Acta Geophys.
DOI 10.1007/s11600-017-0003-3

RESEARCH ARTICLE

Sibsonian and non-Sibsonian natural neighbour interpolation of the total electron content value

Kacper Kotulak[1] · Adam Froń[1] · Andrzej Krankowski[1] · German Olivares Pulido[2] · Manuel Henrandez-Pajares[3]

Received: 3 January 2017 / Accepted: 4 January 2017

Abstract In radioastronomy the interferometric measurement between radiotelescopes located relatively close to each other helps removing ionospheric effects. Unfortunately, in case of networks such as LOw Frequency ARray (LOFAR), due to long baselines (currently up to 1500 km), interferometric methods fail to provide sufficiently accurate ionosphere delay corrections. Practically it means that systems such as LOFAR need external ionosphere information, coming from Global or Regional Ionospheric Maps (GIMs or RIMs, respectively). Thanks to the technology based on Global Navigation Satellite Systems (GNSS), the scientific community is provided with ionosphere sounding virtually worldwide. In this paper we compare several interpolation methods for RIMs computation based on scattered Vertical Total Electron Content measurements located on one thin ionospheric layer (Ionospheric Pierce Points—IPPs). The results of this work show that methods that take into account the topology of the data distribution (e.g., natural neighbour interpolation) perform better than those based on geometric computation only (e.g., distance-weighted methods).

Keywords TEC · Interpolation · Natural neighbour interpolation

Introduction

Information regarding the state of the ionosphere is paramount for the performance of scientific and technological infrastructures such as, for example, power grid networks, pipelines and satellite-based navigation services (e.g., GPS, GLONASS, and soon, Galileo and Beidou too).

It is well known that at short baselines (e.g., less than ~50 km), most of the ionospheric effects may be removed by means of interferometric techniques. Otherwise, for longer baselines, the ionospheric variability must be taken into account.

Interferometric techniques are also widely used in radioastronomy. Nevertheless, radio-telescope networks that operate along baselines longer than 50 km may not be able to remove the ionospheric delay of the signal by means of interferometry between nearby stations. That is the case of LOFAR project (http://www.lofar.org/), a network of radio-telescopes in Europe that operates at the low frequency band of the electromagnetic spectrum (10–240 MHz) with baselines as long as 1500 km. Consequently, LOFAR network requires ionospheric modelling in order to minimize the degradation of the observed signal.

Among several experiments and goals of LOFAR, the ionosphere may affect the observational data that are used

✉ Kacper Kotulak
kacper.kotulak@uwm.edu.pl

Adam Froń
adam.fron@uwm.edu.pl

Andrzej Krankowski
kand@uwm.edu.pl

German Olivares Pulido
g.olivares@bom.gov.au

Manuel Henrandez-Pajares
manuel.hernandez@upc.edu

[1] Space Radio-Diagnostics Research Centre, University of Warmia and Mazury, Olsztyn, Poland

[2] CRC for Spatial Information, Victoria, Australia

[3] Department of Applied Mathematics IV, Ionospheric Determination and Navigation Based on Satellite and Terrestrial Systems Research Team, Barcelona, Spain

Published online: 21 January 2017

to constrain theoretical models of the epoch of the reionization, solar science, space weather and cosmic magnetism (see www.lofar.org for further details). One of the Polish LOFAR part tasks is high-resolution monitoring of radio sources in cooperation with GNSS to study ionosphere influence on low-frequency radiowaves. Possibilities of ionospheric, space weather and solar studies were described by Dąbrowski et al. (2016).

The incapability of interferometric techniques of minimizing the impact of the ionospheric delay on the signal might be overcome by using either GIMs or RIMs which provide insight into the state of the ionosphere. Indeed, the International GNSS Service (IGS) provides GIMs through four IGS Ionosphere Associate Analysis Centres (IAACs); CODE, ESA (European Space Agency), JPL (Jet Propulsion Laboratory) and UPC (Universitat Politecnica de Catalunya) (Komjathy et al. 2010; Schaer 1999).

These global ionospheric maps are computed using Total Electron Content (TEC) data provided by GNSS-based ground receivers (Hernández-Pajares et al. 2011). Due to the lack of homogeneity of the GNSS networks worldwide, interpolation techniques are used for filling the gaps (Orus et al. 2005 and citations therein). However, GIMs do not provide the highest resolution for local areas. In such case, RIMs should be used instead (Jakowski et al. 2005).

There are few GNNS ground receiver networks that may provide high-resolution RIMs, such as, for example, GEONET (http://www.gsi.go.jp/), in Japan and CRTN in California (http://sopac.ucsd.edu/). In Europe, EUREF permanent network (http://www.epncb.oma.be/) has longer inter-station distance than the abovementioned networks.

The accuracy of GIMs and RIMs depends on the accuracy of TEC data and ionospheric modelling. For example, GIMs provided by the UPC are computed using tomographic Slant TEC (STEC) data (Hernandez-Pajares et al. 1999) and the use of fast and accurate ways of solving the associated topological problem has also been studied (see Hernández-Pajares et al. 1997). Then, gaps (due to the lack of receivers in the Southern Hemisphere and the non-homogeneous distribution of GNSS ground receivers) are filled by estimating the correlation between the estimated uncertainties with the kriging method (Orus et al. 2005).

Other ionospheric modelling methods are data-assimilative models (GAIM, at the Jet Propulsion Laboratory—Schunk 2002; Pi et al. 2003, 2004, 2009; Hajj 2004; Wang 2004; Mandrake et al. 2005; Scherliess et al. 2006; Komjathy et al. 2010), harmonic spherical functions (Schaer 1999; Choi et al. 2010; Han et al. 2013) and empirical models (e.g., the International Reference Ionosphere project—IRI, http://iri.gsfc.nasa.gov/) (Strangeways 2009; Buresova et al. 2009).

It is also important to take into account the time scale of the ionospheric variability. The time-scale spectrum spans from seconds (e.g., scintillations) to years (Solar cycle), thus making the ionospheric temporal variability modelling a real challenge (Krankowski et al. 2007).

In general, the criteria to choose amongst several ionospheric modelling methods may be based on their use. For example, for radio-telescope networks, it is important to provide ionospheric information in real time within the area of coverage. In this regard, tomographic models can estimate the state of the ionosphere in real time (Hernández-Pajares et al. 2000, 2011).

As for the spatial resolution, any model or interpolation method degrades the uncertainty of the raw STEC data, no matter how good it may be. Consequently, for the computation of RIMs, it is also of great importance to find the best spatial modelling available, that is to say, the model that yields the minimum discrepancy with real data.

In this regard, this work presents a comparison of several interpolation methods used for computing RIMs over Europe using GPS-ground receiver stations from the EUREF network in Poland. All these methods use STEC data computed by the UPC TOmographic Model of the IONosphere software (UPC TOMION).

This work is organized as follows: the second section presents a description of the interpolation methods; the third section presents the STEC data set and the results; the fourth section contains the solutions to some interpolation problems; finally, the fifth section summarizes the conclusions.

Description of chosen interpolation methods

Inverse distance weighting (IDW)

Inverse distance weighting can be considered as one of the most intuitive and easiest methods of evaluating the influence of values of certain phenomena measured in less or more scattered data points with unknown values at any arbitrary point.

The general idea of inverse distance weighting is the lowering of influence with increasing distance between the points. The result of the interpolation at each point is a weighted average of the values of the sample or data set, where the weights are decreasing proportionally to a power function of increasing distance between the points (Dumitru 2013). As a degree of IDW we consider an exponentiation of X lying in the denominator (meaning x^{-1} as the first degree, x^{-2} as the second, and so on). For our comparison we have used the second degree of IDW, which is also the most commonly used one.

Two scenarios of IDW method implementation can be considered—global and local. The global IDW interpolation assumes that all the data points have an influence on the interpolated points. In our comparison we have used a kind of local approach, where only an arbitrary number of points (7—which was the average number of neighbours in our natural interpolation) selected based on the closest distance, is considered to influence the value of the interpolated point.In our comparison the values at each point are evaluated using the following formula:

$$F(X,Y) = \sum_{i=1}^{n} w_i f_i, \tag{1}$$

where n is the number of neighbouring points, f_i is the function value at each neighbouring point and w_i is a normalized weight:

$$w_i = \frac{\mathrm{d}_i^{-2}}{\sum_{i=1}^{n} w_i f_i}, \tag{2}$$

where d stands for the distance between each data point and the interpolated point.

Polynomial interpolation

Polynomial fitting differs significantly from the other methods. Indeed, the computed surface does not fit the data points, but instead it lies within a certain distance from them.

For the approximation of the polynomial coefficients, a_1, a_2, …, a_n, we have used the Moore–Penrose pseudoinverse (see Moore 1920) based on the whole set of data points for each epoch. Similarly to IDW, polynomials can be evaluated to a certain degree, resulting in a variety of computed surfaces (e.g., a plane, a bilinear surface, a cubic area, etc.). In our comparison we have used a force method to evaluate an optimal polynomial degree for each epoch (of course in a certain range, as the pseudoinverse gives the possibility of creating polynomials of a degree greatly exceeding the number of data points).

Figure 1 shows the number of coefficients of the polynomial equation in terms of its order. For the selected (6^{th}) order, the polynomial equation can be expressed as follows:

$$f(x,y) = \sum_{i=0}^{6} \left(\sum_{j=0}^{6-i} a_{ij} x^i y^j \right), \tag{3}$$

where a_{ij} are the polynomial coefficients and x, y are the coordinates of each point.

For low degree (up to 10), the polynomial method returns a surface which approximates data smoothly, and it is easily computed, resulting in a continuous surface that easily provides a value for any arbitrary point.

