

# Plotting the map projection graticule involving discontinuities based on combined sampling

Tomáš Bayer

*Department of Applied Geoinformatics and Cartography, Faculty of Science, Charles University, Czech Republic*  
bayertom@natur.cuni.cz

---

**Abstract.** This article presents a new algorithm for interval plotting the projection graticule on the interval  $\Omega = \Omega_\varphi \times \Omega_\lambda$  based on the combined sampling technique. The proposed method synthesizes the uniform and adaptive sampling approaches and treats the discontinuities of the coordinate functions  $F, G$ . A full set of the projection constant values represented by the projection pole  $K = [\varphi_k, \lambda_k]$ , two standard parallels  $\varphi'_1, \varphi'_2$  and the central meridian shift  $\lambda'_0$  are supported. In accordance with the discontinuity direction it utilizes a subdivision of the given latitude/longitude intervals  $\Omega_\varphi = [\varphi, \bar{\varphi}]$ ,  $\Omega_\lambda = [\lambda, \bar{\lambda}]$  to the set of disjoint subintervals  $\Omega_{k,\varphi}^g, \Omega_{k,\lambda}^g$  forming tiles without the internal singularities, containing only “good” data; their parameters can be easily adjusted. Each graticule tile borders generated over  $\Omega_k^g = \Omega_{k,\varphi}^g \times \Omega_{k,\lambda}^g$  run along the singularities. For combined sampling with the given threshold  $\bar{\alpha}$  between the adjacent segments of the polygonal approximation, the recursive approach has been used; meridian/parallel offsets are  $\Delta\varphi, \Delta\lambda$ . Finally, several tests of the proposed algorithms are involved.

**Keywords:** digital cartography; mathematical cartography; adaptive sampling; graticule; meridians; parallels; recursive approach; map projection; great circle; discontinuity; visualization; sphere.

---

## 1. Introduction

Maps are an essential part of our history and cultural heritage; a close attention is paid to their study and research. Working with the map content, its geometric and spatial characteristics described by the map projection cannot be ignored. The map projection  $\mathbb{P}$  is defined by the coordinate functions  $F(\varphi, \lambda)$ ,  $G(\varphi, \lambda)$  of two independent variables, the latitude  $\varphi$ , and the longitude  $\lambda$ .  $F, G$  may have a different form; they may not be continuous, or their analytic form may not exist. To avoid the discontinuities, the interval  $\Omega = \Omega_\varphi \times \Omega_\lambda$  will be divided into the disjoint set of  $k$  “good” subintervals  $\Omega_k^g$ ; their “boundaries” run along the singularities.

A current approach concentrated on uniform sampling of the meridians and parallels with the steps  $\delta\varphi$ ,  $\delta\lambda$  may not be sufficient. Despite its popularity, the equally spaced points cannot describe the meridian/parallel course without errors; the problems of undersampling or oversampling are common. Because of the complex/straight shapes of the meridians and parallels, this technique is not generally recommended. Adaptive sampling brings several benefits, it adapts to a different curvature of the function, reduces the amount of data and provides a natural and smooth plot of the function without jumps and unnatural breaks. This technique is popular in computer graphics; recall the deCasteljau or Chaikin’s algorithms for the curve approximation. A combination of the uniform and adaptive sampling techniques synthesizes their advantages, which is discussed in [5].

Taking into account the facts mentioned above, this technique may provide a smooth and natural depiction of the graticule which is less data-redundant. A full set of the projection

constant values represented by the projection pole  $K = [\varphi_k, \lambda_k]$ , two standard parallels  $\varphi'_1, \varphi'_2$ , and the central meridian shift  $\lambda'_0$  is supported. They have a strong influence on the shape of the graticule. The last parameter makes the process of the graticule reconstruction more difficult; an intersection of the meridian  $m(\lambda'_0)$  with the parallel  $p(\varphi)$  as well as with the meridian  $m(\lambda)$  need to be determined. Another essential subproblem is represented by the automatic detection of the discontinuities; this issue refers to many coordinate functions  $F, G$ .

Finally, all algorithms will be tested on the real cartographic data represented by several projections. To illustrate the behavior, properties, and drawbacks of the proposed methods, the projections with more singularities will be preferred; the Fournier I. projection represents a typical candidate. Apart from current outcomes in the cartographic or GIS software tools, several specific applications may occur. Let us mention the problem of the unknown map projection analysis. Providing the visualization of the results of the detection algorithm illustrates its efficiency, which, from the “raw numbers”, may not be apparent.

This paper is organized as follows. In Section 3 the mathematical background of the solution is described. Section 4 is devoted to the proposed combined sampling algorithm, the properties of which in Section 5 will serve to analyze its behavior on real data.

## 2. Related Work

The polygonal approximation of the parametric curve based on adaptive sampling is mentioned in several papers: the refinement criteria in [12], the approximation by polygonal curves in [7], the spatial approximation of implicit curves in [17], [10], the affine arithmetic working in triangulated models in [21]. The curve interpolation algorithms preserving its speed are described in [23], [25], [26], the adaptive-feed rate technique in [27]. However, the map projections are never defined by the implicit equations. There are several papers focused on the approximation and drawing the functions or surfaces containing discontinuities. The interpolation and approximation of the piece-wise smooth functions are discussed in [2], [9]. Several detection methods for the localization of the singularities of surfaces can be found in [13], [22], [19], [11], [18], [1], [8], the combined sampling technique in [5]. The reconstructed projection graticule and its plot may be utilized in many cartographic or GIS applications. An interesting application is represented by the analysis of the unknown map projection described in [6], [4] providing the visualization of results of the detection algorithms, typically the reconstructed projection graticule.

## 3. Map projection and its properties

Let  $S^2$  be a sphere of the radius  $R$  in  $\mathbb{R}^3$  (reference surface, Earth),  $A = (0, 0, R)$ ,  $B = (0, 0, -R)$  North and South Poles of  $S^2$ , and  $\sigma$  a plane. For a current point  $Q = [\varphi, \lambda] \in S^2$ , different from  $A, B$ , and its image  $P' = [X, Y] \in \sigma$ , the map projection  $\mathbb{P} : S^2 - \{A, B\} \rightarrow \sigma$ ,  $\mathbb{P}(Q) = P'$ , is defined by the coordinate functions  $F, G$ , of two independent variables  $\varphi, \lambda$

$$X = F(\varphi, \lambda), \quad Y = G(\varphi, \lambda), \quad (3.1)$$

which are continuous with their first order partial derivatives, finite, and independent. A transformation between the normal and oblique aspects is performed using the laws of spher-

ical trigonometry

$$\sin \varphi' = \sin \varphi_k \sin \varphi + \cos \varphi_k \cos \varphi \cos \Delta\lambda, \quad (3.2)$$

$$\tan \lambda' = \frac{\cos \varphi \sin \Delta\lambda}{\cos \varphi \sin \varphi_k \cos \Delta\lambda - \sin \varphi \cos \varphi_k}, \quad (3.3)$$

where  $\varphi', \lambda'$  are the spherical coordinates related to  $K$ . The inverse transformation has the form of

$$\sin \varphi = \sin \varphi_k \sin \varphi' - \cos \varphi_k \cos \varphi' \cos \lambda', \quad (3.4)$$

$$\tan \Delta\lambda = \frac{\cos \varphi' \sin \lambda'}{\cos \varphi' \sin \varphi_k \cos \lambda' + \sin \varphi' \cos \varphi_k}, \quad (3.5)$$

where  $\Delta\lambda = \lambda - \lambda_k$ . A meridian of the sphere  $S^2$  is a curve  $m(\varphi, \lambda_c)$ ,  $\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ ,  $\lambda_c = \text{const}$ ; its image in  $\mathbb{P}$  is  $\tilde{m}(\varphi, \lambda_c) = (F(\varphi, \lambda_c), G(\varphi, \lambda_c))$ . A parallel of the sphere  $S^2$  in  $\mathbb{R}^3$  is a curve  $p(\varphi_c, \lambda)$ ,  $\lambda \in [-\pi, \pi]$ ,  $\varphi_c = \text{const}$ ; its image in  $\mathbb{P}$  is  $\tilde{p}(\varphi_c, \lambda) = (F(\varphi_c, \lambda), G(\varphi_c, \lambda))$ . In the simplified notation  $m(\lambda)$  represents a meridian of the longitude  $\lambda$ ,  $p(\varphi)$  a parallel of the latitude  $\varphi$  and  $\tilde{m}(\lambda)$ ,  $\tilde{p}(\varphi)$  their projected variants.

### 3.1. Map projection singularities

During the graticule construction, several types of singularities occur; see Fig. 3.2. The sampling algorithm should be adapted to these facts. A meridian or a parallel may intersect a singularity or may be coincident. There are several simple strategies for handling and detecting the discontinuities; their overview can be found in [15], [3], [16], [5]. Our approach is based on the *LR* criterion described in [20].

**Coincidence with the infinite singularity.** If the meridian  $m(\lambda)$  coincides with the infinite singularity  $c$ ,  $\lambda = c$ , then,

$$\lim_{\lambda \rightarrow c^\pm} F(\varphi, \lambda) = \infty \vee \lim_{\lambda \rightarrow c^\pm} F(\varphi, \lambda) = -\infty \vee \lim_{\lambda \rightarrow c^\pm} G(\varphi, \lambda) = \infty \vee \lim_{\lambda \rightarrow c^\pm} G(\varphi, \lambda) = -\infty,$$

(Case A), or,

$$\lim_{\lambda \rightarrow c^\pm} F(\varphi, \lambda) = \pm\infty \vee \lim_{\lambda \rightarrow c^\pm} F(\varphi, \lambda) = \mp\infty \vee \lim_{\lambda \rightarrow c^\pm} G(\varphi, \lambda) = \pm\infty \vee \lim_{\lambda \rightarrow c^\pm} G(\varphi, \lambda) = \mp\infty,$$

(Case B) for any  $\varphi \in (-\pi/2, \pi/2)$ . Using the left- and right-boundaries  $(c + \varepsilon)$  and  $(c - \varepsilon)$  in Case A, the original projected meridian  $\tilde{m}(c)$  is replaced with

$$\tilde{m}(c) = 0.5 [\tilde{m}(c - \varepsilon) + \tilde{m}(c + \varepsilon)].$$

For Case B, the original meridian  $m(\lambda)$  is replaced with two new meridians  $m^-(\lambda \rightarrow c^-)$ , and  $m^+(\lambda \rightarrow c^+)$ . Analogously, Cases C, D occur, if a parallel  $p(\varphi)$  coincides with the infinite singularity  $c$ ,

$$\lim_{\varphi \rightarrow c^\pm} F(\varphi, \lambda) = \infty \vee \lim_{\varphi \rightarrow c^\pm} F(\varphi, \lambda) = -\infty \vee \lim_{\varphi \rightarrow c^\pm} G(\varphi, \lambda) = \infty \vee \lim_{\varphi \rightarrow c^\pm} G(\varphi, \lambda) = -\infty,$$

or, if,

$$\lim_{\varphi \rightarrow c^\pm} F(\varphi, \lambda) = \pm\infty \vee \lim_{\varphi \rightarrow c^\pm} F(\varphi, \lambda) = \mp\infty \vee \lim_{\varphi \rightarrow c^\pm} G(\varphi, \lambda) = \pm\infty \vee \lim_{\varphi \rightarrow c^\pm} G(\varphi, \lambda) = \mp\infty.$$

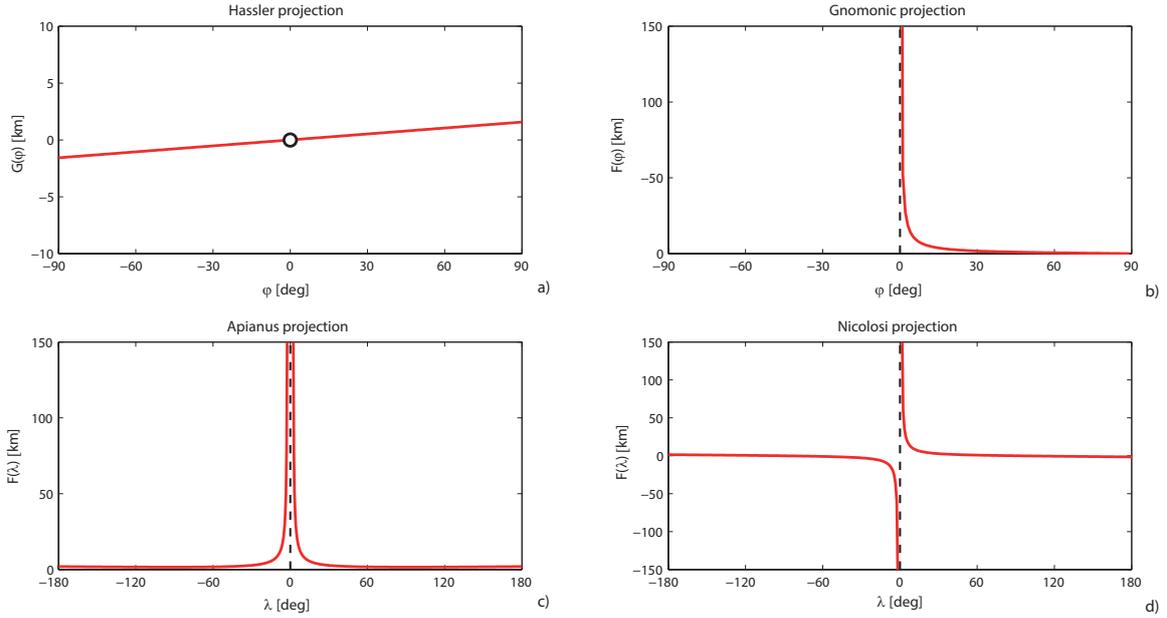


Figure 3.1: Discontinuities of the coordinate functions  $F, G$  and asymptotes: the removable discontinuity (Hassler projection), the infinite discontinuity (gnomonic projection) at  $\varphi = 0$ , the infinite discontinuity (Apianus and Nicolosi projections) at  $\lambda = 0$ .

For Case C, the originally projected parallel  $\tilde{p}(c)$  may be replaced with the average

$$\tilde{p}(c) = 0.5 [\tilde{p}(c - \varepsilon) + \tilde{p}(c + \varepsilon)], \quad (3.6)$$

for Case D the parallel  $p(\varphi)$  is replaced with two new parallels  $p^-(\varphi \rightarrow c^-)$ , and  $p^+(\varphi \rightarrow c^+)$ . The removable discontinuities may be treated analogously to Cases A, C, the jump discontinuities analogously to Cases B, D. An example of the coincidence with the discontinuities can be found in Fig. 3.2.

**Intersection of the singularity.** The singularity may not affect the entire meridian or parallel, but only some inferior points. The meridian  $m(\lambda)$  intersects the infinite singularity  $c$  at the point  $q_i = [\varphi_i = c, \lambda]$ , if

$$\lim_{\varphi_i \rightarrow c^\pm} F(\varphi_i, \lambda) = \infty \vee \lim_{\varphi_i \rightarrow c^+} F(\varphi_i, \lambda) = -\infty \vee \lim_{\varphi_i \rightarrow c^\pm} G(\varphi_i, \lambda) = \infty \vee \lim_{\varphi_i \rightarrow c^\pm} G(\varphi_i, \lambda) = -\infty,$$

(Case E), or, if

$$\lim_{\varphi_i \rightarrow c^\pm} F(\varphi_i, \lambda) = \pm\infty \vee \lim_{\varphi_i \rightarrow c^\pm} F(\varphi_i, \lambda) = \mp\infty \vee \lim_{\varphi_i \rightarrow c^\pm} G(\varphi_i, \lambda) = \pm\infty \vee \lim_{\varphi_i \rightarrow c^\pm} G(\varphi_i, \lambda) = \mp\infty,$$

representing Case F. For Case E, the projected meridian point  $q_i$  is replaced with the average

$$X_i = 0.5 (F(c - \varepsilon, \lambda) + F(c + \varepsilon, \lambda)), \quad Y_i = 0.5 (G(c - \varepsilon, \lambda) + G(c + \varepsilon, \lambda)).$$

Treating Case F is more difficult, the entire meridian  $m(\lambda)$  needs be split into two parts at  $c$

$$m(\lambda) = \left\langle \left\langle m^-(\varphi_i, \lambda), \varphi \leq \varphi_i < c \right\rangle, \left\langle m^+(\varphi_i, \lambda), c < \varphi_i \leq \bar{\varphi} \right\rangle \right\rangle,$$

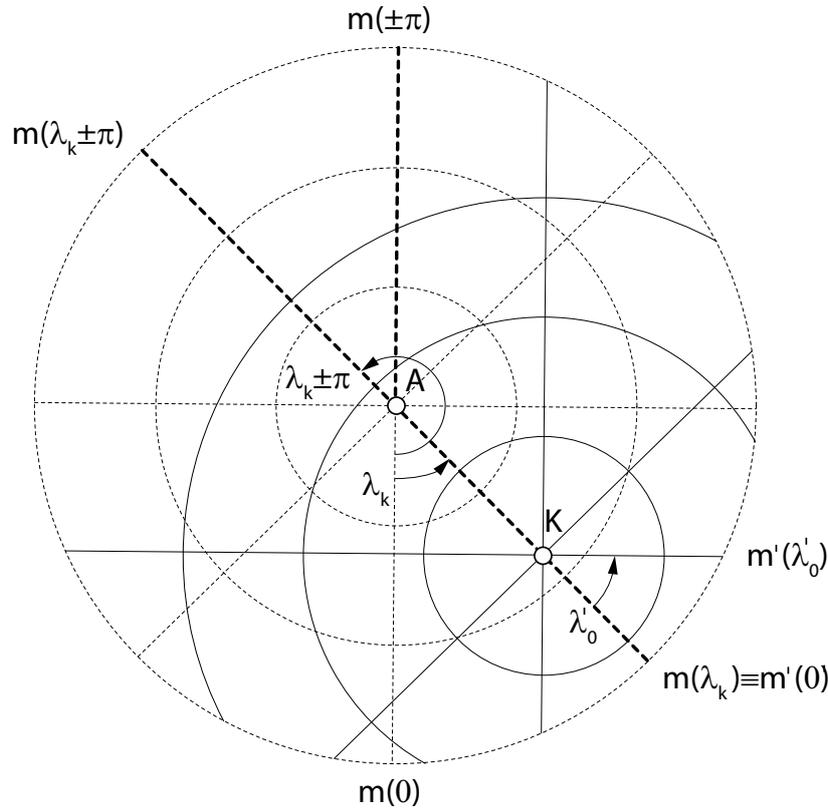


Figure 3.2: Coincidence with the singularities: the meridian  $m(\lambda_k)$  passing the transformed pole  $K$ , the meridian  $m(\lambda_k \pm \pi)$  “opposite” the transformed pole, and the meridian  $m(\pm\pi)$ ; the azimuthal projection.

projected separately. Analogously, Cases G, H occur, if a parallel  $p(\varphi)$  intersects the infinite singularity at the point  $q_i = [\varphi, c = \lambda_i]$ ,

$$\lim_{\lambda_i \rightarrow c\pm} F(\varphi, \lambda_i) = \infty \vee \lim_{\lambda_i \rightarrow c\pm} F(\varphi, \lambda_i) = -\infty \vee \lim_{\lambda_i \rightarrow c\pm} G(\varphi, \lambda_i) = \infty \vee \lim_{\lambda_i \rightarrow c\pm} G(\varphi, \lambda_i) = -\infty,$$

or, if,

$$\lim_{\lambda_i \rightarrow c\pm} F(\varphi, \lambda_i) = \pm\infty \vee \lim_{\lambda_i \rightarrow c\pm} F(\varphi, \lambda_i) = \mp\infty \vee \lim_{\lambda_i \rightarrow c\pm} G(\varphi, \lambda_i) = \pm\infty \vee \lim_{\lambda_i \rightarrow c\pm} G(\varphi, \lambda_i) = \mp\infty.$$

For Case G, the projected parallel point  $q_i$  is replaced with the average

$$X_i = 0.5(F(\varphi, c - \varepsilon) + F(\varphi, c + \varepsilon)), \quad Y_i = 0.5(G(\varphi, c - \varepsilon) + G(\varphi, c + \varepsilon)),$$

however, for Case H, the parallel  $p(\varphi)$  needs to be split into two parts at  $c$

$$p(\varphi) = \left\langle \left\langle p^-(\varphi, \lambda_i), \underline{\lambda} \leq \lambda_i < c \right\rangle, \left\langle p^+(\varphi, \lambda_i), c < \lambda_i \leq \bar{\lambda} \right\rangle \right\rangle.$$

The removable discontinuities may be treated analogously to Cases E, G, the jump discontinuities analogously to Cases F, H.

