for Lenstra, Lenstra, Lovász, the authors of the famous LLL algorithm (Lenstra, Lenstra & Lovász, 1982).

IV.6.2 Definition of w_k

If we assume for instance that all the receivers of the network are active (see Fig. IV.1), the receiver phase bias $w_{\mathrm{r}k}(i)$ is then the entry of $\boldsymbol{w}_k$ with index $i-1$. The indices are shifted by -1 since we used the convention $\varphi_{\mathrm{r}k}(1)=0$ (Eq. (IV.7)).

The satellite phase bias $w_{\mathrm{s}k}(j)$ is then the entry of $\boldsymbol{w}_k$ with index $(n_{\mathrm{r}}-1+j)$; see for example the first two epochs of Fig. IV.1. To handle the cases of disappearance of one or more satellite(s), we introduce an index function $\varpi_{\mathrm{s}k}$ defined so that $w_{\mathrm{s}k}(j)$ is the entry of $\boldsymbol{w}_k$ with index $(n_{\mathrm{r}}-1+\varpi_{\mathrm{s}k}(j))$. For example, at epoch $k=3$ of Fig. IV.1, for $j>3$, we have $\varpi_{\mathrm{s}k}(j)=j-1$.

Difficulties with some receivers would be handled in a similar manner by introducing index functions $\varpi_{\mathrm{r}k}(i)$.

IV.6.3 Definition of v

The entries of $\boldsymbol{v}$ in Eq. (IV.82) are the closure ambiguities to be taken into account from epochs 1 to k included. The corresponding closure-edge list $_{\mathrm{c},k}$ is defined in this section; see for example the lower grid of Fig. IV.5. In the implementation of the QR method, those entries are to be put at the top of the column matrix $\boldsymbol{v}$ at the epoch $\kappa\le k$ where they appear; see Sect. C3. This means that for all $\kappa'<\kappa$, the matrices $\boldsymbol{A}_{\kappa'}$ have then (on their left-hand side) implicit additional columns whose entries are zero. We now concentrate on the closure-edge lists $_{\mathrm{c},\kappa}$ for $\kappa=1,\ldots,k$.

At epoch $k=1$, the spanning tree $\mathscr{G}_{\mathrm{st},1}$ of $\mathscr{G}_1=\mathscr{H}_1$ is built as specified in Sect. G.2. For example, for the scenario defined in Fig. IV.1, the edges of $\mathscr{G}_{\mathrm{st},1}$ then correspond to the large dots of the upper grid of Fig. IV.5; $_{\mathrm{c},1}$ then includes three closure edges ordered as they are encountered when scanning that grid line by line:

$$_{\mathrm{c},1}=\{(\mathrm{r}_3,\mathrm{s}_3),(\mathrm{r}_3,\mathrm{s}_4),(\mathrm{r}_4,\mathrm{s}_4)\}\qquad(k=1)$$

At epoch $k=2$, the spanning tree $\mathscr{G}_{\mathrm{st},2}$ of $\mathscr{G}_2=\mathscr{H}_1\cup\mathscr{H}_2$ is completed by adding the edges corresponding to the first active edges involving the new satellites, for example $(\mathrm{r}_1,\mathrm{s}_7)$, and $(\mathrm{r}_2,\mathrm{s}_6)$ in Fig. IV.5. When implementing the QR method, the new closure edges must then be the first terms of $_{\mathrm{c},2}$; see Fig. I.5 in Sect. I.2.1. The previous terms of that list are then shifted rightwards. In the example of Fig. IV.1, we then have

$$_{\mathrm{c},2}=\{(\mathrm{r}_2,\mathrm{s}_7),(\mathrm{r}_3,\mathrm{s}_2),(\mathrm{r}_3,\mathrm{s}_6),(\mathrm{r}_4,\mathrm{s}_5),(\mathrm{r}_3,\mathrm{s}_3),(\mathrm{r}_3,\mathrm{s}_4),(\mathrm{r}_4,\mathrm{s}_4)\}\quad(k=2)$$

We proceed similarly for the next epochs; see the lower grid of Fig. IV.5.

By construction, the number of closure edges n_{c} is a non-decreasing function of k; see Eq. (IV.29) and Fig. IV.5. In fact, this number defines the dimension of the NLP problem (I.9) to be solved at epoch k.

IV.7 Equivalent ambiguity solutions: related methodological aspects

In this section, we analyse some other methods which are used for solving the GNSS calibration problem, and compare the corresponding solutions with the CAA solution. This done

for Blewitt (1989) in Sect. IV.7.1, for Collins et al. (2010) in Sect. IV.7.2, and for Loyer et al. (2012) in Sect. IV.7.3. Finally, we suggest some possible improvements of the existing methods in Sect. IV.7.4.

IV.7.1 Blewitt's approach

In his original contribution published in 1989, Blewitt proposes a resolution of the problem in three stages. In the first one, the float solution is obtained in UD mode. In the second one, double differencing of the float biased ambiguities thus obtained provides DD ambiguities. The latter are then fixed via some sequential adjustment algorithm. In the third and final stage, the UD ambiguities are derived from those fixed DD ambiguities. This is done via the inverse of the operator D defined in the appendix B of that paper. The estimates of station-receiver locations, orbital parameters, etc., are then updated. Briefly, one may therefore say that the Blewitt approach is hybrid: it is a 'UD-DD-UD approach.'

The D-matrix of our analysis is the matrix $[D_\diamond]$ introduced in Sect. IV.3.6: Eq. (IV.54). Given some DD ambiguity set N_d (such as that defined in Eq. (IV.55)), some spanning tree $\mathscr{G}_\mathrm{st}$, and some ambiguity point $\breve{N}_\mathrm{st}$ arbitrarily chosen in $E_\mathrm{st}(\mathbb{Z})$, it is possible to retrieve N up to a vector of $E_\mathrm{b}(\mathbb{Z})$. More precisely (see Fig. IV.6 together with Fig. IV.3), the UD ambiguity obtained via Eq. (IV.57),

$$\breve{N} := [D_\diamond]^{-1} \begin{bmatrix} [\breve{N}_\mathrm{st}] \\ [N_\mathrm{d}] \end{bmatrix}$$

is equal to $N_\mathrm{c} + B\mu^{(\breve{N}_\mathrm{st})}$. Removing the rank defect via the inverse of $[D_\diamond]$ therefore amounts to imposing the constraint $Q_\mathrm{st} v = \breve{N}_\mathrm{st}$ where $\breve{N}_\mathrm{st}$ can be chosen in an arbitrary manner.

IV.7.2 Collins's approach

In the last statement of Sect. IV.7.1, one recognizes the concept of *ambiguity datum fixing* of the UD approach of Collins et al. (2010); the latter is thus closely linked to that of Blewitt. However, with regard to Eq. (IV.16), the 'direct problem' to be solved in the LS sense is then

$$\left| \begin{array}{l} B_\kappa w_\kappa + \mathcal{R}^\mathrm{e}_\kappa v = b_\kappa \qquad (\kappa = 1, \ldots, k) \\ \text{subject to the constraint } Q_\mathrm{st} v = \breve{N}_\mathrm{st} \end{array} \right. \tag{IV.83}$$

Let us denote by $(\breve{w}_1, \ldots, \breve{w}_k, \breve{v})$ the solution of this problem, i.e., the Blewitt/Collins solution. Let $(\breve{w}_1, \ldots, \breve{w}_k, \breve{v})$ now be the CAA solution obtained with the same spanning tree $\mathscr{G}_\mathrm{st}$. As shown below, we then have

$$\breve{w}_\kappa = \breve{\varphi}_\kappa - \mathcal{R}^\mathrm{v}_\kappa \mu^{(\breve{N}_\mathrm{st})}, \qquad \breve{v} = \breve{v} + B\mu^{(\breve{N}_\mathrm{st})} \tag{IV.84}$$

Proof. The LS solutions of Eq. (IV.5) are of the form (IV.18)-(IV.19). Equation (IV.84) then follows from the fact that by construction $\breve{v} + B\mu^{(\breve{N}_\mathrm{st})}$ satisfies the constraint of Eq. (IV.83).

Indeed, as $Q_{\rm st}\breve{v} = 0$, we have

$$\begin{aligned} Q_{\rm st}\big(\breve{v} + B\mu^{(\breve{N}_{\rm st})}\big) &= Q_{\rm st}B\mu^{(\breve{N}_{\rm st})} \\ &= B_{\rm st}\mu^{(\breve{N}_{\rm st})} \\ &= B_{\rm st}(B_{\rm st}^{-1}Q_{\rm st}\breve{N}_{\rm st}) = Q_{\rm st}\breve{N}_{\rm st} \\ &= \breve{N}_{\rm st} \qquad \square \end{aligned}$$

Remark.7.4: In the special case where $\breve{N}_{\rm st}$ is set equal to zero on all the edges of $\mathscr{G}_{\rm st}$, the Blewitt/Collins solution coincides with the CAA solution; see Eq. (IV.84) and Fig. IV.6. This explicitly shows that the Blewitt and Collins approaches can lead to the same results as the CAA method. However, even in that case, Blewitt's approach is not equivalent in terms of efficiency. The direct implementation of the Collins/CAA principle is *a priori* preferable: the results are then obtained without any differencing operation, and without any generalized inversion of the fixed ambiguity set; see Sects. IV.4 and IV.6. As illustrated in Fig. IV.6, the reference ambiguity solutions $\bar{v}$ obtained with the Blewitt, Collins and CAA methods must of course be identical on every edge of $\mathscr{G}$, and this for any choice of $\mathscr{G}_{\rm st}$ and $\breve{N}_{\rm st}$; see 2.8 ⊡

IV.7.3 Related approach

With regard to the way the narrow-lane (NL) ambiguity N_1 is fixed, the zero-difference approach of Loyer et al. (2012) can be considered as a variant of Collins' approach. We now clarify this point.

In that particular approach, the rank defect is removed after having obtained the *'float ambiguity solution'* $\hat{N}_1$ via the introduction of some additional constraints; see Sect. 2.6 of that paper. These authors then implicitly choose some spanning tree $\mathscr{G}_{\rm st}$, and set the ambiguities to $\lfloor \hat{N}_1(i,j) \rceil$ on the edges $(\mathrm{r}_i, \mathrm{s}_j)$ of $\mathscr{G}_{\rm st}$. The remaining UD ambiguities (i.e., the $n_{\rm c}$ ambiguities associated with the closure edges) are fixed via some 'bootstrapping operation.' This operation provides the 'Babai point' of the discrete-search algorithms presented in Jazaeri et al. (2012), and Lannes & Prieur (2013).

Setting $N := N_1$ and $\breve{N}_{\rm st} := \lfloor Q_{\rm st}\hat{N}_1 \rceil$, we thus have $N - B\mu = \breve{N}_{\rm st}$ on $\mathscr{G}_{\rm st}$, hence $\mu := B_{\rm st}^{-1}Q_{\rm st}(N - \breve{N}_{\rm st})$. It then follows from Eqs. (IV.42) and (IV.58) that the ambiguity solution of the Loyer et al. is

$$N - B\mu = (N - B\mu^{(N)}) + B\mu^{(\breve{N}_{\rm st})} = N_{\rm c} + B\mu^{(\breve{N}_{\rm st})}$$

Provided that the solution provided by the bootstrapping process is the right NLP solution $\breve{v}$, we then have (like for the Blewitt and Collins solutions in Fig. IV.6)

$$\breve{v} = \check{v} + B\mu^{(\breve{N}_{\rm st})}$$

where $\check{v}$ is the CAA solution obtained with the same spanning tree. The choice $\breve{N}_{\rm st} := 0$ would lead to a 'mild version' of the corresponding NL closure-ambiguity approach; see how the NLP problem is solved in Sect. IV.6.1.

IV.7.4 Suggested improvements of the existing methods

In the previous sections, we have analysed some of the most frequently-used GNSS calibration methods. This analysis was made on the basis of published papers; it is thereby partial, since there are unfortunately many important topics which are not described in the papers presenting those methodologies. Blewitt (1989)'s method is rather well established from a theoretical point of view, but its UD-DD-UD principle is not optimal. The UD approaches of Collins et al. (2010) and Loyer et al. (2012) are closer to our CAA-(S-system) principle, but their introduction is rather intuitive: no theoretical justification is clearly given.

For the first time, all those methods have been presented in a unified framework. We thus provided a theoretical justification of what is done for removing the rank defect in those methods. Our study also showed that despite their apparent differences, they are rather close to one another; see Fig. IV.6.

Taking profit of the theoretical analysis presented in this chapter, we now propose some modifications aimed at improving those methodologies. First of all, in the implementation of those methods, on the edges of the selected spanning tree, the ambiguities to be fixed should be set equal to zero: $\check{N}_{\mathrm{st}} := 0$. It is indeed the simplest choice, and the remaining ambiguities are then the closure ambiguities of N. This concept, which is also used in astronomy, has a clear meaning; see Sects. IV.3.3, and II. As these ambiguities depend on the selected spanning tree, it is also recommended to provide the reference set of every ambiguity solution; see Sect. IV.2.5 and Fig. IV.6. Furthermore, this particular solution is also convenient for plotting and comparing the time variations of the clock-phase biases; see 2.7.

A suggestion for improving the performances of existing network-calibration methods, in particular for real-time PPP applications, would be the progressive construction of graph $\mathscr{G}$; see Sect. IV.2.1, Figs. IV.1 and IV.5. This would allow one to determine the first epoch at which the fixed ambiguities can be validated; see Sect. IV.6.1 and 6.3.

Another related aspect is the use of modern integer-programming techniques. In particular, we suggest that the NLP/ILS problems should be solved via 'reduction methods' which benefit from the last developments of the LLL algorithm; see, e.g., Jazaeri et al. (2012); Lannes (2013); Lannes & Prieur (2013). This should lead to a substantial gain in reliability and processing time. Note that these methods can be even used for huge networks. Indeed, parallel versions of the LLL algorithm have already been implemented; see Luo & Qiao (2011) and Lannes (2013) for the LAMBDA-decorrelation versions of such algorithms.

Remark.7.5: With regard to the NLP/ILS problems to be solved, the 'traditional ionosphere-free approach' (such as that of Collins et al. (2010), or Loyer et al. (2012)) is not recommended. This approach leads to unnecessary developments. Furthermore, the physical arguments for using it were called to question many years ago. Indeed, as explicitly shown by Teunissen (1997), the wide-lane decorrelation effect of that approach is far from being optimal $\square$

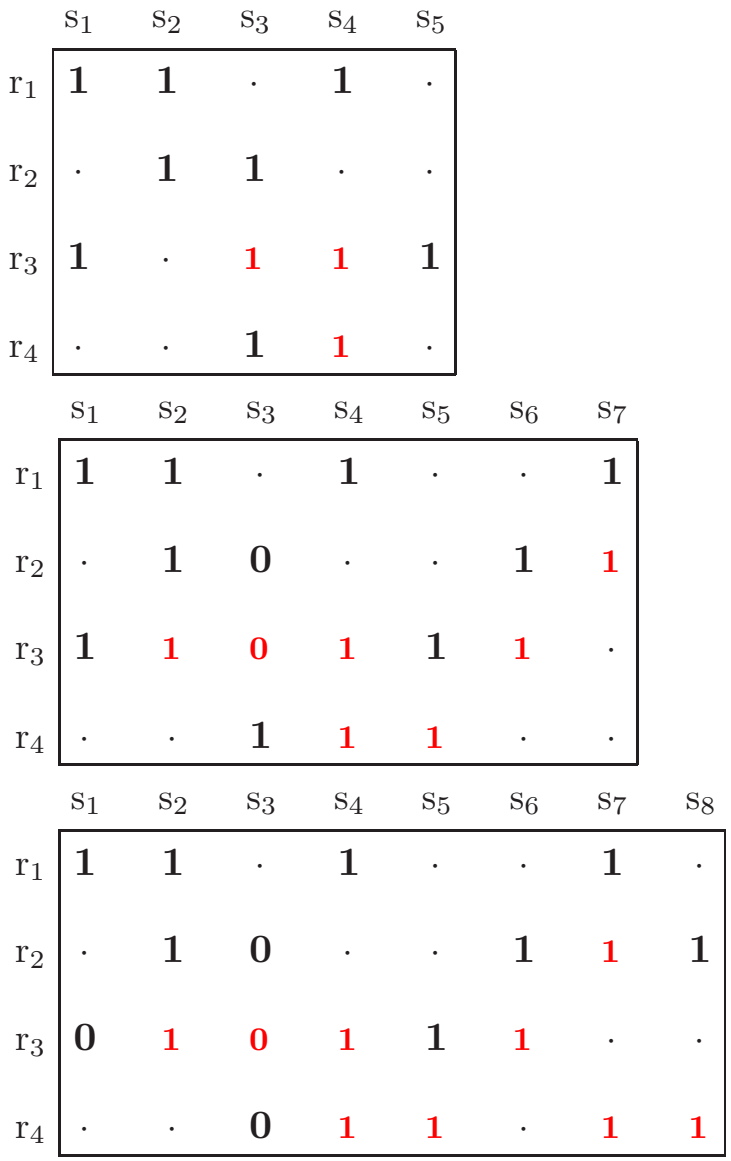

Figure IV.1: Characteristic functions of $\mathscr{H}_k$ with regard to $\mathscr{G}_k$ for $k = 1, 2, 3$ (example). From top to bottom, $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$. The dots define the edges $(\mathrm{r}_i, \mathrm{s}_j)$ for which no data have been obtained until epoch k included. Here, $n_{\mathrm{e}1} = 11$, $n_{\mathrm{e}2} = 15$, and $n_{\mathrm{e}3} = 16$. By definition, $\mathscr{G}_k$ is the union of the observational graphs until epoch k included. The number of the edges of $\mathscr{G}_k$ is 11 at epoch 1, 17 at epoch 2, and 20 at epoch 3. Six edges appear at epoch 2: $(\mathrm{r}_1, \mathrm{s}_7)$, $(\mathrm{r}_2, \mathrm{s}_6)$, $(\mathrm{r}_2, \mathrm{s}_7)$, $(\mathrm{r}_3, \mathrm{s}_2)$, $(\mathrm{r}_3, \mathrm{s}_6)$ and $(\mathrm{r}_4, \mathrm{s}_5)$; two edges disappear: $(\mathrm{r}_2, \mathrm{s}_3)$ and $(\mathrm{r}_3, \mathrm{s}_3)$. Note that satellites s_6 and s_7 are then detected by the network. Three edges appear at epoch 3: $(\mathrm{r}_2, \mathrm{s}_8)$, $(\mathrm{r}_4, \mathrm{s}_7)$ and $(\mathrm{r}_4, \mathrm{s}_8)$; two edges disappear: $(\mathrm{r}_3, \mathrm{s}_1)$ and $(\mathrm{r}_4, \mathrm{s}_3)$. Satellite s_3 then disappears. At each epoch, the large-sized numbers define the edges of $\mathscr{G}_{\mathrm{st},k}$, the selected spanning tree of $\mathscr{G}_k$; see Fig. IV.5 further on.

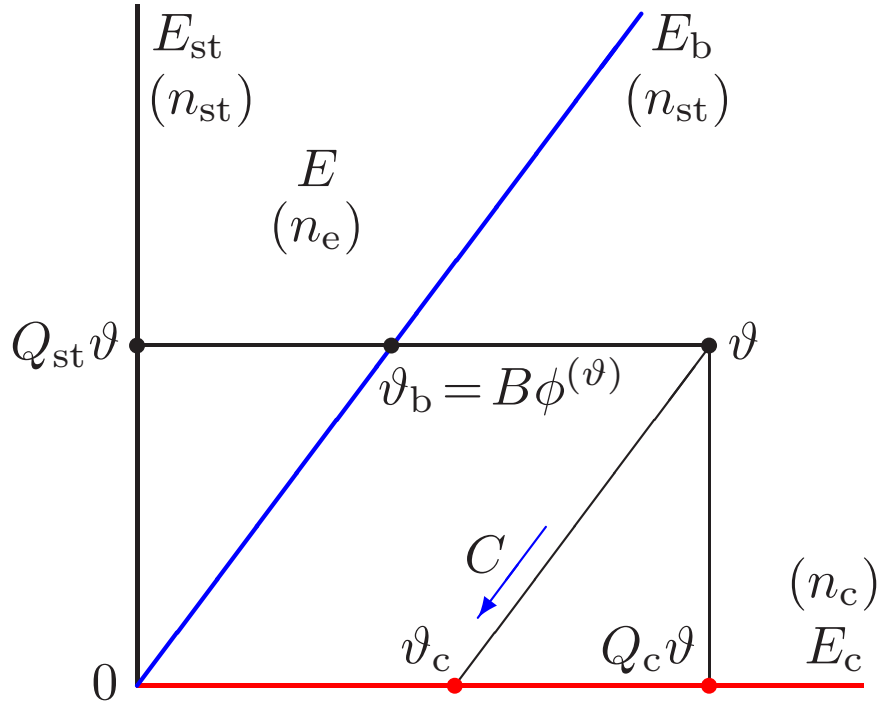

Figure IV.2: Geometrical illustration of Property P1. In this geometrical representation of the edge-delay space $E \cong \mathbb{R}^{n_{\mathrm{e}}}$, E_{st} is the spanning-tree delay space. This space is isomorphic to the vertex-bias space $V_b \cong \mathbb{R}^{n_{\mathrm{st}}}$. The orthogonal complement of E_{st} in the Euclidean space E is the closure-delay space E_{c}. The range of the bias operator B, the edge-bias space, is a subspace of E denoted by E_{b}. This space is isomorphic to E_{st} and thereby to V_b. (The dimensions of these spaces are written within parentheses.) As illustrated here, E is the oblique direct sum of E_{b} and E_{c}. The closure operator C is the oblique projection of E onto E_{c} along E_{b}.