Voronoi diagram and Delaunay triangulation

Natural neighbour interpolation is based on the Voronoi diagram and Delaunay triangulation, which defines the topology in terms of the distribution of nearest neighbours around each point (Sukumar et al. 2001).

To illustrate the idea of Voronoi diagram (often called Dirichlet tessellation or Thiessen diagram) we will consider dataset that consisted of scattered points, called Voronoi sites.

Each site corresponds to a certain Voronoi cell (also called Voronoi face), which defines the range of points that are closer to the chosen site than to other sites. The Voronoi diagram is simply the sum of cells corresponding to all Voronoi sites (Fortune 1995). That leads to the conclusion that a Voronoi diagram is actually a realization of the nearest neighbour interpolation (see Fig. 2).

Voronoi diagram is strictly related to the Delaunay triangulation. Delaunay triangulation is the realization of a set of triangles created in such way that each triangle circumcircle does not have any Voronoi site in its interior (Fortune 1995).

Voronoi edge is a mid-perpendicular of a Delaunay triangle edge. In other words, the Delaunay edge existence determines the existence of Voronoi edge and vice versa.

Voronoi diagrams determine natural neighbourhood—two Voronoi sites are each other's natural neighbour if there exists a Voronoi edge between them. The same rule can be applied to the Delaunay triangulation—two points are their own natural neighbours if they are linked by a Delaunay edge (see Fig. 3).

Both Voronoi diagram and Delaunay triangulation can be used in interpolation—but only Delaunay triangulation can be used separately, as the easiest way to compute the Voronoi diagram requires also a Delaunay triangulation.

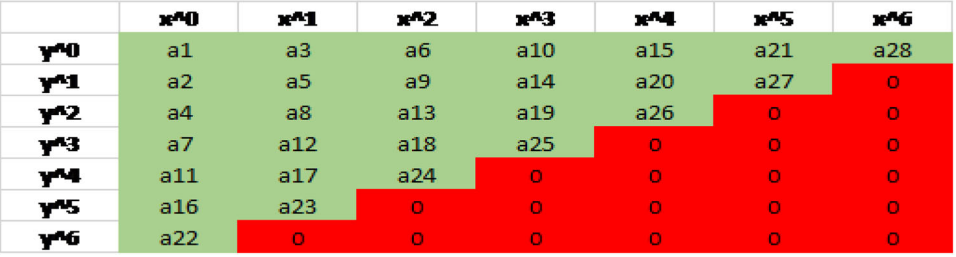

Fig. 1 The polynomial parameters for 6th order polynomial equation in regard to two variables (coordinates)

	x^0	x^1	x^2	x^3	x^4	x^5	x^6
y^0	a1	a3	a6	a10	a15	a21	a28
y^1	a2	a5	a9	a14	a20	a27	0
y^2	a4	a8	a13	a19	a26	0	0
y^3	a7	a12	a18	a25	0	0	0
y^4	a11	a17	a24	0	0	0	0
y^5	a16	a23	0	0	0	0	0
y^6	a22	0	0	0	0	0	0

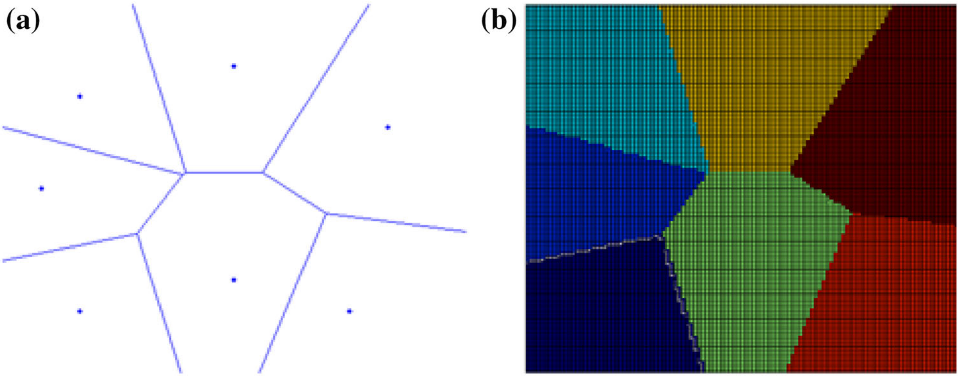

Fig. 2 Graphical comparison between Voronoi diagram (**a**) and discrete nearest neighbour interpolation of regular grid set of 10,000 points (**b**)

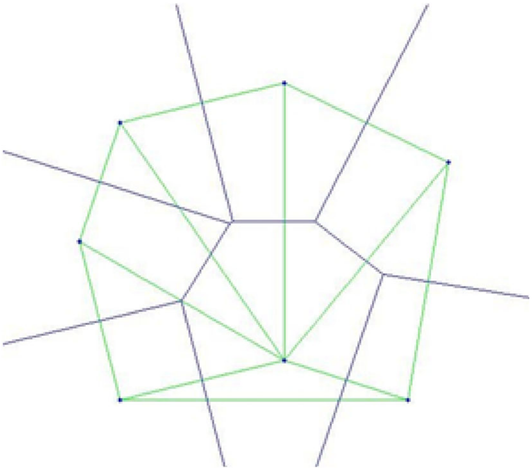

Fig. 3 Voronoi diagram and Delaunay triangulation put together (Voronoi edges are Delaunay edges' mid-perpendiculars, but not the reverse). Not every Voronoi edge can be seen

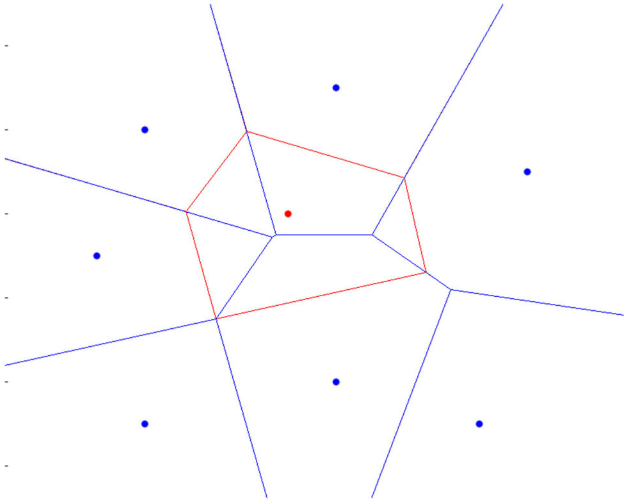

Fig. 4 Two orders of Voronoi diagram—the first (*blue*) consisting of known points and their cells, and an overlapping cell of the inserted (and then interpolated) point (*red*) of the second order

Natural neighbour interpolation

As previously said, natural neighbour interpolation is a method based on the Voronoi diagram. In calculations we consider two orders of the diagram—the first containing only scattered known data points, and the second order, which is based on the first, but with one of the interpolated points inserted into the data (Fig. 4).

The insertion of a new point into an existing Voronoi diagram results in the creation of a new cell, the area of which 'steals' parts of the neighbouring cells, while the whole rest of the diagram remains unchanged.

The weights of all neighbours of the inserted point are then evaluated as a ratio of the stolen (overlapping) area to the area of the whole new cell, as shown by Eqs. 4 and 5 below (Sibson 1981):

$$f_x = \sum_{m=1}^{M} \alpha_m f_m, \tag{4}$$

where f_x is the value for the interpolated point, M is the number of natural neighbours (number of neighbouring Voronoi cells), f_m describes the value of the *m-th* neighbour and α_m is a weight coefficient computed with Eq. 5 as follows:

$$\alpha_m = \frac{s_m}{s_0}, \tag{5}$$

where s_m is the stolen area of m-th cell and s_0 is the whole area of the new cell (see Fig. 5).

Overlapping areas can be computed in two different ways. In the first scenario, the overlapping area is acquired by subtracting the neighbouring cells area before entering the new point and the same cell after the introduction of such point (Harman 2008). The second scenario is to compute the overlapping area as the area of the intersection of the new cell and the neighbouring cell.

The first scenario is easier to calculate, but much more sensitive and exposed to errors caused by open Voronoi cells. Those errors will be further discussed in later sections of this work.

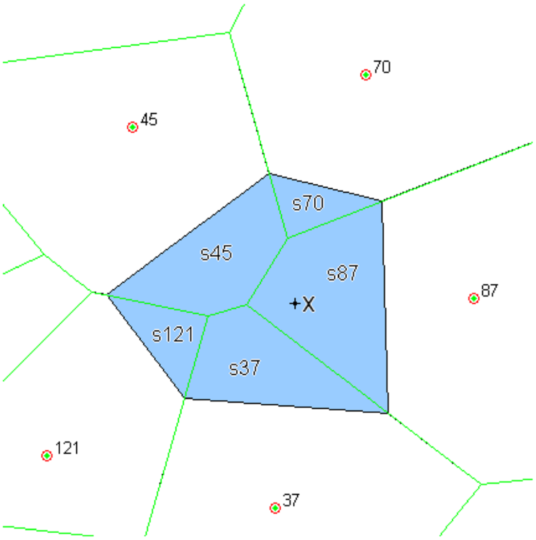

Fig. 5 The area of the new cell divided into sections 'stolen' from each neighbour

Non-Sibsonian interpolation

The non-Sibsonian interpolation method is actually a natural neighbour method; therefore, it is also based on Voronoi diagram computation. Nevertheless, unlike the Sibsonian method, this method uses linear quantities instead of areas. This approach solves some of the problems the Sibsonian method faces (e.g., open Voronoi cells).