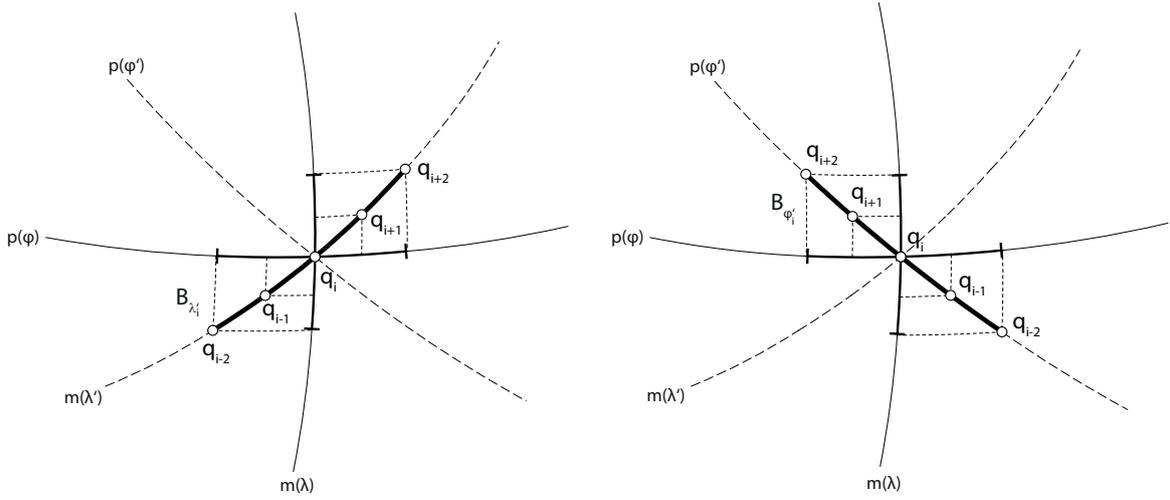


Figure 3.3: Boundaries  $B_{\varphi'_i}, B_{\lambda'_i}$  and their backward projections to the  $(\varphi, \lambda)$  directions.

**Detection of singularities.** For parametric functions  $F, G$  it is necessary to recognize whether a singularity refers to  $\varphi$  or  $\lambda$  coordinate. Therefore, each sampled point  $q_i = [\varphi_i, \lambda_i]$  and its boundary  $B(q_i, \varepsilon)$  will be checked for a discontinuity. Suppose  $B_{\varphi_i} = [\varphi_i - \varepsilon, \varphi_i + \varepsilon]$ ,  $B_{\lambda_i} = [\lambda_i - \varepsilon, \lambda_i + \varepsilon]$  to be the projections of  $B(q_i, \varepsilon)$  to the  $(\varphi, \lambda)$  directions and  $q_{\varphi_{i+l}} = [\varphi_i + lh, \lambda_i] \in B_{\varphi_i}$ ,  $q_{\lambda_{i+l}} = [\varphi_i, \lambda_i + lh] \in B_{\lambda_i}$ , where  $h = \varepsilon/2$ ,  $l = -2, -1, 0, 1, 2$ , to be 5 sampled points. If any point of  $B_{\varphi_i}$  is singular, the singularity has a direction of the meridian  $m(\lambda_i)$  and analogously for  $B_{\lambda_i}$ .

For the *oblique aspect* of  $\mathbb{P}$ ,  $B_{\varphi'_i} = [\varphi'_i - \varepsilon, \varphi'_i + \varepsilon]$ ,  $B_{\lambda'_i} = [\lambda'_i - \varepsilon, \lambda'_i + \varepsilon]$  represent the projections of  $B(q_i, \varepsilon)$  to the  $(\varphi', \lambda')$  directions and  $q_{\varphi'_{i+l}} = [\varphi'_i + lh, \lambda'_i] \in B_{\varphi'_i}$ ,  $q_{\lambda'_{i+l}} = [\varphi'_i, \lambda'_i + lh] \in B_{\lambda'_i}$  are the 5 sampled points. If all points of  $B_{\varphi'_i}$  are singular, the singularity has a direction of the meridian  $m(\lambda'_i)$  and analogously for  $B_{\lambda'_i}$ . While  $B_{\varphi_i}, B_{\lambda_i}$  refer to directions of the geographic meridian/parallel,  $B_{\varphi'_i}, B_{\lambda'_i}$  are aligned with the transformed meridian/parallel directions, see Fig. 3.3.

Depending on the direction, the coordinates  $x_i, y_i$  are labeled with  $\varphi, \lambda$ . The infinite discontinuities of coordinates  $x_{\varphi'_{i-k}}, y_{\varphi'_{i-k}}$  in the latitude/longitude directions are tested by comparing with the coordinate thresholds  $\bar{x}, \bar{y}$ . Subsequently, the jump discontinuity is detected using  $LR$  criteria

$$LR_x(\varphi'_i, \lambda'_i) = \frac{|F_r^2 - F_l^2|}{F_r^2 + F_l^2}, \quad LR_y(\varphi'_i, \lambda'_i) = \frac{|G_r^2 - G_l^2|}{G_r^2 + G_l^2},$$

where  $F_r = 3F_i - 4F_{i+1} + F_{i+2}$ ,  $F_l = 3F_i - 4F_{i-1} + F_{i-2}$ , and  $G_r = 3G_i - 4G_{i+1} + G_{i+2}$ ,  $G_l = 3G_i - 4G_{i-1} + G_{i-2}$ . If  $LR_x(\varphi'_i, \lambda'_i + lh) > \overline{LR} \vee LR_y(\varphi'_i, \lambda'_i + lh) > \overline{LR}$ ,  $l = -2, -1, 0, 1, 2$ , then,  $F$  or  $G$  (or both) are probably not smooth at  $q_i$ ; a discontinuity is aligned with the  $\lambda'$  direction. Otherwise, if  $LR_x(\varphi'_i + lh, \lambda'_i) > \overline{LR} \vee LR_y(\varphi'_i + lh, \lambda'_i) > \overline{LR}$ , a discontinuity is aligned with the  $\varphi'$  direction, where  $\overline{LR} = 0.8$  represents the threshold. If a discontinuity is detected, the backward projection of  $B_{\varphi'_i}, B_{\lambda'_i}$  to  $B_{\varphi_i}, B_{\lambda_i}$  is performed; see Fig. 3.3. In practice, only the internal points  $q_{\varphi'_{i+l}}$  of  $B_{\varphi'_i}$ , and  $q_{\lambda'_{i+l}}$  of  $B_{\lambda'_i}$  are converted to the normal aspect using Eqs. 3.4, 3.5 and their direction is checked. Subsequently, the meridian/parallel is shifted or split at  $q_i = [\varphi_i, \lambda_i]$ .

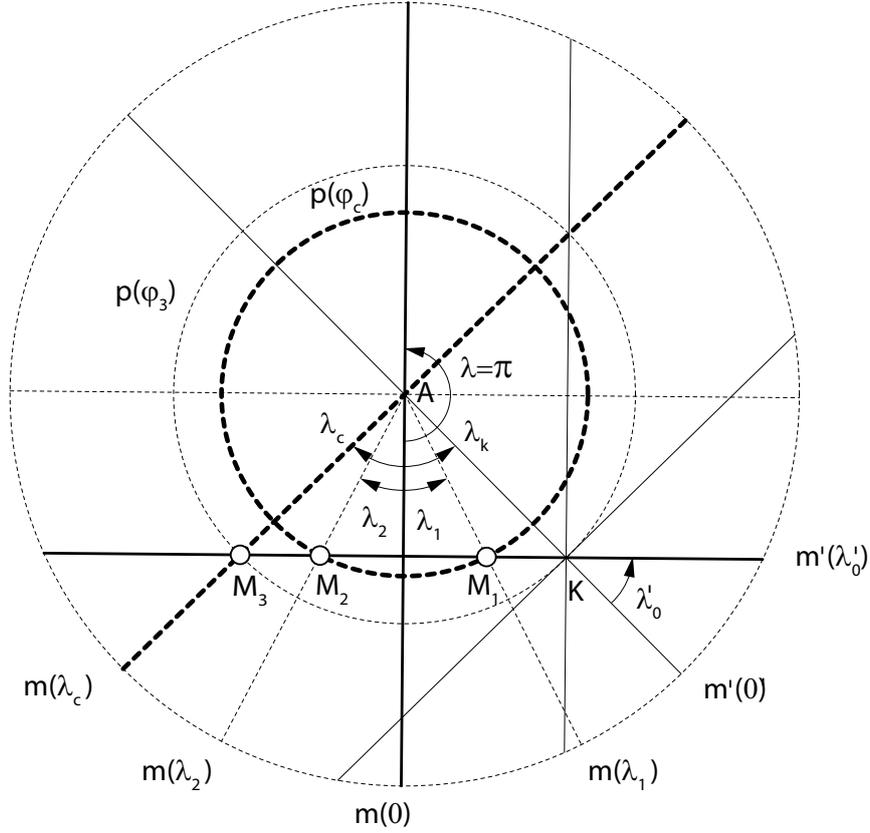


Figure 3.4: The intersections of the meridian  $m'(\lambda'_0)$ , parallel  $p(\varphi_c)$ , and the meridian  $m(\lambda_c)$  lead to the  $\varphi/\lambda$  interval partition into two/three subintervals;  $A$  represents the North Pole,  $K$  the transformed pole.

### 3.2. Intersection of the great circle and meridian/parallel arcs

Finding an intersection of the great circle arc and the meridian/parallel arc on the sphere represents an essential problem of the graticule construction algorithm. While  $p(\varphi), m(\lambda)$  refer to the normal aspect, commas in the labels indicate the oblique aspect.

The parallel  $p(\varphi_c)$  sampled on  $\Omega_\lambda = [-\pi, \pi]$  intersects  $m'(\lambda'_0)$  in at most two points  $M_1 = [\varphi_1, \lambda_1]$ ,  $M_2 = [\varphi_2, \lambda_2]$  where  $\lambda_1 \geq \lambda_2$ , and  $\varphi_1 = \varphi_2 = \varphi_c$ . The interval will be split into the three parts given by the longitude subintervals

$$\Omega_{\lambda,1} = [-\pi, \lambda_2), \quad \Omega_{\lambda,2} = (\lambda_2, \lambda_1), \quad \Omega_{\lambda,3} = (\lambda_1, \pi].$$

If  $\lambda'_0 = 0$ , then  $M_1 = [\varphi_c, \lambda_k]$ ,  $M_2 = [\varphi_c, \lambda_k \pm \pi]$ . Depending on  $\lambda_k$ , the longitude interval  $\Omega_\lambda$  will be split into three parts  $\Omega_{\lambda,1}, \Omega_{\lambda,2}, \Omega_{\lambda,3}$ . If  $\lambda_k \geq 0$

$$\Omega_{\lambda,1} = [-\pi, \lambda_k - \pi), \quad \Omega_{\lambda,2} = (\lambda_k - \pi, \lambda_k), \quad \Omega_{\lambda,3} = (\lambda_k, \pi],$$

otherwise,

$$\Omega_{\lambda,1} = [-\pi, \lambda_k), \quad \Omega_{\lambda,2} = (\lambda_k, \lambda_k + \pi), \quad \Omega_{\lambda,3} = (\lambda_k + \pi, \pi].$$

For  $\lambda'_0 = 0$  the meridians  $m'(0)$  and  $m(\lambda_c)$  have just two intersections at the poles:  $M_3 = [\frac{\pi}{2}, \cdot] \equiv A$ ,  $M_4 = [-\frac{\pi}{2}, \cdot] \equiv B$ ; the latitude interval  $\Omega_\varphi$  of the meridian  $m(\lambda_c)$  does not need to be split, see Fig. 3.4. If  $\lambda'_0 \neq 0$ , at most one intersection,  $M_3 = [\varphi_3, \lambda_c]$  exists. To avoid the singularity, the meridian  $m(\lambda_c)$  sampled in  $\Omega_\varphi = [-\frac{\pi}{2}, \frac{\pi}{2}]$  will be split into two parts given by the latitude subintervals

$$\Omega_{\varphi,1} = [-\frac{\pi}{2}, \varphi_3), \quad \Omega_{\varphi,2} = (\varphi_3, \frac{\pi}{2}].$$

**Intersection of two planes.** Suppose the planes  $\rho_1(n_1)$ ,  $\rho_2(n_2)$  given by the normal vectors  $n_1 = (n_{1,x}, n_{1,y}, n_{1,z})$ ,  $n_2 = (n_{2,x}, n_{2,y}, n_{2,z})$ , the points  $Q_1 \in \rho_1$ ,  $Q_2 \in \rho_2$  and an arbitrary point  $Q_0 = [x_0, y_0, z_0]$ . The intersection point  $M = [x_M, y_M, z_M]$  which lies on both planes  $\rho_1, \rho_2$  will be chosen to minimize the norm [14]

$$\min \|M - Q_0\|_2^2, \quad \text{subject to } (M - Q_1)n_1 = 0, \quad (M - Q_2)n_2 = 0.$$

It leads to the system of linear equations with the Lagrange multipliers  $\lambda_1, \lambda_2$  and the solution  $s = A^{-1}b$ , where

$$s = \begin{bmatrix} x_M \\ y_M \\ z_M \\ \lambda_1 \\ \lambda_2 \end{bmatrix}, \quad A = \begin{bmatrix} 2 & 0 & 0 & n_{1,x} & n_{2,x} \\ 0 & 2 & 0 & n_{1,y} & n_{2,y} \\ 0 & 0 & 2 & n_{1,z} & n_{2,z} \\ n_{1,x} & n_{1,y} & n_{1,z} & 0 & 0 \\ n_{2,x} & n_{2,y} & n_{2,z} & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 2x_0 \\ 2y_0 \\ 2z_0 \\ x_1n_{1,x} + y_1n_{1,y} + z_1n_{1,z} \\ x_2n_{2,x} + y_2n_{2,y} + z_2n_{2,z} \end{bmatrix}.$$

Because  $A$  is sparse and symmetric, its inversion can be found in the analytic form. Let us put

$$\begin{aligned} k_0 &= n_{1,x}^2, & k_1 &= n_{1,y}^2, & k_2 &= n_{1,z}^2, & k_3 &= n_{2,x}^2, & k_4 &= n_{2,y}^2, \\ k_5 &= n_{2,z}^2, & k_6 &= k_0 + k_1, & k_7 &= k_3 + k_4, & k_8 &= k_3 + k_5, & k_9 &= k_4 + k_5, \\ k_{10} &= n_{1,x}n_{2,x}, & k_{11} &= n_{1,x}n_{2,y}, & k_{12} &= n_{1,x}n_{2,z}, & k_{13} &= n_{1,y}n_{2,x}, & k_{14} &= n_{1,y}n_{2,y}, \\ k_{15} &= n_{1,y}n_{2,z}, & k_{16} &= n_{1,z}n_{2,x}, & k_{17} &= n_{1,z}n_{2,y}, & k_{18} &= n_{1,z}n_{2,z}, & k_{19} &= k_{15}n_{2,y}, \\ k_{20} &= k_{17}n_{2,z}, & k_{21} &= k_{15}n_{1,x}, & k_{22} &= k_{13}n_{1,y}, & k_{23} &= k_0 + k_1 + k_2, & k_{24} &= k_3 + k_4 + k_5, \end{aligned}$$

then

$$\det(A) = 2(k_0k_9 - 2k_{10}k_{12} + k_2k_7 - 2k_{11}(k_{10} + k_{12}) + k_1k_8),$$

and the elements of  $A^{-1}$  are

$$\begin{aligned} a_{11} &= \frac{(k_{18} - k_{16})(k_{18} - k_{16})}{\det(A)}, & a_{12} &= \frac{(k_{14} - k_{17})(k_{18} - k_{16})}{\det(A)}, & a_{13} &= \frac{(k_{13} - k_{15})(k_{16} - k_{18})}{\det(A)}, \\ a_{14} &= \frac{2(-k_{19} - k_{20} + n_{1,x}k_9)}{\det(A)}, & a_{15} &= \frac{2[n_{1,y}k_{15} - k_{22} + n_{1,z}(k_{17} - k_{14})]}{\det(A)}, & a_{22} &= \frac{(k_{17} - k_{14})(k_{17} - k_{14})}{\det(A)}, \\ a_{23} &= \frac{(k_{13} - k_{15})(k_{17} - k_{14})}{\det(A)}, & a_{24} &= \frac{2[n_{1,y}k_8 - n_{2,y}(k_{10} + k_{12})]}{\det(A)}, & a_{25} &= \frac{2[n_{1,x}k_{13} - k_{21} + n_{1,z}(k_{18} - k_{16})]}{\det(A)}, \\ a_{33} &= \frac{(k_{15} - k_{13})(k_{15} - k_{13})}{\det(A)}, & a_{34} &= \frac{2[n_{1,z}k_7 - n_{2,z}(k_{10} + k_{11})]}{\det(A)}, & a_{35} &= \frac{2[n_{2,z}k_6 - n_{1,z}(k_{10} + k_{11})]}{\det(A)}, \\ a_{44} &= \frac{-4k_{24}}{\det(A)}, & a_{45} &= \frac{4(k_{10} + k_{11} + k_{12})}{\det(A)}, & a_{55} &= \frac{-4k_{23}}{\det(A)}. \end{aligned}$$

The direction  $n$  of the line of intersection is given by the cross product  $n = n_1 \times n_2$ , its parametric equation is  $(x, y, z) = (x_M, y_M, z_M) + \alpha(n_x, n_y, n_z)$ , where  $\alpha \in [0, 1]$ .

**Intersection of the line and sphere.** It is necessary to test, whether the line of intersection intersects a sphere  $S^2$  of the radius  $R$  with the center  $C = [x_c, y_c, z_c]$ , given by  $(x_M + \alpha n_x - x_c)^2 + (y_M + \alpha n_y - y_c)^2 + (z_M + \alpha n_z - z_c)^2 = R^2$ . It leads to the quadratic equation for  $\alpha$ , where

$$\begin{aligned} a &= n_x^2 + n_y^2 + n_z^2, \\ b &= 2(x_M n_x - n_x x_c + y_M n_y - n_y y_c + z_M n_z - z_M z_c), \\ c &= (x_m - x_c)^2 + (y_m - y_c)^2 + (z_m - z_c)^2 - R^2. \end{aligned}$$

For the discriminant  $D > 0$ , there are two intersections  $M_1 = [x_1, y_1, z_1]$ , and  $M_2 = [x_2, y_2, z_2]$ , determined from  $(x_{1,2}, y_{1,2}, z_{1,2}) = (x_M, y_M, z_M) + \alpha_{1,2}(n_x, n_y, n_z)$ . If  $D = 0$ , then  $M_1 \equiv M_2$ ; for  $D < 0$  the line does not intersect the sphere. The Cartesian coordinates of intersections need to be converted to the spherical, where  $\varphi_{1,2} = \arcsin(z_{1,2}/\sqrt{x_{1,2}^2 + y_{1,2}^2 + z_{1,2}^2})$ ,  $\lambda_{1,2} = \text{atan2}(y_{1,2}, x_{1,2})$ .