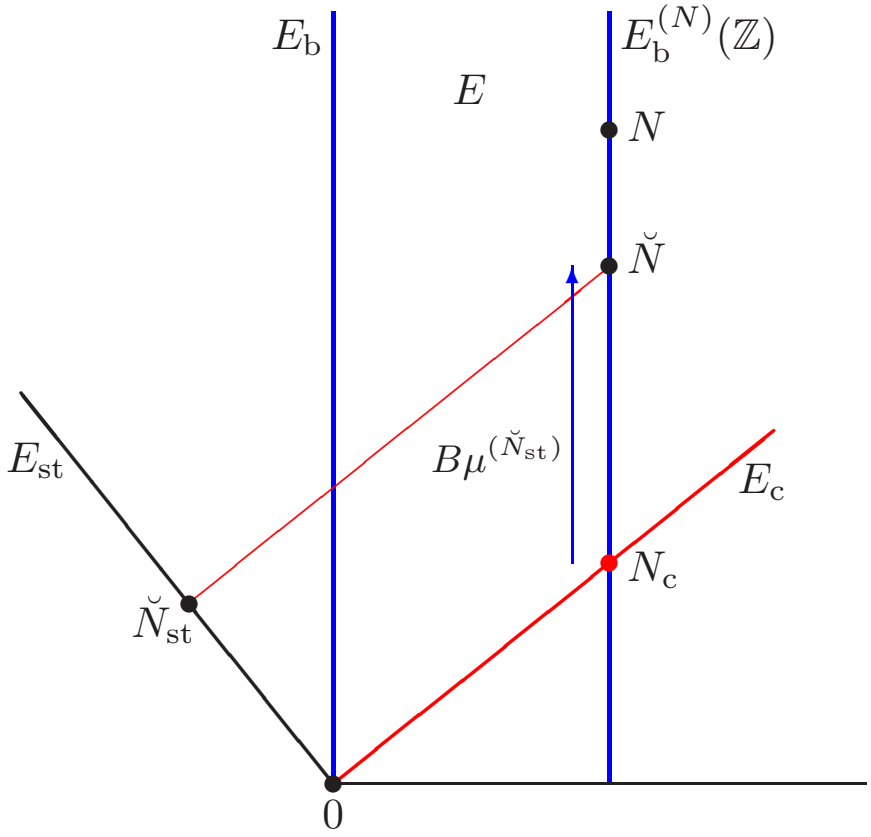

Figure IV.3: Geometrical illustration of Property P2 In this symbolic representation of the edge-delay space E, $E_{\mathrm{b}}^{(N)}(\mathbb{Z})$ is the affine lattice passing through N and parallel to the integer lattice $E_{\mathrm{b}}(\mathbb{Z})$ of the edge-bias space E_{b} (here, for clarity, the vertical axis); $\breve{N}$ is the UD ambiguity obtained via the relationship (IV.50) in which $\breve{N}_{\mathrm{st}}$ is arbitrarily fixed in E_{st}, and N_{c} is the CD ambiguity point of N (the closure ambiguity of N). In the important special case where $\breve{N}_{\mathrm{st}}$ is set equal to 0, $\breve{N}$ reduces to N_{c}.

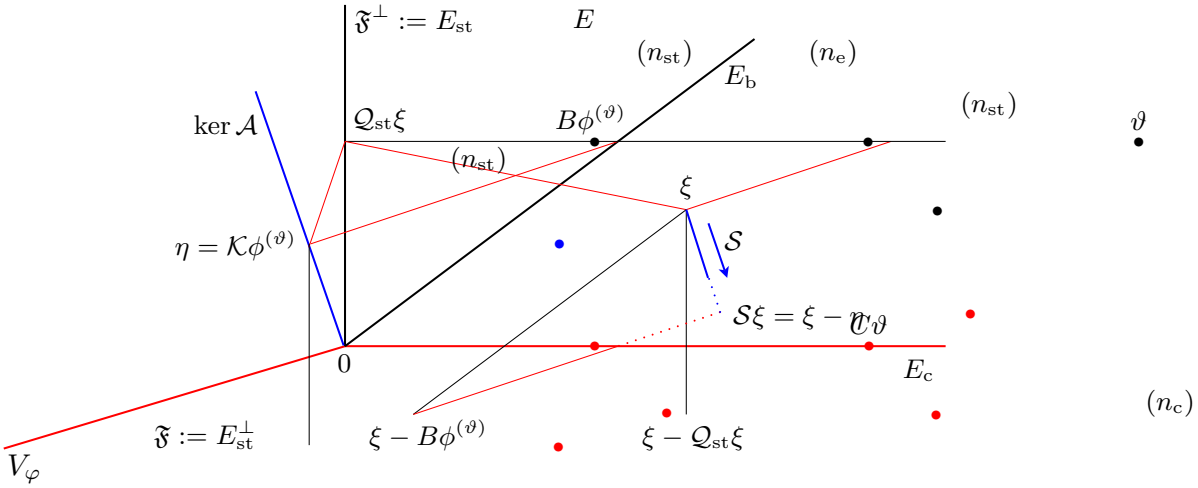

Figure IV.4: Geometrical illustration of the CAA principle in the S-system approach. In this 3D-geometrical illustration, which completes Fig. IV.2, the Euclidean space $\mathfrak{E}$ of the functional variable ξ involved in Eqs. (IV.67) and (IV.68) is represented as a 3D-space; $\mathfrak{E}$ is the orthogonal sum of V_φ and E, where V_φ is the direct sum of the vertex-bias spaces V_{φ_κ} (see Eq. (IV.62)), and E is the edge-delay space of the float ambiguity variable ϑ (the projection of ξ on E). According to the definitions introduced in Sect. IV.3.1, E can be regarded as the orthogonal sum of the spanning-tree delay space $E_{\rm st}$ and the closure-delay space $E_{\rm c}$. Note that the projection of ξ on $E_{\rm st}$ coincides with that of ϑ: $\mathcal{Q}_{\rm st}\xi = Q_{\rm st}\vartheta$. As shown in Sect. IV.3.2, we have $\phi^{(\vartheta)} = B_{\rm st}^{-1}Q_{\rm st}\vartheta$ (Eq. (IV.35)). The operator $\mathcal{A}$ defined by Eq. (IV.67) is not of full rank. Its null space is a subspace of $\mathfrak{E}$ of dimension $n_{\rm st}$; more precisely, $\ker\mathcal{A}$ is the range of the operator $\mathcal{K}$; see Eq. (IV.69). The edge-bias space $E_{\rm b}$ is the projection of $\ker\mathcal{A}$ on E. In the CAA approach, the following condition is imposed: the functional variable to be estimated lies in $\mathfrak{F}$, the orthogonal complement of $E_{\rm st}$ in $\mathfrak{E}$. As $\mathfrak{E}$ is the direct sum of $\mathfrak{F}$ and $\ker\mathcal{A}$ (see text), the estimable functional variable is the oblique projection of ξ on $\mathfrak{F}$ along $\ker\mathcal{A}$, i.e., $\S\xi$ with $\vartheta := N$; see Eqs. (IV.78) and (IV.79). The projection of $\S\xi$ on E, $\vartheta_{\rm c} \stackrel{\rm def}{=} C\vartheta$, is the closure component of ϑ, i.e., the closure ambiguity $N_{\rm c} \stackrel{\rm def}{=} CN$ when $\vartheta := N$.

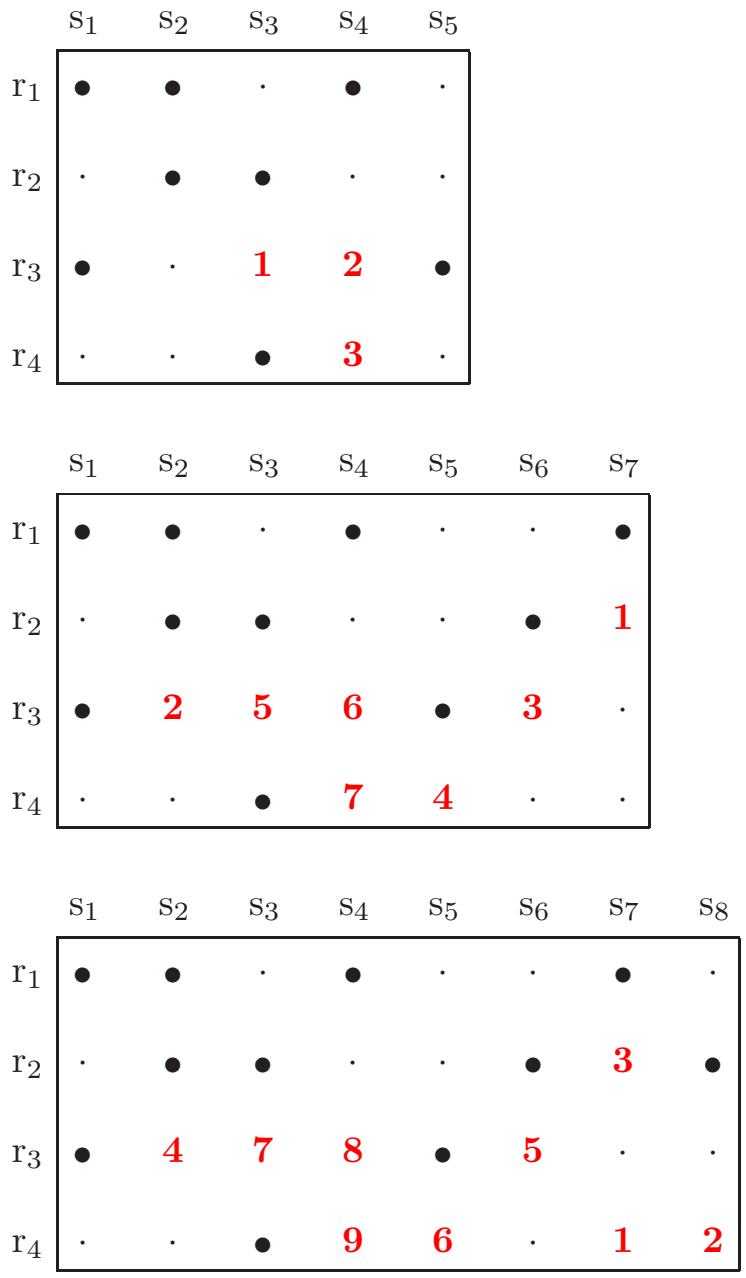

Figure IV.5: Spanning trees $\mathscr{G}_{\mathrm{st},k}$ and closure-edge lists $_{\mathrm{c},k}$ The example shown here corresponds to that introduced in Fig. IV.1. From top to bottom, epoch $k = 1$ ($n_{\mathrm{st},1} = 8$), epoch $k = 2$ ($n_{\mathrm{st},2} = 10$), and epoch $k = 3$ ($n_{\mathrm{st},3} = 11$). The large dots correspond to the edges of these spanning trees; the small dots define the edges $(\mathrm{r}_i, \mathrm{s}_j)$ that do not appear in $\mathscr{G}_k$ at those epochs. The spanning tree of $\mathscr{G}_2$ is obtained from that of $\mathscr{G}_1$ by adding the edges $(\mathrm{r}_1, \mathrm{s}_7)$ and $(\mathrm{r}_2, \mathrm{s}_6)$. Likewise, the spanning tree of $\mathscr{G}_3$ is obtained from that of $\mathscr{G}_2$ by adding the edge $(\mathrm{r}_2, \mathrm{s}_8)$. The closure edges of $_{\mathrm{c},k}$ are ordered as shown in red; see text.

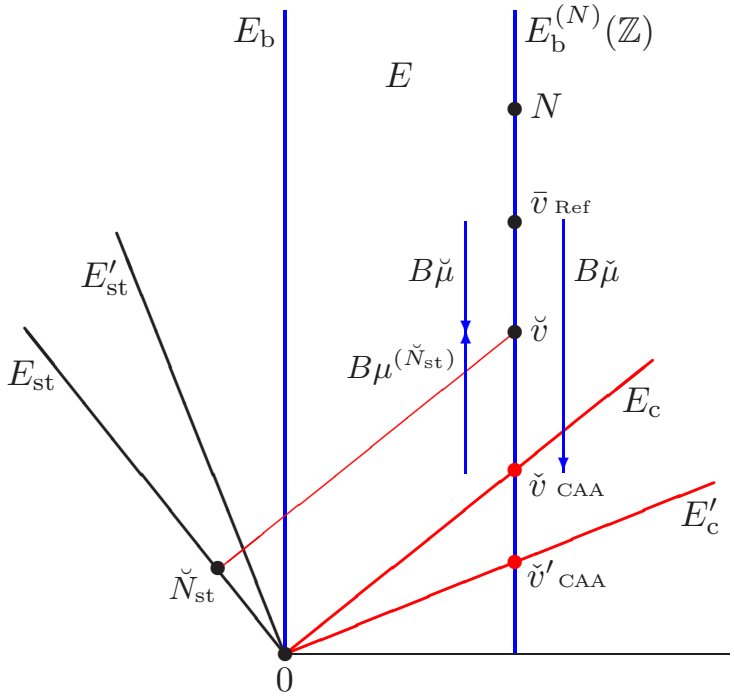

Figure IV.6: Links between some equivalent ambiguity solutions In this geometrical representation of the edge-delay space E, the equivalent ambiguity solutions lie in the affine lattice $E_{\rm b}^{(N)}(\mathbb{Z})$ passing through N and parallel to the integer lattice $E_{\rm b}(\mathbb{Z})$ of the edge-bias space $E_{\rm b}$ (here for clarity the vertical axis). The Blewitt-Collins solution $\check{v}$ obtained by imposing the constraint $Q_{\rm st}v = \check{N}_{\rm st}$ (for some $\check{N}_{\rm st}$ arbitrarily fixed in $E_{\rm st}$) corresponds to the intersection of $E_{\rm b}^{(N)}(\mathbb{Z})$ with the affine space passing through $\check{N}_{\rm st}$ and parallel to $E_{\rm c}(\mathbb{Z})$ (the closure-ambiguity lattice induced by the choice of $\mathscr{G}_{\rm st}$); see Fig. IV.3. The CAA solutions induced by the choice of the spanning trees $\mathscr{G}_{\rm st}$ and $\mathscr{G}_{\rm st'}$, $\check{v}$ and $\check{v}'$, correspond to the intersections of $E_{\rm b}^{(N)}(\mathbb{Z})$ with $E_{\rm c}(\mathbb{Z})$ and $E'_{\rm c}(\mathbb{Z})$, respectively. Note that $\check{v} = N - B\mu^{(N)}$ from Property P1, and $\check{v} = \check{v} + B\mu^{(\check{N}_{\rm st})}$ from Property P2. Here, $\bar{v}$ represents the reference solution: $\bar{v} = \check{v} - B\check{\mu}$; see Eqs. (IV.21), (IV.23) and (IV.25). Likewise, $\bar{v}$ is obtained from the Blewitt or Collins solution $\check{v}$ via the relation $\bar{v} = \check{v} - B\check{\mu}$.

Chapter V

Conclusion

In chapter II, we have presented an analysis of the phase-calibration problem encountered in astronomy when mapping incoherent sources with aperture-synthesis devices. More precisely, we concentrated on the phase-calibration operation involved in the self-calibration procedures of phase-closure imaging. In radio-astronomy, the related optimization problems have been stated and solved hitherto at the phasor level. By conducting our analysis at the phase level, we derived a new method for diagnosing and solving the difficulties of the phasor approach.

In chapter III, we have presented the new method we proposed in Lannes & Prieur (2013) for solving the nearest-lattice point (NLP) problems arising in astronomy, geodesy and GNSS. The main theoretical aspects of the matter were also analysed. This contribution concerns both the preconditioning stage, and the discrete-search stage in which the integer ambiguities are finally fixed. We have proposed several algorithms whose efficiency was shown via intensive numerical tests on GNSS data. The same algorithms can be used in the astronomical self-calibration procedures. The related NLP problems are indeed very similar.

Concerning the preconditioning stage, we have shown in Sect. III.3.3 that the LLL-type algorithms with delayed size-reduction lead to a gain of the order of two relative to the standard LLL algorithm. We have also particularly optimized the discrete-search (DS) algorithms (see Sect. sec:integ-5.3). Our DS algorithms run also about twice as fast as the state-of-the-art DS algorithms of Jazaeri et al. (2012). We have thus been able to perform intensive calculations on large-size problems with our old computers. This would be particularly interesting for real-time data processing of world-wide global GNSS networks. As explicitly shown in Lannes (2013), parallel versions of our LLL-type algorithms could also be implemented for those extreme cases.

In astronomy, our self-calibration approach could lead to a substantial gain in computing time for large interferometric arrays. Another important asset of our approach is to propose a method for validating the calibration solution. For each phase-calibration operation, we determine the global minimum of the arc functional and the first secondary minima (if any); see Sects. III.1.1, III.4.3 and Sect. II.4. In the case of multiple minima, the relative discrepancy between the values the global and secondary minima provides a measure against which the reliability of the process can be appreciated. This is an innovative approach which could promote the use of the self-calibration procedures in radio imaging.

In chapter IV, we have examined the problem of calibrating the clock (or pseudo-clock) phase biases of GNSS networks. In the context specified in Sect. IV.1, the basic rank defect

of this problem is related to the way these phase biases and the carrier-wave ambiguities are involved in the observational equations. We have analyzed the different ways of removing this rank defect, and defined an efficient strategy for obtaining clock-phase bias estimates in a standard form.

This rank defect is intrinsically related to the structure of Eq. (IV.1), and cannot be resolved by additional experimental data. As a result, according to Eq. (IV.13), any variation of the vertex ambiguity vector μ can be compensated by a variation of the edge-ambiguity function N; μ is a vector of $\mathbb{Z}^{n_{\rm st}}$; see Sect. IV.2.2. For PPP applications, the satellite clock-phase biases can be estimated up to constant integers. The choice of μ in $\mathbb{Z}^{n_{\rm st}}$ does not therefore affect the significant part of these bias values. The retrieved ambiguities are of course affected by this choice, but this has no actual impact on the GNSS results such as the estimates of the station-position parameters, for example.

We propose a new approach in chapter IV, the closure-ambiguity approach (CAA), which is a simple and efficient way for finding a solution; see Sects. IV.4 to IV.6. It is an undifferenced method based on some particular constraints. The related choice of μ, which is associated with the notion of closure ambiguity, is similar to that implicitly made by de Jonge (1998); it defines the very principle of the closure-ambiguity approach. Thanks to the introduction of graph $\mathscr{G}$, the union of the observational graphs until the current epoch, the closure ambiguities are dealt with in an optimal manner. In particular, compared to the approaches presented in de Jonge (1998) and Lannes & Gratton (2009), no graph transition is to be performed. These new aspects have been illustrated with some simple and generic examples; see Figs. IV.1 and IV.5.

We have analysed the main classes of other methods used for calibrating GNSS networks, and established the link between those methods and our CAA method. More precisely, we present a unified theoretical framework in which all those methods can be understood and compared more easily (see Sect. IV.7). We have thus been able to derive functional relations between the solutions provided by the methods of Blewitt (1989); Collins et al. (2010); Loyer et al. (2012). Those solutions are displayed in Fig. IV.6 which gives a synthetic representation of the results provided by our approach (CAA) and all those methods.

We have also shown that the intermediate differencing stage of Blewitt's approach can be avoided, without any counterpart, by removing the rank defect via our approach or that of Teunissen (1984): the closure ambiguities to be fixed then appear in the very formulation of the UD problem to be solved. The NLP techniques of ambiguity resolution can thus be directly applied to the float solution; see Sect. IV.6.1. Compared to the hybrid UD-DD-UD methods deriving from the basic contribution of Blewitt (1989), the technical implementation of the CAA method is simpler and more efficient; some CPU time can thus be saved.

It also appeared that the concept of ambiguity datum fixing of Collins et al. (2010) comes within our CAA framework. When the ambiguities are fixed at zero on the edges of some arbitrary spanning tree of $\mathscr{G}$, the remaining ambiguities to be fixed are nothing else than the closure ambiguities of the ambiguity functional N under consideration. Compared to Blewitt's approach, one may therefore say that the UD approaches of Collins et al. (2010) and Loyer et al. (2012) are closer to our CAA-(S-system) principle.

To compare the methods providing LS estimates of clock-phase biases, we have introduced a reference particular solution. For this solution, when a clock-phase bias is estimated for the first time, its fractional part is confined to the one-cycle width interval centred on zero; the integer-ambiguity function is modified accordingly. The notion of reference solu-

tion is very useful for testing the compatibility of all those GNSS methods: pertinent methods should lead to the same reference ambiguity solution; see Fig. IV.6. This test is independent of the selected spanning tree.

The QR implementation of the CAA method has been presented in Sect. IV.6 and I, in an exhaustive manner. Completed by the NLP algorithms developed for huge networks (Lannes, 2013; Lannes & Prieur, 2013), all the elements are now gathered for implementing very powerful techniques. The implications of our approach may concern the software packages used for processing most GNSS networks, and particularly global networks using GPS, GLONASS, Galileo, IRNSS or BeiDou/Compass satellites. The CAA principle is well suited for handling the integer-ambiguity problem of all those networks. As outlined in 6.3, it could lead to applications in real-time kinematic precise-point positioning (RTK-PPP).

In order to facilitate its integration in existing GNSS software packages, our approach has been described in a precise manner. The gain of such an implementation would be both in terms of performance and reliability. Indeed, our undifferenced approach proposes an appropriate procedure for ambiguity resolution and clock-phase bias calibration, and this with a rigorous handling of the rank defect to be removed. Another possible application of this work would be to use the method described in Sect. IV.2.5 for comparing the solutions obtained by different software packages.

Appendix A

On some class of integer matrices

In Sect. II.2, we have introduced integer matrices whose entries are equal to ± 1 or 0; see, for example, Eqs. (II.31), (II.32) and (II.47). These matrices lie in a particular class of integer matrices denoted by $\mathcal{M}$: an $[m \times n]$ matrix $[A]$ lies in $\mathcal{M}$ if its matrix elements $a_{i,j}$ are equal to ± 1 or 0. In this appendix, to analyse these matrices, we introduce two subclasses of $\mathcal{M}$: $\mathcal{M}_1$ and $\mathcal{M}_2$ (Sects. A.1 and A.2, respectively). With the aid of the 'founding algorithm' presented in Sect. A.3, we are then be able to give a deeper insight into their structure, rank and inverse if any (Sect. A.4).

A.1 Class $\mathcal{M}_1$

Let $[A]$ be some matrix in $\mathcal{M}$ such that for some i, i' and j, we have either $a_{i,j} = \pm 1$ with $a_{i',j} = \mp 1$ (Case 1), or $a_{i,j} = \pm 1$ with $a_{i',j} = \pm 1$ (Case 2). We then denote by $[\mathcal{S}]$ the unimodular matrix that reduces $a_{i',j}$ to 0. In Case 1, $[\mathcal{S}]$ is the unimodular matrix $[\mathcal{S}_+]$ that adds the entries of line i to the corresponding entries of line i'. In Case 2, $[\mathcal{S}]$ is the unimodular matrix $[\mathcal{S}_-]$ that subtracts the entries of line i from the corresponding entries of line i'. Note that when $a_{i,j} = -a_{i',j}$ for all j, $[\mathcal{S}_+]$ sets to 0 all the elements of line i'. Likewise, when $a_{i,j} = a_{i',j}$ for all j, $[\mathcal{S}_-]$ annihilates line i'.

By definition, $\mathcal{M}_1$ is the class of matrices $[A]$ for which the following property holds: *under successive actions of appropriate* $[\mathcal{S}]$ *operators,* $[A]$ *remains in* $\mathcal{M}$.