According to Belikov et al. (1997), the interpolated point can be computed as a linear combination of the neighbours as follows:

$$f_x = \sum_{m=1}^{M} \alpha_m f_m, \tag{6}$$

where f_x is the value of the interpolated point x, M is the number of natural neighbours (i.e., number of Voronoi cell edges), f_m is the value of the m-th neighbour and α_m is a weight coefficient described by the equation given below:

$$\alpha_m = \frac{\mathrm{d}_m}{h_m} \tag{7}$$

with d_m the Voronoi edge length and h_m standing for half the distance between the interpolated point x and the Voronoi edge m (see Fig. 6).

Belikov et al. (1997) note that such a way of computing weights is far easier and more efficient than the Sibsonian approach, which is based on polygons' areas. Moreover, the non-Sibsonian method does not require building two-order Voronoi diagrams. Finally, it is also free from some area calculation problems, which are present in the Sibsonian natural neighbour method and will be further discussed in posterior sections.

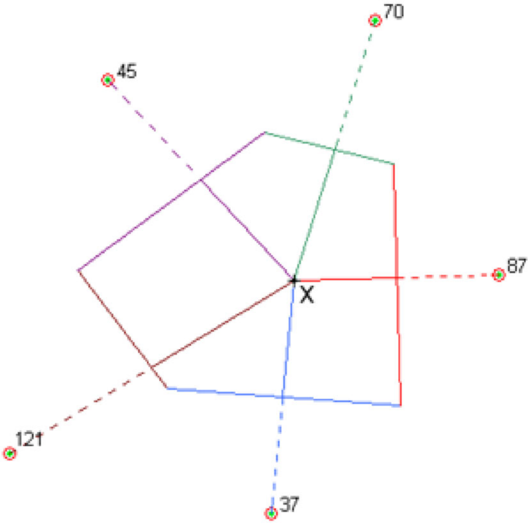

Fig. 6 Non-Sibsonian interpolation mechanism. Edges and distances of the *same colour* correspond to pairs of d_m and h_m, respectively, used to compute α_m

Quasi-natural interpolation

Quasi-natural method is a composition of the inverse distance weighting interpolation method and Delaunay triangulation. We use prefix „quasi", as the method is not actually based on Voronoi diagram and its features, like classical natural methods. Delaunay triangulation is used to find the target set of neighbour, which will substitute the „random set of points" described by the Shepard method (Dumitru 2013). Natural neighbours chosen this way will gain weights equal to those in Sheprad method described with the following equation:

$$w_m = \frac{1}{\mathrm{d}_{x,m}^{p}}, \tag{8}$$

where w_m is the weight corresponding to the neighbour m, $\mathrm{d}_{x,m}$ is the distance between the target point x and its neighbour m and p is a power factor. We will use $p = 2$ as it is one of the most commonly used (Dumitru 2013).

A very important issue is the differentiation between natural and nearest neighbours, as even in the local solution the nearest neighbours could be not the same as natural ones (see Figs. 7, 8).

Differentiation between those two groups of neighbours is extremely important for irregularly scattered data.

Comparison between methods

Dataset

Exemplary computations were based on the dataset that consisted of TEC value observations from 19 EUREF

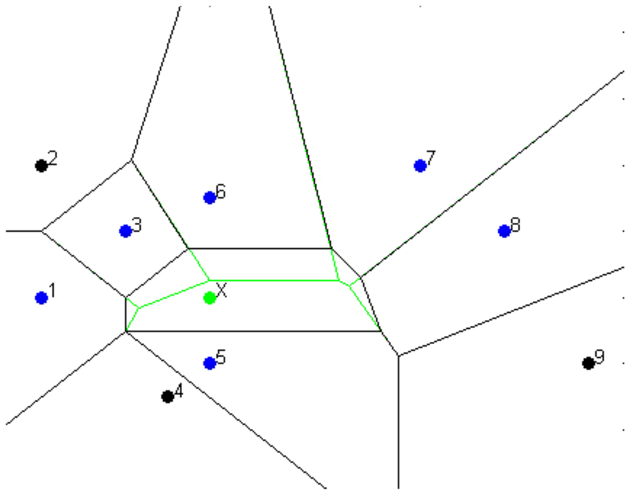

Fig. 7 Natural neighbours (*blue*) of a certain point X

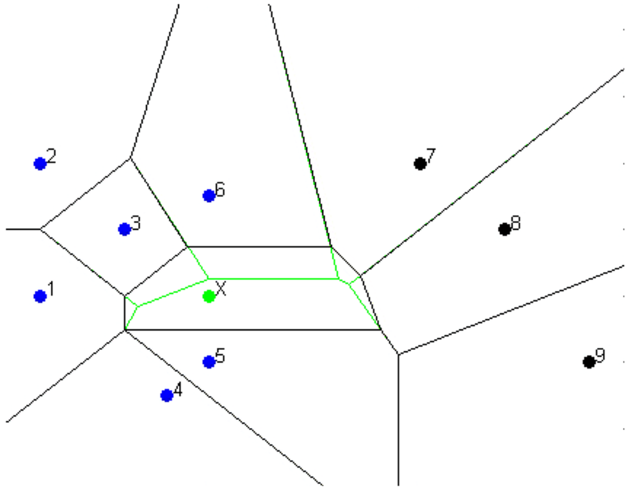

Fig. 8 Nearest neighbours (*blue*) of a certain point X

Permanent Network (EPN) stations located within and nearby the area of Poland (see Fig. 9). The data set provided by the UPC included Slant TEC (STEC) values for each IPP. The STEC values were observed alongside the lines of sight between the EPN station and every satellite in view. Then it was computed into Vertical TEC (VTEC) related to the proper IPP assuming a 450-km thin-shell height as Fig. 10 shows. The mapping function used for the computation of the VTEC is as follows:

$$\mathrm{VTEC} = \mathrm{STEC} * \sqrt{1 - \left(\frac{R_e}{R_e + h_{\mathrm{ion}}} \cos \varepsilon\right)^2} \qquad (9)$$

where ε is the elevation between the receiver and satellite, h_{ion} is the height of ionospheric thin layer and R_e is the Earth's radius.

The IPPs of three stations: Borowiec (bor1), Bydgoszcz (bydg) and Lodz (lodz) were taken as unknown for the interpolation. Then their known VTEC values were used to assess the performance of the interpolation method.

Those three stations have been chosen due to their locations. They are indeed surrounded by other stations from all sides, thus avoiding extrapolation scenarios, which are out of the scope of this work. As we show later, natural neighbour method does not work well for extrapolation. Although inner stations may have IPPs surrounded by open Voronoi cells, the likelihood of such scenario happening is smaller than for any other station closer to the border of the network.

Moreover, the interpolation was performed on an epoch-basis and every unknown point was interpolated separately and never added to the known dataset. This procedure was aimed at avoiding potential error.

Considering the fact that the dataset contains 30-s interval data for the whole day (24 h observations from 15 June 2015, during quiet geomagnetic conditions after 2013 solar activity peak), the observations of 16 stations from the EUREF network provided 74400 interpolated points divided into 2880 one-epoch subsets, which is a good statistical sample for assessing the performance of the proposed methods.

Results

For each point in each interpolation method we established a level of confidence, including relative errors, standard deviations (as we have the „true" values for the interpolated variables) and root means squares for every whole epoch.

Results are presented below (see Tables 1, 2, 3; Figs. 11, 12, 13). Computation time was also taken into account, as we will try to evaluate the possibility of using natural methods to compute rapid, near-real time local TEC values. Root mean squares, relative errors and standard deviations are shown only for the second scenario of natural neighbour interpolation, as the results in both scenarios are the same, but they are more distorted in the first one by lost points (point-losing problem will be further analysed in other sections).

Also, as it was said earlier, we have taken computation time into account. Results are shown below (see Table 4).

The fastest method is undoubtedly the quasi-natural neighbour method. The slowest methods are the Sibsonian natural neighbour methods. Indeed, as they build two Voronoi diagrams for each point they require longer computational time. The difference of both Sibsonian natural neighbour methods in terms of computational time stems from the fact that the second scenario requires the computation of two polygons intersection. Consequently, it takes longer than simple subtraction of two numbers representing areas of the corresponding Voronoi cells, which is the method used in the first scenario. The non-Sibsonian natural neighbour method requires building only one