**Practical computation.** The meridian plane  $\varrho_1$  containing  $m(\lambda_c)$  passes through the points  $P_5 = [\varphi_5, \lambda_c]$ ,  $P_6 = [\varphi_6, \lambda_c]$ , and  $P_7 = [\varphi_7, \lambda_c]$ , where  $\varphi_5 = -\frac{\pi}{2} + \varepsilon$ ,  $\varphi_6 = 0$ ,  $\varphi_7 = \frac{\pi}{2} - \varepsilon$ . Analogously, the parallel plane  $\varrho_2$  containing  $p(\varphi_c, \lambda)$  is given by the points  $P_8 = [\varphi_c, \lambda_8]$ ,  $P_9 = [\varphi_c, \lambda_9]$ , and  $P_{10} = [\varphi_c, \lambda_{10}]$ , where  $\lambda_8 = -\pi + \varepsilon$ ,  $\lambda_9 = 0$ ,  $\lambda_{10} = \pi - \varepsilon$ , and the meridian plane  $\varrho_3$  containing  $m'(\varphi', \lambda'_0)$  centered at  $\lambda'_0$  is given by three points  $P_{11} = [\varphi'_{11}, \lambda'_0]$ ,  $P_{12} = [\varphi'_{12}, \lambda'_0]$ , and  $P_{13} = [\varphi'_{13}, \lambda'_0]$ , where  $\varphi'_{11} = -\frac{\pi}{2} + \varepsilon$ ,  $\varphi'_{12} = 0$ ,  $\varphi'_{13} = \frac{\pi}{2} - \varepsilon$ .

1. *The inverse transformation*

Initially, the spherical coordinates  $[\varphi'_i, \lambda'_i]$  of points  $P_{11}, P_{12}, P_{13}$  related to the pole  $K = [\varphi_k, \lambda_k]$  are transformed to the normal aspect using Eqs. 3.4, 3.5, where  $i = 11, 12, 13$ .

2. *The transformation to the geocentric coordinates*

The spherical coordinates  $[\varphi_i, \lambda_i]$  of points  $P_5 - P_{13}$  are transformed to the Cartesian geocentric coordinates  $x_i, y_i, z_i$ , where  $x_i = \cos \varphi_i \cos \lambda_i$ ,  $y_i = \cos \varphi_i \sin \lambda_i$ ,  $z_i = \sin \varphi_i$ ,  $i = 5, \dots, 13$ .

3. *Find the analytic representation of planes*

The meridian plane of  $m'(\lambda'_0)$  is given by the vectors  $u_1, v_1$ , where  $u_1 = P_{12} - P_{11}$ ,  $v_1 = P_{13} - P_{11}$ . The vectors  $u_2, v_2$  of the second meridian plane  $m(\lambda_c)$  are  $u_2 = P_6 - P_5$ ,  $v_2 = P_7 - P_5$ , and the vectors  $u_3, v_3$  of the parallel plane  $p(\varphi_c)$  are  $u_3 = P_9 - P_8$ ,  $v_3 = P_{10} - P_8$ . Find the vectors  $n_1, n_2, n_3$  perpendicular to  $u_1, v_1$ ,  $u_2, v_2$ , and  $u_3, v_3$  using the cross products  $n_1 = u_1 \times v_1$ ,  $n_2 = u_2 \times v_2$ ,  $n_3 = u_3 \times v_3$ .

4. *Compute the line of intersection*

For the meridian planes  $m'(\lambda'_0)$ , and  $m(\lambda_c)$ , set:  $Q_1 \equiv P_{11}$ ,  $Q_2 \equiv P_5$ ; for the meridian  $m'(\lambda'_0)$  and parallel  $p(\varphi_c)$  planes, set:  $Q_1 \equiv P_{11}$ ,  $Q_2 \equiv P_8$ , compute  $Q_0 = (P_{11} + P_{12} + P_{13})/3$ . Evaluate the line of intersection, compute the cross products  $n_{1,2} = n_1 \times n_2$ ,  $n_{1,3} = n_1 \times n_3$ , where the direction  $n$  of both lines of intersection is  $n \equiv n_{1,2}$ , or  $n \equiv n_{1,3}$ .

5. *Intersection of the line and sphere*

Let  $S^2$  be the sphere of the radius  $R = 1$  centered at  $C = [0, 0, 0]$ . Depending on the sign of the discriminant  $D$ , evaluate the intersections  $M_1 = [x_1, y_1, z_1]$ ,  $M_2 = [x_2, y_2, z_2]$  of the

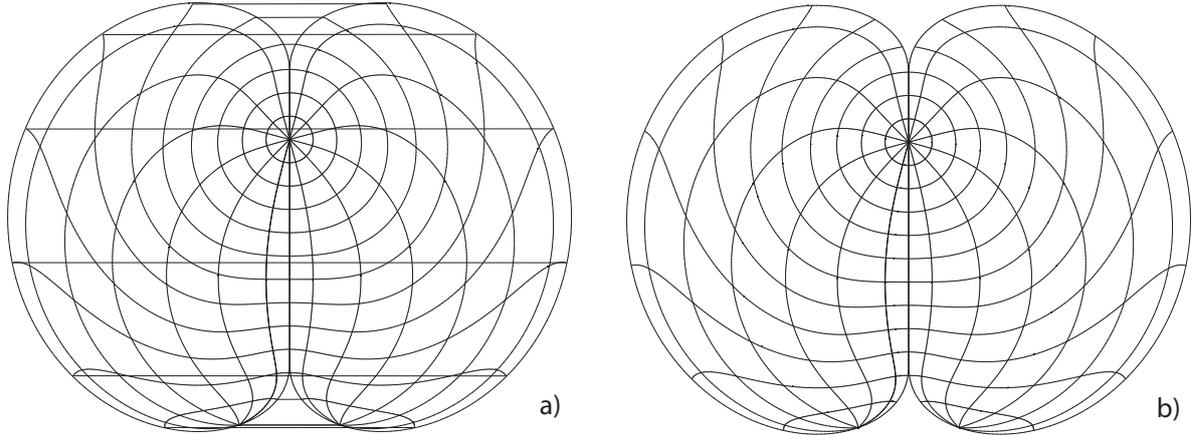


Figure 4.1: The Hassler projection in the oblique aspect,  $K = [50^\circ, 15^\circ]$ ; the singularities involved a) and treated b).

planes  $\varrho_2$  and  $\varrho_3$ . The planes  $\varrho_1$ ,  $\varrho_3$  have two intersections denoted as  $M_3 = [x_3, y_3, z_3]$  and  $M_4 = [x_4, y_4, z_4]$ .

#### 6. Transformation to spherical coordinates

Finally, the Cartesian coordinates of the intersections  $M_1$ ,  $M_2$ ,  $M_3$  will be converted to the spherical.

### 4. Graticule reconstruction using the combined sampling

Let  $\Omega = \Omega_\varphi \times \Omega_\lambda$ , where  $\Omega_\varphi = [\varphi, \bar{\varphi}]$ , and  $\Omega_\lambda = [\lambda, \bar{\lambda}]$ , be the subdomain, inside which the polygonal approximation of the meridians/parallels is constructed. The points  $p_i = (F(\varphi_i, \lambda_c), G(\varphi_i, \lambda_c))$ ,  $1 \leq i \leq n_\varphi$ ,  $\varphi_i \in \Omega_\varphi$  approximate the meridian  $m(\lambda_c)$ , and analogously for the parallel. The combined sampling procedure represents a generalization of the curve sampling procedure discussed in [5]. It starts with the uniform sampling procedure, which after  $\underline{d}$  steps transforms to adaptive sampling. A refinement criterion is based on the angular difference  $\alpha_i$  of segments formed by three adjacent points  $(p_{i-1}, p_i, p_{i+1})$ .

#### 4.1. Graticule construction with singularities

The proposed algorithm based on the stack  $\mathcal{S}$  implementation is described in Alg. 1. The basic idea is to find a set of disjoint subsets  $\Omega_\varphi^g, \Omega_\varphi^g \subseteq \Omega_\varphi$ , and  $\Omega_\lambda^g, \Omega_\lambda^g \subseteq \Omega_\lambda$ , where  $\Omega^g = \Omega_\varphi^g \times \Omega_\lambda^g$ , containing “good” data without singularities that allow for adaptive sampling. Over each  $\Omega_j^g$ , the borders of which run along discontinuities the graticule fragment is constructed. The entire graticule is put together from these fragments.

The point  $q_i = [\varphi_i, \lambda_i]$  is “good” if no singularity at  $F(q_i)$  and  $G(q_i)$  occurs. Unlike the one-dimensional version of the problem,  $\Omega$  is partitioned in two orthogonal directions  $(\varphi, \lambda)$ . Let the  $j$ -th interval  $\Omega_j = \Omega_{\varphi,j} \times \Omega_{\lambda,j}$ , where  $\Omega_{\varphi,j} = [\varphi_j, \bar{\varphi}_j]$ ,  $\Omega_{\lambda,j} = [\lambda_j, \bar{\lambda}_j]$ , contain a singularity  $c$ ,  $c \in \Omega$ , with the unknown direction stored in the stack  $\mathcal{S}$ , and  $\varepsilon$ ,  $\varepsilon > 0$ , be the numerical threshold. Suppose  $\Delta\varphi$  to be the latitude offset between parallels and  $\Delta\lambda$  to be the longitude offset between meridians.

---

**Algorithm 1** Combined sampling with the singularities, the stack implementation.

---

```

1: function GRATICULE( $L_\varphi, L_\lambda, \Omega_\varphi, \Omega_\lambda, \Delta\varphi, \Delta\lambda, \mathbb{P}, \bar{s}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
2:    $S = \emptyset, s = 0$ 
3:    $S \leftarrow \Omega_j = \Omega_\varphi \times \Omega_\lambda$ 
4:   while  $S \neq \emptyset$  do
5:      $\Omega_j = \Omega_{\varphi,j} \times \Omega_{\lambda,j} = S.pop()$ 
6:      $\Omega_{j,1} = \Omega_{\varphi,j,1} \times \Omega_{\lambda,j,1} = \Omega_j, \Omega_{j,2} = \Omega_{\varphi,j,2} \times \Omega_{\lambda,j,2} = \Omega_j$ 
7:      $k = 0$ 
8:     try
9:        $L_{\varphi,j} = \emptyset, L_{\lambda,j} = \emptyset$ 
10:      Meridians( $L_{\lambda,j}, \Omega_{\varphi,j}, \Omega_{\lambda,j}, \Delta\lambda, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
11:      Parallels( $L_{\varphi,j}, \Omega_{\varphi,j}, \Omega_{\lambda,j}, \Delta\varphi, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
12:       $L_\varphi \leftarrow L_{\varphi,j}, L_\lambda \leftarrow L_{\lambda,j}$ 
13:    catch (SingularityLatError e)
14:      if ( $s < \bar{s}$ ) then
15:        processInt( $\Omega_{\varphi,j}, e.\varphi_i, \Omega_{\varphi,j,1}, \Omega_{\varphi,j,2}, k, s, \varepsilon$ )
16:      catch (SingularityLonError e)
17:        if ( $s < \bar{s}$ ) then
18:          processInt( $\Omega_{\lambda,j}, e.\lambda_i, \Omega_{\lambda,j,1}, \Omega_{\lambda,j,2}, k, s, \varepsilon$ )
19:      catch (SingularityError e)
20:        if ( $s > \bar{s}$ ) then
21:           $L_\varphi = \emptyset, L_\lambda = \emptyset$ 
22:          return
23:        else
24:          if ( $k > 0$ ) then
25:             $S \leftarrow \Omega_{j,1}$ 
26:          else if ( $k > 1$ ) then
27:             $S \leftarrow \Omega_{j,2}$ 

```

---

Initially,  $\Omega^g \equiv \Omega$  is set, so  $\Omega_\varphi^g \equiv \Omega_\varphi$ ,  $\Omega_\lambda^g \equiv \Omega_\lambda$ . All adaptively sampled meridian/parallel points  $q_i$  are checked for the discontinuities. If a discontinuity  $c$  is found, it is classified, and its direction is determined. Depending on  $c$  value, the lower/upper bound of  $\Omega_{\varphi,j}$  or  $\Omega_{\lambda,j}$  is shifted, or, a split to two disjoint intervals is performed:

$$\begin{aligned} \Omega_{\varphi,j} &= \langle \Omega_{\varphi,j,1}, \Omega_{\varphi,j,2} \rangle, & \Omega_{\varphi,j,1} &= [\underline{\varphi}_j, c - \varepsilon], \Omega_{\varphi,j,2} = [c + \varepsilon, \bar{\varphi}_j], \\ \Omega_{\lambda,j} &= \langle \Omega_{\lambda,j,1}, \Omega_{\lambda,j,2} \rangle, & \Omega_{\lambda,j,1} &= [\underline{\lambda}_j, c - \varepsilon], \Omega_{\lambda,j,2} = [c + \varepsilon, \bar{\lambda}_j]. \end{aligned}$$

Otherwise, the polygonal approximation  $L_{\lambda,j}$  of the meridian  $m(\lambda)$  or  $L_{\varphi,j}$  of the parallel  $p(\varphi)$  is constructed. All disjoint polygonal approximations  $L_j$  are stored in the lists of meridians  $L_\lambda$  and parallels  $L_\varphi$ . Recall  $s$  to be the amount of  $\Omega_{\varphi,j}$ , and  $\Omega_{\lambda,j}$  splits. The procedure can be summarized as follows:

### 1. *The initial phase*

Initialize the empty stack  $S = \emptyset$ ,  $\Omega^g = \Omega$  and push  $S \leftarrow \Omega$ . Set the number of splits to  $s = 0$ .

**Algorithm 2** The construction of meridians.

---

```

1: function MERIDIANS( $L_{\lambda,j}, \Omega_{\varphi,j}, \Omega_{\lambda,j}, \Delta\lambda, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
2:    $\lambda_{j,start} = \Delta\lambda[1 + \text{floor}(\frac{\lambda_j}{\Delta\lambda})]$ 
3:    $\lambda_{j,end} = \Delta\lambda[\text{ceil}(\frac{\bar{\lambda}_j}{\Delta\lambda}) - 1]$ 
4:   meridianPart( $L_{\lambda,j}, \Omega_{\varphi,j}, \lambda_{j,start}, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
5:   for ( $\lambda = \lambda_{j,start}; \lambda \leq \lambda_{j,end}; \lambda = \lambda + \Delta\lambda$ ) do
6:     meridianPart( $L_{\lambda,j}, \Omega_{\varphi,j}, \lambda, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
7:   if ( $\lambda_k \in \Omega_{\lambda,j}$ ) then
8:     meridianPart( $L_{\lambda,j}, \Omega_{\varphi,j}, \lambda - \varepsilon, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
9:     meridianPart( $L_{\lambda,j}, \Omega_{\varphi,j}, \lambda + \varepsilon, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
10:   $\lambda_{k,o} = (\lambda_k > 0 ? \lambda_k - \pi : \lambda_k + \pi)$ 
11:  meridianPart( $L_{\lambda,j}, \Omega_{\varphi,j}, \lambda_{k,o} - \varepsilon, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
12:  meridianPart( $L_{\lambda,j}, \Omega_{\varphi,j}, \lambda_{k,o} + \varepsilon, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
13:  meridianPart( $L_{\lambda,j}, \Omega_{\varphi,j}, \lambda_{j,end}, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )

```

---

## 2. Recursive steps

Repeat the following steps until  $\mathbf{S}$  is empty:

(a) *Pop the actual interval  $\Omega_j$*

Pop the actual good interval  $\Omega_j \leftarrow S$  from  $\mathbf{S}$ , where  $\Omega_j^g = \Omega_{\varphi,j} \times \Omega_{\lambda,j}$ , and  $\Omega_{\varphi,j} = [\underline{\varphi}_j, \bar{\varphi}_j]$ ,  $\Omega_{\lambda,j} = [\underline{\lambda}_j, \bar{\lambda}_j]$ .

(b) *Create the empty lists*

Create new empty lists  $L_{\lambda,j} = \emptyset$ ,  $L_{\varphi,j} = \emptyset$  storing the polygonal approximation of the meridians and parallels.

(c) *Create the intervals for feasible splitting*

Initialize the newly created intervals  $\Omega_{j,1} = \Omega_{\varphi,j,1} \times \Omega_{\lambda,j,1}$ , and  $\Omega_{j,2} = \Omega_{\varphi,j,2} \times \Omega_{\lambda,j,2}$ , as  $\Omega_{j,1} = \Omega_j$ ,  $\Omega_{j,2} = \Omega_j$ , where  $\Omega_{\varphi,j,1} = [\underline{\varphi}_{j,1}, \bar{\varphi}_{j,1}]$ ,  $\Omega_{\lambda,j,1} = [\underline{\lambda}_{j,1}, \bar{\lambda}_{j,1}]$ , and  $\Omega_{\varphi,j,2} = [\underline{\varphi}_{j,2}, \bar{\varphi}_{j,2}]$ ,  $\Omega_{\lambda,j,2} = [\underline{\lambda}_{j,2}, \bar{\lambda}_{j,2}]$  used for feasible splitting of  $\Omega_j^g$ . Set the amount of newly created intervals as  $k = 0$ .

(d) *Create the temporary polygonal approximation  $L_{\varphi,j}, L_{\lambda,j}$  of the graticule*

Create the temporary polygonal approximation of the meridians and parallels on  $\Omega_j^g$  using adaptive sampling stored in  $L_{\varphi,j}, L_{\lambda,j}$ .

(e) *Copy the temporary polygonal approximation*

If no discontinuity appears, add  $L_{\lambda,j}$  to  $\mathbf{L}_\lambda$ :  $\mathbf{L}_\lambda \leftarrow L_{\lambda,j}$ , and  $L_{\varphi,j}$  to  $\mathbf{L}_\varphi$ :  $\mathbf{L}_\varphi \leftarrow L_{\varphi,j}$ , and go to step (a). Otherwise,  $c$  represents the discontinuity of the given type and direction detected. The following cases, when  $c = \varphi_i$ , or,  $c = \lambda_i$ , must be treated in Steps i-iii).

(f) *Resolve the singularities by splitting  $\Omega_{\varphi,j}, \Omega_{\lambda,j}$*

Depending on the singularity  $c$  direction, do the following steps:

- i. If  $c = \varphi_i$ , then  $\Omega_{\varphi,j}$  needs to be split. Call the function `processInt()` with parameters:  $\Omega_{\varphi,j}$ ,  $c$ ,  $\Omega_{\varphi,j,1}$ ,  $\Omega_{\varphi,j,2}$ ,  $k$ ,  $\mathbf{S}$ ,  $\varepsilon$ . The subintervals  $\Omega_{\varphi,j,1}$ ,  $\Omega_{\varphi,j,2}$  are passing by reference.
- ii. If  $c = \lambda_i$ , then  $\Omega_{\lambda,j}$  needs to be split. Call the function `processInt()` with parameters:  $\Omega_{\lambda,j}$ ,  $c$ ,  $\Omega_{\lambda,j,1}$ ,  $\Omega_{\lambda,j,2}$ ,  $k$ ,  $\mathbf{S}$ ,  $\varepsilon$ . The subintervals  $\Omega_{\lambda,j,1}$ ,  $\Omega_{\lambda,j,2}$  are passing by reference.
- iii. If  $c = \varphi_i \vee c = \lambda_i$ , do the following steps. If  $s > \bar{s}$ , the maximum allowed recursion depth is exceeded without a reasonable solution, clear the polygonal approximations  $\mathbf{L}_\varphi$ ,  $\mathbf{L}_\lambda$ . Otherwise, if  $k > 0$ , at least one new interval needs to be created; push  $\Omega_{j,1}$  to the stack:  $\mathbf{S} \leftarrow \Omega_{j,1}$ . If  $k > 1$ , push the second interval  $\Omega_{j,2}$  to the stack:  $\mathbf{S} \leftarrow \Omega_{j,2}$ .

