Journal of Geodesy (1998) 72: 586–599

Journal of Geodesy
© Springer-Verlag 1998

Spherical panel clustering and its numerical aspects

W. Freeden, O. Glockner, M. Schreiner

University of Kaiserslautern, Geomathematics Group, P.O. Box 3049, D-67653 Kaiserslautern, Germany
e-mail: freeden@mathematik.uni-kl.de

Received: 1 October 1997 / Accepted: 1 April 1998

Abstract. In modern approximation methods, linear combinations in terms of (space localizing) radial basis functions play an essential role. Areas of application are numerical integration formulae on the unit sphere Ω corresponding to prescribed nodes, spherical spline interpolation and spherical wavelet approximation. The evaluation of such a linear combination is a time-consuming task, since a certain number of summations, multiplications and the calculation of scalar products are required. A generalization of the panel clustering method in a spherical setup is presented. The economy and efficiency of panel clustering are demonstrated for three fields of interest, namely upward continuation of the Earth's gravitational potential, geoid computation by spherical splines and wavelet reconstruction of the gravitational potential.

Key words. Panel clustering · Numerical integration · Spline and wavelet-based determination of the geoid and the gravitational potential

1 Introduction

Modern satellites provide us with a huge amount of observational data. This requires powerful algorithms for the preparation and interpretation of the observations. Since there is a demand on fine local resolutions in the Earth's gravitational field determination, new approximation methods using space-localizing radial basis functions such as spherical splines (cf. the survey article by Freeden et al. 1997) or spherical wavelets (cf. Freeden and Windheuser 1997; Freeden and Schreiner 1998) have been developed (see also the monograph of Freeden et al. 1998). Both mathematical techniques have in common that they lead to the evaluation of linear combinations of spherical radial basis functions of type

$$F(\xi) = \sum_{i=1}^{N} a_i K(\xi \cdot \eta_i), \quad a_i \in \mathbb{R}, \quad \xi \in \Omega \tag{1}$$

where K is a (space-localizing) radial basis function and $\eta_1, \ldots, \eta_N$ are locations on the unit sphere Ω, at which the measurements are taken. Since the satellite experiments provide us with large data sets and many observations over a period of a few years, such an evaluation may become a very time-consuming task that has to be repeated for each data set. That is the reason why we are concerned with the development of a fast algorithm for the evaluation of Eq. (1). The presented algorithm is based on the idea of the panel clustering method for the fast evaluation of radial basis functions on the unit square in the two-dimensional Euclidean space $\mathbb{R}^2$ (cf. Beatson and Newsam 1992). We essentially take advantage of the fact that a radial basis function only depends on the spherical distance of two locations (unit vectors), i.e. it can be handled simply as function of one variable over the interval $[-1, +1]$. Furthermore, the contribution of the nodal points η_i to the total value of F decreases with an increasing distance. Thus the panel clustering method calculates the most influencing part, i.e. the contribution of the close environment, explicitly, and gives a fast approximation for the remaining part. This leads to a separation of the entire sphere into a near- and a far-field, respectively; the partition is performed by a hierarchical subdivision mechanism. We start by dividing the sphere Ω, for example in the case of the icosahedron, into 20 different spherical triangles. Every triangle is repeatedly subdivided into four different daughter triangles down to a certain level. The given data points $\eta_1, \ldots, \eta_N$ are sorted into the triangles and are combined to clusters of points. Due to the fact that a radial basis function can be considered as a univariate function on the interval $[-1, +1]$, we are able to interpolate it in one dimension with the help of

Correspondence to: W. Freeden

the Legendre polynomials, for which an immediate connection to the spherical harmonics can be established via the well-known addition theorem. This fact enables us to calculate the so-called far-field coefficients for each triangle.

It should be pointed out that the preliminary work presented up to now can be completed before the actual evaluation starts. After deciding which triangles belong to the near- or far-field, respectively, the evaluation of the sum F at a single point $\xi \in \Omega$ reduces to explicitly calculating the near-field contribution and to adding the contributions of all far-field triangles, represented by the far-field coefficients. A desirable feature of our algorithm is that it can be applied to an iterative solution of a system of linear equations occurring in spherical spline interpolation since a matrix–vector multiplication leads to a sum representation of the considered type so that many iterative solvers can be efficiently implemented. While we have been restricted to handling only a few thousand linear equations until now, e.g. in spherical spline interpolation, the presented algorithm now allows us to solve linear systems with 80 000 equations in 80 000 unknowns or even more.

Remark. A traditional method for the evaluation of geodetic integral formulae is the subdivision of the integration domain into a near-zone and a far-zone. Due to the rapid decay of the kernels occurring in the formulae, the contribution of the near-zone has to be computed very accurately, whereas the contribution of the far-zone may be approximated coarsely (see e.g. Heiskanen and Moritz 1967). From our comments given above it is obvious that this paper addresses a related task: the efficient evaluation of a spherical function by linear combinations of space-localizing base functions in the form of a hierarchical panel clustering of the sphere. The relevance of the work is caused by the fact that multi-resolution methods (by splines, wavelets) for the treatment of very large sets of data points on the sphere will command more and more attention in the near future.