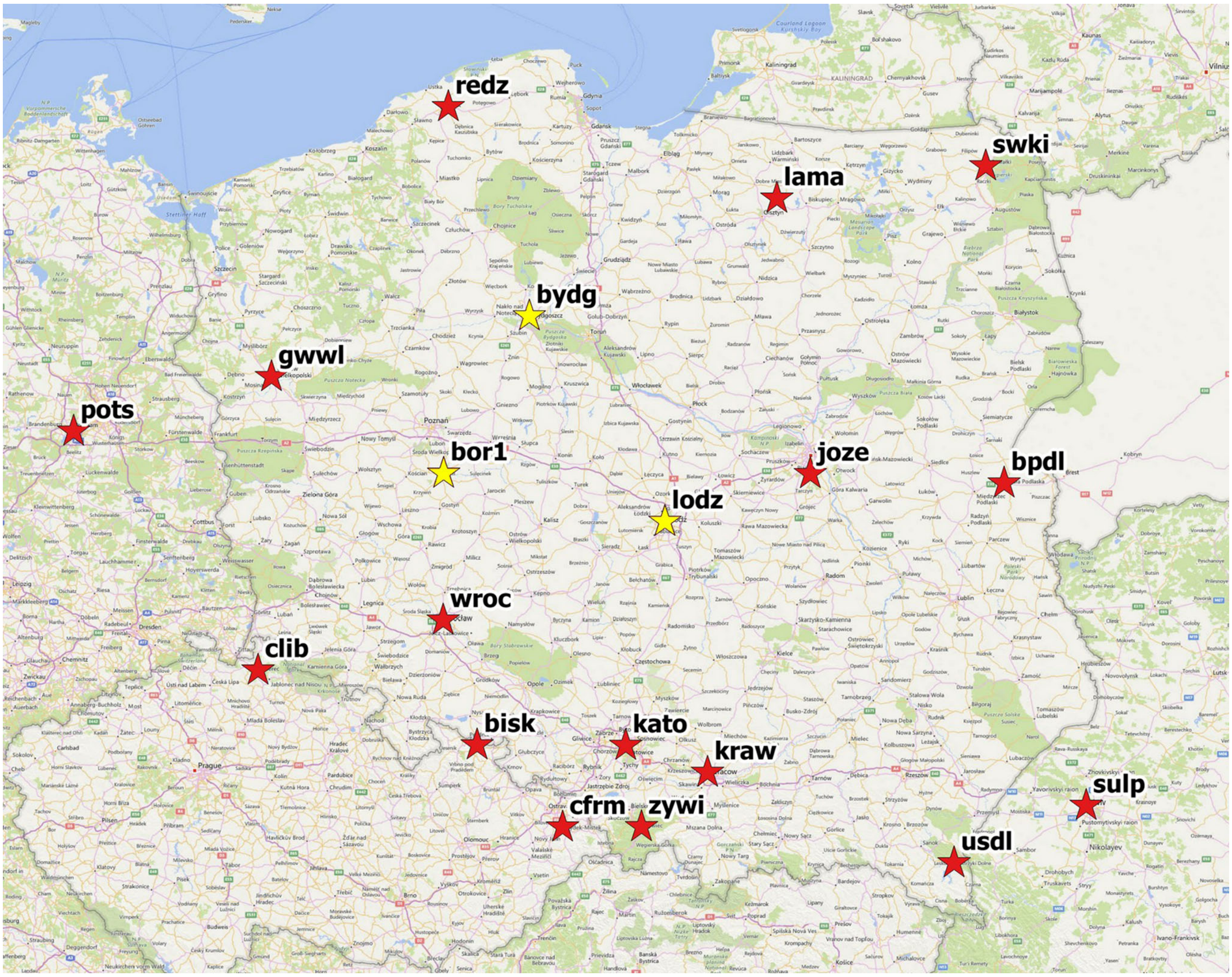

Fig. 9 Location of analysed EPN stations (stations taken as unknowns are marked with *yellow symbol*)

Voronoi diagram and does not require any area computation at all. Weights are based on simple linear quantities (distance between points and length of the Voronoi edge). The fastest quasi-natural neighbour method does not build any Voronoi diagrams at all as it is based on the easier (and quicker) Delaunay triangulation method.

As shown above, we received some statistical information about efficiency of each presented interpolation method, but before we set final efficiency marks, we should turn the attention to some problems we faced while conducting interpolation using natural neighbour methods.

Troubleshooting

Losing points

The first problem with natural neighbour methods is that not every point can be interpolated. For example, during the process of computation some of them (see Table 5) have not received any value.

The method based on the first scenario lost over 2500 times more points than the second one. Such huge difference between both Sibsonian methods suggests that the computational method systematically fails under some circumstances for the first scenario. In further analysis we will show that the second scenario method loses far more points than it was presented in Table 5, i.e., the losses are simply hidden.

Six lost points of the natural neighbour method (second scenario) are actually located outside the network (we based our computations on IPPs, which are dependent on rapidly changing position of satellite on the sky)—hence the interpolation method failure.

The first scenario method fails to interpolate a point when at least one of the neighbours corresponds with an open Voronoi cell. Indeed, in such a case the area of the Voronoi cell is actually infinite (see Fig. 14), which corresponds to a not-a-number computational result.

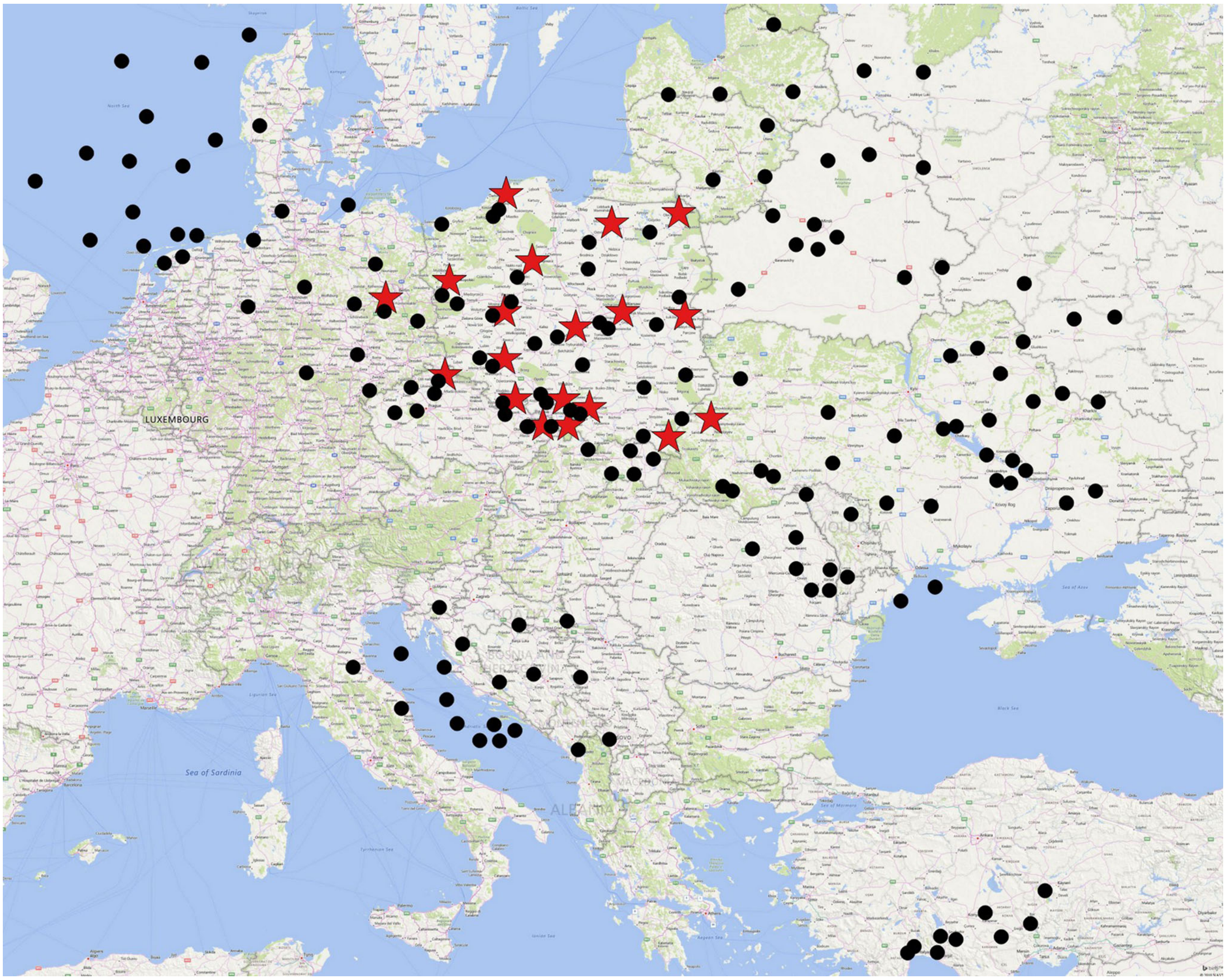

Fig. 10 Exemplary location of observed IPPs (*black dots*) for chosen EPN stations (*red stars*) for epoch 1440 (12:00 UT)

Table 1 Averages of root mean squares for each analysed method (TECU)

Interpolation method	Mean	Min	1st Quartile	Median	3rd Quartile	Max
Natural	0.063	0.013	0.040	0.055	0.069	0.789
IDW	0.123	0.033	0.097	0.122	0.148	0.279
Quasi-natural	0.075	0.024	0.057	0.072	0.088	0.196
Non-Sibsonian	0.054	0.017	0.042	0.053	0.064	0.113
Polynomial	0.088	0.032	0.066	0.084	0.103	0.200

Table 2 Averages of relative errors for each analysed method (%)

Interpolation method	Mean	Min	1st Quartile	Median	3rd Quartile	Max
Natural	0.239	0.000	0.046	0.127	0.301	24.822
IDW	0.538	0.000	0.167	0.379	0.745	7.909
Quasi-natural	0.327	0.000	0.105	0.233	0.433	7.263
Non-Sibsonian	0.242	0.000	0.066	0.156	0.318	2.950
Polynomial	0.434	0.000	0.144	0.321	0.604	4.214

Table 3 Averages of standard deviations for each analysed method (TECU)

Interpolation method	Mean	Min	1st Quartile	Median	3rd Quartile	Max
Natural	0.039	0.000	0.008	0.021	0.050	3.937
IDW	0.092	0.000	0.027	0.063	0.127	1.020
Quasi-natural	0.055	0.000	0.017	0.039	0.074	0.865
Non-Sibsonian	0.039	0.000	0.011	0.027	0.054	0.391
Polynomial	0.071	0.000	0.024	0.054	0.099	0.606