For the implementation, see Alg. 1. Its design is robust against common numerical failures. The singularity value  $c$  is stored in the thrown exceptions `SingularityLatError`, `SingularityLonError`, which are derived from the base class `SingularityError`. The splitting procedure is realized in accordance with the singularity direction.

**Generate meridians.** The procedure creates all meridians inside the given interval  $\Omega_j = \Omega_{\varphi,j} \times \Omega_{\lambda,j}$ , where  $\Omega_{\varphi,j} = [\underline{\varphi}_j, \overline{\varphi}_j]$ ,  $\Omega_{\lambda,j} = [\underline{\lambda}_j, \overline{\lambda}_j]$ . Initially, the bounding meridians  $m(\underline{\lambda}_j)$ , and  $m(\overline{\lambda}_j)$  are created. To find the longitude  $\lambda_{start}$  of the first meridian  $m(\lambda_{start})$ , which is the smallest multiplier of  $\Delta\lambda$  higher than  $\underline{\lambda}_j$ , and the last meridian  $m(\lambda_{end})$ , which is the largest multiplier of  $\Delta\lambda$  lower than  $\overline{\lambda}_j$ , use

$$\lambda_{j,start} = \Delta\lambda[1 + \text{floor}(\frac{\underline{\lambda}_j}{\Delta\lambda})], \quad \lambda_{j,end} = \Delta\lambda[\text{ceil}(\frac{\overline{\lambda}_j}{\Delta\lambda}) - 1].$$

Other meridians  $m(\lambda_{j,i})$  are generated inside the interval  $[\lambda_{start}, \lambda_{end}]$  so that  $\lambda_j = \lambda_{j,start} + i\Delta\lambda$ ,  $\lambda_{start} < \lambda < \lambda_{end}$ . Due to possible discontinuities, the meridian consists of several parts. If a discontinuity at  $c = \varphi_j$  is found, the meridian needs to be split at  $c$ . If a discontinuity at  $c = \lambda_j$  is detected, a meridian needs to be shifted. This is achieved by splitting  $\Omega_j$  at  $c = \varphi_j$ , or, at  $c = \lambda_j$  with the additional shift  $\varepsilon$ . Otherwise, a current meridian  $m(\lambda_j)$  is constructed. Finally, the meridians passing along the pole  $m(\lambda_k \pm \varepsilon)$  and the opposite meridians  $m(\lambda_k \pm \pi \mp \varepsilon)$ , for  $\lambda_k \leq 0$  are added. The procedure is summarized in Alg. 2.

**Generate parallels.** The procedure generates all parallels inside the interval  $\Omega_j = \Omega_{\varphi,j} \times \Omega_{\lambda,j}$ , where  $\Omega_{\varphi,j} = [\underline{\varphi}_j, \overline{\varphi}_j]$ ,  $\Omega_{\lambda,j} = [\underline{\lambda}_j, \overline{\lambda}_j]$ . Initially, the bounding parallels  $p(\underline{\varphi}_j)$ , and  $p(\overline{\varphi}_j)$  are created. The latitude  $\varphi_{start}$  of the first parallel  $p(\varphi_{start})$  is the smallest multiplier of  $\Delta\varphi$  higher than  $\underline{\varphi}_j$ , the last parallel  $p(\varphi_{end})$  is the largest multiplier of  $\Delta\varphi$  lower than  $\overline{\varphi}_j$ . Other parallels  $p(\varphi_{j,i})$  are generated inside the interval  $[\varphi_{start}, \varphi_{end}]$  so that  $\varphi_j = \varphi_{start} + i\Delta\varphi$ ,  $\varphi_{start} < \varphi < \varphi_{end}$ . If a discontinuity at  $c = \lambda_j$  is found, the parallel needs to be split at  $c$ . If a discontinuity at  $c = \varphi_j$  is detected, the parallel needs to be shifted. This is achieved by splitting  $\Omega_j$  at  $c = \varphi_j$ , or, at  $c = \lambda_j$ . Otherwise, a current parallel  $p(\varphi_j)$  is constructed. The procedure is summarized in Alg. 3.

**Generate a meridian.** Combined sampling provides a polynomial approximation of the meridian  $m(\lambda)$  by the refinement  $\bar{\alpha}$  and the recursion depth  $d$  over the interval  $\Omega_{\varphi,j} = [\underline{\varphi}_j, \overline{\varphi}_j]$ .

---

**Algorithm 3** The construction of parallels.

---

```

1: function PARALLELS( $L_{\lambda,j}, \Omega_{\varphi,j}, \Omega_{\lambda,j}, \Delta\varphi, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
2:    $\varphi_{j,start} = \Delta\varphi[1 + \text{floor}(\frac{\varphi_j}{\Delta\varphi})]$ 
3:    $\varphi_{j,end} = \Delta\varphi[\text{ceil}(\frac{\bar{\varphi}_j}{\Delta\varphi}) - 1]$ 
4:   parallelPart( $L_{\varphi,j}, \Omega_{\varphi,j}, \varphi_{j,start}, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
5:   for ( $\varphi = \varphi_{j,start}; \varphi \leq \varphi_{j,end}; \varphi = \varphi + \Delta\varphi$ ) do
6:     parallelPart( $L_{\varphi,j}, \Omega_{\varphi,j}, \varphi, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
7:   parallelPart( $L_{\varphi,j}, \Omega_{\varphi,j}, \varphi_{j,end}, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )

```

---

Suppose  $m'(\lambda'_0)$  to be the central meridian. The polygonal approximation consists of two steps:

1. *The partition of  $m(\lambda)$  to the fragments.*

Depending on the position of the pole  $K = [\varphi_k, \lambda_k]$  and the central meridian  $m'(\lambda'_0)$ , the original interval  $\Omega_{\varphi,j}$  of the meridian  $m(\lambda)$  needs to be split along the intersection point  $M_3 = [\varphi_3, \lambda]$ . If  $\lambda'_0 = 0$  the meridians  $m'(0)$  and  $m(\lambda)$  have just two intersections  $M_3 = [\pi/2, \cdot] \equiv A$ ,  $M_4 = [-\pi/2, \cdot] \equiv B$  at poles  $A, B$  (this case is not important for the algorithm). Otherwise, the solution  $M_3 = [\varphi_3, \lambda_3]$  found from the intersection of two planes (see Sec. 3.2) holds

$$\varphi_3 \in [\underline{\varphi}_j + \varepsilon, \bar{\varphi}_j - \varepsilon] \wedge (|\lambda_3 - \lambda| < \varepsilon).$$

If an intersection  $M_3$  is found, the splitting procedure

$$\Omega_{\varphi,j,1} = [\underline{\varphi}_j, \varphi_3 - \varepsilon], \quad \Omega_{\varphi,j,2} = [\varphi_3 + \varepsilon, \bar{\varphi}_j],$$

is undertaken; see Fig. 3.4 and Alg. 4.

2. *A combined sampling of the fragments.*

The sampling methods will be discussed in Sec. 4.2.

**Generate a parallel.** The parallel  $p(\varphi)$  is intersected by the meridian  $m'(\lambda'_0)$  in at most two points. The original interval  $\Omega_{\lambda,j}$  needs to be split along the intersection points  $M_1 = [\varphi_1, \lambda_1]$ ,  $M_2 = [\varphi_2, \lambda_2]$ ; the solution is found from the intersection of two planes (see Sec. 3.2). In general, the following cases are solved:

- If  $M_1 \in \Omega_{\lambda,j}$ , then  $\lambda_1 \in [\underline{\lambda}_j + \varepsilon, \bar{\lambda}_j - \varepsilon]$  and  $\Omega_{\lambda,j}$  is split to

$$\Omega_{\lambda,j,1} = [\underline{\lambda}_j, \lambda_1 - \varepsilon], \Omega_{\lambda,j,2} = [\lambda_1 + \varepsilon, \bar{\lambda}_j].$$

- If  $M_2 \in \Omega_{\lambda,j}$ , then  $\lambda_2 \in [\underline{\lambda}_j + \varepsilon, \bar{\lambda}_j - \varepsilon]$  and  $\Omega_{\lambda,j}$  is split to

$$\Omega_{\lambda,j,1} = [\underline{\lambda}_j, \lambda_2 - \varepsilon], \Omega_{\lambda,j,2} = [\lambda_2 + \varepsilon, \bar{\lambda}_j].$$

- If  $M_1 \in \Omega_{\lambda,j} \wedge M_2 \in \Omega_{\lambda,j}$ , then  $\Omega_{\lambda,j}$  is split to

$$\Omega_{\lambda,j,1} = [\underline{\lambda}_j, \lambda_1 - \varepsilon], \Omega_{\lambda,j,2} = [\lambda_1 + \varepsilon, \lambda_2 - \varepsilon], \Omega_{\lambda,j,3} = [\lambda_2 + \varepsilon, \bar{\lambda}_j].$$

For practical computation, the coordinates are sorted so that  $\lambda_1 \leq \lambda_2$ ; see Alg. 5.

---

**Algorithm 4** The construction of a meridian from the fragments.

---

```

1: function MERIDIANPART( $L_{\lambda,j,k}, \Omega_{\varphi,j}, \lambda, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
2:    $\langle M_3^1, M_3^2 \rangle = m(\lambda) \times m'(\varphi', \lambda'_0)$ 
3:    $L_{\lambda,j,1} = \emptyset, L_{\lambda,j,2} = \emptyset$ 
4:   if  $(\varphi_3^1 > \underline{\varphi}_j + \varepsilon) \wedge (\varphi_3^1 < \bar{\varphi}_j - \varepsilon) \wedge (|\lambda_3^1 - \lambda| < \varepsilon)$  then
5:      $\bar{\varphi}_{j,1} = \varphi_3^1 - \varepsilon, \underline{\varphi}_{j,2} = \varphi_3^1 + \varepsilon$ 
6:     csMerInit( $\mathbb{P}, L_{\lambda,j,1}, \underline{\varphi}_{j,2}, \bar{\varphi}_{j,1}, \lambda, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
7:      $L_{\lambda,j} \leftarrow L_{\lambda,j,1}$ 
8:     csMerInit( $\mathbb{P}, L_{\lambda,j,2}, \underline{\varphi}_{j,2}, \bar{\varphi}_j, \lambda, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
9:      $L_{\lambda,j} \leftarrow L_{\lambda,j,2}$ 
10:  else if  $(\varphi_3^2 > \underline{\varphi}_j + \varepsilon) \wedge (\varphi_3^2 < \bar{\varphi}_j - \varepsilon) \wedge (|\lambda_3^2 - \lambda| < \varepsilon)$  then
11:     $\bar{\varphi}_{j,1} = \varphi_3^2 - \varepsilon, \underline{\varphi}_{j,2} = \varphi_3^2 + \varepsilon$ 
12:    csMerInit( $\mathbb{P}, L_{\lambda,j,1}, \underline{\varphi}_j, \bar{\varphi}_{j,1}, \lambda, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
13:     $L_{\lambda,j} \leftarrow L_{\lambda,j,1}$ 
14:    csMerInit( $\mathbb{P}, L_{\lambda,j,2}, \underline{\varphi}_{j,2}, \bar{\varphi}_j, \lambda, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
15:     $L_{\lambda,j} \leftarrow L_{\lambda,j,2}$ 

```

---

#### 4.2. Combined sampling of meridian and parallel fragments

Three sampling techniques denoted as S1, S2, S3 representing a natural extension of the sampling methods described in [5] will be illustrated on the meridian/parallel approximation.

**Sampling method S1.** It provides a polynomial approximation of the meridian  $m(\lambda)$  by the refinement criteria  $\bar{\alpha}$  and the recursion depth  $d$ . Both  $\Omega_{\varphi,j,1}, \Omega_{\varphi,j,2}$  are partitioned into four disjoint subintervals with the randomly shifting borders.

##### 1. The initial phase

Let  $L_{\lambda,j,k} = \emptyset$  be the empty set. Compute  $x_a = F(a, \lambda)$ ,  $y_a = G(a, \lambda)$  and  $x_b = F(b, \lambda)$ ,  $y_b = G(b, \lambda)$ , where  $p_a = [x_a, y_a]$  and  $p_b = [x_b, y_b]$ . If a singularity in  $a$  or  $b$  is detected, throw an exception. Add the initial vertex to  $L_{\lambda,j,k}$ :  $L_{\lambda,j,k} \leftarrow p_a$ , where  $p_a = (x_a, y_a)$ . Set the recursion depth  $d = 1$ . The procedure is summarized in Alg. 6.

##### 2. The recursive procedure

Enter the recursive procedure and do the following substeps:

(a) If  $d > \bar{d}$  or  $b - a < \varepsilon$ , stop the recursive procedure and go to Step 3.

(b) For a given  $\Omega_{\varphi,j,k} = [a, b]$ , the interval is split by three points

$$\varphi_1 = a + \frac{1}{2}r_1(b - a), \quad \varphi_2 = a + r_2(b - a), \quad \varphi_3 = a + \frac{3}{2}r_3(b - a),$$

into the approximate quarters

$$\Omega_{\varphi,j,k,1} = [a, \varphi_1], \quad \Omega_{\varphi,j,k,2} = [\varphi_1, \varphi_2] \quad \Omega_{\varphi,j,k,3} = [\varphi_2, \varphi_3], \quad \Omega_{\varphi,j,k,4} = [\varphi_3, b],$$

where  $r_1, r_2, r_3$  are the random numbers inside the interval [0.45, 0.55].

**Algorithm 5** The construction of a parallel from the fragments.

---

```

1: function PARALLELPART( $L_{\varphi,j,k}, \Omega_{\lambda,j}, \varphi, \mathbb{P}, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
2:    $\langle M_1, M_2 \rangle = p(\varphi) \times m'(\varphi', \lambda'_0)$ 
3:    $L_{\varphi,j,1} = \emptyset, L_{\varphi,j,2} = \emptyset, L_{\varphi,j,3} = \emptyset$ 
4:   if  $(\lambda_1 > \underline{\lambda}_j + \varepsilon) \wedge (\lambda_1 < \bar{\lambda}_j - \varepsilon) \wedge (\lambda_2 > \underline{\lambda}_j + \varepsilon) \wedge (\lambda_2 < \bar{\lambda}_j - \varepsilon)$  then
5:      $\underline{\lambda}_{1,2} = \min(\lambda_1, \lambda_2), \bar{\lambda}_{1,2} = \max(\lambda_1, \lambda_2)$ 
6:      $\bar{\lambda}_{j,1} = \underline{\lambda}_{1,2} - \varepsilon, \underline{\lambda}_{j,2} = \underline{\lambda}_{1,2} + \varepsilon, \bar{\lambda}_{j,2} = \bar{\lambda}_{1,2} - \varepsilon, \underline{\lambda}_{j,2} = \bar{\lambda}_{1,2} + \varepsilon$ 
7:     csParInit( $\mathbb{P}, L_{\varphi,j,1}, \underline{\lambda}_j, \bar{\lambda}_{j,1}, \varphi, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
8:      $L_{\varphi,j} \leftarrow L_{\varphi,j_1}$ 
9:     csParInit( $\mathbb{P}, L_{\varphi,j,2}, \underline{\lambda}_{j,2}, \bar{\lambda}_{j,2}, \varphi, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
10:     $L_{\varphi,j} \leftarrow L_{\varphi,j_2}$ 
11:    csParInit( $\mathbb{P}, L_{\varphi,j,3}, \underline{\lambda}_{j,3}, \bar{\lambda}_j, \varphi, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
12:     $L_{\varphi,j} \leftarrow L_{\varphi,j_3}$ 
13:  else if  $(\lambda_1 > \underline{\lambda}_j + \varepsilon) \wedge (\lambda_1 < \bar{\lambda}_j - \varepsilon)$  then
14:     $\bar{\lambda}_{j,1} = \lambda_1 - \varepsilon, \underline{\lambda}_{j,2} = \lambda_1 + \varepsilon$ 
15:    csParInit( $\mathbb{P}, L_{\varphi,j,1}, \underline{\lambda}_j, \bar{\lambda}_{j,1}, \varphi, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
16:     $L_{\varphi,j} \leftarrow L_{\varphi,j_1}$ 
17:    csParInit( $\mathbb{P}, L_{\varphi,j,2}, \underline{\lambda}_{j,2}, \bar{\lambda}_j, \varphi, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
18:     $L_{\varphi,j} \leftarrow L_{\varphi,j_2}$ 
19:  else if  $(\lambda_2 > \underline{\lambda}_j + \varepsilon) \wedge (\lambda_2 < \bar{\lambda}_j - \varepsilon)$  then
20:     $\bar{\lambda}_{j,1} = \lambda_2 - \varepsilon, \underline{\lambda}_{j,2} = \lambda_2 + \varepsilon$ 
21:    csParInit( $\mathbb{P}, L_{\varphi,j,1}, \underline{\lambda}_j, \bar{\lambda}_{j,1}, \varphi, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
22:     $L_{\varphi,j} \leftarrow L_{\varphi,j_1}$ 
23:    csParInit( $\mathbb{P}, L_{\varphi,j,2}, \underline{\lambda}_{j,2}, \bar{\lambda}_j, \varphi, \underline{d}, \bar{d}, \varepsilon, \bar{\alpha}$ )
24:     $L_{\varphi,j} \leftarrow L_{\varphi,j_2}$ 

```

---

- (c) If a singularity in  $[\varphi_1, \lambda]$ ,  $[\varphi_2, \lambda]$ , or  $[\varphi_3, \lambda]$  occurs, throw a new exception according to the discontinuity direction.
- (d) Otherwise, evaluate the function values  $x_1 = F(\varphi_1, \lambda)$ ,  $y_1 = G(\varphi_1, \lambda)$ ,  $x_2 = F(\varphi_2, \lambda)$ ,  $y_2 = G(\varphi_2, \lambda)$ ,  $x_3 = F(\varphi_3, \lambda)$ ,  $y_3 = G(\varphi_3, \lambda)$ , at new vertices  $p_1, p_2, p_3$ .
- (e) For  $d \leq \underline{d}$ , this step begins with uniform sampling of the meridian points. If  $d > \bar{d}$ , it transforms to the adaptive method. Check the refinement criteria  $\alpha_1 = \alpha(p_a, p_1, p_2)$ ,  $\alpha_2 = \alpha(p_1, p_2, p_3)$ ,  $\alpha_3 = \alpha(p_2, p_3, p_b)$ , and the recursive depth  $d$ . When the meridian is not sufficiently smooth, or  $d \leq \underline{d}$ , it needs to be refined. Hence, the recursive subdivision is necessary.
- (f) If  $\alpha_1 > \bar{\alpha}$ , call the recursive procedure with the increased depth  $d = d + 1$  for the interval  $[a, \varphi_1]$ .
- (g) Add the new point  $p_1$  to the polynomial approximation of  $\tilde{m}(\lambda) : L_{\lambda,j,k} \leftarrow p_1$ .
- (h) If  $\alpha_1 > \bar{\alpha} \vee \alpha_2 > \bar{\alpha}$ , call the recursive procedure with the increased depth  $d = d + 1$  for the interval  $[\varphi_1, \varphi_2]$ .
- (i) Add the new point  $p_2$  to the polynomial approximation of  $\tilde{m}(\lambda) : L_{\lambda,j,k} \leftarrow p_2$ .