Example 1. Baseline matrices of type B. Let $\mathscr{V}$ be a set of $n_{\rm v}$ (vertices or) pupil elements such as that defined in Sect. II.1.1: $\mathscr{V} \stackrel{\rm def}{=} \{{\rm v}_1, {\rm v}_2, \ldots, {\rm v}_{n_{\rm v}}\}$. Now, consider a matrix $[A]$ of $\mathcal{M}$ built as follows. For each line, choose two distinct pupil elements of $\mathscr{V}$: ${\rm v}_p$ and ${\rm v}_q$. Then, set $a_{i,p} = 1$, $a_{i,q} = -1$, and $a_{i,j} = 0$ for $j \neq p$ and $j \neq q$. Each line of $[A]$ is thus associated with a directed (edge or) baseline $({\rm v}_p, {\rm v}_q)$. When two distinct baselines $({\rm v}_p, {\rm v}_q)$ and $({\rm v}_{p'}, {\rm v}_{q'})$ share only one pupil element, the action of either $\mathcal{S}_+$ or $\mathcal{S}_-$ amounts to associating another directed baseline with line i'. As a result, $[A]$ lies in $\mathcal{M}_1$. In this framework, it is also clear that if some column(s) of such a matrix is (are) removed, the matrix thus obtained also lies in $\mathcal{M}_1$. This shows, in particular, that the matrices $[B]$ and $[B_{\rm st}]$ of Sect. II.2.2 lie in $\mathcal{M}_1$; see Eqs. (II.31) and (II.32) for example.

Example 2. Closure matrices of type $\mathcal{C}_3^\circ$. Let $\mathbb{G} \stackrel{\rm def}{=} (\mathscr{V}, \mathscr{E})$ be some phase-calibration graph; see Sect. II.1.1. A triangle whose edges lie in $\mathscr{E}$ defines a loop of order 3. All the triangles of $\mathbb{G}$ can be listed. For example, the graph of Fig. II.4 includes the loops of order three

$(1,2,3)$, $(1,2,4)$, $(1,3,4)$ and $(2,3,4)$. Clearly, such loops are involved in the definition of the usual closure phases. Consider, for example, loop $(1,2,3)$. For any $\beta_{\rm b}$ in $E_{\rm b}$ (see Sect. II.2.2), we have for this first loop:

$$\beta_{\rm b}(1,2) + \beta_{\rm b}(2,3) - \beta_{\rm b}(1,3) = 0$$

i.e.,

$$[_1]\beta_{\rm b} = 0$$

where $[_1]$ is a line matrix with 6 elements. The entries of $[_1]$ associated with the edges $(1,2)$, $(2,3)$ and $(1,3)$ are 1, 1 and -1, respectively; the other entries are zero. We can thus build a matrix $[\mathcal{C}_3^\circ]$ of $\mathcal{M}$ whose lines $_i$ are associated with the loops of order 3. In this example, $[\mathcal{C}_3^\circ]$ includes 4 lines and 6 columns. The baselines being ordered as specified in Sect. II.2.6, we then have [Eq. (II.47)]:

$$[\mathcal{C}_3^\circ] = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix} \tag{A.1}$$

By construction, such a 'closure matrix' annihilates the pupil phases.

When two loops of order 3 associated with lines i and i' share some baseline, the action of either $[\mathcal{S}_+]$ or $[\mathcal{S}_-]$ amounts to associating with line i' a loop of order 4. By construction, the entries of that line are still equal to ± 1 or 0. Without giving here further details, we may therefore surmise that the closure matrices $[A] \overset{\rm set}{=} [\mathcal{C}_3^\circ]$ lie in $\mathcal{M}_1$: under successive actions of appropriate $[\mathcal{S}]$ operators, $[A]$ remains in $\mathcal{M}$. This can be checked *a posteriori* by applying to $[A]$ the founding algorithm described in Sect. A1.3. Like in Example 1, if some column(s) of such a matrix is (are) removed, the matrix thus obtained also lies in $\mathcal{M}_1$.

A.2 Class $\mathcal{M}_2$

Let $[A]$ be some matrix in $\mathcal{M}$. For each line i of $[A]$, we denote by j_i the index j of the first column for which $a_{i,j}$ is not zero. When $a_{i,j}$ is equal to 0 for all $j \leq n$, we set $j_i \overset{\rm set}{=} n+1$. Matrix $[A]$ lies in $\mathcal{M}_2$ if the following conditions are satisfied:

1) For $i = 1, \ldots, m$: when $j_i \leq n$, then $a_{i,j_i} = 1$.

2) For $i = 1, \ldots, m-1$: when $j_i < n$, then $j_{i+1} > j_i$; when $j_i = n$ (or $n+1$), then $j_{i'} = n+1$ for $i' = i+1, \ldots, m$.

Examples of such matrices are to be found in Sect. A1.3 and A1.4: see Eqs. (A.2), (A.6) and (A.11).

A.3 Founding algorithm

Before stating the following property, we recall that a unimodular matrix is a square integer matrix with determinant ± 1.

Property. *For any* $[A]$ *in* $\mathcal{M}_1$, *there exists a unimodular matrix* $[\mathcal{Z}]$ *such that* $[\mathcal{Z}][A]$ *lies in* $\mathcal{M}_2$.

The following algorithm, which yields $[\mathcal{Z}][A]$ and $[\mathcal{Z}]$ explicitly, can be regarded as a proof of this property.

Step 0: *Initialization*
Set $i_0 \overset{\text{set}}{=} 1$, $j_0 \overset{\text{set}}{=} 1$, and $[\mathcal{Z}] \overset{\text{set}}{=} [\mathcal{I}_m]$, where $[\mathcal{I}_m]$ is the $[m \times m]$ identity matrix.

Step 1: *Define the column* j_0 *to be processed*
While $a_{i,j_0} = 0$ for all $i \geq i_0$ with $j_0 < n$, set $j_0 \overset{\text{set}}{=} j_0 + 1$. If $a_{i,j_0} = 0$ for all $i \geq i_0$ with $j_0 = n$, terminate the process.

Step 2: *If need be, perform some line permutation*
Denote by i_1 the smallest integer $i \geq i_0$ for which a_{i_1,j_0} is equal to ± 1. When i_1 is not equal to i_0, permute lines i_0 and i_1. The corresponding operation can be written in the form $[A] \overset{\text{set}}{=} [\P][A]$ where $[\P]$ is a permutation operator, a trivial unimodular matrix. Update $[\mathcal{Z}]$: $[\mathcal{Z}] \overset{\text{set}}{=} [\P][\mathcal{Z}]$.

Step 3: *If need be, set* $a_{i_0,j_0} = 1$
If $a_{i_0,j_0} = -1$, set $[A] \overset{\text{set}}{=} [\mathcal{S}][A]$ and $[\mathcal{Z}] \overset{\text{set}}{=} [\mathcal{S}][\mathcal{Z}]$, where $[\mathcal{S}]$ is the trivial unimodular operator that changes the sign of the entries of line i_0.

Step 4: *If need be, set to* 0 *the lower part of column* j_0
For all $i > i_0$, if $a_{i,j_0} = \pm 1$, then combine lines i and i_0 so that $a_{i,j_0} = 0$. In other terms, set $[A] \overset{\text{set}}{=} [\mathcal{S}_\mp][A]$, and update $[\mathcal{Z}]$ consequently: $[\mathcal{Z}] \overset{\text{set}}{=} [\mathcal{S}_\mp][\mathcal{Z}]$.

Step 5: *Termination test*
If $i_0 = m$ or $j_0 = n$, terminate the process. Otherwise, set $i_0 \overset{\text{set}}{=} i_0 + 1$, $j_0 \overset{\text{set}}{=} j_0 + 1$, and go to *Step 1*.

Example 3: Rank of the closure matrices of type $\mathcal{C}_3^\circ$. Let us consider, for example, the closure matrix $[\mathcal{C}_3^\circ]$ of Fig. II.4: Eq. (A.1). As already pointed out, this type of matrix lies in $\mathcal{M}_1$; see Example 2 of Sect. A1.1. The founding algorithm then yields, without any line permutation, the following matrix of $\mathcal{M}_2$:

$$[\mathcal{Z}][\mathcal{C}_3^\circ] = \begin{bmatrix} \mathbf{1} & -1 & 0 & 1 & 0 & 0 \\ 0 & \mathbf{1} & -1 & -1 & 1 & 0 \\ 0 & 0 & \mathbf{0} & \mathbf{1} & -1 & 1 \\ 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{A.2}$$

The column and row ranks of this matrix are equal to 3. As a result, $n_{c3} \overset{\text{def}}{=} \text{rank}\,[\mathcal{C}_3^\circ] = \text{rank}\,[\mathcal{C}_3^\circ]^{\text{t}} = 3$. The closure phases defined via the first three lines of $\mathcal{C}_3^\circ$, i.e., those associated with the triangles $(1,2,3)$, $(1,2,4)$ and $(1,3,4)$, are linearly independent. Indeed, the first three lines of matrix (A.2) were obtained from matrix (A.1) without any permutation. The matrix $[\mathcal{C}_3]$ that provides the closure phases of a vector β of E on these loops is therefore the following [Eq. (II.48)]:

$$[\mathcal{C}_3] = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 \end{bmatrix} \tag{A.3}$$

In the case of Fig. II.1, the founding algorithm applied to the corresponding closure matrix $\mathcal{C}_3^\circ$ shows that $n_{c3} = 5$; $\mathcal{C}_3$ then includes 5 lines. Note that we then also have $n_{c3} = n_c$.

A.4 Unimodular matrices of $\mathcal{M}_1$

Let us consider the special case where $[A]$ is an invertible $[n \times n]$-matrix of $\mathcal{M}_1$. There then exists some unimodular matrix $[\mathcal{Z}]$ such that $[\mathcal{Z}][A]$ is an upper-triangular matrix $[U]$ with rank n and determinant 1; see Sect. A1.3. The determinant of A is then the same as that of $[\mathcal{Z}]$: ± 1; $[A]$ is therefore unimodular; its inverse is then given by the relation

$$[A]^{-1} = [U]^{-1}[\mathcal{Z}] \tag{A.4}$$

Indeed,

$$\begin{aligned} ([U]^{-1}[\mathcal{Z}])\,[A] &= [U]^{-1}([\mathcal{Z}][A]) \\ &= [U]^{-1}[U] \\ &= [\mathcal{I}_n] \end{aligned}$$

As $[U]$ is an upper-triangular matrix, the j^{th} column of $[A]^{-1}$ can be obtained from the j^{th} column of $[\mathcal{Z}]$ by back-substitution.

Example 4. Inverse of $[B_{st}]$ matrices. Consider the special case of Example 1 (Sect. A1.1) where $[A]$ is the matrix $[B_{st}]$ defined by Eq. (II.32):

$$[A] = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \tag{A.5}$$

Then,

$$[U] = \begin{bmatrix} \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & -1 \\ 0 & 0 & \mathbf{1} \end{bmatrix} \tag{A.6}$$

and

$$[\mathcal{Z}] = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & -1 \end{bmatrix} \tag{A.7}$$

hence [the alternative way of getting Eq. (II.37)]:

$$[A]^{-1} = [U]^{-1}[\mathcal{Z}] = \begin{bmatrix} -1 & 0 & 0 \\ -1 & 1 & -1 \\ -1 & 0 & -1 \end{bmatrix} \tag{A.8}$$

Example 5. Computation of the closure phases $\beta_c(i_\ell, j_\ell)$ from a maximum set of independent closure phases of order three. In Sect. II.2.6, we have shown that when $n_{c3} = n_c$, then

$\ker \mathcal{C}_3 = \ker C = E_\mathrm{b}$; see Eqs. (II.51) and (II.52). Let us then consider the operator from E_c into $\mathbb{R}^{n_\mathrm{c}}$ that provides the closure-phase vector of order three of a vector of E_c:

$$A\colon E_\mathrm{c} \to \mathbb{R}^{n_\mathrm{c}}, \qquad A\beta_\mathrm{c} \stackrel{\mathrm{def}}{=} \mathcal{C}_3\beta_\mathrm{c}$$

We then have $\ker A = E_\mathrm{b} \cap E_\mathrm{c}$, hence $\ker A = \{0\}$ from Property 1. Furthermore, $\dim E_\mathrm{c} = n_\mathrm{c}$ [Eq. (II.30)]; A is then invertible. The closure phases $\beta_\mathrm{c}(i_\ell, j_\ell)$ can then be expressed as linear combinations of independent closure phases of order three.

To clarify this point in a concrete manner, let us consider the special case of Fig. II.4. The matrix $[\mathcal{C}_3]$ that provides the closure phases of a vector β of E on the loops $(1,2,3)$, $(1,2,4)$ and $(1,3,4)$ is that of Eq. (A.3). The column vectors of $[\mathcal{C}_3]$ relative to the loop-closure baselines $(2,3)$, $(1,3)$ and $(1,4)$ of Fig. II.4 form, in this order, the following matrix $[A]$:

$$[A] \stackrel{\mathrm{def}}{=} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix} \tag{A.9}$$

As $\beta = \beta_\mathrm{b} + \beta_\mathrm{c}$ with $\beta_\mathrm{c} = C\beta$ [Eq. (II.42)], and $\mathcal{C}_3\beta_\mathrm{b} = 0$ (since here, in particular, $\ker \mathcal{C}_3 = E_\mathrm{b}$), we thus have

$$[\mathcal{C}_3\beta] = [A][C\beta] \tag{A.10}$$

Here, $[C\beta] = \begin{bmatrix}\beta_\mathrm{c}(2,3) & \beta_\mathrm{c}(1,3) & \beta_\mathrm{c}(1,4)\end{bmatrix}^\mathrm{t}$. Clearly, $[A]$ is a unimodular matrix of $\mathcal{M}_1$. Applied to this matrix, the founding algorithm then yields

$$[U] = \begin{bmatrix} \mathbf{1} & -1 & 0 \\ 0 & \mathbf{1} & -1 \\ 0 & 0 & \mathbf{1} \end{bmatrix} \tag{A.11}$$

and

$$[\mathcal{Z}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \tag{A.12}$$

hence

$$[A]^{-1} = [U]^{-1}[\mathcal{Z}] = \begin{bmatrix} 1 & -1 & 1 \\ 0 & -1 & 1 \\ 0 & -1 & 0 \end{bmatrix} \tag{A.13}$$

We thus have, from Eq. (A.10),

$$[C\beta] = [A]^{-1}[\mathcal{C}_3\beta] \tag{A.14}$$

The closure phases $\beta_\mathrm{c}(i_\ell, j_\ell)$ can thus be computed via the closure phases of order three.

As illustrated through this example, when $n_{\mathrm{c}3} = n_\mathrm{c}$, there exists an operator that provides the closure phases $\beta_\mathrm{c}(i_\ell, j_\ell)$ from a maximum set of independent closure phases of order three. However, that does not mean that the closure phases $\beta_\mathrm{c}(i_\ell, j_\ell)$ are to be computed that way. In all cases, it is much simpler to compute them directly from β via Eqs. (II.40) and (II.35).

Appendix B

Proof of Property 2

Let α_g be a τ-constrained minimizer of g; see Sect. II.4.1. Then,

$$\begin{aligned}\varepsilon_{\alpha_g} &\stackrel{\text{def}}{=} \operatorname{arc}(\widehat{\varphi}_{\text{c}} - B\alpha_g) \\ &= (\widehat{\varphi}_{\text{c}} - B\alpha_g) - 2\pi\nu\end{aligned}$$

where [see Eq. (II.20) in terms of functions]

$$\nu \stackrel{\text{def}}{=} \left\lfloor \frac{\widehat{\varphi}_{\text{c}} - B\alpha_g}{2\pi} \right\rceil$$

Moreover, $|\varepsilon_{\alpha_g}| \leq 2\pi\tau$ all over $\mathscr{E}$ [Eq. (II.86)]. There then exists a small ball of V_b centred on α_g whose points α are such that

$$\left\lfloor \frac{\widehat{\varphi}_{\text{c}} - B\alpha}{2\pi} \right\rceil = \nu$$

In this ball, we therefore have

$$\begin{aligned}\operatorname{arc}(\widehat{\varphi}_{\text{c}} - B\alpha) &= (\widehat{\varphi}_{\text{c}} - B\alpha) - 2\pi\nu \\ &= (\widehat{\varphi}_{\text{c}} - 2\pi\nu) - B\alpha\end{aligned}$$

and thereby $g(\alpha) = \|(\widehat{\varphi}_{\text{c}} - 2\pi\nu) - B\alpha\|_w^2$. As α_g is a minimizer of g, it then follows from Eq. (II.57) that

$$\alpha_g = B^+(\widehat{\varphi}_{\text{c}} - 2\pi\nu)$$

According to Property 1, ν can be decomposed in the form

$$\nu = B\mu_\nu + \nu_{\text{c}}$$

where $\mu_\nu \stackrel{\text{def}}{=} B_{\text{st}}^{-1} Q_{\text{st}} \nu$; see Eq. (II.36). As a result,

$$\alpha_g = B^+\widehat{\varphi}_{\text{c}} - 2\pi B^+(B\mu_\nu + \nu_{\text{c}})$$

i.e., $\alpha_g = B^+(\widehat{\varphi}_{\text{c}} - 2\pi\nu_{\text{c}}) - 2\pi\mu_\nu$. Denoting by $\dot{v}$ the point of $\mathbb{Z}^{n_{\text{c}}}$ defined by ν_{c}, we therefore have, from the definition of $\hat{v}$ [Eq. (II.88)],

$$\alpha_g = 2\pi B^+(\hat{v} - \dot{v}) - 2\pi\mu_\nu$$

As a result [see the definition of $\dot{\alpha}$ in Eq. (II.89)],

$$\alpha_g = \dot{\alpha} - 2\pi\mu_\nu$$

On each pupil element of $\mathscr{V}$, α_g is therefore equal to $\dot{\alpha}$ modulo 2π. As a corollary, we of course have $e^{\mathrm{i}\alpha_g} = e^{\mathrm{i}\dot{\alpha}}$. As $B\alpha_g = B\dot{\alpha} - 2\pi B\mu_\nu$, we also have

$$\begin{aligned}\varepsilon_{\alpha_g} &\stackrel{\text{def}}{=} \operatorname{arc}(\widehat{\varphi}_\mathrm{c} - B\alpha_g)\\ &= \operatorname{arc}(\widehat{\varphi}_\mathrm{c} - B\dot{\alpha})\\ &= \varepsilon_{\dot{\alpha}}\end{aligned}$$

It then follows from Eq. (II.82) that $g(\alpha_g) = g(\dot{\alpha})$. Furthermore, using the relations established in this appendix, we have

$$\begin{aligned}\varepsilon_{\alpha_g} &= (\widehat{\varphi}_\mathrm{c} - B\alpha_g) - 2\pi\nu\\ &= (\widehat{\varphi}_\mathrm{c} - 2\pi\nu) - 2\pi BB^+(\hat{v} - \dot{v}) + 2\pi B\mu_\nu\\ &= \widehat{\varphi}_\mathrm{c} - 2\pi(\nu - B\mu_\nu) - 2\pi BB^+(\hat{v} - \dot{v})\\ &= (\widehat{\varphi}_\mathrm{c} - 2\pi\nu_\mathrm{c}) - 2\pi BB^+(\hat{v} - \dot{v})\\ &= 2\pi[(\hat{v} - \dot{v}) - BB^+(\hat{v} - \dot{v})]\end{aligned}$$

hence, from Eq. (II.60),

$$\varepsilon_{\alpha_g} = 2\pi S(\hat{v} - \dot{v})$$

As $|\varepsilon_{\alpha_g}| \leq 2\pi\tau$, we therefore have $|S(\hat{v} - \dot{v})| \leq \tau$; $\dot{v}$ therefore lies in Γ_τ; see Eq. (II.90). We have thus shown that every τ-constrained minimizer of g was associated with a point $\dot{v}$ of Γ_τ.

It remains to prove the converse, i.e., if $\dot{v}$ is a point of Γ_τ, there then exists a τ-constrained minimizer of g associated with this point. Let us consider the particular pupil-phase function [see Eq. (II.89)]

$$\dot{\alpha} \stackrel{\text{def}}{=} 2\pi B^+(\hat{v} - \dot{v})$$

where $\dot{v}$ lies in Γ_τ. We then have, from Eq. (II.60),

$$2\pi(\hat{v} - \dot{v}) - B\dot{\alpha} = 2\pi S(\hat{v} - \dot{v})$$

hence, since $\dot{v}$ is in Γ_τ: $|2\pi(\hat{v} - \dot{v}) - B\dot{\alpha}| \leq 2\pi\tau$ all over $\mathscr{E}$. There therefore exists a small ball of V_b centred on $\dot{\alpha}$ whose points α are such that

$$\left\lfloor \frac{2\pi\hat{v} - B\alpha}{2\pi} \right\rceil = \dot{v}$$

In this ball

$$\begin{aligned}\operatorname{arc}(\widehat{\varphi}_\mathrm{c} - B\alpha) &= (2\pi\hat{v} - B\alpha) - 2\pi\dot{v}\\ &= 2\pi(\hat{v} - \dot{v}) - B\alpha\end{aligned}$$

From the definition of $\dot{\alpha}$ and Eq. (II.57), it follows that $\dot{\alpha}$ is a τ-constrained minimizer of g.

Appendix C

Point search in integer lattices

This appendix describes the method that we presented in the appendix 3 of Lannes & Prieur (2011). A more recent and complementary analysis of the NLP problem can be found in Sect. III.

Let us consider the ellipsoid defined by Eq. (II.93),

$$\mathfrak{E}(\mathfrak{c}) \stackrel{\text{def}}{=} \{v \in \mathbb{R}^{n_\text{c}} : \omega(v - \hat{v}) \leq \mathfrak{c}\} \tag{C.1}$$

where ω is the quadratic form

$$\omega(v) \stackrel{\text{def}}{=} \|T_\text{c} v\|^2 \tag{C.2}$$

Clearly, we have

$$\omega(v) = (v \mid \Omega v)_{\mathbb{R}^{n_\text{c}}} \quad \text{with} \quad \Omega = [T_\text{c}]^\text{t}[T_\text{c}] \tag{C.3}$$

As $[T_\text{c}]^\text{t}[T_\text{c}]$ is positive definite (see Sect. 3.8), ω is a positive-definite quadratic form.