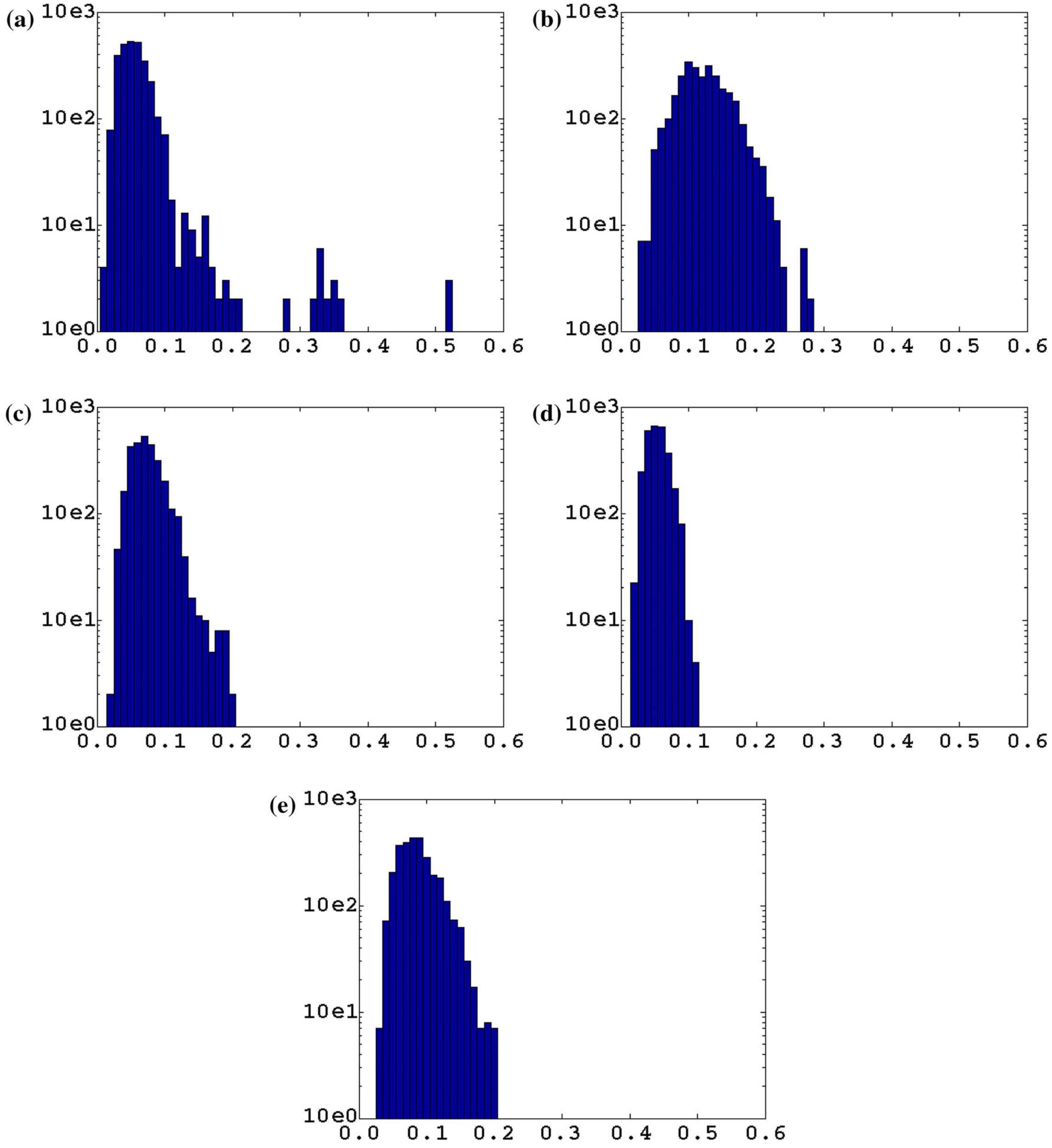

Fig. 11 Histograms of root mean squares for **a** natural Sibsonian, **b** IDW, **c** quasi-natural, **d** non-Sibsonian and **e** polynomial methods (with logarithmic scale on the *y* (*vertical*) axis)

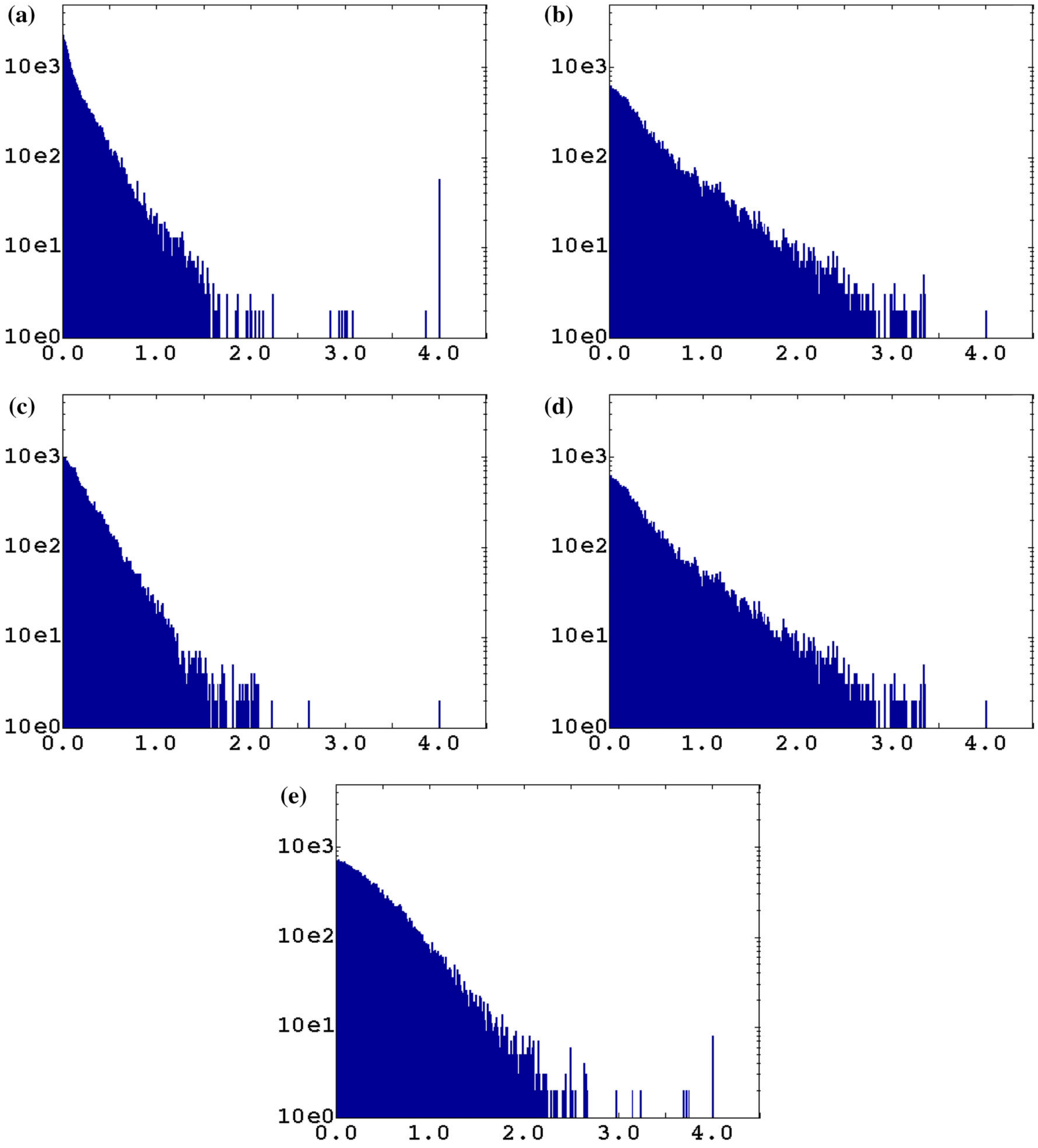

Fig. 12 Histograms of relative errors for **a** natural Sibsonian, **b** IDW, **c** quasi-natural, **d** non-Sibsonian and **e** polynomial methods (with logarithmic scale on the *y* (*vertical*) axis)

Similarly to the second-scenario Sibsonian method, the non-Sibsonian natural method fails not only in case of an extrapolation situation, but also when the target point is too close to the network border. In this situation the Voronoi cell corresponding to the interpolated point is open (see Fig. 15).

The interpolated point (marked with red boldface) is not actually located outside the network, so one would expect the method would be able to compute it. Nevertheless, the open area of the Voronoi cell is infinite, thus preventing the computation of the weight coefficients (based on the length of those edges).

Efficiency drops

Another problem with the natural methods is that occasionally the efficiency drops. This affects only the second scenario of the Sibsonian method and is highly correlated with hidden point losses, which were pointed in the previous section.