---

**Algorithm 6** Combined sampling of the meridian, the initial phase.

---

```

1: function CSMERINIT( $\mathbb{P}, L_{\lambda,j,k}, a, b, \lambda, d, \underline{d}, \bar{d}, \epsilon, \bar{\alpha}$ )
2:    $L_{\lambda,j,k} = \emptyset$ 
3:   if discontinuity in  $(a, \lambda)$  in  $\varphi$  direction then
4:     throw LatSingularityException ( $a$ )
5:   if discontinuity in  $(a, \lambda)$  in  $\lambda$  direction then
6:     throw LonSingularityException ( $\lambda$ )
7:   if discontinuity in  $(b, \lambda)$  in  $\varphi$  direction then
8:     throw LatSingularityException ( $b$ )
9:   if discontinuity in  $(b, \lambda)$  in  $\lambda$  direction then
10:    throw LonSingularityException ( $\lambda$ )
11:   $x_a = F(a, \lambda), y_a = G(a, \lambda)$ 
12:   $L_{\lambda,j,k} \leftarrow Point(x_a, y_a)$ 
13:  csMer( $\mathbb{P}, L_{\lambda,j,k}, a, b, x_a, y_a, x_b, y_b, d, \underline{d}, \bar{d}, \epsilon, \alpha$ )
14:   $x_b = F(b, \lambda), y_b = G(b, \lambda)$ 
15:   $L_{\lambda,j,k} \leftarrow Point(x_b, y_b)$ 

```

---

(j) If  $\alpha_2 > \bar{\alpha} \vee \alpha_3 > \bar{\alpha}$ , call the recursive procedure with the increased depth  $d = d + 1$  for the interval  $[\varphi_2, x_3]$ .

(k) Add the new point  $p_3$  to the polynomial approximation of  $\tilde{m}(\lambda) : L_{\lambda,j,k} \leftarrow p_3$ .

(l) If  $\alpha_3 > \bar{\alpha}$ , call the recursive procedure for the interval  $[\varphi_3, b]$ .

### 3. Add the last point

Add the last point  $p_b$  to the polynomial approximation of  $\tilde{m}(\lambda) : L_{\lambda,j,k} \leftarrow p_b$  and finish the adaptive sampling procedure.

The procedure is summarized in Alg. 7. Unfortunately, this technique does not involve the angles  $\alpha$  between the adjacent segments from the previous iteration. While the newly created segments of the polygonal approximation of  $m(\lambda)$  can fulfill the condition of  $\alpha_i \leq \bar{\alpha}$ , their joining to the previous/next segments may not hold this requirement. In general, this issue affects curves of the complex shape, where the average maximum value  $\alpha_i \doteq 1.25\bar{\alpha}$ .

**Sampling method S2.** The first improvement is represented by the refinement of the recursive condition. The original conditions  $\alpha_1 > \bar{\alpha}$ ,  $\alpha_1 > \bar{\alpha} \vee \alpha_2 > \bar{\alpha}$ ,  $\alpha_2 > \bar{\alpha} \vee \alpha_3 > \bar{\alpha}$ , and  $\alpha_3 > \bar{\alpha}$ , are replaced with the new

$$\alpha_1 > \bar{\alpha} \vee \alpha_2 > \bar{\alpha} \vee \alpha_3 > \bar{\alpha}, \quad (4.1)$$

identical for all recursive calls. This approach brings a slightly increased amount of the recursive calls, while the average maximum value falls to  $\alpha_i \doteq 1.15\bar{\alpha}$ . In most situations, this issue does not significantly affect the smoothness of the polynomial approximation and its perception by the user. However, exceeding the threshold may not always be acceptable.

**Sampling method S3.** Unlike S1, S2, the improved sampling method S3 takes into account angles between the newly created and adjacent segments from the previous iteration.

---

**Algorithm 7** Combined sampling of the meridian points, the recursive phase, method S1.

---

```

1: function CSMER( $\mathbb{P}, L_{\lambda,j,k}, a, b, x_a, y_a, x_b, y_b, d, \underline{d}, \bar{d}, \epsilon, \alpha$ )
2:   if  $(d > \bar{d}) \vee (b - a < \epsilon)$  then
3:     return
4:    $r_1 = \text{rand}(0.45, 0.55), r_2 = \text{rand}(0.45, 0.55), r_3 = \text{rand}(0.45, 0.55)$ 
5:    $\varphi_1 = a + \frac{1}{2}r_1(b - a), \varphi_2 = a + r_2(b - a), \varphi_3 = a + \frac{3}{2}r_3(b - a)$ 
6:   if discontinuity in  $(\varphi_i, \lambda)i$  in  $\varphi$  direction then
7:     throw LatSingularityException  $(\varphi_i), i = 1, 2, 3$ 
8:   if discontinuity in  $(\varphi_i, \lambda)i$  in  $\lambda$  direction then
9:     throw LonSingularityException  $(\lambda), i = 1, 2, 3$ 
10:   $x_i = F(\varphi_i, \lambda), y_i = G(\varphi_i, \lambda), i = 1, 2, 3$ 
11:   $p_a = \text{Point}(x_a, y_a), p_b = \text{Point}(x_b, y_b), p_i = \text{Point}(x_i, y_i), i = 1, 2, 3$ 
12:   $\alpha_1 = \alpha(p_a, p_1, p_2), \alpha_2 = \alpha(p_1, p_2, p_3), \alpha_3 = \alpha(p_2, p_3, p_b)$ 
13:  if  $(\alpha_1 > \bar{\alpha}) \vee (d \leq \underline{d})$  then
14:    CSMER( $\mathbb{P}, L_{\lambda,j,k}, a, \varphi_1, x_a, y_a, x_1, y_1, d + 1, \underline{d}, \bar{d}, \bar{\alpha}$ );
15:     $L_{\lambda,j,k} \leftarrow \text{Point}(x_1, y_1)$ 
16:    if  $(\alpha_1 > \bar{\alpha}) \vee (\alpha_2 > \bar{\alpha}) \vee (d \leq \underline{d})$  then
17:      CSMER( $\mathbb{P}, L_{\lambda,j,k}, \varphi_1, \varphi_2, x_1, y_1, x_2, y_2, d + 1, \underline{d}, \bar{d}, \bar{\alpha}$ );
18:       $L_{\lambda,j,k} \leftarrow \text{Point}(x_2, y_2)$ 
19:      if  $(\alpha_2 > \bar{\alpha}) \vee (\alpha_3 > \bar{\alpha}) \vee (d \leq \underline{d})$  then
20:        CSMER( $\mathbb{P}, L_{\lambda,j,k}, \varphi_2, \varphi_3, x_2, y_2, x_3, y_3, d + 1, \underline{d}, \bar{d}, \bar{\alpha}$ );
21:         $L_{\lambda,j,k} \leftarrow \text{Point}(x_3, y_3)$ 
22:        if  $(\alpha_3 > \bar{\alpha}) \vee (d \leq \underline{d})$  then
23:          CSMER( $\mathbb{P}, L_{\lambda,j,k}, \varphi_3, b, x_3, y_3, x_b, y_b, d + 1, \underline{d}, \bar{d}, \bar{\alpha}$ );

```

---

For a given  $\Omega_{\varphi,j,k} = [a_k, b_k]$ , its predecessor  $\Omega_{\varphi,j,k-1} = [a_{k-1}, b_{k-1}]$  and successor  $\Omega_{\varphi,j,k+1} = [a_{k+1}, b_{k+1}]$  are  $\Omega_{\varphi,j,k-1} = \text{pred}(\Omega_{\varphi,j,k}), \Omega_{\varphi,j,k+1} = \text{succ}(\Omega_{\varphi,j,k})$ , where  $b_{k-1} = a_k, a_{k+1} = b_k$ . Analogously, for the polygonal approximation of  $\Omega_{\varphi,j,k}$  bounds are  $p_{a,k-1} = \text{pred}(p_{a,k}), p_{b,k+1} = \text{succ}(p_{b,k})$ , where  $p_{a,k-1} = [x_{a,k-1}, y_{a,k-1}], p_{a,k} = [x_{a,k}, y_{a,k}] \equiv p_{b,k-1}, p_{b,k} = [x_{b,k}, y_{b,k}] \equiv p_{a,k+1}, p_{b,k+1} = [x_{b,k+1}, y_{b,k+1}]$ , and  $x_{a,k-1} = F(a_{k-1}, \lambda), x_{a,k} = F(a_k, \lambda), x_{b,k} = F(b_k, \lambda), x_{b,k+1} = F(b_{k+1}, \lambda)$ ; see Fig. 4.2

The approximate quarters of  $\Omega_{\varphi,j,k}$  denoted as  $\Omega_{\varphi,j,k,l}, l = 1, \dots, 4$ , have the form of

$$\Omega_{\varphi,j,k,1} = [a_k, \varphi_{1,k}], \quad \Omega_{\varphi,j,k,2} = [\varphi_{1,k}, \varphi_{2,k}] \quad \Omega_{\varphi,j,k,3} = [\varphi_{2,k}, \varphi_{3,k}], \quad \Omega_{\varphi,j,k,4} = [\varphi_{3,k}, b_k],$$

The idea of modification is straightforward, an information about previous and subsequent quarters of  $\Omega_{\varphi,j,k,l}$  is stored

$$\Omega_{\varphi,j,k,l-1} = \text{pred}(\Omega_{\varphi,j,k,l}), \quad \Omega_{\varphi,j,k,l+1} = \text{succ}(\Omega_{\varphi,j,k,l}).$$

While the first quarter lacks its predecessor, the last quarter lacks the successor. For a current recursive depth  $d$ , the segment  $(p_{a,k}, p_{b,k})$  of the polygonal approximation is split to the quarters, where  $p_{1,k}, p_{2,k}, p_{3,k}$  are the newly created vertices at the recursive depth  $d + 1$  representing the ‘‘projected bounds’’ of newly created intervals  $\Omega_{\varphi,j,k,1}, \dots, \Omega_{\varphi,j,k,4}$ . Unlike S1, S2, the angles  $\alpha_i$  are measured not only between the newly created segments

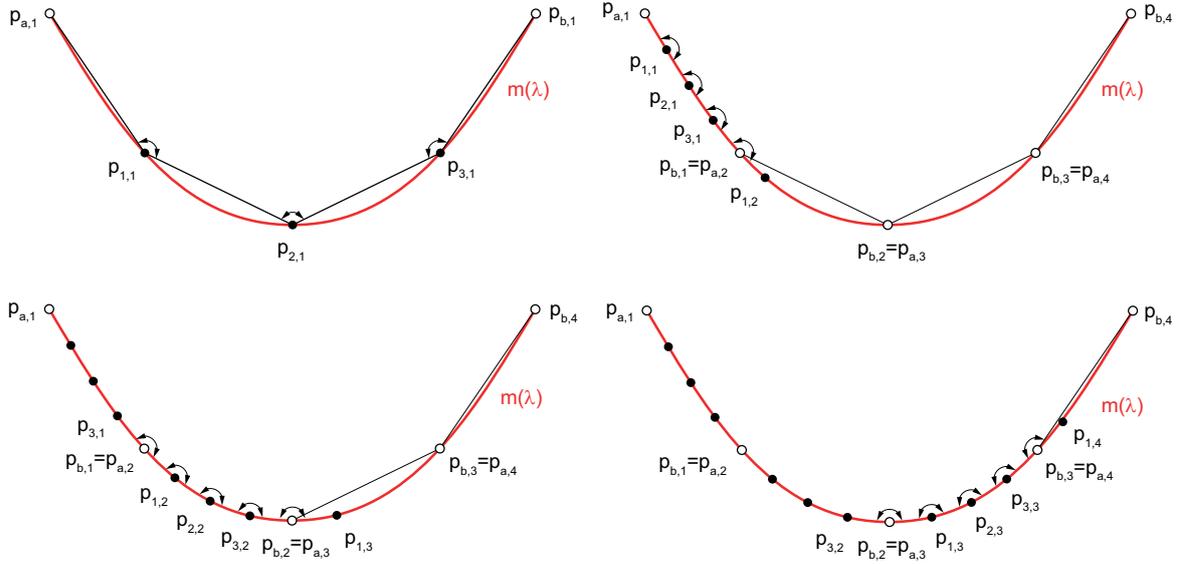


Figure 4.2: Illustration of the combined sampling method S3 for the meridian  $m(\lambda)$ .

$$\alpha_1 = \alpha(p_{a,k}, p_{1,k}, p_{2,k}), \quad \alpha_2 = \alpha(p_{1,k}, p_{2,k}, p_{3,k}), \quad \alpha_3 = \alpha(p_{2,k}, p_{3,k}, p_{b,k}),$$

in the recursive depth  $d + 1$ , but also between the adjacent segments

$$\alpha_0 = \alpha(p_{3,k-1}, p_{a,k}, p_{1,k}), \quad \alpha_4 = \alpha(p_{3,k}, p_{b,k}, p_{1,k+1}),$$

where  $p_{3,k-1}$  is the projected lower bound of  $\Omega_{\varphi,j,k-1,4}$  (a projected third quarter of the interval  $\Omega_{\varphi,j,k-1}$  previous to  $\Omega_{\varphi,j,k}$ ) with the latitude

$$\varphi_{3,k-1} = a_k + \frac{1}{2}r_0(a_{k-1} - a_k),$$

$p_{1,k+1}$  is the projected first-quarter  $\Omega_{\varphi,j,k+1,1}$  of the next interval  $\Omega_{\varphi,j,k+1}$  with the latitude

$$\varphi_{1,k+1} = b_k + \frac{1}{2}r_4(b_{k+1} - b_k),$$

and  $r_0, r_4$  are the random numbers generated according to the principles mentioned above. The modified condition for the recursive call has the form of

$$\alpha_0 > \bar{\alpha} \vee \alpha_1 > \bar{\alpha} \vee \alpha_2 > \bar{\alpha} \vee \alpha_3 > \bar{\alpha} \vee \alpha_4 > \bar{\alpha}. \quad (4.2)$$

In general, the improved performance of S3 occurs only in specific situations when the sampled function has a complex shape (meridians of Bonne projections). However, the significant disadvantage is represented by the increased amount of sampled points in the highly curved areas. The recursive step has a slightly different form:

1. If  $d > \bar{d}$  or  $b_k - a_k < \varepsilon$ , stop the recursive procedure and go to Step 3.
2. For a given  $\Omega_{\varphi,j,k} = [a_k, b_k]$ , its predecessor  $\Omega_{\varphi,j,k-1} = [a_{k-1}, b_{k-1}]$ , and successor  $\Omega_{\varphi,j,k+1} = [a_{k+1}, b_{k+1}]$ , the interval is split by three points, the positions are close to the quarters of the interval

$$\varphi_{1,k} = a_k + \frac{1}{2}r_1(b_k - a_k), \quad \varphi_{2,k} = a_k + r_2(b_k - a_k), \quad \varphi_{3,k} = a_k + \frac{3}{2}r_3(b_k - a_k).$$

**Algorithm 8** Combined sampling of the meridian points, the recursive phase, method S3.

---

```

1: function CSMER( $\mathbb{P}, L_{\lambda,j,k}, a_k, b_k, a_{k-1}, b_{k+1}, x_a, y_a, x_b, y_b, d, \underline{d}, \bar{d}, \epsilon, \alpha$ )
2:   if ( $d > \bar{d}$ )  $\vee$  ( $b_k - a_k < \epsilon$ ) then
3:     return
4:    $r_i = \text{rand}(0.45, 0.55), i = 0, \dots, 4$ 
5:    $\varphi_{1,k} = a_k + \frac{1}{2}r_1(b_k - a_k), \varphi_{2,k} = a + r_2(b_k - a), \varphi_{3,k} = a + \frac{3}{2}r_3(b_k - a_k)$ 
6:    $\varphi_{3,k-1} = a_k + \frac{1}{2}r_1(b_{k-1} - a_k), \varphi_{1,k+1} = b_k + r_4(b_{k+1} - b_k)$ 
7:   if discontinuity in  $(\varphi_{i,k}, \lambda)$  in  $\varphi$  direction then
8:     throw LatSingularityException  $(\varphi_{i,k}), i = 1, 2, 3$ 
9:   if discontinuity in  $(\varphi_{i,k}, \lambda)$  in  $\lambda$  direction then
10:    throw LonSingularityException  $(\lambda), i = 1, 2, 3$ 
11:    $x_{3,k-1} = F(\varphi_{3,k-1}, \lambda), y_{3,k-1} = G(\varphi_{3,k-1}, \lambda)$ 
12:    $x_{i,k} = F(\varphi_{i,k}, \lambda), y_{i,k} = G(\varphi_{i,k}, \lambda), i = 1, 2, 3$ 
13:    $x_{1,k+1} = F(\varphi_{1,k+1}, \lambda), y_{1,k+1} = G(\varphi_{1,k+1}, \lambda)$ 
14:    $p_{a,k} = \text{Point}(x_{a,k}, y_{a,k}), p_{b,k} = \text{Point}(x_{b,k}, y_{b,k})$ 
15:    $p_{i,k} = \text{Point}(x_{i,k}, y_{i,k}), i = 1, 2, 3$ 
16:    $p_{3,k-1} = \text{Point}(x_{3,k-1}, y_{3,k-1}), p_{1,k+1} = \text{Point}(x_{1,k+1}, y_{1,k+1})$ 
17:    $\alpha_1 = \alpha(p_{a,k}, p_{1,k}, p_{2,k}), \alpha_2 = \alpha(p_{1,k}, p_{2,k}, p_{3,k}), \alpha_3 = \alpha(p_{2,k}, p_{3,k}, p_{b,k})$ 
18:    $\alpha_0 = \alpha(p_{3,k-1}, p_{a,k}, p_{1,k}), \alpha_4 = \alpha(p_{3,k}, p_{b,k}, p_{1,k+1})$ 
19:   if ( $\alpha_0 > \bar{\alpha}$ )  $\vee$  ( $\alpha_1 > \bar{\alpha}$ )  $\vee$  ( $\alpha_2 > \bar{\alpha}$ )  $\vee$  ( $\alpha_3 > \bar{\alpha}$ )  $\vee$  ( $\alpha_4 > \bar{\alpha}$ )  $\vee$  ( $d \leq \underline{d}$ ) then
20:     csMer( $\mathbb{P}, L_{\lambda,j,k}, a_k, \varphi_{1,k}, \varphi_{3,k-1}, \varphi_{2,k}, x_{a,k}, y_{a,k}, x_{1,k}, y_{1,k}, d + 1, \underline{d}, \bar{d}, \bar{\alpha}$ )
21:    $L_{\lambda,j,k} \leftarrow \text{Point}(x_{1,k}, y_{1,k})$ 
22:   if ( $\alpha_0 > \bar{\alpha}$ )  $\vee$  ( $\alpha_1 > \bar{\alpha}$ )  $\vee$  ( $\alpha_2 > \bar{\alpha}$ )  $\vee$  ( $\alpha_3 > \bar{\alpha}$ )  $\vee$  ( $\alpha_4 > \bar{\alpha}$ )  $\vee$  ( $d \leq \underline{d}$ ) then
23:     csMer( $\mathbb{P}, L_{\lambda,j,k}, \varphi_{1,k}, \varphi_{2,k}, \varphi_{a,k}, \varphi_{3,k}, x_{1,k}, y_{1,k}, x_{2,k}, y_{2,k}, d + 1, \underline{d}, \bar{d}, \bar{\alpha}$ )
24:    $L_{\lambda,j,k} \leftarrow \text{Point}(x_{2,k}, y_{2,k})$ 
25:   if ( $\alpha_0 > \bar{\alpha}$ )  $\vee$  ( $\alpha_1 > \bar{\alpha}$ )  $\vee$  ( $\alpha_2 > \bar{\alpha}$ )  $\vee$  ( $\alpha_3 > \bar{\alpha}$ )  $\vee$  ( $\alpha_4 > \bar{\alpha}$ )  $\vee$  ( $d \leq \underline{d}$ ) then
26:     csMer( $\mathbb{P}, L_{\lambda,j,k}, \varphi_{2,k}, \varphi_{3,k}, \varphi_{1,k}, b_k, x_{2,k}, y_{2,k}, x_{3,k}, y_{3,k}, d + 1, \underline{d}, \bar{d}, \bar{\alpha}$ )
27:    $L_{\lambda,j,k} \leftarrow \text{Point}(x_{3,k}, y_{3,k})$ 
28:   if ( $\alpha_0 > \bar{\alpha}$ )  $\vee$  ( $\alpha_1 > \bar{\alpha}$ )  $\vee$  ( $\alpha_2 > \bar{\alpha}$ )  $\vee$  ( $\alpha_3 > \bar{\alpha}$ )  $\vee$  ( $\alpha_4 > \bar{\alpha}$ )  $\vee$  ( $d \leq \underline{d}$ ) then
29:     csMer( $\mathbb{P}, L_{\lambda,j,k}, \varphi_{3,k}, b_k, \varphi_{2,k}, \varphi_{1,k+1}, x_{3,k}, y_{3,k}, x_{b,k}, y_{b,k}, d + 1, \underline{d}, \bar{d}, \bar{\alpha}$ )

```

---

Subsequently, the third quarter  $\varphi_{3,k-1}$  of the previous interval  $\Omega_{\varphi,j,k-1}$  representing the lower bound of  $\Omega_{\varphi,j,k-1,4}$  and the first quarter  $\varphi_{1,k+1}$  of the next intervals  $\Omega_{\varphi,j,k+1}$  representing the upper bound of  $\Omega_{\varphi,j,k+1,1}$ , are evaluated.