In this appendix, we show how to list the points $\dot{v}$ of $\mathbb{Z}^{n_\text{c}} \cap \mathfrak{E}(\mathfrak{c})$, and for each of them (if need be), how to compute $\omega(\dot{v} - \hat{v})$, and thereby $g(\dot{\alpha}) = 4\pi^2\omega(\dot{v} - \hat{v})$; see Eq. (II.96). For clarity, we now set

$$n \stackrel{\text{set}}{=} n_\text{c}$$

In most cases encountered in practice, the standard basis of lattice $\mathbb{Z}^n$ is far from being orthogonal for the inner product induced by Ω. In other terms, in this basis, Ω is far from being diagonal. When n is large, listing the points of $\mathbb{Z}^n \cap \mathfrak{E}(\mathfrak{c})$ may then be time consuming.

To circumvent this difficulty, we need to find a basis of $\mathbb{Z}^n$ in which the matrix of ω, $\bar{\Omega}$, is as diagonal as possible, with diagonal terms of the same order of magnitude (as much as possible). Such a basis is said to be a 'reduced basis' of lattice $(\mathbb{Z}^n, \omega)$. As Ω is the inverse of the variance-covariance matrix of the closure-phase vector [see Eqs. (C.3) and (II.70)], exhibiting a reduced basis amounts to performing a decorrelation process, hence the title of Sect. A3.1. As specified in Sect. A3.2, the discrete search is performed in the selected reduced basis.

C.1 Decorrelation method

The decorrelation methods to be implemented must somehow refer to the principle of the LLL algorithm, an algorithm devised by Lenstra, Lenstra & Lovász (1982) (see, e.g., Cohen

(1996)). The approach presented in this section corresponds to the 'new implementation' of this algorithm devised by Luk & Tracy (2008).

One then starts from the Cholesky factorization

$$\Omega = R^{\mathrm{t}} R \tag{C.4}$$

in which R is an upper-triangular matrix with positive diagonal elements. Note that this Cholesky factor R can be directly obtained from T_{c} by QR factorization; see, e.g., Sect. 7.1 in Lannes & Gratton (2009). The procedure proposed by Luk & Tracy (2008) provides a reduced upper-triangular matrix of the form

$$\bar{R} = \mathcal{Q} R \mathcal{Z} \tag{C.5}$$

in which $\mathcal{Q}$ is an orthogonal matrix ($\mathcal{Q}^{-1} = \mathcal{Q}^{\mathrm{t}}$), and $\mathcal{Z}$ is a unimodular matrix. (A unimodular matrix is a square integer matrix with determinant ± 1; its inverse is also an integer matrix.) Denoting by $\bar{r}_{\ell,m}$, the matrix elements of $\bar{R}$, let us set

$$\bar{\kappa}_\ell \overset{\text{def}}{=} \frac{\bar{r}_{\ell,\ell+1}}{\bar{r}_{\ell,\ell}} \qquad (1 \leq \ell < n)$$

Luk and Tracy define $\mathcal{Q}$ and $\mathcal{Z}$ so that the following LLL conditions are satisfied:

(i) $\bar{r}_{\ell,\ell} > 2|\bar{r}_{\ell,m}|$ (for $1 \leq \ell < m \leq n$)

(ii) $\bar{r}^2_{\ell,\ell} \geq (\gamma - \bar{\kappa}^2_{\ell-1})\bar{r}^2_{\ell-1,\ell-1}$ (for $2 \leq \ell \leq n$)
with $1/4 < \gamma < 1$

In practice, for optimal decorrelation, γ is set equal to 0.999. From Condition (i), we have

$$\bar{\kappa}^2_\ell < 1/4$$

It then follows from Condition (ii) that

$$\bar{r}^2_{\ell,\ell} \geq \frac{3}{4}\,\bar{r}^2_{\ell-1,\ell-1} \qquad (\text{for } 2 \leq \ell \leq n) \tag{C.6}$$

This pointed out, $\bar{r}_{\ell,\ell}$ is generally less than $\bar{r}_{\ell-1,\ell-1}$.

According to Eqs. (C.2) to (C.5), we have, since $\mathcal{Q}^{\mathrm{t}}\mathcal{Q}$ is the identity on $\mathbb{R}^n$,

$$\begin{aligned} \omega(\dot{v} - \hat{v}) &= \|R(\dot{v} - \hat{v})\|^2_{\mathbb{R}^n} \\ &= \|\bar{R}\big(\bar{v} - \bar{\hat{v}}\big)\|^2_{\mathbb{R}^n} \end{aligned} \tag{C.7}$$

where

$$\bar{v} \overset{\text{def}}{=} \mathcal{Z}^{-1}\dot{v}, \qquad \bar{\hat{v}} \overset{\text{def}}{=} \mathcal{Z}^{-1}\hat{v} \tag{C.8}$$

The selected reduced basis is defined by the column vectors of $\mathcal{Z}$. The entries of $\bar{v}$ are therefore the components of the integer-ambiguity vector $\dot{v}$ in that reduced basis, and likewise for $\bar{\hat{v}}$. The integer matrices $\mathcal{Z}$ and $\mathcal{Z}^{-1}$ are progressively built through the process. Let us finally note that $\mathcal{Q}$ is not explicitly built.

C.2 Discrete search process

In this section, we identify the points of $\mathbb{Z}^n \cap \mathfrak{E}(\mathfrak{c})$ in the reduced basis defined by $\mathcal{Z}$.

From Eq. (C.7), we have $\omega(\dot{v} - \hat{v}) = \bar{\omega}(\bar{\dot{v}} - \bar{\hat{v}})$, where $\bar{\omega}$ is the quadratic form with matrix $\bar{\Omega} = \bar{R}^{\mathrm{t}}\bar{R}$. For clarity, in the remainder of this section, we will omit the bars on $\dot{v}$, $\hat{v}$, ω, Ω and R. As

$$\omega(\dot{v} - \hat{v}) = \big[R(\dot{v} - \hat{v})\big]^{\mathrm{t}}\big[R(\dot{v} - \hat{v})\big]$$

we have

$$\omega(\dot{v} - \hat{v}) = \sum_{\ell=1}^{n} w_\ell^2 \tag{C.9}$$

where

$$w_\ell \overset{\text{def}}{=} \sum_{m=\ell}^{n} r_{\ell,m} v^{(m)} \tag{C.10}$$

with

$$v^{(m)} \overset{\text{def}}{=} \dot{v}^{(m)} - \hat{v}^{(m)} \tag{C.11}$$

Here, we have implicitly set $v \overset{\text{def}}{=} \dot{v} - \hat{v}$ and $w \overset{\text{def}}{=} Rv$; $\dot{v}^{(m)}$ is the mth component of $\dot{v}$ (in the selected basis), i.e., the mth integer ambiguity (in that basis), whereas $\hat{v}^{(m)}$ is the mth component of the float solution $\hat{v}$ (in the same basis).

We now show that the ambiguity vectors $\dot{v}$ of $\mathfrak{E}(\mathfrak{c})$ can be identified by 'conditioning,' successively, their components $\dot{v}^{(n)}, \dot{v}^{(n-1)}, \ldots, \dot{v}^{(1)}$. Given some $p \leq n$, assume that

1) when p is strictly less than n, the ambiguities $\dot{v}^{(\ell)}$ for $\ell = n, n-1, \ldots, p+1$ have already been conditioned properly;

2) the ambiguities $\dot{v}^{(\ell)}$ for $\ell = p, p-1, \ldots, 1$ have not been conditioned yet.

The problem is to define the values of $\dot{v}^{(p)}$ so that the ambiguity vector $\dot{v}$ thus progressively defined (if possible) lies in ellipsoid $\mathfrak{E}(\mathfrak{c})$.

Conditioning interval for $\dot{v}^{(p)}$. The following condition must then be satisfied [see Eqs. (C.1), (C.7) and (C.9)]:

$$\begin{cases} w_p^2 \leq \mathfrak{c} & \text{if } p = n; \\ w_p^2 + \sum_{\ell=p+1}^{n} w_\ell^2 \;\leq\; \mathfrak{c} & \text{if } 1 \leq p < n \end{cases}$$

By setting

$$y_p \overset{\text{def}}{=} w_p^2 \qquad (1 \leq p \leq n) \tag{C.12}$$

and

$$z_p \overset{\text{def}}{=} \begin{cases} \mathfrak{c} & \text{if } p = n; \\ \mathfrak{c} - \sum_{\ell=p+1}^{n} w_\ell^2 & \text{if } 0 \leq p < n \end{cases} \tag{C.13}$$

the above conditions are equivalent to

$$y_p \leq z_p \qquad (1 \leq p \leq n) \tag{C.14}$$

For $0 \leq p < n$, z_p can be written in the form

$$z_p = \left(\mathfrak{c} - \sum_{\ell=p+2}^{n} w_\ell^2\right) - w_{p+1}^2$$

We thus have the recurrence formula

$$z_p = z_{p+1} - y_{p+1} \qquad (0 \leq p < n) \tag{C.15}$$

Note that

$$y_n = \left[r_{n,n} v^{(n)}\right]^2, \qquad z_n = \mathfrak{c} \tag{C.16}$$

From Eqs. (C.14) and (C.12), $|w_p|$ must be less than or equal to $\sqrt{z_p}$; we must therefore have

$$-\sqrt{z_p} \leq w_p \leq \sqrt{z_p} \tag{C.17}$$

Let us now expand w_p in the form [see Eq. (C.10)]

$$w_p = r_{p,p} v^{(p)} + \widetilde{w}_p \tag{C.18}$$

where

$$\widetilde{w}_p \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } p = n; \\ \displaystyle\sum_{m=p+1}^{n} r_{p,m} v^{(m)} & \text{if } 1 \leq p < n \end{cases} \tag{C.19}$$

As $r_{p,p}$ is positive [see Condition (i)], Equation (C.17) is therefore equivalent to

$$-\frac{1}{r_{p,p}}(\sqrt{z_p} + \widetilde{w}_p) \leq v^{(p)} \leq \frac{1}{r_{p,p}}(\sqrt{z_p} - \widetilde{w}_p)$$

It then results from Eq. (C.11) that $\dot{v}^{(p)}$ must lie in the 'conditioning interval'

$$\mathcal{J}_p \stackrel{\text{def}}{=} \left[t_p - \frac{\sqrt{z_p}}{r_{p,p}} \qquad t_p + \frac{\sqrt{z_p}}{r_{p,p}}\right] \qquad (1 \leq p \leq n) \tag{C.20}$$

where

$$t_p \stackrel{\text{def}}{=} \hat{v}^{(p)} - \frac{\widetilde{w}_p}{r_{p,p}} \tag{C.21}$$

The width of this interval (which centred on t_p) is equal to $2\sqrt{z_p}/r_{p,p}$. In the conditioning procedure, as z_{p-1} is less than or equal to z_p [see Eqs. (C.15) and (C.12)], and $r_{p-1,p-1}$ is generally greater than $r_{p,p}$, the width of $\mathcal{J}_p$ is gradually reduced. In the case where there exists no integer in $\mathcal{J}_p$, ambiguity $\dot{v}^{(p)}$ cannot be conditioned. This means that ellipsoid $\mathfrak{E}(\mathfrak{c})$ does not include any ambiguity vector of $\mathbb{Z}^n$ whose last $n - p$ components are the $\dot{v}^\ell$ for $\ell = p+1, \ldots, n$.

Listing the ambiguity vectors $\dot{v}$ of $\mathfrak{E}(\mathfrak{c})$. We are now ready to start listing the points of $\mathbb{Z}^n \cap \mathfrak{E}(\mathfrak{c})$. To follow the process described below, it is convenient to consider $\dot{v}$ as a column vector with $\dot{v}^{(n)}$ at the top, and $\dot{v}^{(1)}$ at the bottom. In practice, of course, the entries are left in their original order, from 1 to n.

We first set $p \overset{\text{set}}{=} n$. From Eqs. (C.20) and (C.21), we have, since $\widetilde{w}_n = 0$ and $z_n = \mathfrak{c}$ [see Eqs. (C.19) and (C.13)],

$$\mathcal{J}_n \overset{\text{def}}{=} \left[\hat{v}^{(n)} - \frac{\sqrt{\mathfrak{c}}}{r_{n,n}} \qquad \hat{v}^{(n)} + \frac{\sqrt{\mathfrak{c}}}{r_{n,n}}\right]$$

Provided that $\mathfrak{c}$ is sufficiently large, $\mathcal{J}_n$ includes some integer(s). Ambiguity $\dot{v}^{(n)}$ is then set equal to the 'integer rounded centre' of $\mathcal{J}_n$, i.e., the integer of $\mathcal{J}_n$ the closest to its centre; $\dot{v}^{(n)}$ being thus conditioned, we compute [see successively Eqs. (C.11), (C.16), (C.15) and (C.19)]

$$v_n, \quad y_n, \quad z_{n-1} = z_n - y_n, \quad \widetilde{w}_{n-1} = r_{n-1,n} v^{(n)}$$

We then set $p \overset{\text{set}}{=} n - 1$, and define the conditioning interval $\mathcal{J}_{n-1}$; see Eq. (C.20). Likewise, $\dot{v}^{(n-1)}$ being set equal to the integer rounded centre (if any) of $\mathcal{J}_{n-1}$, we compute

$$v_p, \quad y_p, \quad z_{p-1} = z_p - y_p, \quad \widetilde{w}_{p-1} = \sum_{m=p}^{n} r_{p-1,m} v^{(m)}$$

We then set $p \overset{\text{set}}{=} n - 2$, and define the conditioning interval for ambiguity $\dot{v}^{(n-2)}$; ambiguity $\dot{v}^{(n-2)}$ is then conditioned. We then proceed similarly for conditioning the ambiguities $\dot{v}^{(n-3)}, \ldots, \dot{v}^{(1)}$.

Without going into further details, a sequence of n entries $\dot{v}^{(p)}$ from $p = n$ to $p = 1$ (if any) can thus be obtained. Such a sequence can be visualized as a complete downward walk of a 'tree-data structure.' Clearly, from the root node (or vertex) $\dot{v}^{(n)}$, we now know the way of creating its descendants (or child nodes) if any. We are thus able to generate an 'ordered directed tree' whose nodes are (associated with) the coordinates $\dot{v}^{(p)}$ of the points $\dot{v}$ of $\mathbb{Z}^n \cap \mathfrak{E}(\mathfrak{c})$, if any. Note that some paths may be incomplete. Indeed, some conditioning intervals $\mathcal{J}_p$ may include no integer; see Eqs. (C.20) and (C.21).

There exists a one-to-one map between the 'directed complete paths' in that tree and the points $\dot{v}$ of $\mathbb{Z}^n \cap \mathfrak{E}(\mathfrak{c})$. Searching for all the points contained in $\mathfrak{E}(\mathfrak{c})$ amounts therefore to finding all the existing complete paths in this ordered directed tree. Many possible 'tree-walking' algorithms can be devised. We present below the principle of the algorithm that we have implemented to obtain the results presented in Sect. II.6. Note that, by convention, trees are drawn growing downwards (as in genealogical trees). By definition, a subtree from a given node $\dot{v}^{(p)}$ is the tree having $\dot{v}^{(p)}$ as root node and all its descendants in the original tree.

For simplicity, let us assume that we have obtained a point $\dot{v}$ (if any) corresponding to one path from the root level $p = n$ to the lowest level $p = 1$.

We then add the points (if any) corresponding to the complete paths obtained by simply using the brothers of $\dot{v}^{(1)}$.

We then successively consider all the brothers at level $p = 2$ of the parent node $\dot{v}^{(2)}$, and for each of them walk down the corresponding subtree (if any).

We repeat the process at level $p = 3$ by exploring, successively, all the brothers of $\dot{v}^{(3)}$, and walking down all the corresponding subtrees.

We continue this 'up-and-down' process at levels $p = 4$, $p = 5$, and so on, until reaching level $p = n$, thus ending our exploration of all the possible complete paths of this tree.

Values of $\omega(\dot{v} - \hat{v})$ for $\dot{v}$ in $\mathbb{Z}^n \cap \mathfrak{E}(\mathfrak{c})$. Let us now concentrate on the conditioning process at level $p = 1$. Denoting by k_0 the integer rounded centre of $\mathcal{J}_1$, we first set $\dot{v}^{(1)} \overset{\text{set}}{=} k_0$; see Eqs. (C.20) and (C.21) for $p = 1$. We then compute [see Eqs. (C.18) and (C.11)]

$$w_1 = r_{1,1}(k_0 - \hat{v}^{(1)}) + \widetilde{w}_1 \tag{C.22}$$

and $y_1 = w_1^2$. According to Eqs. (C.9), (C.13) and (C.15), the value of $\omega(\dot{v} - \hat{v})$ at the integer ambiguity vector $\dot{v}$ thus conditioned is given by the formula

$$\begin{aligned} \omega(\dot{v} - \hat{v}) &= \mathfrak{c} - z_0 \\ &= \mathfrak{c} - (z_1 - y_1) \end{aligned} \tag{C.23}$$

The brothers of $\dot{v}$ (if any) are of the form $\dot{v} + k\dot{u}_1$, where $\dot{u}_1$ is the vector with components $\dot{u}_1^{(1)} = 1$ and $\dot{u}_1^{(m)} = 0$ for $m > 1$. The corresponding values of ω are then obtained via the variational formula [see Eq. (C.7)]

$$\begin{aligned} \omega\big((\dot{v} - \hat{v}) + k\dot{u}_1\big) &= \omega(\dot{v} - \hat{v}) \\ &\quad + 2k(\dot{v} - \hat{v})^{\mathrm{t}} R^{\mathrm{t}} R\dot{u}_1 + k^2\omega(\dot{u}_1) \end{aligned}$$

As

$$(\dot{v} - \hat{v})^{\mathrm{t}} R^{\mathrm{t}} R\dot{u}_1 = [R(\dot{v} - \hat{v})]^{\mathrm{t}} R\dot{u}_1 = w_1 r_{1,1}$$

and

$$\omega(\dot{u}_1) = \dot{u}_1^{\mathrm{t}} R^{\mathrm{t}} R\dot{u}_1 = [R\dot{u}_1]^{\mathrm{t}} R\dot{u}_1 = r_{1,1}^2$$

we therefore have

$$\omega\big((\dot{v} + k\dot{u}_1) - \hat{v}\big) = \omega(\dot{v} - \hat{v}) + 2(kr_{1,1})w_1 + (kr_{1,1})^2$$

i.e.,

$$\omega\big((\dot{v} + k\dot{u}_1) - \hat{v}\big) = \omega(\dot{v} - \hat{v}) + b_k(c + b_k) \tag{C.24}$$

where $b_k \overset{\text{def}}{=} kr_{11}$ and $c \overset{\text{def}}{=} 2w_1$. In our algorithms, this formula is intensively used.

Remark A3.2.1: Search for $\check{v}$ only. When one simply wants to identify the point $\check{v}$ for which the smallest value of $\omega(\dot{v} - \hat{v})$ is attained, the up-and-down algorithm can often be shortened. Indeed, from Eqs. (C.13) and (C.15), we have

$$\begin{aligned} \textstyle\sum_{\ell=p}^{n} w_\ell^2 &= \mathfrak{c} - z_{p-1} \\ &= \mathfrak{c} - (z_p - y_p) \end{aligned} \qquad (2 \le p \le n)$$

Consequently, when for some node $\dot{v}^{(p)}$ (with $p \ge 2$), $[\mathfrak{c} - (z_p - y_p)]$ is greater than (or equal to) the smallest value of $\omega(\dot{v} - \hat{v})$ already attained in the process, the subtree from that node can be ignored (pruning).

To save CPU time, it then also strongly recommended to span each ambiguity interval in an alternate manner around its integer rounded centre. Moreover, from the outset, the value of $\mathfrak{c}$ may be reduced via the 'bootstrapping operation' described in Remark A3.2.2. As it is the case in GNSS (see, e.g., Lannes & Teunissen (2011)), finding $\check{v}$ with $n = 100$ (for example) can then be done in real time.

Remark A3.2.2: Bootstrapping operation. Consider the smallest ellipsoid $\mathfrak{E}(\mathfrak{c})$ including some integer-ambiguity point $\dot{v}$, for instance the zero point of $\mathbb{Z}^n$: $\dot{v}_0$ [Eq. (II.97)]. From Eqs. (C.1), (C.2) and (C.7), we have

$$\mathfrak{c} \overset{\text{set}}{=} \mathfrak{c}(\dot{v}) = \|w_*\|^2_{\mathbb{R}^n} \quad \text{where} \quad w_* \overset{\text{def}}{=} R(\hat{v} - \dot{v}) \tag{C.25}$$

To reduce the size of $\mathfrak{E}(\mathfrak{c})$, we now consider the impact on $\mathfrak{c}$ of an integer variation of the pth component of $\dot{v}$.

Let $\dot{u}_p$ be the integer vector whose components are all zero, except $\dot{u}_p^{(p)}$ which is equal to unity. Denoting by A the column matrix formed by the pth column of R, we have, for any k in $\mathbb{Z}$,

$$\mathfrak{c}(\dot{v} + k\dot{u}_p) = \|w_* - Ak\|^2_{\mathbb{R}^n} \tag{C.26}$$

If k was a real number x, the minimum of $\mathfrak{c}$ in x would be obtained for the solution of the normal equation $A^{\mathrm{t}}Ax = A^{\mathrm{t}}w_*$. Here, this 'float solution' is therefore given by the formula

$$x_\circ = b/a \quad \text{where} \quad a \overset{\text{def}}{=} A^{\mathrm{t}}A, \; b \overset{\text{def}}{=} A^{\mathrm{t}}w_* \tag{C.27}$$

As $w_* - Ak \equiv (w_* - Ax_\circ) - A(k - x_\circ)$, we have

$$\begin{aligned} \|w_* - Ak\|^2_{\mathbb{R}^n} &= \|w_* - Ax_\circ\|^2_{\mathbb{R}^n} + \|A(k - x_\circ)\|^2_{\mathbb{R}^n} \\ &= \|w_* - Ax_\circ\|^2_{\mathbb{R}^n} + a(k - x_\circ)^2 \end{aligned}$$

Indeed, $(w_* - Ax_\circ \mid Ah)_{\mathbb{R}^n} = 0$ for any h in $\mathbb{R}$, and in particular for $h = k - x_\circ$. The minimum of $\mathfrak{c}$ in k is therefore obtained for the integer $k_\circ$ closest to $x_\circ$:

$$k_\circ = \lfloor b/a \rceil \tag{C.28}$$

As $\|w_* - Ak\|^2_{\mathbb{R}^n} = \|w_*\|^2_{\mathbb{R}^n} + \|Ak\|^2_{\mathbb{R}^n} - 2(w_* \mid Ak)_{\mathbb{R}^n}$, we have [see Eqs. (C.26) and (C.25)]

$$\mathfrak{c}(\dot{v} + k_\circ\dot{u}_p) = \mathfrak{c}(\dot{v}) + \mathfrak{d}$$

where

$$\begin{aligned} \mathfrak{d} &= \|Ak_\circ\|^2_{\mathbb{R}^n} - 2[A^{\mathrm{t}}w_*]k_\circ \\ &= ak_\circ^2 - 2bk_\circ \\ &= k_\circ(ak_\circ - 2b) \end{aligned} \tag{C.29}$$

When $k_\circ$ is different from zero [see Eq. (C.28)], $\mathfrak{d}$ is negative. We then successively set

$$\left| \begin{array}{l} \mathfrak{c} \overset{\text{set}}{=} \mathfrak{c} + \mathfrak{d} \\ \dot{v} \overset{\text{set}}{=} \dot{v} + k_\circ\dot{u}_p \\ w_* \overset{\text{set}}{=} w_* - Ak_\circ \end{array} \right. \tag{C.30}$$

We have thus found a way of reducing the size of $\mathfrak{E}(\mathfrak{c})$ in an elementary manner. The bootstrapping operation is based on this principle. One starts with $\mathfrak{c} \overset{\text{set}}{=} \mathfrak{c}_0 \equiv \mathfrak{c}(\dot{v}_0)$ [for which $w_* = R\hat{v}$ from Eq. (C.25)], the operations (C.30) being performed for $p = n, n-1, \ldots, 1$ in a cyclical manner, until $\mathfrak{c}$ does not change all over a cycle. In the context of Remark A3.2.1, the discrete search for $\check{v}$ is restricted to the ellipsoid thus obtained.