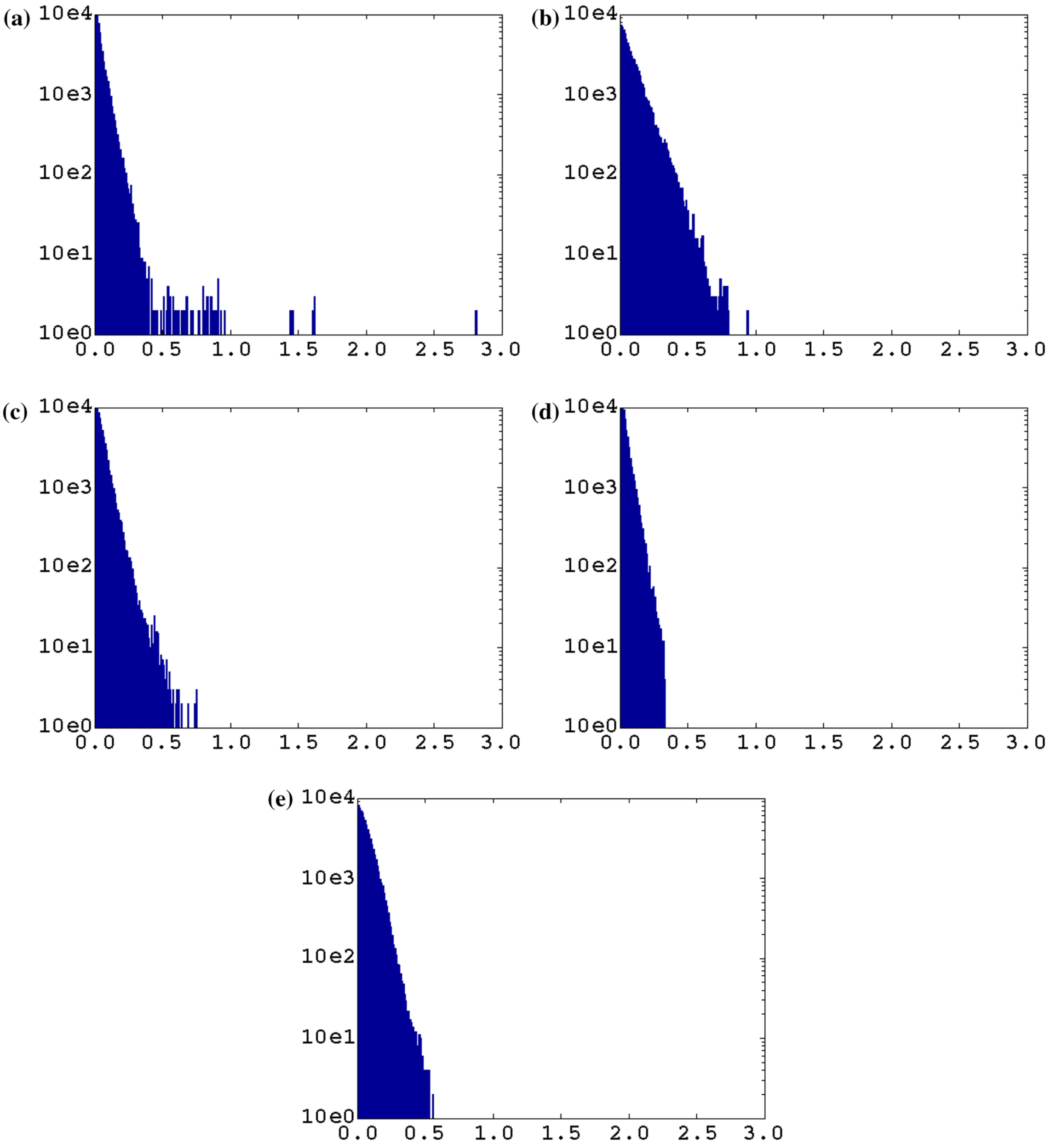

Fig. 13 Histograms of standard deviations for **a** natural Sibsonian, **b** IDW **c** quasi-natural, **d** non-Sibsonian and **e** polynomial methods (with logarithmic scale on the *y* (*vertical*) axis)

Table 4 Computation time (in seconds) of each analysed method for the full set of 74,400 points, with the first epoch containing 23 points and one single IPP (Intel i7, 4 GB RAM, 500 GB HDD)

Interpolation method	Full set (74,400 points)	One epoch (23 points)	One point
Natural (1st scenario)	581.2	0.1566	0.0075
Natural (2nd scenario)	2020.44	0.4773	0.0195
IDW	67.94	0.0311	0.0208
Quasi-natural	36.22	0.0174	0.0008
Non-Sibsonian	585.79	0.1564	0.0076
Polynomial	96.32	0.0442	0.0192

Table 5 Lost points at each epoch (absolute and relative numbers in whole set of 74,400 points)

Interpolation method	Lost points (nominally)	Lost points (%)
Natural (1st scenario)	16,017	21.53
Natural (2nd scenario)	6	0.01
IDW	0	0.00
Quasi-natural	0	0.00
Non-Sibsonian	72	0.10
Polynomial	0	0.00

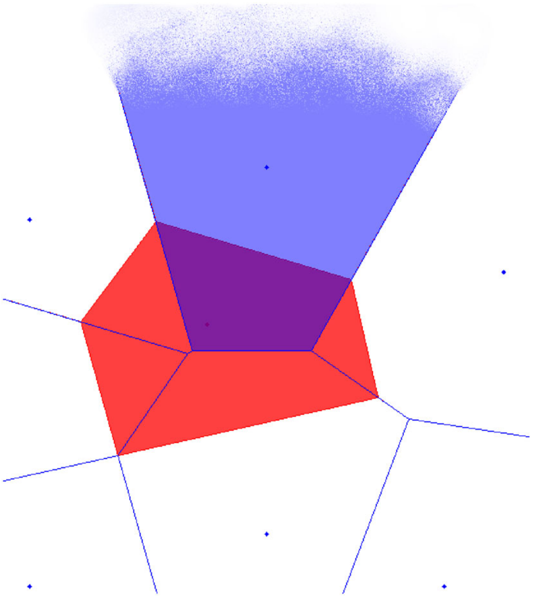

Fig. 14 Open neighbouring cell problem in Sibsonian natural method. Searched overlapping area (*purple*) is computed as the difference of the whole cell area (before putting *red* cell into diagram) and the diminished cell area (*light-blue*)

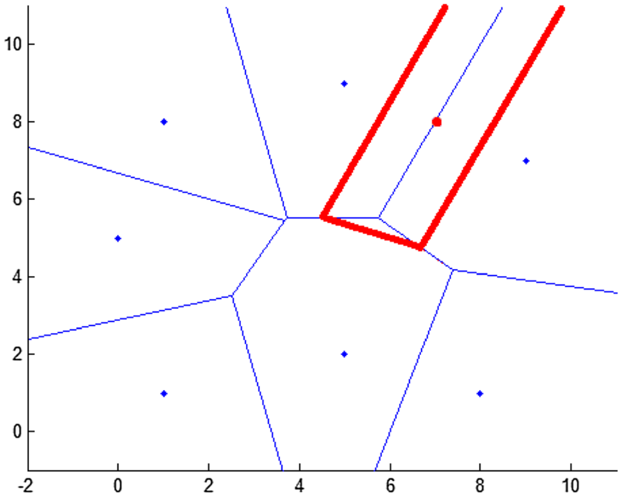

Fig. 15 Open searched cell problem in non-Sibsonian natural method

The same problems affecting those previous two methods also affect the second scenario natural method. In this case, the points are not lost but they provide wrong values. Relative errors and standard deviation values (see Figs. 11, 12, 13) can be clearly seen on histograms of RMS, Accidentally, there occur larger values than in other methods.

First, we should check the problem associated with an open cell for the searched point when using the non-Sibsonian method. Figure 16 contains standard deviations for the 72 points lost in non-Sibsonian natural neighbour interpolation.

As shown, standard deviations (with the maximum value of 3.937 TEC units −1 TECU = 10^{16} electrons/m^2) can reach almost 3 TECU. For some points the standard deviation gets a smaller value, but as their behaviour is impossible to predict, it is much safer to discard the open cell points from the solution (there are very few of them—less than 0.1% of all points) or at last flag them as ‘possible to be wrong’. Detecting such points is very simple to perform and will be shown afterwards.

The second problem—open neighbouring cell problem—is far more complex. To understand its nature, we have to look closer at the geometry of Voronoi cells. Overlapping areas should look like in Fig. 17.

When a neighbouring cell is open (i.e., with at least one of its vertices laying to infinity), the geometry of overlapping areas is changed. The infinity-point is excluded by the algorithm and the polygons are closed thereby ignoring it. This causes the loss of one of the target cell vertices and, consequently, it changes the geometry of the overlapping areas. Such situation is presented in Fig. 18.

This leads to the undesirable reduction of one or more weights. The easiest way to decect that is to compute the relative difference between the sum of overlapping areas

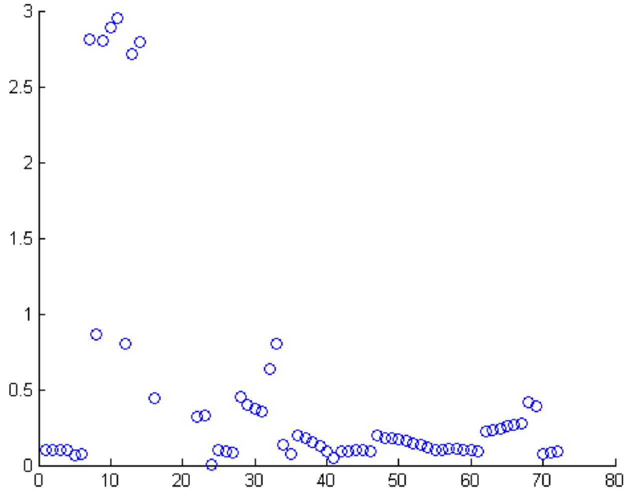

Fig. 16 Standard deviations (in TECU) distribution for 72 open searched cells

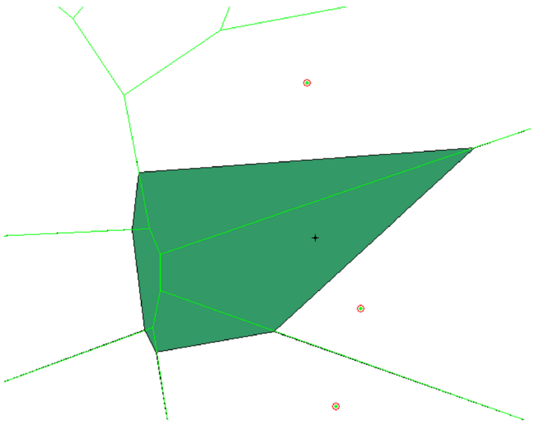

Fig. 17 Properly located overlapping areas

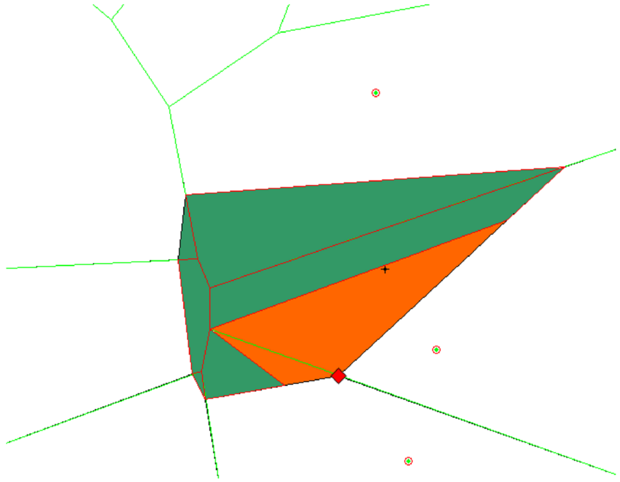

Fig. 18 Deformed overlapping area geometry caused by infinity-vertex problem. Overlapping areas are computed as the *green areas*. Orange ones are lost. Point marked with *red diamond* is excluded because it lies on the infinity-heading line

and the expected value equal to the Voronoi cell of interpolating point, as in Eq. 10.