2 The panel clustering method

Let $K : t \mapsto K(t), t \in [-1, +1]$ be a continuous (space-localizing) function [for a discussion of space and frequency localization on the sphere in terms of expectation value and variance and the development of the uncertainty relation the reader is referred to the article by Freeden and Windheuser (1997)]. In the approach presented here, however, space localization may be understood in the sense that, for every $\xi \in \Omega, K(\xi \cdot \xi)$ is essentially larger than $K(\xi \cdot \eta)$ for all $\eta \in \Omega$ outside a fixed spherical cap around the centre ξ. Evaluating the expression F of Eq. (1) at a certain point $\xi \in \Omega$ takes advantage of the fact that $K(\cdot \eta)$ only depends on the spherical distance from $\eta \in \Omega$. The influence of nearby nodes (near-field) is calculated explicitly, but only an approximation is used to determine the contribution of faraway nodes (far-field). In the following, ξ is said to be far away from η if $\xi \cdot \eta$ is smaller than a certain $t^* \in [-1, +1]$, i.e. if the spherical distance between ξ and η is greater than arccos (t^*). A simple illustration of this idea is given in Fig. 1. The blank area represents the near-field of ξ_1, and the remaining area represents the far-field. Fundamental to our algorithm is that, for example, all nodes in the black area Q belong to the far-field of ξ_1 as well as that of ξ_2 and ξ_3. This gives rise to the accumulation of the nodes into clusters of points, which can be treated as parts of the far-field of a large number of points. For the purpose of accumulation, a subdivision of the entire sphere is performed. Later on it will be shown that a hierarchical subdivision scheme is very useful for our purpose. The complete algorithm consists of two stages, a setup stage and an evaluation stage. In the setup stage the hierarchical subdivision is performed and the so-called *far-field coefficients* are calculated to render a rapid computation in the evaluation stage.

Furthermore, in the case of spline interpolation, the equations $F(\eta_i) = y_i, \quad i = 1, \dots, N$ associated to a prescribed data set $(\eta_1, y_1), \dots, (\eta_N, y_N)$ must be fulfilled and, therefore, the following system of linear equations must be solved somehow:

$$\sum_{i=1}^{N} a_i K(\eta_i \cdot \eta_j) = y_j, \quad j = 1, \dots, N \tag{2}$$

A problem is that the matrix $C = (c_{i,j})_{i,j=1,\dots,N}$, $c_{i,j} = K(\eta_i \cdot \eta_j)$ is full-sized for a large number of (space-localizing) kernels K. If the matrix C is positive definite and symmetric (e.g. in the case of a reproducing kernel K of a suitable Hilbert space), the Cholesky algorithm would be suitable to solve the linear Eq. (1), provided that N is not too large. However, for a large N, direct solution of the system is unpractical and iterative methods like the conjugate-gradient method require at least one matrix–vector multiplication in each iteration step. For this reason, fast algorithms that take advantage of the special structure of the matrix C should be developed. Since the appearance of a matrix–vector product $Cp = q, \sum_{i=1}^{N} p_i K(\eta_i \cdot \eta_j) = q_j,\ j = 1, \dots, N,$

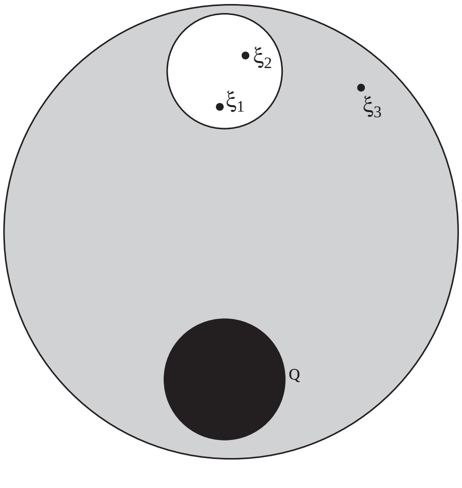

Fig. 1. Idea of the algorithm

corresponds to an N-times evaluation of a function of the type of Eq. (1), an algorithm for solving the linear Eq. (2) can be constructed from a method for the fast evaluation of the function of Eq. (1).

This paper is organized as follows. Section 2.1 introduces an approach to subdivide the sphere hierarchically, followed by the approximation of the far-field contribution in Sect. 2.2. In Sect. 2.3 we summarize the algorithm for the fast evaluation of a linear combination of the type of Eq. (1). Finally, in Sect. 2.4 our method is applied to the conjugate-gradient method for the solution of linear equations of the type of Eq. (2). The paper ends with some test examples.

2.1 Triangulation of the sphere

First of all a suitable subdivision of the unit sphere has to be developed. As mentioned before, this subdividing process should be done hierarchically. Due to this fact a partition of the sphere by circular panels (spherical caps) as shown in Fig. 1 is not appropriate for our purpose. The idea of using a rectangular partition in the polar coordinate plane may arise, but this method is not advisable, because it would lead to very long and narrow panels near the poles. Thus, the subdivision is started by an icosahedron (Fig. 2) (or an octahedron, Fig. 3)) with 20 (or eight) spherical triangles, respectively. Henceforth the starting level is called the *root* or the *zeroth level* of subdivision.