Appendix D

Trust-region method

For the functional f of the problem considered in Sect. II.5, the principle the trust-region method can be presented as follows (for further details, see Moré (1983)).

Let us consider the following quadratic expansion of f at some pupil-phase point α [see Eqs. (II.102) and (II.103)]:

$$q(h) \overset{\text{def}}{=} c + (b \mid h)_{V_b} + \frac{1}{2}(h \mid Ah)_{V_b} \tag{D.1}$$

where

$$c \overset{\text{def}}{=} f(\alpha), \quad b \overset{\text{def}}{=} f'(\alpha), \quad A \overset{\text{def}}{=} f''(\alpha) \tag{D.2}$$

According to these definitions, c is the value of the chord functional at α, b is its gradient, and A is its Hessian [see Eq. (II.104)]; $(\cdot \mid \cdot)_{V_b}$ is the inner product on the Euclidean space V_b [see Eq. (II.25)]. In this appendix, as no confusion may arise, subscript V_b is now omitted.

The iteration of the trust-region method is of the traditional form $\alpha \overset{\text{set}}{=} \alpha + h$. The originality of this method is to choose h, at each iteration, so that the smallest value of q is attained in the largest possible ball centred on the origin; the trust ball is the corresponding ball centred on α. The degree of confidence in the approximation $f(\alpha + h) \simeq q(h)$, and thereby the size of the trust ball, is controlled throughout the process by means of the Armijo test; see Sect. A4.2. One thus goes down 'continuously' to a local minimum. We now clarify these points.

D.1 Local quadratic minimization

Given $\varrho \geq 0$, let us consider the functional

$$q_\varrho(h) \overset{\text{def}}{=} q(h) + \frac{1}{2}\varrho\|h\|^2 \tag{D.3}$$

Note that

$$q_\varrho(h) = c + (b \mid h) + \frac{1}{2}(h \mid A_\varrho h) \tag{D.4}$$

where

$$A_\varrho \overset{\text{def}}{=} A + \varrho I \tag{D.5}$$

Here, I is the identity operator (on V_b). The gradient and the Hessian of q_ϱ at h are equal to $b + A_\varrho h$ and A_ϱ, respectively.

Let $\lambda_{\min}$ be the smallest eigenvalue of A. Clearly [see Eq. (D.5)], the eigenvalues of A_ϱ are equal to those of A translated by ϱ. Subject to the condition

$$\varrho \geq \varrho_0 \stackrel{\text{def}}{=} \max(-\lambda_{\min} + \epsilon, 0) \tag{D.6}$$

where ϵ is a positive number arbitrarily small, A_ϱ is therefore positive definite, and thereby invertible. The minimum of q_ϱ is then attained at the point h_ϱ such that $b + A_\varrho h_\varrho = 0$; h_ϱ is therefore the solution of the equation

$$A_\varrho h_\varrho = -b \tag{D.7}$$

Note that the conjugate-gradient method is well suited for solving this equation.

D.2 Trust ball

For any $h \neq h_\varrho$, we have (for any $\varrho \geq \varrho_0$)

$$q(h) + \frac{1}{2}\varrho\|h\|^2 > q(h_\varrho) + \frac{1}{2}\varrho\|h_\varrho\|^2$$

hence: $q(h) > q(h_\varrho) + \frac{1}{2}\varrho(\|h_\varrho\|^2 - \|h\|^2)$. This relation shows that in the ball $\mathfrak{B}_\varrho$ centred on the origin with radius $\|h_\varrho\|$, the smallest value of q is attained for $h = h_\varrho$. More precisely, for any h lying in $\mathfrak{B}_\varrho$, and different from h_ϱ, we have $q(h) > q(h_\varrho)$. From Eqs. (D.7) and (D.5), the size of $\mathfrak{B}_\varrho$ is a decreasing function of ϱ. This size is controlled via the Armijo test.

Armijo Test. Given some positive number < 1 (of the order of 0.3), this test consists in verifying whether the following inequality is satisfied or not:

$$f(\alpha) - f(\alpha + h_\varrho) > [q(0) - q(h_\varrho)] \tag{D.8}$$

Note that $q(0) = f(\alpha)$; see Eqs. (D.1) and (D.2). Clearly,

$$q(0) - q(h_\varrho) = -(b \mid h_\varrho) - \frac{1}{2}(h_\varrho \mid Ah_\varrho)$$

As $(A + \varrho I)h_\varrho = -b$ [see Eqs. (D.7) and (D.5)], we therefore have

$$q(0) - q(h_\varrho) = \frac{1}{2}\big[\varrho\|h_\varrho\|^2 - (b \mid h_\varrho)\big] \tag{D.9}$$

If the Armijo test is satisfied, h_ϱ is a possible iteration step; otherwise, $\|h_\varrho\|$ is too large; ϱ must then be increased. The trust ball can therefore be determined by a search algorithm in ϱ based on the Armijo test. In principle, the initial value of ϱ must be chosen as small as possible.

D.3 Special case where A is positive definite

For the chord functional of Sect. II.5, the Hessian is given by the formula [see Eq. (E.6) in A5]

$$A = 2B^*[\cos(\widehat{\varphi}_c - B\alpha)]B$$

As explicitly shown in Sect. A5.2, the condition

$$|\arc(\widehat{\varphi}_c - B\alpha)| < \pi/2 \qquad \text{(all over } \mathscr{E}\text{)}$$

is a sufficient condition for A to be positive definite, and thereby invertible. When this condition is satisfied, the initial value of ϱ can be set equal to zero. In fact, for that functional, the Armijo test proves then to be satisfied. As a result, according to Eqs. (D.7), (D.5) and (D.2), the trust-region iteration then reduces to the Newton-Raphson iteration: h is the solution of the equation

$$[f''(\alpha)]h = -f'(\alpha) \tag{D.10}$$

D.4 General case

When the condition $|\arc(\widehat{\varphi}_c - B\alpha)| < \pi/2$ is not satisfied on some baseline(s), the smallest eigenvalue of A is to be computed. For this purpose, we first introduce the following definition: the dominant eigenvalue of A is the eigenvalue of A that is the largest in absolute value. We can then proceed as follows.

Firstly, the following 'power iteration' provides the dominant eigenvalue λ of A (and an eigenvector v such that $Av = \lambda v$):

Step 0. Choose some vector u (in V_b); u must not be an eigenvector of A. For example, for our Hessian, set all the components $u^{(i)}$ of u equal to unity.

Step 1. Compute $v \overset{\text{set}}{=} Au$; then, set λ equal to the dominant component of v, i.e., the largest component of v in absolute value. Then, normalize v as follows: $v \overset{\text{set}}{=} v/\lambda$.

Step 2. If $\max|v^{(i)} - u^{(i)}| < 10^{-5}$ (for example), terminate the process; otherwise, set $u = v$, and go to *Step 1*.

When the eigenvalue λ thus found is negative, λ is the smallest eigenvalue of A: $\lambda_{\min} = \lambda$.

When this is not the case, i.e, when λ is positive, $\lambda_{\min}$ is computed via the dominant eigenvalue λ' of the operator $A' \overset{\text{def}}{=} I - A/\lambda$. Indeed, we then have $\lambda' = 1 - \lambda_{\min}/\lambda$, hence $\lambda_{\min} = \lambda(1 - \lambda')$. The power iteration that yields λ' is of course similar to that used for obtaining λ.

Appendix E

Gradient and Hessian of the chord functional

In this appendix, we give explicit expressions for the gradient and the Hessian of the chord functional (II.101):

$$f\colon V_b \to \mathbb{R}, \qquad f(\alpha) = \|c(\alpha)\|_w^2 \tag{E.1}$$

where

$$c(\alpha) \overset{\text{def}}{=} 2\sin\Big(\frac{\widehat{\varphi}_{\mathrm{c}} - B\alpha}{2}\Big) \tag{E.2}$$

E.1 Gradient

The gradient of f at α is the vector $f'(\alpha)$ of V_b involved in the relation

$$f(\alpha + h) \simeq f(\alpha) + (f'(\alpha) \mid h)_{V_b} \tag{E.3}$$

From Eqs. (E.1) and (E.2), we have

$$\begin{aligned} f(\alpha + h) &\simeq \|c(\alpha) + c'(\alpha)\, h\|_w^2 \\ &\simeq f(\alpha) + 2\big(\, c(\alpha) \mid c'(\alpha)\, h \,\big)_w \end{aligned}$$

where

$$\begin{aligned} c'(\alpha)\, h &\simeq 2\Big[\cos\Big(\frac{\widehat{\varphi}_{\mathrm{c}} - B\alpha}{2}\Big)\Big]\Big[-\frac{1}{2}Bh\Big] \\ &= -\Big[\cos\Big(\frac{\widehat{\varphi}_{\mathrm{c}} - B\alpha}{2}\Big)\Big]Bh \end{aligned}$$

As a result,

$$f(\alpha + h) \simeq f(\alpha) - 2\Big(B^*\Big[\cos\Big(\frac{\widehat{\varphi}_{\mathrm{c}} - B\alpha}{2}\Big)\Big]c(\alpha) \;\Big|\; h\Big)_{V_b}$$

Here, the adjoint of B is the operator from E_w into V_b explicitly defined by Eq. (II.59). From Eq. (E.2), we have

$$\Big[\cos\Big(\frac{\widehat{\varphi}_{\mathrm{c}} - B\alpha}{2}\Big)\Big]\, c(\alpha) = \sin(\widehat{\varphi}_{\mathrm{c}} - B\alpha)$$

It then follows from Eq. (E.3) that the gradient of f at α is given by the formula

$$f'(\alpha) = -2B^* \sin(\widehat{\varphi}_{\mathrm{c}} - B\alpha) \tag{E.4}$$

E.2 Hessian

The Hessian of f at α is the operator $f''(\alpha)$ on V_b involved in the relation

$$f'(\alpha + h) \simeq f'(\alpha) + [f''(\alpha)]h \tag{E.5}$$

From Eq. (E.4), we therefore have

$$[f''(\alpha)]h = 2B^*[\cos(\widehat{\varphi}_{\mathrm{c}} - B\alpha)]\, Bh$$

hence

$$f''(\alpha) = 2B^*[\cos(\widehat{\varphi}_{\mathrm{c}} - B\alpha)]B \tag{E.6}$$

We now introduce a sufficient condition for $f''(\alpha)$ to be positive definite and thereby invertible. For clarity, let us set $\varepsilon_\alpha \overset{\text{def}}{=} \operatorname{arc}(\widehat{\varphi}_{\mathrm{c}} - B\alpha)$ [Eq. (II.81)]. When

$$|\varepsilon_\alpha| < \pi/2 \qquad \text{(all over } \mathscr{E}\text{)} \tag{E.7}$$

we have $f''(\alpha) = 2\big[\sqrt{\cos\varepsilon_\alpha}\, B\big]^* \big[\sqrt{\cos\varepsilon_\alpha}\, B\big]$, hence, for any h in V_b,

$$\begin{aligned}\big(h \,\big|\, [f''(\alpha)]\, h\big)_{V_b} &= 2\big(\sqrt{\cos\varepsilon_\alpha}\, Bh \,\big|\, \sqrt{\cos\varepsilon_\alpha}\, Bh\big)_w \\ &= 2\big\|\sqrt{\cos\varepsilon_\alpha}\, Bh\big\|_w^2\end{aligned}$$

The condition $\big(h \,\big|\, [f''(\dot{\alpha})]\, h\big)_{V_b} = 0$ then implies that $h = 0$; indeed, the null space of B is reduced to $\{0\}$ [Eq. (II.34)]. Condition (E.7) is therefore a sufficient condition for $f''(\alpha)$ to be positive definite.

Appendix F

Proof of Property RSR

The proof of Property REDUCESWAPRESTORE can be obtained as follows.

From Eqs. (III.41) and (III.36), we have

$$(U_j M_j)^{\mathrm{T}} D_j \, (U_j M_j) = \begin{bmatrix} \breve{u} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} d_{j-1} & 0 \\ 0 & d_j \end{bmatrix} \begin{bmatrix} \breve{u} & 1 \\ 1 & 0 \end{bmatrix}$$

i.e., explicitly,

$$(U_j M_j)^{\mathrm{T}} D_j \, (U_j M_j) = \begin{bmatrix} \bar{d}_{j-1} & d_{j-1}\breve{u} \\ d_{j-1}\breve{u} & d_{j-1} \end{bmatrix}$$

Let us now factorize this matrix in the form

$$\begin{aligned} \mathrm{U}^{\mathrm{T}}\mathrm{D}\mathrm{U} &= \begin{bmatrix} 1 & 0 \\ \mathfrak{u} & 1 \end{bmatrix} \begin{bmatrix} c_{j-1} & 0 \\ 0 & c_j \end{bmatrix} \begin{bmatrix} 1 & \mathfrak{u} \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c_{j-1} & c_{j-1}\mathfrak{u} \\ c_{j-1}\mathfrak{u} & c_j + c_{j-1}\mathfrak{u}^2 \end{bmatrix} \end{aligned}$$

By identifying the corresponding terms, we have

$$c_{j-1} = \bar{d}_{j-1} \qquad c_{j-1}\mathfrak{u} = d_{j-1}\breve{u} \qquad c_j + c_{j-1}\mathfrak{u}^2 = d_{j-1}$$

As a result, $\mathfrak{u} = \bar{u}$ and $c_j + \bar{d}_{j-1}\mathfrak{u}^2 = d_{j-1}$, hence

$$\begin{aligned} c_j &= d_{j-1} - \bar{d}_{j-1}\breve{u}^2 \frac{d_{j-1}^2}{\bar{d}_{j-1}^2} \\ &= d_{j-1}\left(1 - \breve{u}^2 \frac{d_{j-1}}{\bar{d}_{j-1}}\right) \\ &= \frac{d_{j-1}}{\bar{d}_{j-1}}(\bar{d}_{j-1} - \breve{u}^2 d_{j-1}) \\ &= \frac{d_{j-1}}{\bar{d}_{j-1}}\, d_j \\ &= \bar{d}_j \end{aligned}$$

Consequently, $(U_j M_j)^{\mathrm{T}} D_j \, (U_j M_j) = \bar{U}_j^{\mathrm{T}} \bar{D}_j \bar{U}_j$.

The corollary results from the fact that (see Eq. (III.41))

$$\bar{U}_j (U_j M_j)^{-1} = \begin{bmatrix} 1 & \bar{u} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & -\breve{u} \end{bmatrix} = \begin{bmatrix} \bar{u} & 1 - \breve{u}\bar{u} \\ 1 & -\breve{u} \end{bmatrix}$$

i.e., $\bar{U}_j (U_j M_j)^{-1} = G_j$, hence $G_j U_j M_j = \bar{U}_j$. We then have

$$\begin{aligned} & (U_j M_j)^{\mathrm{T}} D_j \, (U_j M_j) \\ & = (G_j U_j M_j)^{\mathrm{T}} \, (G_j^{-1})^{\mathrm{T}} D_j G_j^{-1} \, (G_j U_j M_j) \\ & = \bar{U}_j^{\mathrm{T}} \bar{D}_j \bar{U}_j \end{aligned}$$

hence $(G_j^{-1})^{\mathrm{T}} D_j G_j^{-1} = \bar{D}_j$ □

Appendix G

Elementary notions on GNSS graphs

In this appendix, we present some preliminary notions of algebraic graph theory; these elementary notions are used throughout the paper. Further details about the functional spaces and the operators involved in the GNSS problems can be found in Sect. IV.3. We first define the notions of GNSS grid and GNSS graph (Sect. G.1). We then introduce the concepts of spanning tree and loop (Sect. G.2).

G.1 GNSS grid and graph

For our present purpose (see Sects. IV.2.1 and IV.2.2 in particular), we consider a typical situation in which the network has n_r receivers r_i and n_s satellites s_j. (We recall that 'satellite' should be here understood as 'satellite transmitter;' see Sect. IV.1.) The 'network grid' G_o then includes n_r rows, n_s columns, and $n_\mathrm{r} n_\mathrm{s}$ points; see Fig. G.1. A function such as $N(i,j)$ for example takes its values on some points (i,j) of that grid. Those points form a subgrid denoted by G.

In the example presented in Fig. G.1, the points (i,j) of G are displayed as large dots in the upper part of this figure. Those points correspond to the 'edges' $(\mathrm{r}_i, \mathrm{s}_j)$ of the graph associated with the GNSS network; this graph is displayed in the lower part of Fig. G.1; $\mathscr{E}$ denotes the set of its edges; n_e is their number. The receivers and the satellites involved in the definition of these edges define the 'vertices' of this graph; $\mathscr{V}$ denotes the set of its vertices, and n_v their number: $n_\mathrm{v} = n_\mathrm{r} + n_\mathrm{s}$. A graph such as $\mathscr{G}$ is therefore defined by the pair $(\mathscr{V}, \mathscr{E})$: $\mathscr{G} \equiv \mathscr{G}(\mathscr{V}, \mathscr{E})$. We now assume that $\mathscr{G}$ is connected: given any two vertices of $\mathscr{V}$, there exists a path of edges of $\mathscr{E}$ connecting these vertices; see, e.g., Biggs (1996).

G.2 GNSS spanning tree and loops

As illustrated in Fig. G.2, a spanning tree of $\mathscr{G} \equiv \mathscr{G}(\mathscr{V}, \mathscr{E})$ is a subgraph $\mathscr{G}_\mathrm{st} \equiv \mathscr{G}_\mathrm{st}(\mathscr{V}, \mathscr{E}_\mathrm{st})$ formed by n_v vertices and $n_\mathrm{v} - 1$ edges, with no 'cycle' in it. Here, 'cycle' is used in the sense defined in algebraic graph theory; see, e.g., Biggs (1996). In the GNSS community, to avoid any confusion with the usual notion of wave cycle, the term of 'loop' can be substituted for that of 'cycle.' In this context, the number of loops defined through a given fixed (but arbitrary) spanning tree is the number of edges of $\mathscr{E}$ that do not lie in $\mathscr{E}_\mathrm{st}$. These edges,

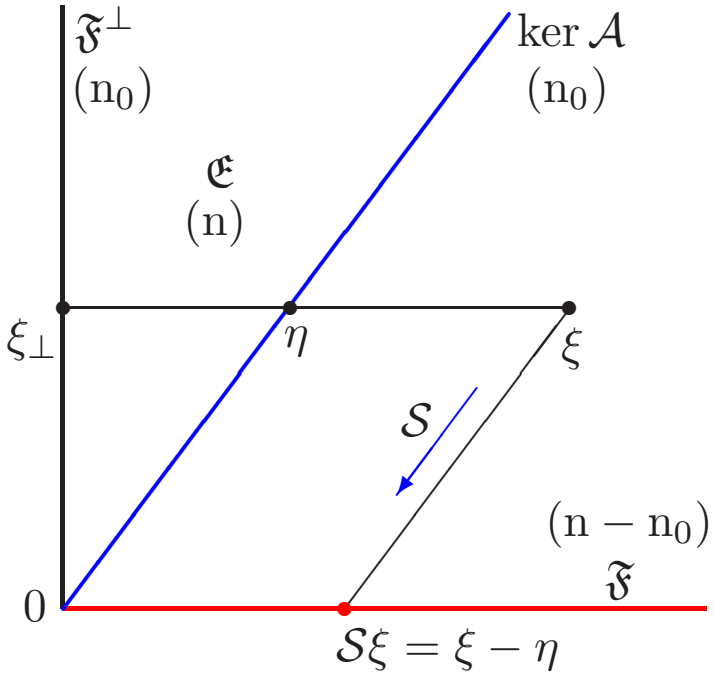

Figure G.1: Subgrid G and graph $\mathscr{G}$. In the example described here, the network grid G_{o} includes twelve points ($n_{\mathrm{r}} = 3$, $n_{\mathrm{s}} = 4$), while its subgrid G includes nine points only; these points are shown as large dots. The corresponding graph $\mathscr{G}$ includes seven vertices and nine edges: $n_{\mathrm{v}} = n_{\mathrm{r}} + n_{\mathrm{s}} = 7$, $n_{\mathrm{e}} = 9$; r_1 does not see s_2, r_2 does not see s_3, and r_3 does not see s_1.

$\mathrm{c}(\ell) \stackrel{\mathrm{def}}{=} (\mathrm{r}_{i_\ell}, \mathrm{s}_{j_\ell})$, are said to be '(loop-)closure edges' (see Fig. G.2). Their number is denoted by n_{c}:

$$n_{\mathrm{c}} = n_{\mathrm{e}} - n_{\mathrm{st}} \tag{G.1}$$

where

$$n_{\mathrm{st}} \stackrel{\mathrm{def}}{=} n_{\mathrm{v}} - 1 = n_{\mathrm{r}} + n_{\mathrm{s}} - 1 \tag{G.2}$$

Many spanning trees of the same graph can be constructed. Here, we are going to present the Kruskal algorithm which is often used in algebraic graph theory; see Biggs (1996). The first step of this algorithm consists in ordering the edges of $\mathscr{E}$, thus generating a sequence of the form $\{(\mathrm{r}_{i_q}, \mathrm{s}_{j_q}) : \ q = 1, \ldots, n_{\mathrm{e}}\}$. The spanning tree is then obtained as follows. Set $q = 0$, $n_{\mathrm{st}} = 0$, and $\mathscr{E}_{\mathrm{st}} = \emptyset$ (the empty set). Then,

(i) if $n_{\mathrm{st}} = n_{\mathrm{v}} - 1$, terminate the process; otherwise, set $q := q + 1$;

(ii) when the vertices of edge $(\mathrm{r}_{i_q}, \mathrm{s}_{j_q})$ are not connected via edges of $\mathscr{E}_{\mathrm{st}}$, set $\mathscr{E}_{\mathrm{st}} := \mathscr{E}_{\mathrm{st}} \cup \{(\mathrm{r}_{i_q}, \mathrm{s}_{j_q})\}$, $n_{\mathrm{st}} := n_{\mathrm{st}} + 1$; then go to step (i).