$$D = (S_x - \Sigma S_i)/S_x \tag{10}$$

where D is the deformation coefficient, S_x is the expected Voronoi cell area and ΣS_i is the sum of overlapping areas (the same method allows to detect points convicted by the previous problem as in such case the deform coefficient will be unable to be computed). Figure 19 illustrates the dependence of standard deviations on the level of deformation.

Table 6 presents the total number of cells with deformed area.

We received over 4600 points (over 6%) with their corresponding deformed cell. Table 7 shows how the statistical distribution of standard deviation changes after excluding all points with uncertain geometry.

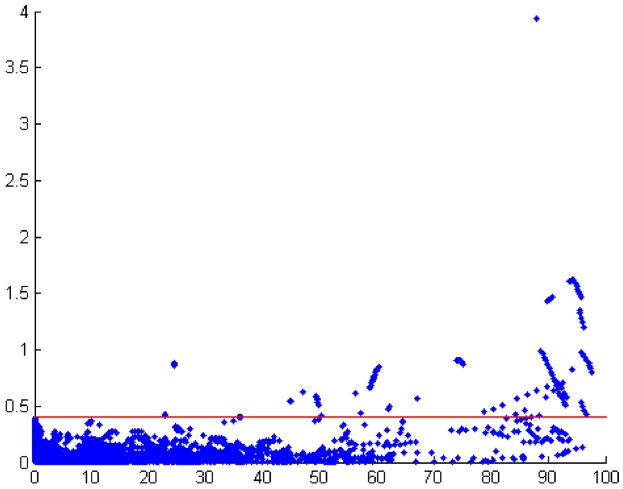

Fig. 19 Dependence of standard deviations (*upwards*; in TECU) on level of deformation (*horizontally*; in percent). The *red line* marks standard deviation level of 0.4 TECU

Table 6 Numbers of points with cells deformed on certain level

Deformation	Number of cells	Percentage
⟨0.9:1⟩	72	0.10
⟨0.8:0.9⟩	54	0.07
⟨0.7:0.8⟩	22	0.03
⟨0.6:0.7⟩	42	0.06
⟨0.5:0.6⟩	133	0.18
⟨0.4:0.5⟩	146	0.20
⟨0.3:0.4⟩	281	0.38
⟨0.2:0.3⟩	528	0.71
⟨0.1:0.2⟩	1042	1.40
⟨0:0.1⟩	2302	3.09
Total	4622	6.21

It is clearly seen that excluding all cell-deformed points solves the problem with large errors. However, such operations require excluding over 6% of total amount of points. Figure 20 shows that the vast majority of points with standard deviation over 0.4 TECU have cells deformed for more than 40%. As can be seen in Table 8, there are 469 of such points, which is only 10% of all excluded points. Table 8 also presents how the standard deviation distribution would look like if we exclude only those points with deformation coefficient above 40%.

The maximum value of standard deviation is not as low as when all cell-deformed points were excluded, but still is reduced from almost 4 TECU to less than 1 TECU and less than 1% of points had to be excluded. This shows that not only flagging suspected points, but also giving them some kind of risk factor may be a good idea.

Table 7 Natural neighbour method standard deviations distribution after excluding all deformed and open cell points

Mean	Min	1st quartile	Median	2nd quartile	Max	Lost points	Open cells points	Deformed cells points	Excluded points total	Percentage
0.036	0.000	0.008	0.021	0.047	0.396	6	72	4622	4694	6.31

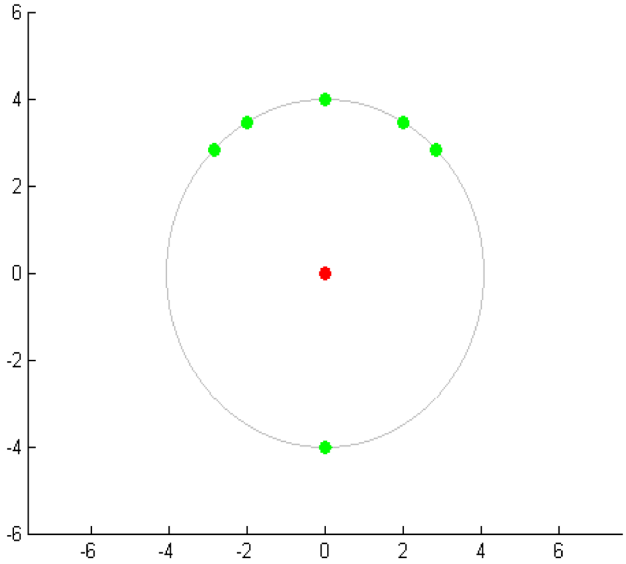

Fig. 20 Exemplary set of equally distant neighbours

It should also be mentioned that this problem is not mathematical. It is caused by limits of computers capabilities—overlapping areas are deformed due to the infinity-computing trouble. This problem could be also solved by including boundaries and computing Voronoi diagrams inside the closed area, but, despite this solution, it requires using a special function, which is not so straightforward and, what is more important, takes over seven times longer to compute (the classic Voronoi based natural neighbour method took about 30 min; see Table 4). Regarding the fact that such a problem occurs rather rarely and only in situations when the dataset is not sufficient, the use of Voronoi limiting function should be considered for each interpolation case.

Final discussion and conclusion

We have presented a comparison of several interpolation methods, namely two methods based on distance-weights (IDW and Quasi-natural), two Voronoi-based methods (natural and non-Sibsonian) and a polynomial method.

For the comparison we have used STEC data computed with UPC TOMION software from GPS ground-receivers from EUREF network over Poland. The 30-s interval dataset spans a period of 24 h from 15 June 2015, during quiet geomagnetic conditions.

The natural neighbour method, in both Sibsonian and non-Sibsonian approaches, provides relative errors over two times smaller than the IDW method (mean value of 0.239, 0.242 and 0.538%, respectively) and almost two times smaller than the outcomes of the polynomial interpolation with mean value of 0.434% (see Table 2). Similar improvement can be seen in standard deviations—0.029 TECU for natural neighbour versus 0.092 TECU for IDW and 0.071 TECU for polynomial interpolation (see Table 3). Highly topology-dependent natural neighbour methods provide the best accuracy as the TEC value depends on its own topology. However, the Sibsonian method in its both scenarios has to cope with many problems and point losses. Moreover, the more effective second scenario-based method requires relatively high computation time (especially when opposed to ultra-rapid quasi natural neighbour method). The solution to the trade-off between accuracy and computational time might be the non-Sibsonian method, which is also very topology-dependent, but least sensitive to the open-cell problem. Another solution may be the quasi-natural method, which provides slightly worse, but still quite promising results (see Tables 1, 2, 3). This method is only partially dependent on the topology, which leads to the drop of accuracy, but, on the other hand, it remains almost unaffected by all topology-caused problems.

Also, a comparison between natural neighbour interpolation and Kriging was considered and will be discussed in future works.

When considering accuracy, the key issue is the way interpolation methods tackle different topologies. To illustrate this problem, let us consider the situation of a

Table 8 Natural neighbour method standard deviations distribution after excluding open cell points and those with corresponding cells deformed for more than 40%

Mean	Min	1st quartile	Median	2nd quartile	Max	Lost points	Open cells points	Deformed cells points	Excluded points total	Percentage
0.036	0.000	0.008	0.021	0.048	0.888	6	72	469	541	0.73

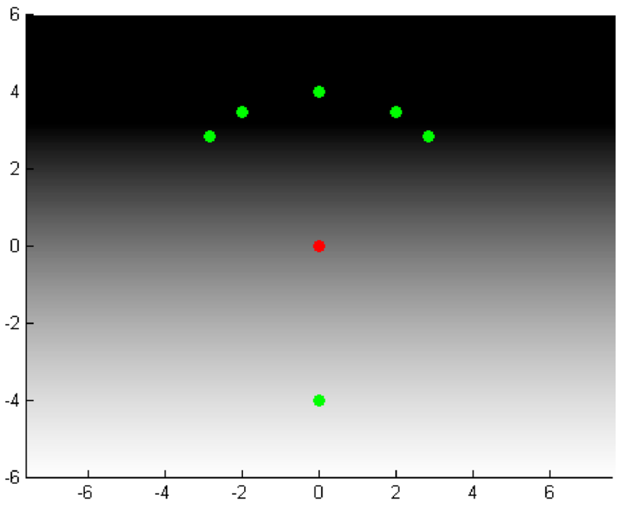

Fig. 21 Exemplary values gradient. Units are arbitrary

point surrounded by six neighbours located in exactly the same distance from the point, like in Fig. 20.