- (a) If a singularity in  $[\varphi_{3,k-1}, \lambda], [\varphi_{1,k}, \lambda], [\varphi_{2,k}, \lambda], [\varphi_{3,k}, \lambda]$ , or  $[\varphi_{1,k+1}, \lambda]$  occurs, throw a new exception according to the discontinuity direction.
- (b) Otherwise, evaluate the function values  $x_{3,k-1} = F(\varphi_{3,k-1}, \lambda), y_{3,k-1} = G(\varphi_{3,k-1}, \lambda), x_{1,k} = F(\varphi_{1,k}, \lambda), y_{1,k} = G(\varphi_{1,k}, \lambda), x_{2,k} = F(\varphi_{2,k}, \lambda), y_{2,k} = G(\varphi_{2,k}, \lambda), x_{3,k} = F(\varphi_{3,k}, \lambda), y_{3,k} = G(\varphi_{3,k}, \lambda), x_{1,k+1} = F(\varphi_{1,k+1}, \lambda), y_{1,k+1} = G(\varphi_{1,k+1}, \lambda)$ , at new vertices  $p_{1,k}, p_{2,k}, p_{3,k}$  and the supplementary vertices  $p_{3,k-1}, p_{1,k+1}$ .
- (c) For  $d \leq \underline{d}$ , this step begins with uniform sampling of the meridian points. If  $d > \bar{d}$ , it transforms to the adaptive method. Check the refinement criteria  $\alpha_0 = \alpha(p_{3,k-1}, p_{a,k}, p_{1,k}), \alpha_1 = \alpha(p_{a,k}, p_{1,k}, p_{2,k}), \alpha_2 = \alpha(p_{1,k}, p_{2,k}, p_{3,k}), \alpha_3 =$

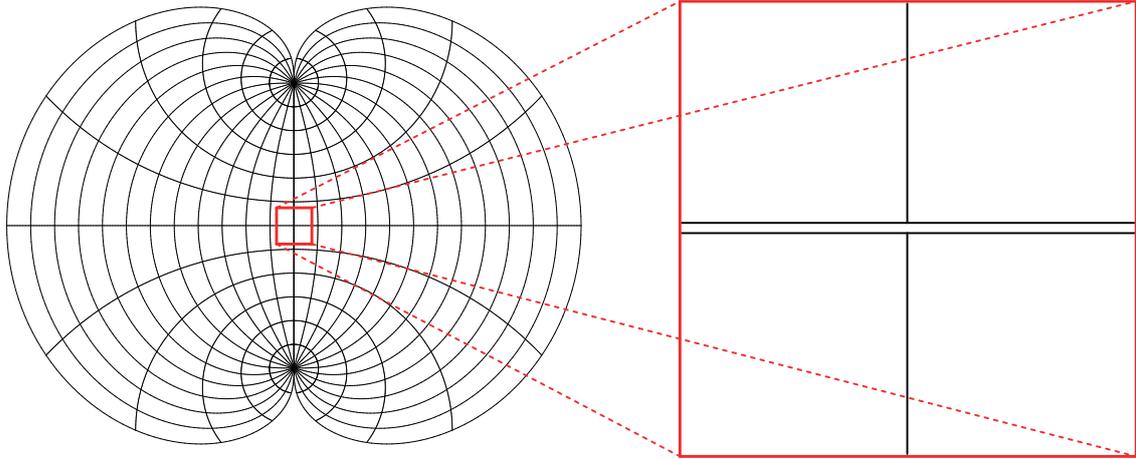


Figure 4.3: The discontinuity treatment without its classification using the proposed algorithm in polyconic projection; the doubled equator is visible under magnification.

$\alpha(p_{2,k}, p_{3,k}, p_{b,k})$ ,  $\alpha_4 = \alpha(p_{3,k}, p_{b,k}, p_{1,k+1})$ , and the recursive depth  $d$ . When a meridian is not sufficiently smooth, or  $d \leq \underline{d}$ , it needs to be refined; the recursive subdivision is necessary.

- (d) If the condition given by Eq. 4.2 is held, call the recursive procedure with the increased depth  $d = d + 1$  for the interval  $\Omega_{\varphi,j,k,1} = [a_k, \varphi_{1,k}]$ , its predecessor  $\Omega_{\varphi,j,k-1,4} = [\varphi_{3,k-1}, a_k]$ , and successor  $\Omega_{\varphi,j,k,2} = [\varphi_{1,k}, \varphi_{2,k}]$ .
- (e) Add the new point  $p_{1,k}$  to the polynomial approximation of  $\tilde{m}(\lambda) : L_{\lambda,j,k} \leftarrow p_{1,k}$ .
- (f) If the condition given by Eq. 4.2 is held, call the recursive procedure with the increased depth  $d = d + 1$  for the interval  $\Omega_{\varphi,j,k,2} = [\varphi_{1,k}, \varphi_{2,k}]$ , its predecessor  $\Omega_{\varphi,j,k,1} = [a_k, \varphi_{1,k}]$ , and successor  $\Omega_{\varphi,j,k,3} = [\varphi_{2,k}, \varphi_{3,k}]$ .
- (g) Add the new point  $p_{2,k}$  to the polynomial approximation of  $\tilde{m}(\lambda) : L_{\lambda,j,k} \leftarrow p_{2,k}$ .
- (h) If the condition given by Eq. 4.2 is held, call the recursive procedure with the increased depth  $d = d + 1$  for the interval  $\Omega_{\varphi,j,k,3} = [\varphi_{2,k}, \varphi_{3,k}]$ , its predecessor  $\Omega_{\varphi,j,k,2} = [\varphi_{1,k}, \varphi_{2,k}]$ , and successor  $\Omega_{\varphi,j,k,4} = [\varphi_{3,k}, b_k]$ .
- (i) Add the new point  $p_{3,k}$  to the polynomial approximation of  $\tilde{m}(\lambda) : L_{\lambda,j,k} \leftarrow p_{3,k}$ .
- (j) If the condition given by Eq. 4.2 is held, call the recursive procedure for the last interval  $\Omega_{\varphi,j,k,4} = [\varphi_{3,k}, b_k]$ , its predecessor  $\Omega_{\varphi,j,k,3} = [\varphi_{2,k}, \varphi_{3,k}]$ , and successor  $\Omega_{\varphi,j,k+1,1} = [b_k, \varphi_{1,l+1}]$ .

The procedure is summarized in Alg. 8.

**A combined sampling of the parallel.** Analogously, the parallel  $p(\varphi)$  is sampled over the interval  $\Omega_{\lambda,j} = [\underline{\lambda}_j, \bar{\lambda}_j]$ . The polygonal approximation consists of two steps: a partition of  $p(\varphi)$  to the fragments and combined sampling of the fragments.

Each parallel fragment needs to be sampled,  $\Omega_{\varphi,j,1}$ ,  $\Omega_{\varphi,j,2}$ ,  $\Omega_{\varphi,j,3}$  are partitioned into four

disjoint subintervals. While for adaptive meridian  $m(\lambda)$  sampling are:  $\lambda = const$ ,  $\varphi \in \Omega_{\varphi,j}$ , for parallel  $p(\varphi)$  sampling are:  $\varphi = const$ ,  $\lambda \in \Omega_{\lambda,j}$ . Therefore, the randomly generated parallel points

$$\lambda_1 = a + \frac{1}{2}r_1(b - a), \quad \lambda_2 = a + r_2(b - a), \quad \lambda_3 = a + \frac{3}{2}r_3(b - a),$$

split the interval  $\Omega_{\lambda,j,k}$  to the approximate quarters

$$\Omega_{\lambda,j,k,1} = [a, \lambda_1], \quad \Omega_{\lambda,j,k,2} = [\lambda_1, \lambda_2] \quad \Omega_{\lambda,j,k,3} = [\lambda_2, \lambda_3], \quad \Omega_{\lambda,j,k,4} = [\lambda_3, b].$$

The remaining steps are analogous. This method can treat all kinds of discontinuities, without specific knowledge of  $\varepsilon$ ; for practical computations is  $\varepsilon = 0.001^\circ$ . It has a disadvantage - the duplication of lines along singularities if the discontinuities are not classified. For the discontinuity  $c = \varphi = 0^\circ$  in the polyconic projection, this issue is illustrated in Fig. 4.3. A hemisphere constructed in the normal aspect is composed of two sub-intervals  $[-\pi/2, 0)$ ,  $(0, \pi/2]$ . Hence, the equator is doubled, which does not look aesthetically pleasing, but it is visible only under magnification.

## 5. Experiments and results

The principles mentioned above will be illustrated on the several tests. While the first analysis compares the behavior and properties of combined sampling, the second test reconstructs the Fournier I. projection graticule in the oblique aspect (several discontinuities in the coordinate functions are involved). Finally, the last experiment illustrates the splitting procedure efficiency depending on the threshold  $\varepsilon$ . The results are summarized in Tabs. 1-5.

### 5.1. Comparison of uniform and combined sampling

During this test, the uniform and combined sampling techniques will be compared regarding the data representation compactness measured by the amount of the sampled meridian points  $n_{mer}$  and parallel points  $n_{par}$ . Additionally, the maximum angles  $\bar{\alpha}_m, \bar{\alpha}_p$  between the sampled meridian and parallel segments together with their mean values  $\tilde{\alpha}_m, \tilde{\alpha}_p$  are measured; these indicators depend on the sampling density. For the uniform sampling, the sampling steps  $\delta_\varphi, \delta_\lambda$  are fixed, while the adaptive sampling is driven by the sampling angle  $\alpha$ . Our experiments will be carried out for combined sampling with  $\alpha = 1^\circ, 2^\circ, 5^\circ$ , the recursive depth is  $d = 1$ , and for the uniform sampling where  $\delta_\varphi, \delta_\lambda$  will be set providing the analogous amount of the sampled points. 8 projections are involved in testing: equidistant cylindrical, conic, azimuthal, Bonne and Hassler ( $\varphi_1 = 50^\circ$ ), Nicolosi, Littrow, and Adams I World conformal. The results are summarized in Tab. 1.

While S1, S2 exceed the  $\bar{\alpha}_m, \bar{\alpha}_p$  values (+20% for S1, +15% for S2), S3 provides values satisfying the given criteria (-7%) without a significant increment of sampled points. Cylindrical, conic and azimuthal projections illustrate the adaptive sampling efficiency; for the straight meridians/parallels, only 2% of points are sufficient for the shape estimation (uniform sampling cannot achieve this efficiency).

Concerning the criteria mentioned above the results are compared in Tab. 2. The sampling steps  $\delta_\varphi, \delta_\lambda$  have been set so that the amount of sampled points  $n_{mer}, n_{par}$  is analogous. In

Table 1: Combined sampling of the graticule for  $\bar{\alpha} = 1^\circ, 2^\circ, 5^\circ$ , methods S1, S2, S3 are compared.

Projection	#	$n_{mer}$	$n_{par}$	$\bar{\alpha}_m$	$\bar{\alpha}_p$	$\tilde{\alpha}_m$	$\tilde{\alpha}_p$
Equidistant cylindrical	S1	195/195/195	190/190/190	0/0/0	0/0/0	0/0/0	0/0/0
	S2	195/195/195	190/190/190	0/0/0	0/0/0	0/0/0	0/0/0
	S3	195/195/195	190/190/190	0/0/0	0/0/0	0/0/0	0/0/0
Equidistant conic	S1	195/195/195	9961/6973/2464	0/0/0	1.02/2.27/4.98	0/0/0	0.53/0.75/2.16
	S2	195/195/195	10066/7678/2470	0/0/0	1.00/2.13/4.06	0/0/0	0.52/0.69/2.16
	S3	195/195/195	10090/8734/2470	0/0/0	0.99/1.99/4.41	0/0/0	0.52/0.60/2.16
Equidistant azimuthal	S1	195/195/195	12352/9091/2494	0/0/0	1.08/2.04/4.87	0/0/0	0.56/0.76/2.78
	S2	195/195/195	13186/9238/2494	0/0/0	1.09/2.06/4.89	0/0/0	0.52/0.74/2.78
	S3	195/195/195	13786/9706/2506	0/0/0	1.00/1.90/4.78	0/0/0	0.50/0.71/2.77
Bonne	S1	14250/7206/2901	6610/3613/1576	1.21/2.27/5.68	1.05/2.06/4.98	0.45/0.88/2.20	0.46/0.84/1.98
	S2	15171/7599/3099	6766/3778/1630	1.05/2.52/5.00	1.00/2.00/4.82	0.42/0.84/2.07	0.45/0.81/1.91
	S3	16839/8367/3483	6958/4330/1654	1.00/1.99/4.98	1.00/1.99/4.96	0.38/0.76/1.84	0.44/0.71/1.82
Hassler	S1	16452/7170/3342	8839/5779/1912	1.14/2.13/5.77	1.22/2.25/4.91	0.41/0.96/2.09	0.51/0.78/2.39
	S2	17391/7479/3519	9370/5974/1966	1.07/2.17/5.20	1.03/2.11/4.99	0.40/0.92/1.99	0.48/0.75/2.32
	S3	19899/8379/4191	9826/6670/2002	1.00/2.00/4.96	1.00/1.99/4.76	0.35/0.82/1.66	0.46/0.67/2.28
Nicolosi	S1	14018/6503/3227	5519/2495/1124	1.14/2.29/5.64	1.15/2.16/5.50	0.43/0.94/1.92	0.42/0.95/2.12
	S2	14468/6944/3320	5804/2564/1208	1.00/2.00/4.96	1.01/2.00/4.94	0.42/0.88/1.86	0.40/0.92/1.96
	S3	15176/8156/3488	6224/2912/1352	1.00/2.00/4.93	1.00/1.99/4.89	0.40/0.74/1.77	0.37/0.81/1.76
Littrow	S1	6513/3279/1470	12011/7052/2507	1.10/2.15/5.43	1.14/2.27/5.17	0.50/1.00/2.30	0.51/0.87/2.48
	S2	6963/3615/1587	12242/7334/2606	1.06/2.22/4.90	1.05/1.99/5.27	0.47/0.91/2.12	0.50/0.84/2.38
	S3	8121/4374/2151	13796/7931/2903	1.00/2.00/4.86	1.00/1.98/4.98	0.40/0.75/1.55	0.45/0.77/2.13
Adams I.	S1	7083/3669/1788	4430/2285/887	1.07/2.00/4.93	1.06/2.10/6.16	0.44/0.86/1.82	0.44/0.83/2.04
	S2	7449/4017/2097	4778/2534/1046	1.07/2.08/4.96	1.02/2.01/4.86	0.42/0.78/1.53	0.41/0.74/1.74
	S3	8085/4689/2148	4562/2618/1058	1.00/2.00/4.89	0.98/1.99/4.44	0.38/0.67/1.49	0.40/0.72/1.72

Table 2: Uniform sampling of the graticule with the determined steps  $\delta\varphi$ ,  $\delta\lambda$  compared to S3; the amounts of sampled points  $n_{mer}$ ,  $n_{par}$  are preserved.

Projection	$\bar{\alpha}$	$\delta\varphi$	$\delta\lambda$	$n_{mer}$	$\Delta n_{mer}$	$n_{par}$	$\Delta n_{par}$	$\bar{\alpha}_m$	$\Delta \bar{\alpha}_m$	$\bar{\alpha}_p$	$\Delta \bar{\alpha}_p$	$\tilde{\alpha}_m$	$\Delta \tilde{\alpha}_m$	$\tilde{\alpha}_p$	$\Delta \tilde{\alpha}_p$
Equidistant cylindrical	1	40.00	45.00	195	+0.0%	190	+0.0%	0	-	0	-	0	-	0	-
	2	40.00	45.00	195	+0.0%	190	+0.0%	0	-	0	-	0	-	0	-
	5	40.00	45.00	195	+0.0%	190	+0.0%	0	-	0	-	0	-	0	-
Equidistant conic	1	45.00	0.68	195	+0.0%	10070	-0.2%	0	-	0.78	-21.2%	0	-	0.52	+0.0%
	2	45.00	0.78	195	+0.0%	8778	+0.5%	0	-	0.90	-54.8%	0	-	0.60	+20.0%
	5	45.00	2.74	195	+0.0%	2470	+0.0%	0	-	3.19	-27.7%	0	-	2.14	-0.9%
Equidistant azimuthal	1	45.00	0.50	195	+0.0%	13718	-0.5%	0	-	0.50	-50.0%	0	-	0.50	+0.0%
	2	45.00	0.71	195	+0.0%	9652	-0.6%	0	-	1.06	-79.3%	0	-	0.71	+0.0%
	5	45.00	2.74	195	+0.0%	2508	+0.1%	0	-	4.11	-14.0%	0	-	2.75	-0.7%
Bonne	1	0.42	0.99	16770	-4.1%	6916	-6.0%	8.53	+753.0%	1.14	+14.0%	0.38	+0.0%	0.44	+0.0%
	2	0.85	1.59	8307	-0.7%	4332	+0.0%	9.96	+295.2%	1.83	-8.0%	0.76	+0.0%	0.71	+0.0%
	5	2.05	4.23	3471	-0.3%	1672	+1.2%	9.61	+93.0%	3.24	-34.7%	1.79	-2.7%	1.87	+2.8%
Hassler	1	0.36	0.70	19539	-1.8%	9842	+0.2%	0.82	-18.0%	0.70	-30.0%	0.35	+0.0%	0.46	+0.0%
	2	0.85	1.04	8307	-0.9%	6650	-0.3%	1.93	-3.5%	1.04	-50.7%	0.83	+1.2%	0.68	+1.5%
	5	1.73	3.43	4134	-1.4%	2014	+0.6%	3.92	-21.0%	5.14	+8.0%	1.69	+1.8%	2.25	-3.0%
Nicolosi	1	0.45	1.17	15200	+0.2%	6200	-0.4%	1.13	+13.0%	0.87	-13.0%	0.36	-10.0%	0.38	-5.0%
	2	0.85	2.51	8132	-0.3%	2920	+0.3%	1.42	-29.0%	1.87	-6.0%	0.75	+1.4%	0.98	+21.0%
	5	2.00	5.40	3496	+0.2%	1360	+0.6%	3.34	-32.2%	5.96	+21.9%	1.78	+0.6%	1.75	-0.6%
Littrow	1	0.77	0.45	8112	-0.1%	13718	-0.6%	4.94	+394.0%	2.88	+288.0%	0.40	+0.0%	0.47	+4.4%
	2	1.66	0.86	4329	-1.0%	7980	+0.6%	9.39	+369.5%	4.95	+150.0%	0.76	+1.3%	0.86	+11.7%
	5	3.28	2.38	2184	+1.5%	2926	+0.8%	9.49	+95.3%	9.54	+91.6%	1.54	-0.7%	2.38	+11.7%
Adams I.	1	0.88	1.51	8073	-0.2%	4598	+0.8%	7.39	+739.0%	5.02	+412.2%	0.43	+13.2%	0.31	-22.5%
	2	1.52	2.73	4719	+0.6%	2622	+0.1%	9.26	+363.0%	8.29	+316.6%	0.74	+10.5%	0.54	-25.0%
	5	3.35	6.49	2145	-0.1%	1064	+0.6%	9.96	+103.7%	9.00	+102.7%	1.62	+8.7%	1.26	-26.7%

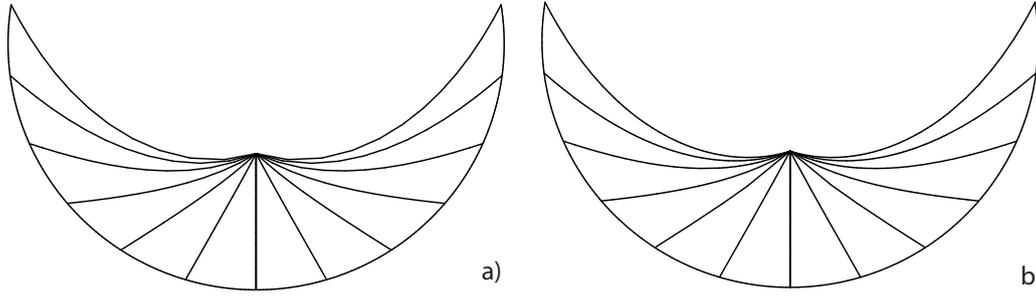


Figure 5.1: Comparison of the uniform a) and combined b) sampling (S3) for the Bonne projection,  $\bar{\alpha} = 5^\circ$ ; the adaptive polygonal approximation is smoother.

the higher-curvature regions, the combined sampling technique provides a smoother approximation of the meridians/parallels. A typical example is represented by the Bonne, Littrow or Adams I. projections, where uniform sampling does not provide good results. The maximum angles between the sampled elements are more than five times higher. If meridians or parallels are represented by the circular arcs, the combined sampling does not bring any advantage and may cause a minor deterioration of the average values  $\bar{\alpha}_m, \bar{\alpha}_p$  (Nicolosi projection). Raise the steps  $\delta\varphi, \delta\lambda$  reduces the advantage of adaptive sampling (and thus  $\bar{\alpha}$ ) even for the high-curvature regions (see Bonne projection in Fig. 5.1). In general, uniform sampling preserves the curvature worse and provides more redundant data.