By construction, the spanning tree thus found depends on how the edges are ordered in the first step. The subgrid of G corresponding to the edges of $\mathscr{G}_{\mathrm{st}}$ is denoted by G_{st}; G_{c} is that corresponding to the closure edges:

$$\mathrm{G}_{\mathrm{c}} \stackrel{\mathrm{def}}{=} \{(i,j) \in \mathrm{G} : (i,j) \notin \mathrm{G}_{\mathrm{st}}\} \tag{G.3}$$

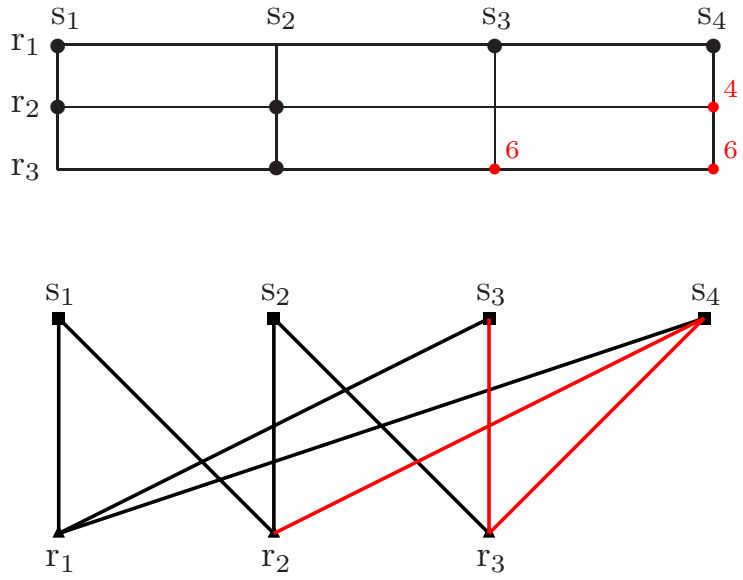

Figure G.2: GNSS spanning tree and loops. Here, the edges of the selected spanning tree $\mathscr{G}_{\rm st}$ of the graph $\mathscr{G}$ introduced in Fig. G.1 are shown as thick lines. The points of the corresponding subgrid $\mathrm{G}_{\rm st}$ are shown as large dots. The remaining points of G (the small dots of G) correspond to the (loop-)closure edges (the thin edges of $\mathscr{G}$). We then have one loop of order four, and 2 loops of order six: $(\mathrm{r}_2,\, \mathrm{s}_4,\, \mathrm{r}_1,\, \mathrm{s}_1)$, $(\mathrm{r}_3,\, \mathrm{s}_3,\, \mathrm{r}_1,\, \mathrm{s}_1,\, \mathrm{r}_2,\, \mathrm{s}_2)$ and $(\mathrm{r}_3,\, \mathrm{s}_4,\, \mathrm{r}_1,\, \mathrm{s}_1,\, \mathrm{r}_2,\, \mathrm{s}_2)$. In G, these orders are shown as small numbers.

Clearly, $\mathrm{G}_{\rm c}$ includes $n_{\rm c}$ loop-closure points; see Eq. (G.1) and Fig. G.2.

To illustrate the action of the Kruskal algorithm, let us consider the graph $\mathscr{G}$ of Fig. G.1. To build a spanning tree of $\mathscr{G}$ from its grid G, let us order the edges of $\mathscr{G}$ by scanning G from left to right and top to bottom. The algorithm examines the edges of G in that order and adds them to the current version of $\mathscr{E}_{\rm st}$ when condition (ii) holds. In this example, this is the case for the first five edges; the vertices $\mathrm{s}_1,\, \mathrm{s}_3,\, \mathrm{s}_4,\, \mathrm{r}_1,\, \mathrm{r}_2$ are thus connected. The sixth edge, $(\mathrm{r}_2,\, \mathrm{s}_4)$, therefore includes two vertices already connected. This edge is therefore the first closure edge: $\text{ς}(1) = (\mathrm{r}_2,\, \mathrm{s}_4)$. The next edge, $(\mathrm{r}_3, \mathrm{s}_2)$, is added to $\mathscr{E}_{\rm st}$ since it corresponds to the first connection of s_2 with the edges of the current version of $\mathscr{E}_{\rm st}$. All the vertices of $\mathscr{G}$ are then connected. The remaining edges are therefore closure edges: $\text{ς}(2) = (\mathrm{r}_3,\, \mathrm{s}_3)$, $\text{ς}(3) = (\mathrm{r}_3,\, \mathrm{s}_4)$. The $\mathscr{G}_{\rm st}$-edge set thus obtained is the following (see Fig. G.2):

$$\mathscr{E}_{\rm st} := \{(\mathrm{r}_1, \mathrm{s}_1), (\mathrm{r}_1, \mathrm{s}_3), (\mathrm{r}_1, \mathrm{s}_4), (\mathrm{r}_2, \mathrm{s}_1), (\mathrm{r}_2, \mathrm{s}_2), (\mathrm{r}_3, \mathrm{s}_2)\}$$

Note that this procedure does not provide the edge path of $\mathscr{E}_{\rm st}$ that links the vertices of the closure edge under consideration. Clearly, closure paths are not needed to be known for the construction of $\mathscr{G}_{\rm st}$. In simple cases such as that of Fig. G.2, such a path can visually be obtained by moving on grid G horizontally and vertically, in alternate manner from the selected closure-edge point; see the related telescoping sums introduced in Sect. IV.3.3. If need be, the edges paths can be obtained automatically in an algebraic manner; see Sect. IV.3.4.

Appendix H

The S-system approach

In this appendix, we give a survey of the general framework of the S-system approach; for further details and related applications, see Baarda (1973); Teunissen (1984); de Jonge (1998); Teunissen & Odijk (2003).

Denoting by $\mathfrak{E}$ a Euclidean space of dimension n, we consider some linear operator $\mathcal{A}$ from $\mathfrak{E}$ into $\mathbb{R}^{\mathrm{m}}$ with $\mathrm{m} \geq \mathrm{n}$. The problem to be solved in a sense to be defined is governed by a relation of the form

$$\mathcal{A}\xi = \beta \tag{H.1}$$

The components of ξ are the unknown parameters of the problem, whereas β is the data vector. In many situations encountered in practice, $\mathcal{A}$ is not of full rank; its null space (i.e., its kernel) is not reduced to $\{0\}$:

$$\mathrm{n}_0 \stackrel{\text{def}}{=} \dim(\ker \mathcal{A}) \geq 1 \tag{H.2}$$

In the S-system approach, this rank defect is removed via an appropriate reduction and redefinition of the unknown parameters. Those new parameters are the 'estimable functions of parameters' of some minimum-constrained problem thus defined; see, e.g., de Jonge (1998). We now give a geometrical interpretation of the S-system principle.

Let us choose some subspace $\mathfrak{F}$ of $\mathfrak{E}$ of dimension $\mathrm{n} - \mathrm{n}_0$ such that $\mathfrak{F} \cap \ker \mathcal{A} = \{0\}$; $\mathfrak{E}$ can then be regarded as the direct sum of $\mathfrak{F}$ and $\ker \mathcal{A}$ (see Fig. H.1):

$$\mathfrak{E} = \mathfrak{F} \oplus \ker \mathcal{A} \tag{H.3}$$

The 'estimable functional variable' is then defined as the oblique projection of ξ on $\mathfrak{F}$ along $\ker \mathcal{A}$: $\S\xi$. The oblique projection (operator) $\S$ is the $\S$-transformation of the S-system method; see, e.g., de Jonge (1998).

We now show how the S-system approach can provide the matrix of $\S$ in the standard basis of $\mathfrak{E}$. The estimable functional variable $\S\xi$, which basically depends on the choice of $\mathfrak{F}$, can thus be explicitly defined.

According to its definition,

$$\S\xi = \xi - \eta \tag{H.4}$$

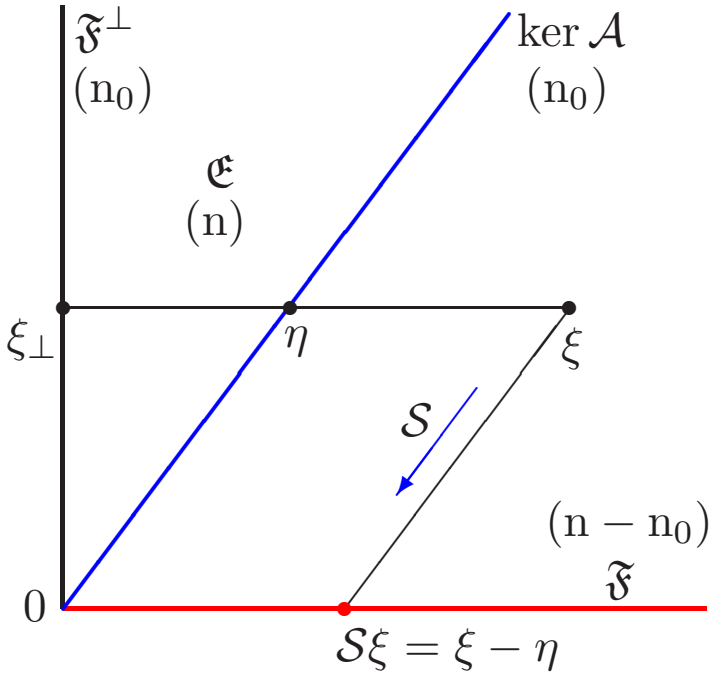

Figure H.1: S-system principle. In this geometrical representation, $\mathfrak{E}$ is a Euclidean space of dimension n. The unknown functional variable ξ is a vector of $\mathfrak{E}$. The null space of the operator $\mathcal{A}$ involved in Eq. (H.1) is of dimension n_0: $\dim(\ker \mathcal{A}) = \mathrm{n}_0$; $\mathfrak{F}$ is a subspace of $\mathfrak{E}$ of dimension $\mathrm{n} - \mathrm{n}_0$ such that $\mathfrak{F} \cap \ker \mathcal{A} = \{0\}$; $\mathfrak{E}$ can then be regarded as the direct sum $\mathfrak{F} \oplus \ker \mathcal{A}$. In the S-system approach, the 'estimable functional variable' is then defined as the oblique projection of ξ on $\mathfrak{F}$ along $\ker \mathcal{A}$: $\S\xi$.

where η is the vector of $\ker \mathcal{A}$ such that $\xi - \eta$ lies in $\mathfrak{F}$; see Fig. H.1. Denoting by $[W]$ a matrix whose column vectors form a basis of $\ker \mathcal{A}$, we have

$$[\eta] = [W][\zeta] \tag{H.5}$$

where ζ is some vector of $\mathbb{R}^{\mathrm{n}_0}$. Clearly, the entries of $[\eta]$, $[W]$ and $[\zeta]$ are expressed in the standard basis of $\mathfrak{E}$. Let $[S_\perp]$ now be a matrix whose column vectors form a basis of $\mathfrak{F}^\perp$, the orthogonal complement of $\mathfrak{F}$ in $\mathfrak{E}$. As $\xi - \eta$ is orthogonal to all the vectors of $\mathfrak{F}^\perp$, we have (in particular) $[S_\perp]^{\mathrm{T}}\big([\xi] - [W][\zeta]\big) = 0$, i.e.,

$$[S_\perp]^{\mathrm{T}}[W][\zeta] = [S_\perp]^{\mathrm{T}}[\xi] \tag{H.6}$$

As shown further on, $[S_\perp]^{\mathrm{T}}[W]$ is invertible. It then follows that $[\zeta] = \big([S_\perp]^{\mathrm{T}}[W]\big)^{-1}[S_\perp]^{\mathrm{T}}[\xi]$, hence from Eqs. (H.4) and (H.5),

$$[\S] = [I] - [W]\left([S_\perp]^{\mathrm{T}}[W]\right)^{-1}[S_\perp]^{\mathrm{T}} \tag{H.7}$$

where $[I]$ is the identity matrix on $\mathfrak{E}$.

We now show that the n_0-by-n_0 matrix $[M_0] \stackrel{\text{def}}{=} [S_\perp]^{\mathrm{T}}[W]$ is invertible.

Proof. Let $\xi_\perp$ be the projection of some vector η of $\ker \mathcal{A}$ on $\mathfrak{F}^\perp$; see Fig. H.1. By considering the case where $\xi = \xi_\perp$, Eq. (H.6) yields $[M_0][\zeta] = [S_\perp]^{\mathrm{T}}[\xi_\perp]$. The condition $[M_0][\zeta] = 0$ implies $[S_\perp]^{\mathrm{T}}[\xi_\perp] = 0$, hence $\xi_\perp = 0$. As a result, η then lies in $\mathfrak{F}$. As $\mathfrak{F} \cap \ker \mathcal{A} = \{0\}$, it follows that $\eta = 0$, hence $\zeta = 0$. The null space of $[M_0]$ is therefore reduced to $\{0\}$; but $[M_0]$ is an n_0-by-n_0 matrix; $[M_0]$ is therefore invertible. □

Remark.0.1: In the S-system approach as it is implemented by de Jonge (1998) for example, one chooses some basis for $\mathfrak{F}$. The corresponding matrix is denoted by S; $\mathfrak{F}$ is then regarded as the range of S; $[S_\perp]^{\mathrm{T}}\xi$ is called the 'S-basis.' Note that $[S_\perp]$ is then denoted by $S^\perp$ ⊡

Appendix I

QR implementation

In Sect. I.1, we first recall the notion of QR factorization. In Sect. I.2, we then show how the float version of Eq. (IV.82) can be solved in the LS sense recursively. In the closure-ambiguity approach, the number of entries of $\boldsymbol{v}$, n_{c}, is a non-decreasing function of k; see Sect. IV.6.3. In Sect. I.2, we consider the case where n_{c} is constant; the cases where at some epochs k, n_{c} increases is dealt with in Sect. I.2.1. In that QR framework, we finally describe (in Sect. I.3) the construction of the matrices $\boldsymbol{A}_k$, $\boldsymbol{B}_k$ and $\boldsymbol{b}_k$ involved in Eq. (IV.82).

I.1 QR factorization

Let us consider the following LS float problem. Given some matrix $\mathrm{A} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$ of full rank n $(\leq \mathrm{m})$, find

$$\hat{x} \stackrel{\text{def}}{=} \underset{x\in\mathbb{R}^{\mathrm{n}}}{\operatorname{argmin}} \|\mathrm{A}x - y\|^2_{\mathbb{R}^{\mathrm{m}}}$$

where $\|\cdot\|_{\mathbb{R}^{\mathrm{m}}}$ is the Euclidean norm of $\mathbb{R}^{\mathrm{m}}$. With regard to numerical accuracy, the best way of solving this problem is to use a method based on the QR factorization of A (see, e.g., Björck (1996)):

$$\mathrm{A} = \mathrm{Q}\begin{bmatrix}\mathrm{R}\\ 0\end{bmatrix} \tag{I.1}$$

where $\mathrm{R} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ is an upper-triangular matrix with positive diagonal terms, and $\mathrm{Q} \in \mathbb{R}^{\mathrm{m}\times\mathrm{m}}$ is an orthogonal matrix: $\mathrm{Q}^{\mathrm{T}}\mathrm{Q} = \mathrm{I}_{\mathrm{m}}$ (the identity matrix on $\mathbb{R}^{\mathrm{m}}$). We thus have

$$\begin{aligned}\|\mathrm{A}x - y\|^2_{\mathbb{R}^{\mathrm{m}}} &= \|\mathrm{Q}^{\mathrm{T}}(\mathrm{A}x - y)\|^2_{\mathbb{R}^{\mathrm{m}}}\\ &= \left\|\mathrm{Q}^{\mathrm{T}}\mathrm{Q}\begin{bmatrix}\mathrm{R}\\ 0\end{bmatrix}x - \mathrm{Q}^{\mathrm{T}}y\right\|^2_{\mathbb{R}^{\mathrm{m}}}\end{aligned}$$

Setting $\mathrm{Q}^{\mathrm{T}}y = z + z'$ where z and z' lie in $\mathbb{R}^{\mathrm{n}}$ and $\mathbb{R}^{\mathrm{m}-\mathrm{n}}$ respectively (see Fig. I.1), it follows that

$$\|\mathrm{A}x - y\|^2_{\mathbb{R}^{\mathrm{m}}} = \|\mathrm{R}x - z\|^2_{\mathbb{R}^{\mathrm{n}}} + \|z'\|^2_{\mathbb{R}^{\mathrm{m}-\mathrm{n}}}$$

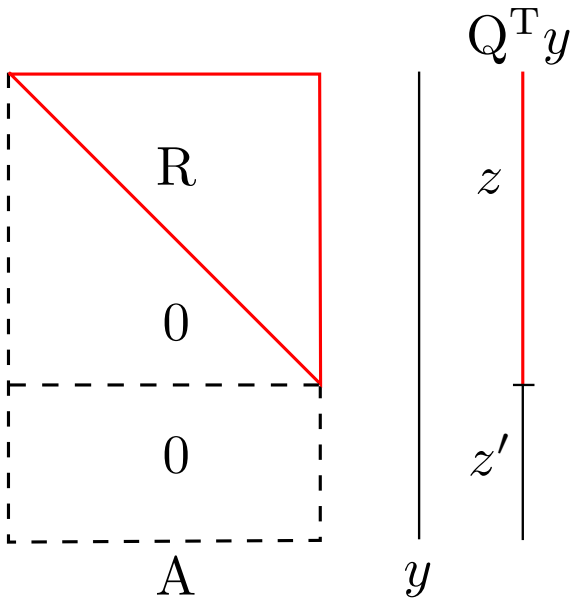

Figure I.1: LS solution via QR factorization. The action of Q^{T} on A and y yields the basic QR structure sketched here: the upper-triangular matrix R and the column matrix z. The solution of the equation $\mathrm{A}x = y$ in the LS sense is then given by Eq. (I.2): $\hat{x} = \mathrm{R}^{-1}z$.

The LS float solution is therefore given by the relation

$$\hat{x} = \mathrm{R}^{-1}z \tag{I.2}$$

As R is an upper-triangular matrix, this solution can be obtained by back substitution.

In the case where x is confined to $\mathbb{Z}^{\mathrm{n}}$, the solution $\check{x}$ of the corresponding nearest-lattice-point (NLP) problem is therefore defined as follows:

$$\check{x} = \underset{x\in\mathbb{Z}^{\mathrm{n}}}{\operatorname{argmin}} \|\mathrm{R}(x-\hat{x})\|^2_{\mathbb{R}^{\mathrm{n}}} \tag{I.3}$$

Indeed, $\mathrm{R}x - z = \mathrm{R}(x-\hat{x})$.

According to Eq. (I.1), QR factorization consists in finding an operator Q^{T} (and thereby an operator Q) such that $\mathrm{Q}^{\mathrm{T}}\mathrm{A}$ has the block structure $[\mathrm{R}\ \ 0]^{\mathrm{T}}$ sketched in Fig. I.1. This operator is defined as a product of elementary orthogonal transformations. In our implementation of the method presented in this paper, the latter are Givens rotations; see Eqs. (2.3.10) to (2.3.13) in Björck (1996)). Premultiplication of A and y by such a rotation matrix affects only row pairs (ℓ_1, ℓ_2) of A and y. Let

$$\mathrm{a} = (\mathrm{a}_{\ell_1}^2 + \mathrm{a}_{\ell_2}^2)^{1/2} \tag{I.4}$$

For $a \neq 0$, this rotation matrix is defined so that

$$\begin{bmatrix} \mathrm{c} & \mathrm{s} \\ -\mathrm{s} & \mathrm{c} \end{bmatrix} \begin{bmatrix} \mathrm{a}_{\ell_1} \\ \mathrm{a}_{\ell_2} \end{bmatrix} = \begin{bmatrix} \mathrm{a} \\ 0 \end{bmatrix} \tag{I.5}$$

It is easy to check that the cosine and sine values c and s are then given by the following formulas

$$\mathrm{c} = \mathrm{a}_{\ell_1}/\mathrm{a} \qquad \mathrm{s} = \mathrm{a}_{\ell_2}/\mathrm{a} \tag{I.6}$$

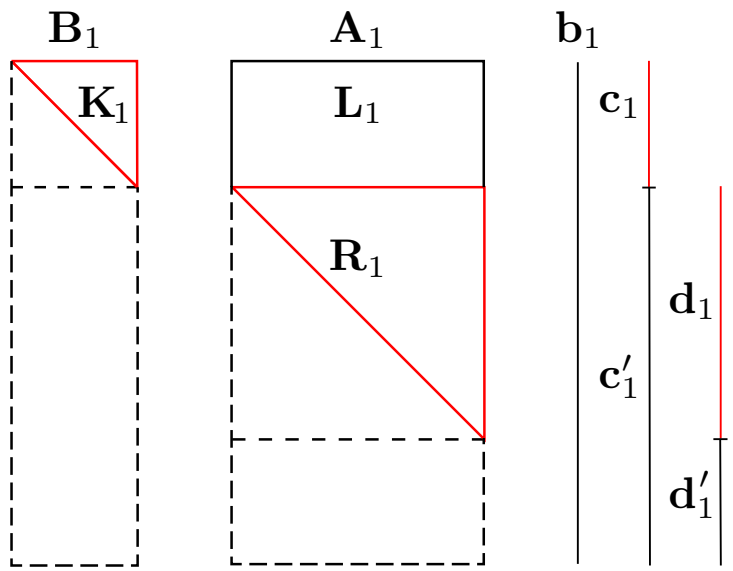

Figure I.2: QR factorization at epoch 1. The principle of the recursive QR method is sketched here for the first epoch with the input block matrices $\boldsymbol{B}_1$, $\boldsymbol{A}_1$ and the data column matrix $\boldsymbol{b}_1$. The initialization process is performed in two steps: $\boldsymbol{K}_1$, $(\boldsymbol{L}_1, \boldsymbol{L}'_1)$, $(\boldsymbol{c}_1, \boldsymbol{c}'_1)$ are built in the first step (see text for $\boldsymbol{L}'_1$), whereas $\boldsymbol{R}_1$, $(\boldsymbol{d}_1, \boldsymbol{d}'_1)$ are built in the second one; for the LS solution thereby obtained at epoch 1, see text.