First, we should consider how the weights would look like in the pure distance-based methods (IDW and quasi-natural methods). In quasi natural method all the weights will be equal to 0.16, as all six points are natural neighbours and they are at the same distance of the interpolated point. The IDW method—which depends on the defined number of neighbours—may behave like quasi-natural method but only if the number of neighbours is equal to six. In other situations (neighbours number different then six), IDW method will discard randomly one or more neighbours or, if defined neighbours number is too large, take some extra, non-neighbouring but nearby points into calculation.

To illustrate the influence of such weight distribution, we should assign some values to the points. Let us consider the example shown in Fig. 21, where there are six values distributed with a linear gradient.

The interpolated point is right in the middle, so it should get a value approximately equal to the mean value of the points at the top and that at the bottom. Nevertheless, in quasi-natural and IDW methods all weights have the same value, thereby biasing the result in favour of the most populated cluster of points at the top.

Figure 22 shows the Voronoi diagram for those points in Fig. 21. The shape of the Voronoi cell (for the non-Sibsonian method) and overlapping areas structure (for the Sibsonian method) display how the influence of the points at the top over the interpolated point is diminished with respect to the one at the bottom. Since the weight coefficients are related to the size of the Voronoi cell and overlapping areas (Eqs. 5 and 7, respectively), the smaller the areas and Voronoi cells in the cluster of points, the lower the values of the weight coefficients are.

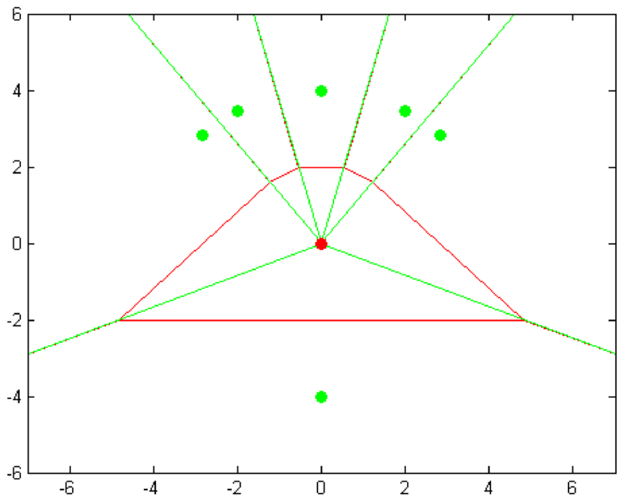

Fig. 22 Two-ordered Voronoi diagram for the set of points. Units are arbitrary

To summarize, by taking into account the topology, the result is more precise. This simple example helps to explain why natural neighbour methods perform better with non-homogenously distributed datasets than methods solely based on relative distances.

References

Belikov VV, Ivanov VD, Kontorovich VK, Korytnik SA, Semenov AYu (1997) The non-Sibsonian interpolation: a new method of interpolation of the value of a function on an arbitrary set of points. Comput Math Math Phys 37(1):9–15

Buresova D, Nava B, Galkin I, Angling M, Stankov SM, Coisson P (2009) Data ingestion and assimilation in ionospheric models. Ann Geophys 52(3/4):235–253

Choi B-K, Lee W-K, Cho S-K, Park J-U, Park P-H (2010) Global GPS ionospheric modelling using spherical harmonic expansion approach. J Astron Space Sci 27(4):359–366. doi:10.5140/JASS.2010.27.4.359

Dąbrowski BP, Krankowski A, Błaszkiewicz L, Rothkaehl H (2016) Prospects for solar and space weather research with polish part of the LOFAR telescope. Acta Geophys 64(3):825–840. doi:10.1515/acgeo-2016-0028

Dumitru PD, Plopeanu M, Badea D (2013) Comparative study regarding the methods of interpolation. In: 1st European conference geodesy and geomatics engineering GENG'13 "Recent Advances in Geodesy and Geomatics Engineering", 8–10 October, 2013, Antalya, Turkey, pp 45–52

Fortune S (1995) Voronoi diagrams and Delaunay triangulations. In: Toth CD, O'Rourke J, Goodman JE (eds) Handbook of discrete and computational geometry. CRC Press Inc, Boca Raton, pp 377–388

Hajj GA, Wilson BD, Wang C, Pi X, Rosen IG (2004) Data assimilation of ground GPS total electron content into a physics-based ionospheric model by use of the Kalman filter. Radio Sci 39:RS1S05. doi:10.1029/2002RS002859

Han D, Yun H, Kee C (2013) Modeling of ionospheric delay for SBAS using spherical harmonics functions. Trans Nav Int J Marine Nav Safety Sea Transport 7(2):205–209. doi:10.12716/1001.07.02.07

Harman C, Johns M (2008) Voronoi natural neighbors interpolation. In: Proceeding of class of 2008 senior conference on computational geometry, Swarthmore College, Swarthmore, USA, 49–53

Hernandez-Pajares M, Juan JM, Sanz J (1999) New approaches in global ionospheric determination using ground GPS data. J Atmos Sol -Terr Phys 61:1237–1247

Hernández-Pajares M, Juan JM, Sanz J (1997) Neural network modeling of the ionospheric electron content at global scale using GPS data. Radiol Sci 32(3):1081–1089

Hernández-Pajares M, Juan JM, Sanz J, Colombo OL (2000) Application of ionospheric tomography to real-time GPS carrier-phase ambiguities resolution at scales of 400–1000 km and with high geomagnetic activity. Geophys Res Lett 27(13):2009–2012

Hernández-Pajares M, Juan JM, Sanz J, Aragón-Àngel A, García-Rigo A, Salazar D, Escudero M (2011) The ionosphere: effects, GPS modeling and the benefits for space geodetic techniques. J Geodyn 85(12):887–907. doi:10.1007/s00190-011-0508-5

Jakowski N, Wilken V, Schlueter S, Stankov S, Heise S (2005) Ionospheric space weather effects monitored by simultaneous ground and space based GNSS signals. J Atmos Sol-Terr Phys 67(12):1074–1084

Komjathy A, Wilson B, Pi X, Akopian Y, Dumett D, Iijima B, Verkhoglya-dova O, Mannucci AJ (2010) JPL/USC GAIM: on the impact of using COSMIC and ground-based GPS measurements to estimate ionospheric parameters. J Geophys Res 115:A02307. doi:10.1029/2009JA014420

Krankowski A, Shagimuratov II, Baran LW, Yakimova G (2007) The structure of the mid- and high-latitude ionosphere during the November 2004 storm event obtained from GPS observations. Acta Geophys 55(4):490–508. doi:10.2478/s11600-007-0033-3

Mandrake L, Wilson B, Wang C, Hajj G, Mannucci A, Pi X (2005) A performance evaluation of the operational Jet Propulsion Laboratory/Univer-sity of Southern California Global Assimilation Ionospheric Model (JPL/USC GAIM). J Geophys Res 110:A12306. doi:10.1029/2005JA011170

Moore EH (1920) On the reciprocal of the general algebraic matrix. Bull Am Math Soc 26(9):394–395

Orus R, Hernandez-Pajares M, Juan JM, Sanz J (2005) Improvement of global ionospheric VTEC maps by using kriging interpolation technique. J Atmos Sol-Terr Phys 67:1598–1609

Pi X, Wang C, Hajj GA, Rosen G, Wilson BD, Bailey GJ (2003) Estimation of E × B drift using a global assimilative ionospheric model: an observation system simulation experiment. J Geophys Res 108(A2):1075. doi:10.1029/2001JA009235

Pi X, Had GA, Wilson BD, Mannucci AJ, Komjathy A, Mandrake L, Wang C (2004). 3-Dimensional assimilative ionospheric modelling for regions of large TEC gradient. In: Proceedings of ION 2004 national technical meeting, Satellite Division of the Institute of Navigation, 26–28 January 2004, San Diego, USA

Pi X, Mannucci AJ, Iijima BA, Wilson BD, Komjathy A, Runge TF, Akopian V (2009) Assimilative modeling of ionospheric disturbances with FORMOSAT-3/COSMIC and ground-based GPS measurements. Terr Atmos Ocean Sci 20(1):273–285. doi:10.3319/TAO.2008.01.04.01(F3C)

Schaer S (1999) Mapping and predicting the earth's ionosphere using the global positioning system. Ph.D. thesis, Astronomical Institute, University of Bern, Switzerland

Scherliess L, Schunk RW, Sojka JJ, Thompson DC, Zhu L (2006) Utah State University global assimilation of ionospheric measurements Gauss-Markov Kalman filter model of the ionosphere: model description and validation. J Geophys Res 111:A11315. doi:10.1029/2006JA011712

Schunk RW (2002) Global assimilation of ionospheric measurements (GAIM). In: Proceedings of international ionospheric effects symposium, 7–9 May 2002, Alexandria, USA

Sibson R (1981) A brief description of natural neighbor interpolation. In: Barnett V (ed), Interpreting multivariate data, Wiley, Chichester, pp 21–36

Strangeways HJ, Kutiev I, Cander LR, Kouris S, Gherm V, Marin D, De La Morena B, Pryse SE, Perrone L, M. Pietrella, S. Stankov, L. Tomasik, Tulunay E, Tulunay Y, Zernov N, Zolesi B (2009) Near-Earth space plasma modelling and forecasting. Ann Geophys 52(3/4):255–271. doi:10.4401/ag-4579

Sukumar N, Moran B, Semenov AYu, Belikov VV (2001) Natural neighbour Galerkin methods. Int J Numer Meth Eng 50:1–27

Wang C, Hajj G, Pi X, Rosen IG, Wilson B (2004) Development of the global assimilative ionospheric model. Radio Sci 39:RS1S06. doi:10.1029/2002RS002854