### 5.2. Detection of discontinuities: Fournier I. projection

The Fournier I. projection equations have a complex form and contain several discontinuities both in the latitudinal and longitudinal directions. In accordance with [24], the projection equations are written as follows. If  $(\lambda - \lambda_0) = 0$  or  $|\varphi| = \pi/2$ ,

$$X = 0, \quad Y = R\varphi.$$

If  $\varphi = 0$ ,

$$X = R(\lambda - \lambda_0), \quad Y = 0.$$

If  $|\lambda - \lambda_0| = \pi/2$ ,

$$X = R(\lambda - \lambda_0) \cos \varphi, \quad Y = R\frac{\pi}{2} \sin \varphi.$$

Otherwise,

$$C = \frac{\pi^2}{4}, \quad P = \pi |\sin \varphi|, \quad S = \frac{C - \varphi^2}{P - 2|\varphi|}, \quad A = \frac{(\lambda - \lambda_0)^2}{C} - 1,$$

$$B = S^2 - A[C - PS - (\lambda - \lambda_0)^2],$$

and

$$Y = R \operatorname{sign} \varphi \frac{\sqrt{B} - S}{A}, \quad X = R(\lambda - \lambda_0) \sqrt{1 - \frac{Y^2}{R^2} C},$$

where  $Y$  takes the sign of  $\varphi$ . The singularities are shown in Fig. 5.2.

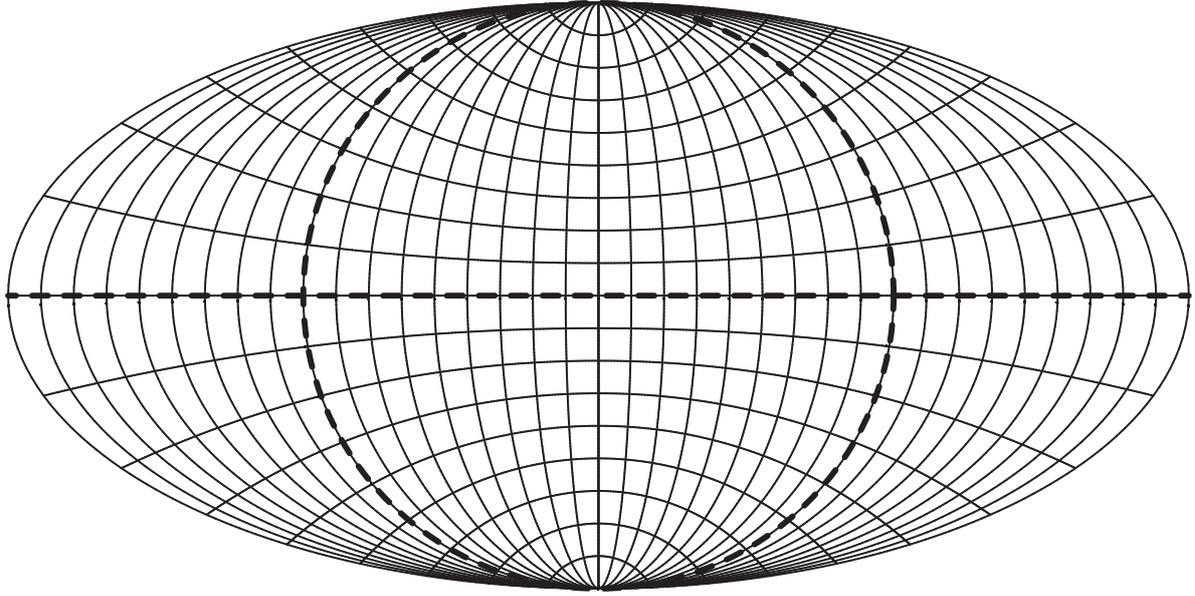


Figure 5.2: Fournier I. projection with the highlighted singularities leading to the subdivision into the tiles; normal aspect of the projection.

**Testing a normal aspect of the projection.** Suppose that the graticule will be generated over the entire planisphere, so  $\Omega = \Omega_\varphi \times \Omega_\lambda$ ,  $\Omega_\varphi = [-\pi/2, \pi/2]$ ,  $\Omega_\lambda = [-\pi, \pi]$  with the offset of meridians and parallels  $\Delta\varphi = \Delta\lambda = 15^\circ$ , the sampling step is  $\alpha = 1^\circ$ . Let us describe the selected steps of the algorithm; see Fig. 5.3:

- Initialize “good data” as  $\Omega_g = \Omega$  and push  $\mathbf{S} \leftarrow \Omega_g$ . Pop the current interval  $\Omega_j = \Omega_{\varphi,j} \times \Omega_{\lambda,j}$  from  $\mathbf{S}$ , detect the removable discontinuity at  $\varphi = 0$ , split  $\Omega_j$  to  $\Omega_{\varphi,j,1} = [-\pi/2, 0)$ ,  $\Omega_{\varphi,j,2} = (0, \pi/2]$ ; see Step b). Create the new subintervals  $\Omega_{j,1} = \Omega_{\varphi,j,1} \times \Omega_{\lambda,j}$ ,  $\Omega_{j,2} = \Omega_{\varphi,j,2} \times \Omega_{\lambda,j}$ , and push  $\mathbf{S} \leftarrow \Omega_{j,1}$ ,  $\mathbf{S} \leftarrow \Omega_{j,2}$ .
- Due to the discontinuity at  $\lambda = -\pi/2$ , the current interval  $\Omega_j$  popped from the stack is split to  $\Omega_{\lambda,j,1} = [-\pi, -\pi/2)$ ,  $\Omega_{\varphi,\lambda,2} = (-\pi/2, \pi]$ ; see Step c). The created subintervals  $\Omega_{j,1} = \Omega_{\varphi,j} \times \Omega_{\lambda,j,1}$ ,  $\Omega_{j,2} = \Omega_{\varphi,j} \times \Omega_{\lambda,j,2}$  are pushed to the stack  $\mathbf{S} \leftarrow \Omega_{j,1}$ ,  $\mathbf{S} \leftarrow \Omega_{j,2}$ .
- Subsequently, the current interval  $\Omega_j = (0, -\pi/2] \times [-\pi, -\pi/2)$  popped out from the stack is free from singularities, the graticule fragment over  $\Omega_j$  is constructed; see Step d).
- Unfortunately, the next current interval  $\Omega_j = (0, -\pi/2] \times (-\pi/2, \pi]$  contains a removable singularity at  $\lambda = \pi/2$ . It is split to  $\Omega_{\lambda,j,1} = [-\pi/2, \pi/2)$ ,  $\Omega_{\varphi,\lambda,2} = (\pi/2, \pi]$ , the created subintervals  $\Omega_{j,1} = \Omega_{\varphi,j} \times \Omega_{\lambda,j,1}$ ,  $\Omega_{j,2} = \Omega_{\varphi,j} \times \Omega_{\lambda,j,2}$  are pushed to the stack  $\mathbf{S} \leftarrow \Omega_{j,1}$ ,  $\mathbf{S} \leftarrow \Omega_{j,2}$ ; see Step e).
- Subsequently, the current interval  $\Omega_j = (0, -\pi/2] \times [-\pi/2, \pi/2)$  popped from the stack is free from singularities, the graticule fragment over  $\Omega_j$  is constructed; see Step f). The lack of discontinuities refers to the next popped interval  $\Omega_j = (0, -\pi/2] \times [\pi/2, \pi)$ , over which the graticule is constructed; see Step g).

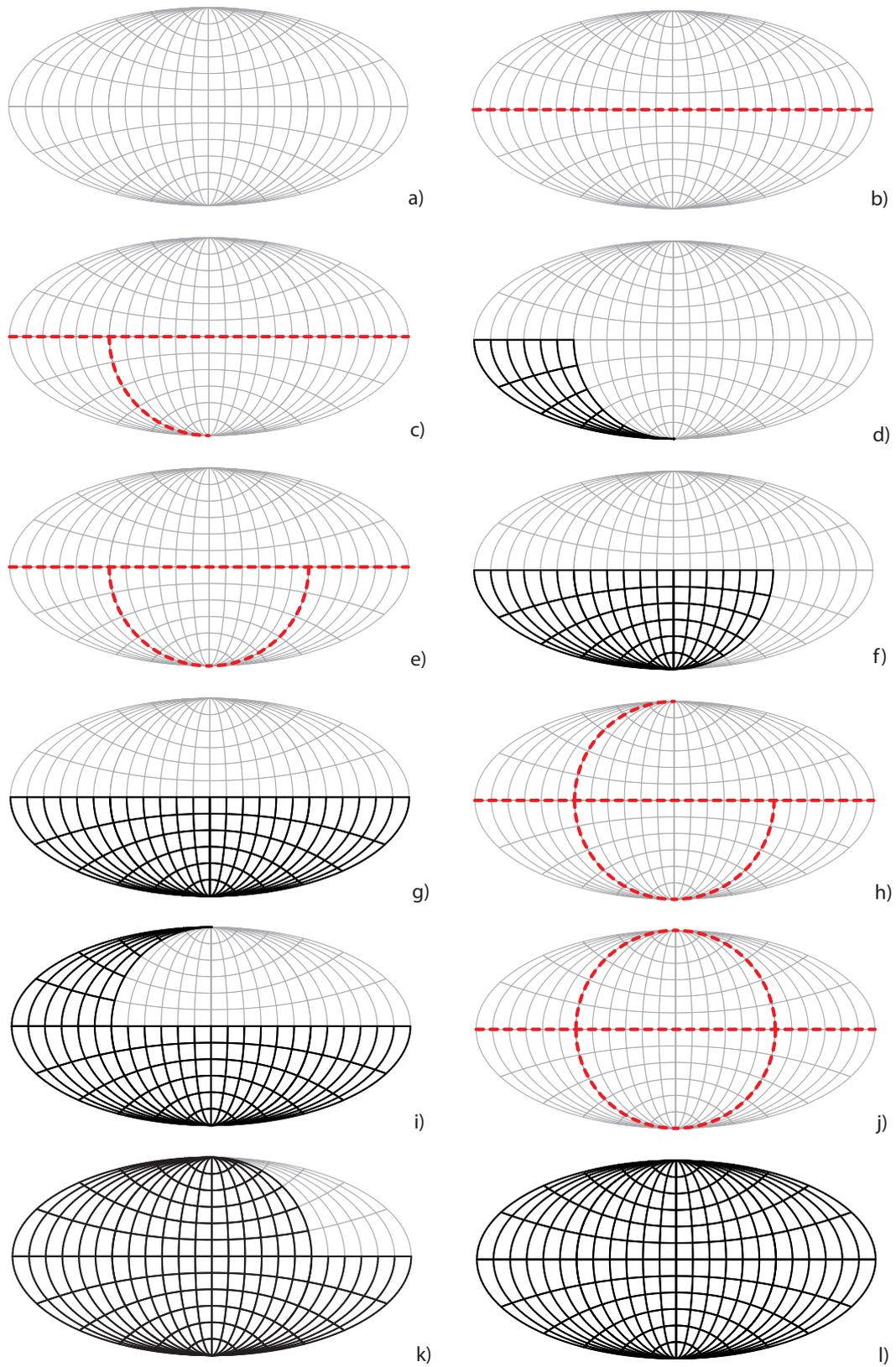


Figure 5.3: Reconstruction of the graticule from the tiles generated along the set of  $\Omega^g$ , Fournier I projection.

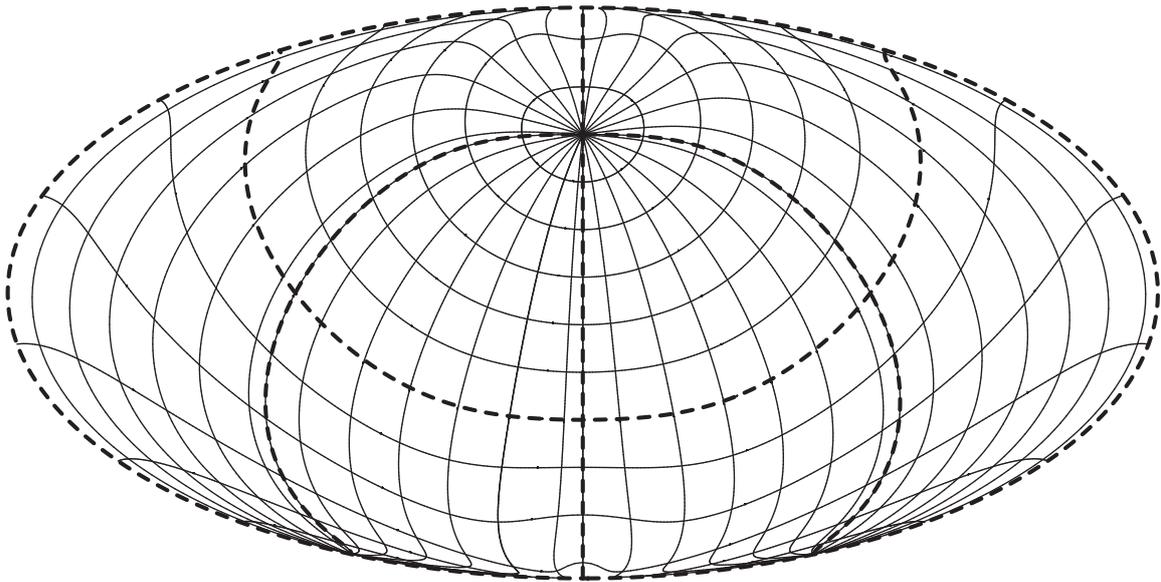


Figure 5.4: Fournier I. projection with the highlighted singularities leading to the subdivision into the tiles; an oblique aspect of the projection,  $K = [50^\circ, 20^\circ]$ .

**Testing an oblique aspect of the projection.** For the oblique aspect of the projection, several new discontinuities appear. Suppose the arbitrary pole  $K = [50^\circ, 20^\circ]$ , the central meridian shift is  $\lambda'_0 = 0^\circ$ . Due to the latitude singularity at  $\varphi_k = 50^\circ$ , the latitude interval  $\Omega_{\varphi,j}$  will be split into two parts:

$$\Omega_{\varphi,j,1} = [-90^\circ, 50^\circ), \quad \Omega_{\varphi,j,2} = (50^\circ, 90^\circ].$$

Analogously, the longitude singularity at  $\lambda_k - 180^\circ = -160^\circ$  splits the interval  $\Omega_{\lambda,j}$  into the three parts:

$$\Omega_{\lambda,j,1} = [-180^\circ, -160^\circ), \quad \Omega_{\lambda,j,2} = (-160^\circ, 15^\circ), \quad \Omega_{\lambda,j,3} = (15^\circ, 180^\circ].$$

The reconstructed graticule with the highlighted singularities is depicted in Fig. 5.4.

### 5.3. Measuring the quantitative parameters of the algorithm

The algorithms have been tested on the construction of the several graticules with the shifted central meridian  $\lambda_0$ . In all cases, the entire planisphere is covered by the graticule with the offset of  $\Delta\varphi = \Delta\lambda = 10^\circ$ . Only the map projections with the singularities are involved in testing: Fournier I., Van der Grinten I., Ortelius Oval, Nicolosi, Hassler. While the Hassler projection has the only singularity at  $\varphi = 0^\circ$ , the Ortelius projection is singular at  $\lambda - \lambda_0 = 0$ , and the Fournier I., Van der Grinten I., and Nicolosi projections add singularities at  $\lambda - \lambda_0 = \pm \frac{\pi}{2}$ . The graphical issues, amount of splits  $n_s$ , processed tiles  $n_t$ , interval resizing  $n_r$ , deleted intervals  $n_d$ , and possible failures of the algorithm will be investigated.

**The impact of the central meridian shift  $\lambda'_0$ .** To verify the properties, behavior and numerical stability of the procedures, the following shifts  $\lambda'_0 = \{0^\circ, 30^\circ, \dots, 150^\circ, 180^\circ\}$  of the

Table 3: The amount of splits  $n_s$ , processed tiles  $n_t$ , interval resizing  $n_r$ , and deleted intervals  $n_d$  for different values of  $\lambda'_0$  representing multiples of  $30^\circ$ .

Projection	$\lambda'_0$	$0^\circ$	$30^\circ$	$60^\circ$	$90^\circ$	$120^\circ$	$150^\circ$	$180^\circ$
Fournier I.	$n_t$	7	3	3	11	3	3	7
	$n_s$	3	1	1	5	1	1	3
	$n_r$	0	0	0	0	0	0	0
	$n_d$	0	0	0	0	0	0	0
Van der Grinten I.	$n_t$	7	3	3	3	3	3	3
	$n_s$	3	1	1	1	1	1	1
	$n_r$	0	0	0	0	0	0	0
Ortelius Oval	$n_t$	3	1	1	1	1	1	1
	$n_s$	1	1	1	1	1	1	1
	$n_r$	0	0	0	0	0	0	0
	$n_d$	0	0	0	0	0	0	0
Nicolosi	$n_t$	7	3	3	7	3	3	3
	$n_s$	3	1	1	3	1	1	1
	$n_r$	0	0	0	0	0	0	0
	$n_d$	0	0	0	0	0	0	0
Hassler	$n_t$	1	1	1	1	1	1	1
	$n_s$	1	1	1	1	1	1	1
	$n_r$	0	0	0	0	0	0	0
	$n_d$	0	0	0	0	0	0	0

Table 4: The amount of splits  $n_s$ , processed tiles  $n_t$ , interval resizing  $n_r$ , and deleted intervals  $n_d$  for different values of  $\lambda'_0$  which are not the exact multiples of  $30^\circ$ .