Note that $\mathrm{m}-1$ Givens rotations are required to fully process the first column of A, $\mathrm{m}-2$ for the second, and so on (see Fig. I.1).

I.2 Recursive QR factorization

As shown in this section, the float version of the following equation (Eq. (IV.82)) can be solved in the LS sense via recursive QR factorization:

$$\begin{bmatrix} \boldsymbol{B}_1 & 0 & 0 & 0 & \boldsymbol{A}_1 \\ 0 & \boldsymbol{B}_2 & 0 & 0 & \boldsymbol{A}_2 \\ 0 & 0 & \ddots & 0 & \vdots \\ 0 & 0 & 0 & \boldsymbol{B}_k & \boldsymbol{A}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_k \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_k \end{bmatrix}$$

Throughout this section, n_c is assumed to be fixed; for related notions, see Sect. 6.3 of Björck (1996), Golub & van Loan (1989), and Bierman (1977).

Initialization: epoch 1

At epoch 1, the problem is to minimize the functional (see the first line of Eq. (IV.82))

$$f_1(\boldsymbol{w}_1, \boldsymbol{v}) \stackrel{\text{def}}{=} \|(\boldsymbol{B}_1\boldsymbol{w}_1 + \boldsymbol{A}_1\boldsymbol{v}) - \boldsymbol{b}_1\|^2_{\mathbb{R}^{n_{\mathrm{e}1}}}$$

The LS solution $(\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{v}})$ is then obtained via two QR factorizations (see Fig. I.2).

1) QR factorization of $\boldsymbol{B}_1$: the Givens rotations of this step are those required for finding the upper-triangular matrix $\boldsymbol{K}_1$; the principle is the same as that described in Sect. I.1 for

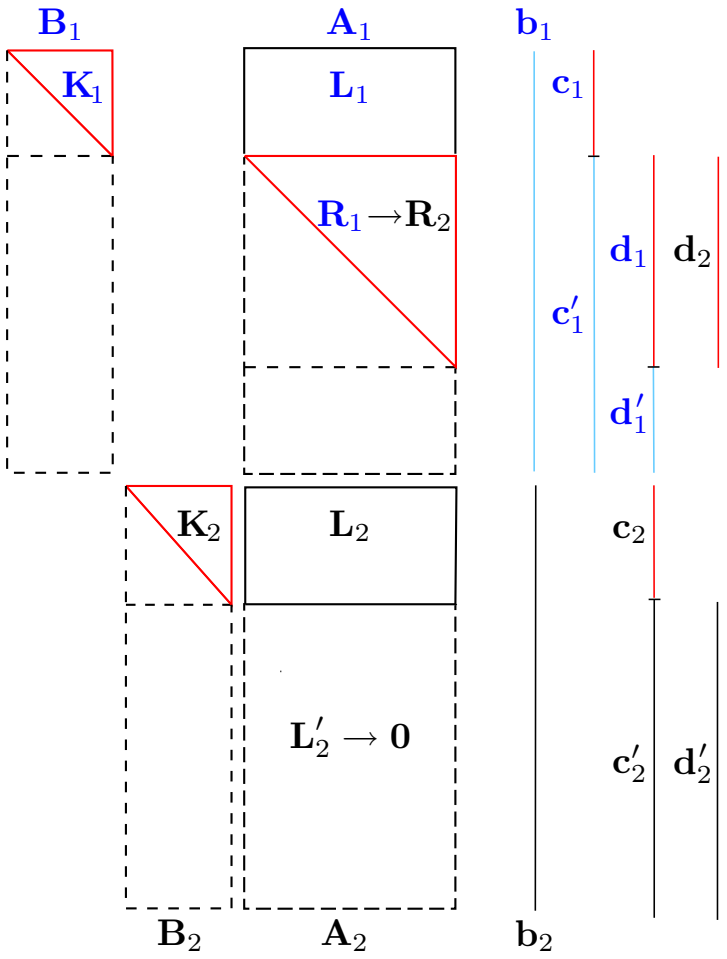

Figure I.3: QR factorization at epochs 1 and 2. The principle of the recursive QR method is sketched here for the first two epochs: epoch 1 with the input block matrices $\boldsymbol{B}_1$, $\boldsymbol{A}_1$ and the data column matrix $\boldsymbol{b}_1$; epoch 2 with the input block matrices $\boldsymbol{B}_2$, $\boldsymbol{A}_2$ and the data column matrix $\boldsymbol{b}_2$. The initialization process is performed in two steps as described in Fig. I.2. At epoch 2, one first builds $\boldsymbol{K}_2$, $(\boldsymbol{L}_2\,,\boldsymbol{L}'_2)$, $(\boldsymbol{c}_2\,,\boldsymbol{c}'_2)$ like at epoch 1, and then $\boldsymbol{R}_2$, $(\boldsymbol{d}_2\,,\boldsymbol{d}'_2)$; for the LS solution thereby obtained at epoch 2, see text.

matrix A. The modified version of $\boldsymbol{A}_1$ thus obtained includes an upper block $\boldsymbol{L}_1$ and a lower block $\boldsymbol{L}'_1$. Likewise, the modified version of $\boldsymbol{b}_1$ includes two column submatrices: $\boldsymbol{c}_1$ and $\boldsymbol{c}'_1$.

2) QR factorization of $\boldsymbol{L}'_1$: the Givens rotations of that step yield the upper-triangular matrix $\boldsymbol{R}_1$. The lower part of $\boldsymbol{L}'_1$ is reduced to 0; $\boldsymbol{c}'_1$ then yields $(\boldsymbol{d}_1\,,\boldsymbol{d}'_1)$; see Fig. I.2. Note that $\boldsymbol{K}_1$, $\boldsymbol{L}_1$ and $\boldsymbol{c}_1$ are not affected by those rotations.

At the end of this initialization stage, we thus have

$$f_1(\boldsymbol{w}_1,\boldsymbol{v}) = \|(\boldsymbol{K}_1\boldsymbol{w}_1 - (\boldsymbol{c}_1 - \boldsymbol{L}_1\boldsymbol{v})\|^2_{\mathbb{R}^{n_{\mathrm{b}1}}} + \|\boldsymbol{R}_1\boldsymbol{v} - \boldsymbol{d}_1\|^2_{\mathbb{R}^{n_{\mathrm{c}}}} + \|\boldsymbol{d}'_1\|^2_{\mathbb{R}^{n_{\mathrm{e}1}-n_{\mathrm{b}1}-n_{\mathrm{c}}}}$$

The float solution in $\boldsymbol{v}$ at epoch 1 is therefore given by the formula $\hat{\boldsymbol{v}} = \boldsymbol{R}_1^{-1}\boldsymbol{d}_1$, hence $\hat{\boldsymbol{w}}_1 = \boldsymbol{K}_1^{-1}(\boldsymbol{c}_1 - \boldsymbol{L}_1\hat{\boldsymbol{v}})$. These solutions can therefore be computed by back substitution. Note that $\|\boldsymbol{d}'_1\|^2_{\mathbb{R}^{n_{\mathrm{e}1}-n_{\mathrm{b}1}-n_{\mathrm{c}}}}$ is the square of the LS residual norm at epoch 1.

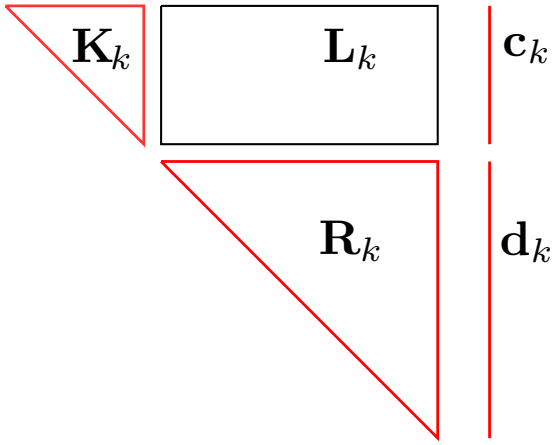

Figure I.4: Recursive QR triangular structure. According to the principle of the recursive QR method sketched in Fig. I.3, the calculation of $\boldsymbol{R}_{k+1}$ and $\boldsymbol{d}_{k+1}$ requires to have kept in memory the upper-triangular matrix $\boldsymbol{R}_k$ and the column matrix $\boldsymbol{d}_k$; see text.

Next epoch: epoch 2

The functional to be minimized is then $f_1(\boldsymbol{w}_1, \boldsymbol{v}) + f_2(\boldsymbol{w}_2, \boldsymbol{v})$ where

$$f_2(\boldsymbol{w}_2, \boldsymbol{v}) \stackrel{\text{def}}{=} \|(\boldsymbol{B}_2\boldsymbol{w}_2 + \boldsymbol{A}_2\boldsymbol{v}) - \boldsymbol{b}_2\|^2_{\mathbb{R}^{n_{\text{e}2}}}$$

As sketched in Fig. I.3, the LS solution $(\hat{\boldsymbol{w}}_1, \hat{\boldsymbol{w}}_2, \hat{\boldsymbol{v}})$ is again obtained via two QR factorizations. The first step of epoch 2 is similar to that of epoch 1; the second one is different.

1) QR factorization of $\boldsymbol{B}_2$

One thus obtains the upper-triangular matrix $\boldsymbol{K}_2$; see Fig. I.3. The modified version of $\boldsymbol{A}_2$ then includes an upper block $\boldsymbol{L}_2$ and a lower block $\boldsymbol{L}'_2$. Likewise, the modified version of $\boldsymbol{b}_2$ includes two column submatrices: $\boldsymbol{c}_2$ and $\boldsymbol{c}'_2$.

2) QR factorization of $\begin{bmatrix} \boldsymbol{R}_1 \\ \boldsymbol{L}'_2 \end{bmatrix}$:

The Givens rotations of the second step then operate on $(\boldsymbol{R}_1\,, \boldsymbol{L}'_2)$ and $(\boldsymbol{d}_1\,, \boldsymbol{c}'_2)$ so as to transform $\boldsymbol{L}'_2$ into a zero-block matrix. One thus gets $\boldsymbol{R}_2$ and $(\boldsymbol{d}_2\,, \boldsymbol{d}'_2)$.

At the end of this stage, we thus have

$$\begin{aligned} f_1(\boldsymbol{w}_1, \boldsymbol{v}) + f_2(\boldsymbol{w}_2, \boldsymbol{v}) = \|(\boldsymbol{K}_1\boldsymbol{w}_1 - (\boldsymbol{c}_1 - \boldsymbol{L}_1\boldsymbol{v})\|^2_{\mathbb{R}^{n_{\text{b}1}}} + \|\boldsymbol{R}_2\boldsymbol{v} - \boldsymbol{d}_2\|^2_{\mathbb{R}^{n_{\text{c}}}} \\ + \|\boldsymbol{d}'_1\|^2_{\mathbb{R}^{n_{\text{e}1}-n_{\text{b}1}-n_{\text{c}}}} + \|(\boldsymbol{K}_2\boldsymbol{w}_2 - (\boldsymbol{c}_2 - \boldsymbol{L}_2\boldsymbol{v})\|^2_{\mathbb{R}^{n_{\text{b}2}}} + \|\boldsymbol{d}'_2\|^2_{\mathbb{R}^{n_{\text{e}2}-n_{\text{b}2}}} \end{aligned}$$

The float solution in $\boldsymbol{v}$ at epoch 2 is therefore given by the formula $\hat{\boldsymbol{v}} = \boldsymbol{R}_2^{-1}\boldsymbol{d}_2$, hence the LS solutions in $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$:

$$\hat{\boldsymbol{w}}_1 = \boldsymbol{K}_1^{-1}(\boldsymbol{c}_1 - \boldsymbol{L}_1\hat{\boldsymbol{v}}), \qquad \hat{\boldsymbol{w}}_2 = \boldsymbol{K}_2^{-1}(\boldsymbol{c}_2 - \boldsymbol{L}_2\hat{\boldsymbol{v}})$$

The square of the LS residual norm at epoch 2 is then equal to $\|\boldsymbol{d}'_1\|^2_{\mathbb{R}^{n_{\text{e}1}-n_{\text{b}1}-n_{\text{c}}}} + \|\boldsymbol{d}'_2\|^2_{\mathbb{R}^{n_{\text{e}2}-n_{\text{b}2}}}$.

Next epochs

In summary, one thus operates, recursively, with the key structure shown in Fig. I.4: $\boldsymbol{K}_k$, $(\boldsymbol{L}_k\,,\boldsymbol{L}'_k)$ and $(\boldsymbol{c}_k\,,\boldsymbol{c}'_k)$ are computed from $\boldsymbol{B}_k$, $\boldsymbol{A}_k$ and $\boldsymbol{b}_k$, the quantities $\boldsymbol{R}_k$ and $(\boldsymbol{d}_k\,,\boldsymbol{d}'_k)$ being then computed from $(\boldsymbol{R}_{k-1}\,,\boldsymbol{L}'_k)$ and $(\boldsymbol{d}_{k-1}\,,\boldsymbol{c}'_\kappa)$. The generalization is straightforward; we then have

$$\begin{aligned}\sum_{\kappa=1}^{k} f_\kappa(\boldsymbol{w}_1,\boldsymbol{v}) = {}& \|(\boldsymbol{K}_1\boldsymbol{w}_1-(\boldsymbol{c}_1-\boldsymbol{L}_1\boldsymbol{v})\|^2_{\mathbb{R}^{n_{\mathrm{b}1}}}+\|\boldsymbol{R}_k\boldsymbol{v}-\boldsymbol{d}_k\|^2_{\mathbb{R}^{n_{\mathrm{c}}}}+\|\boldsymbol{d}'_1\|^2_{\mathbb{R}^{n_{\mathrm{e}1}-n_{\mathrm{b}1}-n_{\mathrm{c}}}}\\ &+\|(\boldsymbol{K}_2\boldsymbol{w}_2-(\boldsymbol{c}_2-\boldsymbol{L}_2\boldsymbol{v})\|^2_{\mathbb{R}^{n_{\mathrm{b}2}}}+\|\boldsymbol{d}'_2\|^2_{\mathbb{R}^{n_{\mathrm{e}2}-n_{\mathrm{b}2}}}\\ &\quad\vdots\\ &+\|(\boldsymbol{K}_k\boldsymbol{w}_k-(\boldsymbol{c}_k-\boldsymbol{L}_k\boldsymbol{v})\|^2_{\mathbb{R}^{n_{\mathrm{b}k}}}+\|\boldsymbol{d}'_k\|^2_{\mathbb{R}^{n_{\mathrm{e}k}-n_{\mathrm{b}k}}}\end{aligned}$$

The float solution in $\boldsymbol{v}$ at epoch k is therefore given by the formula

$$\hat{\boldsymbol{v}}=\boldsymbol{R}_k^{-1}\boldsymbol{d}_k \tag{I.7}$$

hence the LS solutions in $\boldsymbol{w}_1,\ldots,\boldsymbol{w}_k$:

$$\hat{\boldsymbol{w}}_\kappa=\boldsymbol{K}_\kappa^{-1}(\boldsymbol{c}_\kappa-\boldsymbol{L}_\kappa\hat{\boldsymbol{v}})\qquad(\kappa=1,\ldots,k) \tag{I.8}$$

The solution of the corresponding NLP problem is therefore defined as follows:

$$\check{\boldsymbol{v}}=\underset{\boldsymbol{v}\in\mathbb{Z}^{n_{\mathrm{c}}}}{\operatorname{argmin}}\,\|\boldsymbol{R}_k(\boldsymbol{v}-\hat{\boldsymbol{v}})\|^2_{\mathbb{R}^{n_{\mathrm{c}}}} \tag{I.9}$$

Indeed, $\boldsymbol{R}_k\boldsymbol{v}-\boldsymbol{d}_k=\boldsymbol{R}_k(\boldsymbol{v}-\hat{\boldsymbol{v}})$. The phase biases $\check{\boldsymbol{w}}_\kappa$ are then given by the relations

$$\check{\boldsymbol{w}}_\kappa=\boldsymbol{K}_\kappa^{-1}(\boldsymbol{c}_\kappa-\boldsymbol{L}_\kappa\check{\boldsymbol{v}})\qquad(\kappa=1,\ldots,k) \tag{I.10}$$

Their variance-covariance matrix is equal to $\boldsymbol{K}_\kappa^{-1}[\boldsymbol{K}_\kappa^{-1}]^{\mathrm{T}}$.

I.2.1 Handling new components of the closure-ambiguity variable

We now consider the case where ${}_{\mathrm{c},k}$ includes $n_{\mathrm{c}}^{\mathrm{a}}$ new closure edges (see Sect. IV.6.3); superscript a stands for added. One then proceeds in three steps:

1) $n_{\mathrm{c}}^{\mathrm{a}}$ closure-ambiguity entries are added at the top of column matrix $\boldsymbol{v}$;

2) as specified in Sect. I.3.2, $n_{\mathrm{c}}^{\mathrm{a}}$ columns are added on the right-hand side of $\boldsymbol{B}_k$;

3) as shown in Fig. I.5, to build $\boldsymbol{R}_k$, the last $n_{\mathrm{c}}^{\mathrm{a}}$ lines of $\boldsymbol{K}$ and $\boldsymbol{L}$ obtained through the first QR step are added at the top of $\boldsymbol{R}$. Matrices $\boldsymbol{K}_k$, $\boldsymbol{L}_k$ and $\boldsymbol{c}_k$ are then updated accordingly.

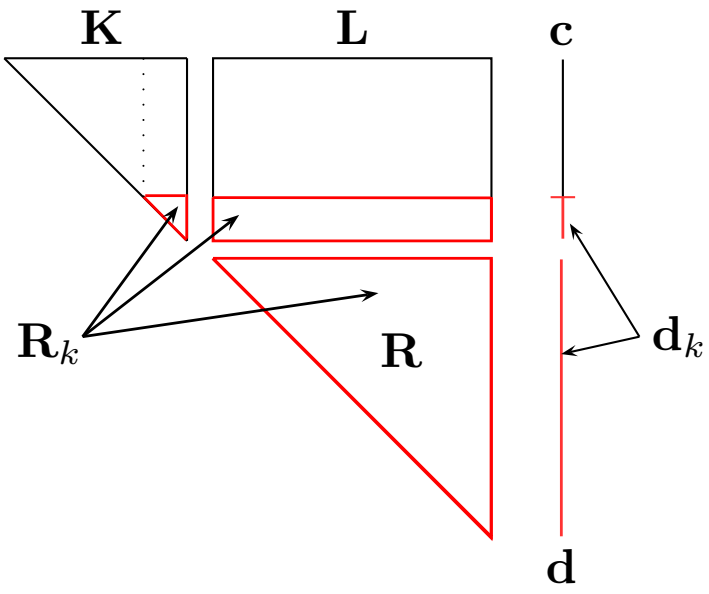

Figure I.5: Handling new components of the closure-ambiguity variable. When new entries of $\boldsymbol{v}$ appear at epoch k, the first columns of $\boldsymbol{A}_k$ are processed as the last columns of $\boldsymbol{B}_k$ (see Fig. I.3). The recursive QR operation then yields the quantities $\boldsymbol{K}$, $\boldsymbol{L}$, $\boldsymbol{c}$, $\boldsymbol{R}$ and $\boldsymbol{d}$. To get $\boldsymbol{R}_k$ and $\boldsymbol{d}_k$, one then proceeds as illustrated here.

I.3 Construction of matrices B_k, A_k and b_k

We first consider the case where the variance-covariance matrix $\boldsymbol{V}_k$ of the data involved in the definition of b_k is the identity: $\boldsymbol{V}_k = \boldsymbol{I}$. Denoting by b_k the column matrix whose entries are the values of b_k on the edges of the observational graph $\mathscr{H}_k$ (see Sect. IV.1), we then have $\boldsymbol{b}_k = \mathrm{b}_k$. To build $\boldsymbol{B}_k = \mathrm{B}_k$ and $\boldsymbol{A}_k = \mathrm{A}_k$, we then distinguish the cases where at epoch k, n_c does not increase (Sect. I.3.1), or increases (Sect. I.3.2). The case $\boldsymbol{V}_k \neq \boldsymbol{I}$ is dealt with in Sect. I.3.3.

I.3.1 Case where n_c does not increase

Matrix B_k, whose number of columns is $n_{\mathrm{b}k}$, is built from the characteristic function $\mathcal{H}_k$ of $\mathscr{H}_k$; see Fig. IV.1. The pth line of B_k corresponds to the pth edge $(\mathrm{r}_i, \mathrm{s}_j)$ on which $\mathcal{H}_k(i,j) = 1$. All the matrix elements of that line are zero, except (one or) two of them (see Eq. (IV.6) and the definition of $\varpi_{\mathrm{s}k}$ in Sect. IV.6.2):

$$\mathrm{B}_k^{p,i-1} = 1 \text{ (for } i > 1), \qquad \mathrm{B}_k^{p,\,n_\mathrm{r}-1+\varpi_{\mathrm{s}k}(j)} = -1$$

Matrix A_k has n_c columns: the number of elements of $_{\mathrm{c},k}$; see Sect IV.6.3 and Fig. IV.5. According to the action of $\mathcal{R}_k^\mathrm{e}$, the entries of the column associated with some closure edge $(\mathrm{r}_i, \mathrm{s}_j)$ are then all zero, except that corresponding to the line associated with that edge if $\mathcal{H}_k(i,j) = 1$; that entry is then set equal to unity. The lines of A_k are of course sorted as the lines of B_k.

I.3.2 Case where n_c increases

We here consider the case where n_c^a new closure edge(s) appear(s) in $_{c,k}$ at some epoch $k > 1$: $n_c := n_c + n_c^a$; see Sect IV.6.3.

Matrix $\mathbf{B}_k^{p,q}$ is defined as in Sect. I.3.1, but n_c^a columns are then added on its right-hand side. (For example, at epoch 2 of Fig. IV.5, $\mathbf{B}_2$ has four additional columns.) The entries of the column of $\mathbf{B}_k$ associated with some new closure edge $(\mathrm{r}_i, \mathrm{s}_j)$ are all zero, except that corresponding to the line associated with that edge; that entry is set equal to unity.

Matrix $\mathbf{A}_k$ is then built as in Sect. I.3.1, except for the new closure-edges, since they are then taken into account in the augmented definition of $\mathbf{B}_k$.