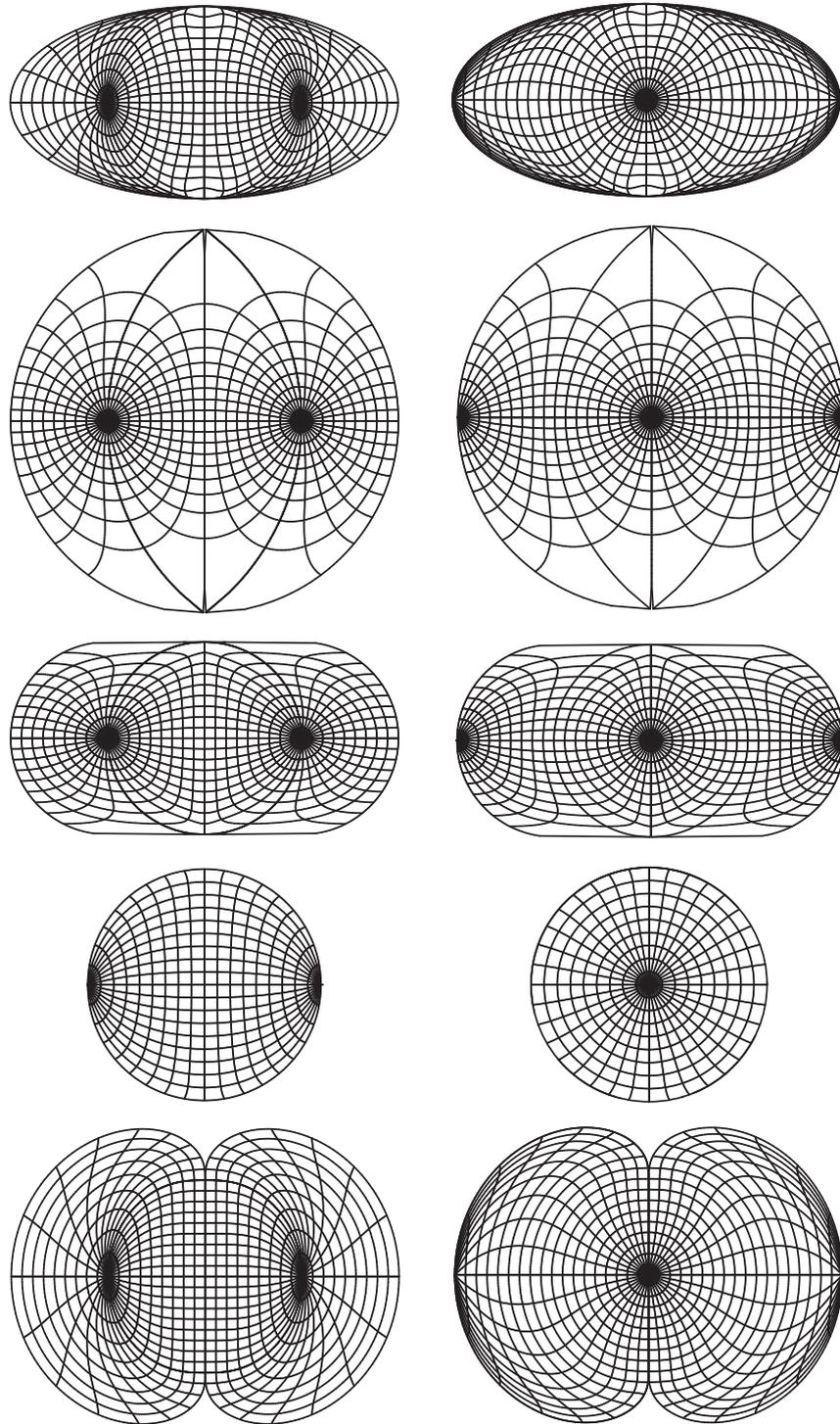
Projection	$\lambda'_0$	$0.001^\circ$	$30.001^\circ$	$60.001^\circ$	$90.001^\circ$	$120.001^\circ$	$150.001^\circ$	$180.001^\circ$
Fournier I.	$n_t$	5	3	3	8	3	3	5
	$n_s$	1	1	1	3	1	1	1
	$n_r$	1	0	0	0	0	0	0
	$n_d$	0	0	0	0	0	0	0
Van der Grinten I.	$n_t$	4	3	3	4	3	3	4
	$n_s$	1	1	1	1	1	1	1
	$n_r$	0	0	0	1	0	0	0
Ortelius Oval	$n_t$	2	1	1	2	1	1	1
	$n_s$	0	0	0	0	0	0	0
	$n_r$	0	0	0	0	0	0	0
	$n_d$	0	0	0	0	0	0	0
Nicolosi	$n_t$	5	3	3	8	3	3	5
	$n_s$	1	1	1	3	1	1	1
	$n_r$	1	0	0	0	0	0	1
Hassler	$n_t$	4	3	3	3	3	3	4
	$n_s$	1	1	1	1	1	1	1
	$n_r$	0	0	0	0	0	0	0
	$n_d$	0	0	0	0	0	0	0

Table 5: The amount of splits  $n_s$ , processed tiles  $n_t$ , interval resizing  $n_r$ , deleted intervals  $n_d$  depending on  $\lambda'_0$  and the threshold  $\varepsilon$ .

Projection	$\lambda'_0$	$0.001^\circ$	$30.001^\circ$	$60.001^\circ$	$90.001^\circ$	$120.001^\circ$	$150.001^\circ$	$179.999^\circ$
Fournier I.	$n_t$	4/17/145/1435	3/17/145/1435	3/17/145/1435	13/65/661/6591	3/17/145/1435	3/17/145/1435	4/17/145/1435
	$n_s$	1/1/1/2	1/1/1/2	1/1/1/2	5/5/5/6	1/1/1/2	1/1/1/2	1/1/1/2
	$n_r$	1/7/71/1212	0/7/71/1212	0/7/71/1212	1/27/325/5786	0/7/71/1212	0/7/71/1212	1/7/71/1212
	$n_d$	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1
	$n_t$	5/27/265/2633	5/27/259/2577	5/27/259/2577	5/27/259/2577	5/27/259/2577	5/27/259/2577	5/27/259/2633
Van der Grinten I.	$n_s$	1/1/3/4	1/1/1/2	1/1/1/2	1/1/1/2	1/1/1/2	1/1/1/2	1/1/1/2
	$n_r$	1/12/130/2311	1/12/128/2283	1/12/128/2283	1/12/128/2283	1/12/128/2283	1/12/128/2283	1/12/128/2311
	$n_d$	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1
	$n_t$	2/2/5/58	1/1/1/1	1/1/1/1	2/2/1/1	1/1/1/1	1/1/1/1	2/2/5/58
Ortelius Oval	$n_s$	0/0/1/3	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/1/3
	$n_r$	0/0/1/38	0/0/0/0	0/0/0/0	1/1/0/0	0/0/0/0	0/0/0/0	1/1/2/39
	$n_d$	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0
	$n_t$	5/47/453/4519	3/47/453/4519	3/47/453/4519	13/95/969/9576	3/47/453/4519	3/47/453/4519	4/47/453/4519
Nicolosi	$n_s$	1/1/1/2	1/1/1/2	1/1/1/2	5/5/5/6	1/1/1/2	1/1/1/2	1/1/1/2
	$n_r$	1/22/225/4254	0/22/225/4254	0/22/225/4254	1/42/479/8828	0/22/225/4254	0/22/225/4254	1/22/225/4254
	$n_d$	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1	0/0/0/1	0/0/1/1
	$n_t$	3/3/5/31	3/3/5/31	3/3/5/31	3/3/5/31	3/3/5/31	3/3/5/31	3/3/5/31
Hassler	$n_s$	1/1/1/1	1/1/1/1	1/1/1/1	1/1/1/1	1/1/1/1	1/1/1/1	1/1/1/1
	$n_r$	0/0/1/14	0/0/1/14	0/0/1/14	0/0/1/14	0/0/1/14	0/0/1/14	0/0/1/14
	$n_d$	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0	0/0/0/0
	$n_t$	3/3/5/31	3/3/5/31	3/3/5/31	3/3/5/31	3/3/5/31	3/3/5/31	3/3/5/31

central meridian representing multiples of  $30^\circ$ , are involved. All projections are proposed in the oblique aspect, where  $K = [50^\circ, 20^\circ]$ . The results are summarized in Tab. 3. It is evident that the quantitative parameters of the algorithm depend on  $\lambda'_0$ . If  $\lambda'_0$  coincides with the singularity, the number of interval corrections as well as the amount of processed tiles increase and vice versa; the maximum values can be found for  $\lambda'_0 = \{0^\circ, 90^\circ, 180^\circ\}$ , which is in accordance with the detected singularities. Taking into account the complexity of equations, the less complicated formulas with few singularities are constructed from a smaller amount of tiles. Due to the position of  $K$  and  $\lambda'_0$ , there are neither resized nor empty intervals. In general, only a few subdivisions are carried out.

**The impact of the numerical inaccuracy.** The impact of numerical inaccuracies will also be illustrated. All projections are proposed in the transverse aspect, where  $K = [0^\circ, 90^\circ]$ . The central meridian shifts are not the exact multiples of  $30^\circ$ , but  $\lambda'_0 = \{0.001^\circ, 30.001^\circ, \dots, 150.001^\circ, 179.999^\circ\}$ . Therefore, some resizing steps are expected. The results are summarized in Tab. 4. Comparing the quantitative parameters, the number of processed tiles  $n_t$  decreased, but some intervals need to be resized. After this refinement they become correct; only a single resizing step is required. However, there was no need for a deletion of the empty interval. It is evident that “simple” equations are insensitive to the projection aspect.



*Figure 5.5: The list of reconstructed graticules of Fournier I., Van der Grinten I., Ortelius Oval, Nicolosi, and Hassler projections using combined sampling,  $\bar{\alpha} = 5^\circ$ ; the central meridian shifts  $\lambda'_0 = 90.001^\circ$ ,  $\lambda'_0 = 179.999^\circ$ .*

**The impact of the central meridian shift  $\lambda'_0$  and the numerical threshold  $\varepsilon$ .** The last test measures the quantitative parameters of the algorithms depending on the value of the threshold  $\varepsilon$ . This value has a strong influence on whether the result will be classified as a singularity. For testing, the following values are involved:  $\varepsilon = \{1.0e^{-6}, 1.0e^{-7}, 1.0e^{-8}, 1.0e^{-9}\}$ ; the cartographic parameters remain unchained. The results are summarized in Tab. 5. It is evident that the amount of tiles  $n_t$  strongly depends on  $\varepsilon$ , especially if a singularity at  $\lambda'$  exists. Increasing  $\varepsilon$  by one order of magnitude, the amount of tiles  $n_t$  changes in the same manner. Taking into account the results, it does not make a sense to use values  $\varepsilon < 1.0e^{-7}$ . The algorithm is slowing down; its computational demands significantly grow up. It also becomes unstable and may lead to the stack overflow. Comparing the analyzed projections, the worst results are provided by the Nicolosi projection, where  $n_t$  almost doesn't depend on  $\lambda'_0$ . In the most pessimistic case, almost 10 000 tiles are required. On the contrary, the Hassler projection with  $n_t < 35$  needs less effort.

Summarize the facts, the reconstruction algorithm depends on the number of singularities as well as on the threshold  $\varepsilon$ . For  $\varepsilon \geq 1.0e^{-6}$ , only a few splits are required. All the reconstructed graticules of the analyzed projections for  $\lambda'_0 = 90.001^\circ$ , and  $\lambda'_0 = 179.999^\circ$  can be found in Fig. 5.5.

## 6. Conclusion

This article presented a new algorithm for the combined sampling of the projection graticule handling all constant values of the projection. The proposed method combines the uniform and adaptive sampling techniques with the triple recursive subdivision. It controls handling the discontinuities and their directions; the detection criterion is presented in Sec. 3.1. In general, the splitting procedure is sensitive to  $\varepsilon$  (if  $\lambda'_0 \rightarrow \varepsilon$ ) as well as to the numerical errors of intersections  $m'(\lambda)$  and  $m(\lambda)$ ,  $m'(\lambda)$  and  $p(\varphi)$ . From the topological point of view, the meridian and parallel intersections lack the vertex, which can cause some problems in GIS processing; the dissolve operation needs to be applied before. The discontinuity detection without its classification (or, a wrong classification) leads to the duplication of lines along the singularities.

The idea of splitting the interval to the set of disjoint intervals without the singularities is efficient and works for any common projection and its coordinate functions. Compared to the uniform sampling, our experiments confirmed that the combined sampling techniques S1-S3 require fewer data in regions with the high curvature (more than five times lower values of  $\bar{\alpha}$ ). For the circular arcs, circles or ellipses the uniform sampling is slightly more efficient. The lowest data redundancy refers to the straight lines, only 1% of the uniform sampling data is sufficient for the polygonal approximation. This is beneficial for the cylindrical, conic, azimuthal and pseudocylindrical projections. A minor disadvantage of the proposed solution is a slightly complicated implementation of the algorithms, the source code in C++/Java is available in the GitHub repository

<https://github.com/bayertom/graticule>.

## References

- [1] Giampietro Allasia, Renata Besenghi, and Roberto Cavoretto. “Adaptive detection and approximation of unknown surface discontinuities from scattered data”. In: *Simulation Modelling Practice and Theory* 17.6 (2009), pp. 1059–1070.
- [2] Francesc Arandiga et al. “Interpolation and Approximation of Piecewise Smooth Functions”. In: *SIAM Journal on Numerical Analysis* 43.1 (2005), pp. 41–57. DOI: [10.1137/S0036142903426245](https://doi.org/10.1137/S0036142903426245).
- [3] Rick Archibald, Anne Gelb, and Jungho Yoon. “Determining the locations and discontinuities in the derivatives of functions”. In: *Applied Numerical Mathematics* 58.5 (2008), pp. 577–592.
- [4] Tomas Bayer. “Advanced methods for the estimation of an unknown projection from a map”. In: *GeoInformatica* 20.2 (2016), pp. 241–284. DOI: [10.1007/s10707-015-0234-x](https://doi.org/10.1007/s10707-015-0234-x).
- [5] Tomas Bayer. “Efficient plotting of functions with discontinuities based on combined sampling”. English. In: *FCE GeoInformatics* 18.3 (2018), pp. 621–669. ISSN: 1384-6175. DOI: [10.1007/s10707-013-0200-4](https://doi.org/10.1007/s10707-013-0200-4).
- [6] Tomas Bayer. “Estimation of an unknown cartographic projection and its parameters from the map”. English. In: *GeoInformatica* 18.3 (2014), pp. 621–669. ISSN: 1384-6175. DOI: [10.1007/s10707-013-0200-4](https://doi.org/10.1007/s10707-013-0200-4).
- [7] P. Binev et al. “Adaptive approximation of curves”. In: *Approximation Theory* (2004), pp. 43–57.
- [8] Mira Bozzini, Licia Lenarduzzi, and Milvia Rossini. “Non-regular Surface Approximation”. In: *Mathematical Methods for Curves and Surfaces: 8th International Conference, MMCS 2012, Oslo, Norway, June 28 – July 3, 2012, Revised Selected Papers*. Ed. by Michael Floater et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 68–87. ISBN: 978-3-642-54382-1. DOI: [10.1007/978-3-642-54382-1\\_5](https://doi.org/10.1007/978-3-642-54382-1_5).
- [9] Mira Bozzini, Licia Lenarduzzi, and Robert Schaback. “Adaptive Interpolation by Scaled Multiquadrics”. In: *Advances in Computational Mathematics* 16.4 (May 2002), pp. 375–387. ISSN: 1572-9044. DOI: [10.1023/A:1014584220418](https://doi.org/10.1023/A:1014584220418).
- [10] Filipe de Carvalho Nascimento et al. “Approximating implicit curves on plane and surface triangulations with affine arithmetic”. In: *Computers & Graphics* 40 (2014), pp. 36–48.
- [11] Andrew Crampton and John C Mason. “Detecting and approximating fault lines from randomly scattered data”. In: *Numerical Algorithms* 39.1 (2005), pp. 115–130.
- [12] Luiz Henrique de Figueiredo. “Adaptive sampling of parametric curves”. In: *Graphics Gems V* 5 (1995), pp. 173–178.
- [13] Tim Gutzmer and Armin Iske. “Detection of discontinuities in scattered data approximation”. In: *Numerical Algorithms* 16.2 (1997), pp. 155–170.
- [14] John Krumm. *Intersection of Two Planes*. 2016.
- [15] David Lee. “Detection, Classification, and Measurement of Discontinuities”. In: *SIAM Journal on Scientific and Statistical Computing* 12.2 (1991), pp. 311–341. DOI: [10.1137/0912018](https://doi.org/10.1137/0912018).

- [16] Licia Lenarduzzi and Robert Schaback. “Kernel-based adaptive approximation of functions with discontinuities”. In: *Applied Mathematics and Computation* 307.Supplement C (2017), pp. 113–123. ISSN: 0096-3003. DOI: <https://doi.org/10.1016/j.amc.2017.02.043>.
- [17] Hélio Lopes, João Batista Oliveira, and Luiz Henrique de Figueiredo. “Robust adaptive polygonal approximation of implicit curves”. In: *Computers & Graphics* 26.6 (2002), pp. 841–852.
- [18] María Cruz López de Silanes, María Cruz Parra, and Juan José Torrens. “On a New Characterization of Finite Jump Discontinuities and Its Application to Vertical Fault Detection”. In: *Math. Comput. Simul.* 77.2-3 (Mar. 2008), pp. 247–256. ISSN: 0378-4754. DOI: [10.1016/j.matcom.2007.08.008](https://doi.org/10.1016/j.matcom.2007.08.008).
- [19] María Cruz López de Silanes, María Cruz Parra, and Juan José Torrens. “Vertical and oblique fault detection in explicit surfaces”. In: *Journal of Computational and Applied Mathematics* 140.1 (2002). Int. Congress on Computational and Applied Mathematics 2000, pp. 559–585. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/S0377-0427\(01\)00601-X](https://doi.org/10.1016/S0377-0427(01)00601-X).
- [20] M Oliveria et al. “Universal high order subroutine with new shock detector for shock boundary layer interaction”. In: *In other words* 10 (2009), pp. 1–2.
- [21] Afonso Paiva et al. “Approximating implicit curves on triangulations with affine arithmetic”. In: *Graphics, Patterns and Images (SIBGRAPI), 2012 25th SIBGRAPI Conference on.* IEEE. 2012, pp. 94–101.
- [22] M. Rossini. “2D-discontinuity detection from scattered data”. In: *Computing* 61.3 (Sept. 1998), pp. 215–234. ISSN: 1436-5057. DOI: [10.1007/BF02684351](https://doi.org/10.1007/BF02684351).
- [23] Moshe Shpitalni, Yoram Koren, and CC Lo. “Realtime curve interpolators”. In: *Computer-Aided Design* 26.11 (1994), pp. 832–838.
- [24] John P. Snyder. *Map projections – A working manual*. Tech. rep. 1395. U. S. Geological Survey, 1987. URL: <http://pubs.er.usgs.gov/usgspubs/pp/pp1395>.
- [25] Daniel C.H. Yang and Tom Kong. “Parametric interpolator versus linear interpolator for precision CNC machining”. In: *Computer-Aided Design* 26.3 (1994). Special Issue:NC machining and cutter-path generation, pp. 225–234. ISSN: 0010-4485. DOI: [https://doi.org/10.1016/0010-4485\(94\)90045-0](https://doi.org/10.1016/0010-4485(94)90045-0).
- [26] S-S Yeh and P-L Hsu. “The speed-controlled interpolator for machining parametric curves”. In: *Computer-Aided Design* 31.5 (1999), pp. 349–357.
- [27] Syh-Shiuh Yeh and Pau-Lo Hsu. “Adaptive-feedrate interpolation for parametric curves with a confined chord error”. In: *Computer-aided design* 34.3 (2002), pp. 229–237.