I.3.3 Case where V_k is not the identity

We here consider the general case where the variance-covariance matrix $\boldsymbol{V}_k$ is to be taken into account. In the QR implementation under consideration, the inverse of $\boldsymbol{V}_k$ is then factorized in the form

$$\boldsymbol{V}_k^{-1} = \boldsymbol{U}_k^{\mathrm{T}} \boldsymbol{U}_k \tag{I.11}$$

where $\boldsymbol{U}_k$ is an upper-triangular matrix. As

$$(\mathbf{B}_k \boldsymbol{w}_k + \mathbf{A}_k \boldsymbol{v} - \mathbf{b}_k)^{\mathrm{T}} \boldsymbol{V}_k^{-1} (\mathbf{B}_k \boldsymbol{w}_k + \mathbf{A}_k \boldsymbol{v} - \mathbf{b}_k) = \|\boldsymbol{U}_k (\mathbf{B}_k \boldsymbol{w}_k + \mathbf{A}_k \boldsymbol{v} - \mathbf{b}_k)\|^2_{\mathbb{R}^{n_{ek}}}$$

matrices $\boldsymbol{B}_k$, $\boldsymbol{A}_k$ and $\boldsymbol{b}_k$ are then given by the relations

$$\boldsymbol{B}_k = \boldsymbol{U}_k \mathbf{B}_k, \quad \boldsymbol{A}_k = \boldsymbol{U}_k \mathbf{A}_k, \quad \boldsymbol{b}_k = \boldsymbol{U}_k \mathbf{b}_k \tag{I.12}$$

The problem is then to solve Eq. (IV.82) in the Euclidean LS sense.

Bibliography

Closest point search in lattices
Agrell, E., Eriksson, T., Vardy, A., Zeger, K., 2002, IEEE T Inform Theory, 48, 2201-2214.

The optimal LLL algorithm is still polynomial in fixed dimension
Akhavi, A., 2003, Theor Comput Sci, 297, 3-23.

S-transformations and criterion matrices
Baarda, HA, 1973, Publications in Geodesy 1. Netherlands Geodetic Commission, Delft. 5

On Lovász lattice reduction and the nearest lattice point problem
Babai, L., 1986, Combinatorica, 6, 1–13.

Single receiver phase ambiguity resolution with GPS data
Bertiger, W., Desai, S.D., Haines, B., Harvey, N., Moore A.W., 2010, J Geod 84: 327-337

Factorization methods for discrete sequential estimation
Bierman, G.J., 1977, Vol. 128 in Mathematics in science and engineering, Academic Press, Inc. New-York

Algebraic Graph Theory
Biggs, N., 1996, Cambridge U. Press, Cambridge, UK

Numerical methods for least-squares problems
Björck, A., 1996, SIAM

Carrier phase ambiguity resolution for the global positioning system applied to geodetic baselines up to 2000 km
Blewitt, G., 1989, J Geophys Res 94(B8), 10187-10203

Principles of Optics
Born, M., Wolf, E., 1970, Oxford, Pergamon Press

A Course in Computational Algebraic Number Theory
Cohen, H., 1996, Springer-Verlag, Berlin

Undifferenced GPS ambiguity resolution using the decoupled clock model and ambiguity datum fixing
Collins, P., Bisnath, S., Lahaye, F., Heroux, P., 2010, J Navigation 57, 123-135

A new method for making maps with unstable interferometers
Cornwell, T.J., Wilkinson, P.N., 1981, MNRAS 196, 1067–1086

A processing strategy for the application of the GPS in networks
de Jonge, P.J., 1998, PhD dissertation 46, Netherlands Geodetic Commission, Delft

Improving carrier-phase ambiguity resolution in global GPS network solutions
Ge, M., Gendt, G., Dick, G., Zhang, F.P., 2005, J Geod 79, 103-110

Resolution of GPS carrier-phase ambiguities in Precise Point Positioning (PPP) with daily observations
Ge, M., Gendt, G., Rothacher, M., Shi, C., Lui, J., 2008, J Geod 82, 389-399

A novel real-time precise positioning service system: global precise point positioning with regional augmentation
Ge, M., Douša, J., Li, X., Ramatschi, M., Nischan, T., Wickert, J., 2012, J GPS 11, 2-10

Integer ambiguity resolution in precise point positioning: method comparison
Geng, J., Meng, X., Dodson, A.H., Teferle, F.N., 2010, J Geod 84, 569–581

Matrix computations
Golub, G.H., van Loan, C.F., 1989, 2nd edition, The Johns Hopkins University Press, Baltimore, Maryland

Fast integer least-squares estimation for GNSS high dimensional ambiguity resolution using lattice theory
Jazaeri, S., Amiri-Simkooei, A.R., Sharifi, M.A., 2012, J Geod, 86, 123–136.

Understanding Radio Polarimetry, IV. The full-coherency analogue of scalar self-calibration: Self-alignment, dynamic range and polarimetric fidelity
Hamaker, J.P., 2000, Astron. and Astrophys. Suppl. 143, 515–534

CLEAN *and* WIPE
Lannes, A., Anterrieu, E., & Maréchal, P. 1997, Astron. Astrophys. Suppl. Ser. 123, 183–198

Résolution d'ambiguïtés entières sur graphes interférométriques et GPS
Lannes, A., 2001, CR Acad Sci I-Math 333, 707–712

A global analysis of the phase calibration operation
Lannes, A., 2005, JOSA A 22, 697–707

QR implementation of GNSS centralized approaches
Lannes, A., Gratton S., 2008, J GPS 7, 133–147

GNSS networks in algebraic graph theory
Lannes, A., Gratton S., 2009, J GPS 8, 53–75

Pseudo-clock biases for PPP. The algebraic approach
Lannes, A., Gratton S., Durand, S., 2010, J GPS 9, 68

GNSS algebraic structures
Lannes, A., Teunissen, P.J.G., 2011, J Geod 85, 273–290

Algebraic analysis of the phase-calibration problem in the self-calibration procedures
Lannes, A., Prieur, J.-L., 2011, AN, 332, 759–784

On the theoretical link between LLL-reduction and LAMBDA*-decorrelation*
Lannes, A., 2013, J Geod 87, 323

Calibration of the clock-phase biases of GNSS networks: the closure-ambiguity approach
Lannes, A., Prieur, J.-L., 2013, J Geod 87, 709

Integer-ambiguity resolution in astronomy and geodesy
Lannes, A., Prieur, J.-L., 2014, Astron. Nach., 335, 198–209 [lien vers HAL]

Integer ambiguity resolution on undifferenced GPS phase measurements and its applications to PPP
Laurichesse, D., Mercier, F., 2007, ION GNSS 2007 20th international technical meeting of the satellite division, 25-28 Sept 2007, Forth Worth, TX, pp 839-848

Factorizing polynomials with rational coefficients
Lenstra, A.K., Lenstra, H.W., Lovász, L., 1982, Math Ann 261, 515–534.

Regional reference network augmented precise point positioning for instantaneous ambiguity resolution
Li, X., Zhang, X., Ge, M., 2011, J Geod 85, 151-158

A method for improving uncalibrated phase delay estimation and ambiguity-fixing in real-time precise point positioning
Li, X., Ge, M., Zhang, H., Wickert, J., 2014, J Geod (in press)

Concepts and performance results on the combination of different integrity methods using UAIM and GNSS without SA
Loehnert, E., Wolf, R., Pielmeier, J., Werner, W., Zink, T., 2000, Proc. ION GPSS-2000, Salt Lake City, USA, 2831-2840

Zero-difference GPS ambiguity resolution at CNES-CLS IGS Analysis Center
Loyer, S., Perosanz, F., Mercier, F., Capdeville, H., Marty, J.-C., 2012, J Geod 86, 991-1003

Linear algebra and its applications
Luk, F.T., & Tracy, D.M. 2008, pp 428–441

A parallel LLL algorithm
Luo, L., Qiao, S., 2011, Proceedings of the Fourth International C* Conference on Computer Science and Software Engineering. ACM Int Conf P Series, ACM Press, 93-101

A parallel LLL algorithm
Luo, L., Qiao, S., 2011, ACM Int Conf P Series, ACM Press, 93-101.

Zero-difference ambiguity blocking. Properties of receiver-satellite biases
Mercier, F., Laurichesse, D., 2008, Proc. ENC GNSS-2008. Toulouse, France.

Mathematical Programming, the State of the Art
Moré, J., 1983, Eds Bachem, A., Grötschel, M., & Korte, B., Springer-Verlag, Berlin, p 258

An LLL algorithm with quadratic complexity
Nguyen, P.Q., Stehlé, D., 2009, SIAM J Comput, 39, 874-903

Single-frequency integer ambiguity resolution enabled Precise Point Positioning
Odijk, D., Teunissen, P.J.G., Zhang, B., 2012, J Surv Eng 138, 193-202

Lattice basis reduction: improved practical algorithms and solving subset sum problems
Schnorr, C.P., Euchner, M., 1994, Math Program, 66, 181-199.

Generalized inverses, adjustment, the datum problem and S-transformations
Teunissen, P.J.G., 1984, In E. Grafarend and F. Sanso (eds), Optimization of Geodetic Networks, Springer, Berlin, pp 11-55

The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation
Teunissen, P.J.G., 1995, J Geod, 70, 65-82.

On the GPS widelane and its decorrelating property
Teunissen, P.J.G., 1997, J Geod 71, 577-587

GPS for Geodesy
Teunissen, P.J.G., Kleusberg, A., 1998 2nd edn, Springer Verlag.

Rank-defect integer estimation and phase-only modernized GPS ambiguity resolution
Teunissen, P.J.G., Odijk, D., 2003, J Geod 76, 523-535

PPP-RTK: Results of CORS network-based PPP with integer ambiguity resolution
Teunissen, P.J.G., Odijk, D., Zhang, B., 2010, J Aeronautics, Astronautics and Aviation 42, 223-229

Recursive data processing for kinematic GPS surveying
Tiberius, C.C.J.M., 1998, Publications on Geodesy, New series: ISSN 0165 1706, Number 45, Netherlands Geodetic Commission, Delft

The GNSS integer ambiguities: estimation and validation
Verhagen, S., 2005, Ph.D. thesis, Univ. Deft, The Netherlands.

New global satellite system ambiguity resolution methods compared to existing approaches
Verhagen, S., Teunissen, P.J.G., 2006, J Guid Control Dynam 29, 891-991

Reduced ambiguity calibration for LOFAR
Yatawatta, S., 2012, Exp. Astron., 34, 89-103

A novel un-differenced PPP-RTK concept
Zhang, B., Teunissen, P.J.G., Odijk, D., 2011, J Navigation 64, S180-S191

URTK: undifferenced network RTK positioning
Zou, X., Ge, M., Tang, W., Shi, C., Liu, J., 2012, GPS Solut

Precise point positioning for the efficient and robust analysis of the GPS data from large networks
Zumberge, J.F., Heflin, M.B., Jefferson, D.C., Watkins, M.M., Webb, F.H., 1997, J Geophys Res 102(B3), 5005-5017

Index

Glossary

$\mathscr{V}$: set of pupil elements used for the definition of the baselines of $\mathscr{E}$. 8, 13, 14, 25, 151

$\mathscr{E}$: set of baselines (i, j) with $i < j$, on which the phases $V_{\mathrm{d}}(i, j)$ are "well defined" (see Sect. II.1.1). 7, 149

$\mathscr{E}_{\mathrm{c}}$: set of loop-entry baselines, i.e., baselines of $\mathscr{E}$ that do not lie in $\mathscr{E}_{\mathrm{st}}$. 13

$\mathscr{E}_{\mathrm{st}}$: set of baselines of the spanning tree $\mathscr{G}_{\mathrm{st}}$. 10, 13

$\mathscr{G}$: phase-calibration graph ($\mathscr{G} \overset{\text{set}}{=} \mathscr{G}(\mathscr{V}, \mathscr{E})$). 8, 18, 38, 40, 149, 151

$\mathscr{G}_{\mathrm{st}}$: spanning tree of the phase-calibration graph $\mathscr{G}$ ($\mathscr{G}_{\mathrm{st}} \overset{\text{set}}{=} \mathscr{G}(\mathscr{V}, \mathscr{E}_{\mathrm{st}})$). 38, 149

E : Selfcal: baseline-phase space, i.e., space of real-valued functions on $\mathscr{E}$. GNSS: edge-delay space.. 12

E_{b} : baseline-bias phase space, i.e., range of the B operator from V_b into E. It corresponds to the functions $\beta_{\mathrm{b}}(i, j)$ of the form $\alpha(i) - \alpha(j)$ with $\alpha(1) = 0$. 13

E_{c} : closure-phase space, i.e., space of the functions of E that vanish on $\mathscr{E}_{\mathrm{st}}$. 13

E_{st} : spanning tree phase space, i.e., space of the functions of E that vanish on $\mathscr{E}_{\mathrm{c}}$. 13

E_w : weighted baseline-phase space. It corresponds to E endowed with the inner product $(\beta_1, \beta_2)_w$ (see Eq. II.53). 20, 37, 38

V_b : Selfcal: pupil-phase space, i.e., space of real-valued functions on $\mathscr{V}$. GNSS: vertex-bias space.. 12, 13, 15

n_{c} : number of loop-entry baselines of $\mathscr{E}$ (= number of baselines of $\mathscr{E}_{\mathrm{c}}$). 11, 19

n_{c3} : maximum number of independent closure phases of order three of $\mathscr{G}$ (= rank of $\mathcal{C}_3^{\circ}$). 18

n_{e} : number of edges of $\mathscr{G}$ (= number of baselines of $\mathscr{E}$). 8

n_{st} : number of edges of the spanning tree $\mathscr{G}_{\mathrm{st}}$ (= number of baselines of $\mathscr{E}_{\mathrm{st}}$). 13

n_{v} : number of vertices of $\mathscr{G}$ (= number of pupil elements of $\mathscr{V}$). 8

B : baseline-bias phase operator from V_b into E; $(B\alpha)(i, j) = \alpha(i) - \alpha(j)$. 9

B_{c} : operator from V_b into E_{c} induced by B. $(B_{\mathrm{c}}\alpha)(i, j) = (B\alpha)(i, j)$ for the loop-entry baselines and $(B_{\mathrm{c}}\alpha)(i, j) = 0$ on the spanning tree $\mathscr{G}_{\mathrm{st}}$. 13

$B_{\rm st}$: operator from V_b into $E_{\rm st}$ induced by B. $(B_{\rm c}\alpha)(i,j) = (B\alpha)(i,j)$ for the spanning tree baselines and $(B_{\rm c}\alpha)(i,j) = 0$ for the loop-entry baselines. 13

B^* : adjoint of B, relative to the inner product w (see Eqs. II.53, II.59). 20

C : phase-closure operator from E to $E_{\rm c}$ (i.e., oblique projection onto $E_{\rm c}$ along $E_{\rm b}$). 15

$\mathcal{C}_3$: usual phase-closure operator. It provides the maximum set of independent closure phases of order three of $\mathscr{G}$. 18

$\mathcal{C}_3^\circ$: phase-closure operator. It provides all the closure phases of order three of $\mathscr{G}$. 18

$\mathcal{F}$: Fourier-transform operator (see Eq. II.1). 6

$\varphi(i,j)$: baseline phase vector used as input for the phase-calibration procedure: $\varphi(i,j) = \varphi_{\rm d}(i,j) - \varphi_{\rm m}(i,j)$. 152

$Q_{\rm c}$: orthogonal projection of E onto $E_{\rm c}$ (see Fig. II.5). 17

$Q_{\rm st}$: orthogonal projection of E onto $E_{\rm st}$ (see Fig. II.5). 14

$V_{\rm d}(i,j)$: complex visibility data function $V_{\rm d}(i,j) \approx V_{\rm o}(i,j)\, e^{{\rm i}[\alpha_{\rm d}(i) - \alpha_{\rm d}(j)]}$ (Eq. II.4). 6, 149

$V_{{\rm d}\star}(i,j)$: calibrated visibility function $V_{{\rm d}\star}(i,j) \overset{\rm def}{=} V_{\rm d}(i,j)\, e^{-{\rm i}[\alpha_{d\star}(i) - \alpha_{d\star}(j)]}$ (Eq. II.7). 7

$V_{\rm m}(i,j)$: visibility function of the model $V_{\rm m}(i,j) \approx V_{\rm o}(i,j)\, e^{{\rm i}[\alpha_{\rm m}(i) - \alpha_{\rm m}(j)]}$ (Eq. II.5). 7

$V_{\rm o}(i,j)$: complex visibility function of the object $V_{\rm o}(i,j) \overset{\rm def}{=} (\mathcal{F} s_{\rm o})\{\boldsymbol{u}(i,j)\}$ (Eq. II.3). 6

$\alpha_{\rm d}(i)$: pupil-bias phase of pupil element $\#i$ (see Eqs. II.4 and II.5). 7

$\alpha_d(i)$: pupil-bias phase of pupil element $\#i$ taking $\alpha_{\rm d}(1)$ as the phase reference: $\alpha_d(i) \overset{\rm def}{=} \alpha_{\rm d}(i) - \alpha_{\rm d}(1)$ (Eq. II.6). 7

α : pupil phase function defined as $\alpha = \alpha_d - \alpha_\varphi$ (see Eq. II.74). It is used as variable of the functionals f and g (see Eqs. II.77 and II.78). 23

α_φ : pupil phase function defined as $\alpha_\varphi = B_{\rm st}^{-1} Q_{\rm st}\varphi$ (see Eq. II.73). 23

φ : baseline phase function defined as $\varphi(i,j) \overset{\rm def}{=} \varphi_{\rm d}(i,j) - \varphi_{\rm m}(i,j)$ (Eq. II.10). 8

$\varphi_{\rm d}$: (experimental) baseline phase function defined with (Eq. II.9): $V_{\rm d} = \rho_{\rm d}\, e^{{\rm i}\varphi_{\rm d}}$. 8

$\varphi_{\rm m}$: (model) baseline phase function defined with (Eq. II.9): $V_{\rm m} = \rho_{\rm d}\, e^{{\rm i}\varphi_{\rm m}}$. Note that the model visibility is progressively determined in the self-calibration procedure). 8

$\varphi_{\rm c}(i_\ell, j_\ell)$: baseline closure phase function. Only relevant for the loop-entry baselines since $\varphi_{\rm c}(i,j) = 0$ for the baselines (i,j) belonging to the spanning tree $\mathscr{G}_{\rm st}$. 22

$\mathfrak{e}_\ell$: characteristic function of the loop-entry baseline (i_ℓ, j_ℓ). 21

ε_α : phase-calibration residual defined as $\varepsilon_\alpha = \arc(\widehat{\varphi}_{\rm c} - B\alpha)$ (Eq. II.81). 24

$\arc$: arc function defined as $\arc\theta = \theta - 2\pi\left\lfloor\frac{\theta}{2\pi}\right\rceil$ (Eq. II.20). 10

$\chord$: chord function defined as $\chord\theta = 2|\sin(\theta/2)|$ (Eq. II.17). 10

$f_\circ(\alpha_d)$: chord functional $f_\circ(\alpha_d) = |\chord(\varphi - B\alpha_d)|_w^2$ (Eq. II.18) . 8

$f(\alpha)$: chord functional $f(\alpha) = |\chord(\widehat{\varphi}_\mathrm{c} - B\alpha)|_w^2$ (Eq. II.77) . 23

$g_\circ(\alpha_d)$: arc functional $g_\circ(\alpha_d) = |\arc(\varphi - B\alpha_d)|_w^2$ (Eq. II.80) . 10, 23

$g(\alpha)$: arc functional $g(\alpha) = |\arc(\widehat{\varphi}_\mathrm{c} - B\alpha)|_w^2$ (Eq. II.78) . 23

closure phase : for any function β of E, its closure phase function is a function of E_c defined by $\beta_\mathrm{c} \stackrel{\mathrm{def}}{=} \beta - \beta_\mathrm{b}$ with $\beta_\mathrm{b} = B\alpha_\beta$ in E_b and $\alpha_\beta = B_\mathrm{st}^{-1}Q_\mathrm{st}\beta$ (see Eqs. II.35, II.38 and II.39). It is null on the spanning tree $\mathscr{G}_\mathrm{st}$. For the loop-entry baselines, $\beta_\mathrm{c}(i_\ell, j_\ell)$ corresponds to the sum of the phases $\beta(i,j)$ of the baselines (i,j) making up the loop closed by (i_ℓ, j_ℓ) (see Sect. II.2.4). 150, 152

Fourier-synthesis : image restoration produre using both the phase obtained after a phase-calibration procedure and the Fourier modulus derived from the experimental visibilities. In general this information does not lead to a unique solution. A piori information is then needed to constrain this solution. 152

loop-entry baseline : baseline of $\mathscr{E}$ that do not lie in $\mathscr{E}_\mathrm{st}$ (see Eq. II.22). 11, 21, 149–151

phase-calibration : calibration of the phases of the experimental visibilities that have been altered, for instance by the atmospheric turbulence. More precisely, it consists in determining the pupil-bias phases α_d of all the elements of $\mathscr{V}$ that minimize the functionnal the size of $\arc(\varphi - B\alpha_d)$ on $\mathscr{E}$; see Eq. (II.19). 151, 152

phase-calibration graph : the phase-calibration graph $\mathscr{G}$ of an experimental setup is the couple $(\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ is the set of its vertices (its antennas or telescopes), and $\mathscr{E}$ that of its edges (its baselines): see Fig. II.1. 149

phase-calibration residual : quantity defined as $\varepsilon_\alpha = \arc(\widehat{\varphi}_\mathrm{c} - B\alpha)$ (Eq. II.81) It is used for clarifying the expression of the phase calibration residuals in the arc approach (see sect. II.3.2). In particular $g(\alpha_g) = |\varepsilon_{\alpha_g}|_w^2$ (see Eq. II.82). 24, 150

phase-closure : combination of the phases of the experimental visibilities in such a way that the contribution of the atmospheric turbulence cancels to zero. 152

phase-closure operator : oblique projection of E onto E_c along E_b (see Fig. II.5). 150

pupil-bias phase : instantaneous phase of a pupil element of $\mathscr{V}$. In the context of interferometric observations through fast turbulence, this phase cannot be measured experimentally and is therefore considered as a bias. 151

scalar :electromagnetic field. In this appendix, we limit our study to the scalar case, and consider the global intensity of the electro-magnetic field only (individual polarimetric channels are not taken into account). 43, 44

self-calibration : iteration procedure made of successive phase-calibration and Fourier-synthesis steps. The phase-calibration steps rely on the phase-closure relations on the baseline phases $\varphi(i,j)$. In this way, the object brightness distribution itself is used for the phase calibration. 150

spanning tree : subgraph of a phase-calibration graph formed by n_{v} vertices and $n_{\mathrm{v}}-1$ edges, with no 'cycle' in it. It is unique in the case of a weighted graph when the baseline weights are all different (see Sect. II.2.1). 10, 149–151

usual phase-closure operator : closure operator whose matrix that provides a maximum set of independent closure phases of order three. 150