In the following levels the triangles are repeatedly subdivided into four new spherical triangles, called the daughter triangles. The triangles of a certain level are disjoint and form a partition of the entire sphere, i.e.

$$\Delta_l^{k_1} \cap \Delta_l^{k_2} = \emptyset \quad \text{for} \quad k_1 \neq k_2$$

$$\bigcup_{k=1}^{N_l} \Delta_l^k = \Omega \ ,$$

where Δ_l^k denotes the kth triangle in the lth level and N_l denotes the number of triangles in level l. In this case, disjoint is understood in the sense that each point belongs to one triangle, i.e. a point lying on the boundary of two neighbouring triangles is not related to both triangles. Furthermore, if $\Delta_l^k, k = 1, \ldots, 4$, are the daughter panels of Δ_{l-1}^m, the following equation is valid:

$$\bigcup_{k=1}^{4} \Delta_l^k = \Delta_{l-1}^m \ .$$

The subdividing process is stopped at some level L.

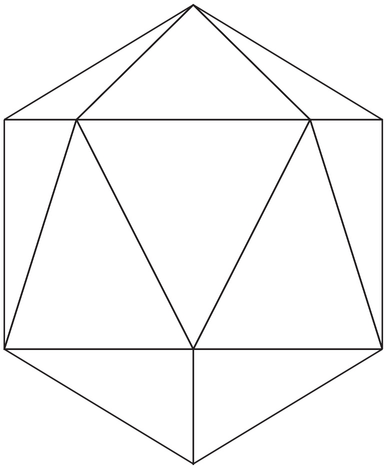

Fig. 2. Icosahedron

Figures 4–7 show the subdivision levels 1, 2, 3 and 4 with an icosahedron as root level. We note that there are

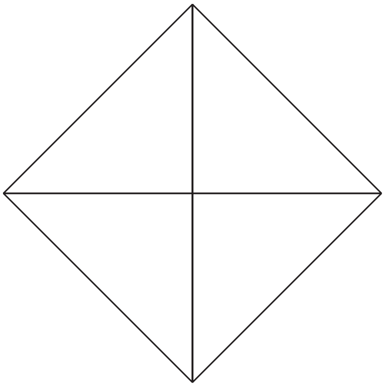

Fig. 3. Octahedron

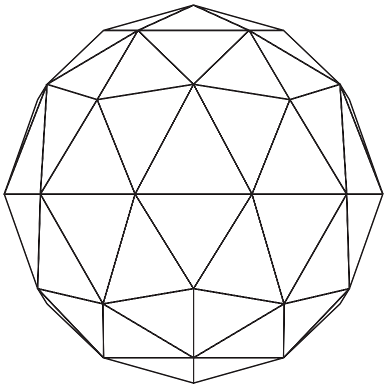

Fig. 4. Subdivision level 1

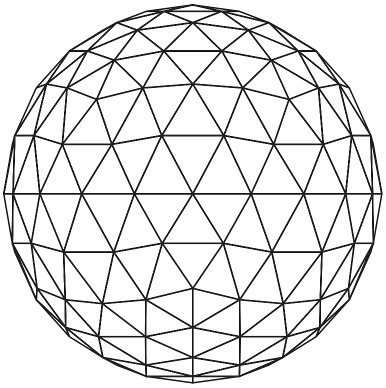

Fig. 5. Subdivision level 2

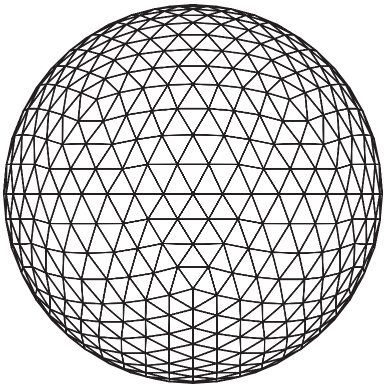

Fig. 6. Subdivision level 3

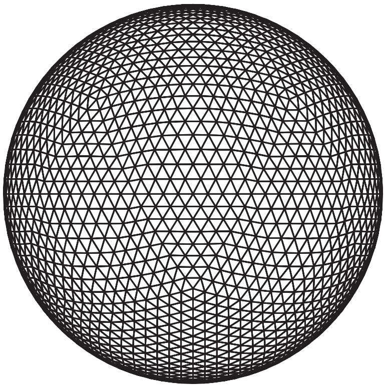

Fig. 7. Subdivision level 4

$2 + 5 \cdot 2^{l+1}$ nodes and $N_l = 5 \cdot 2^{2l+2}$ triangles, in the lth refinement (subdivision level).

Subsequently, the nodes must be sorted into the triangles. The verification as to whether a point belongs to a spherical triangle or not is done by a spherical *angle-sum test* (see Appendix).

2.2 Far-field approximation

In this section we discuss how to find an appropriate approximation of the far-field of a linear combination F defined by Eq. (1). To be more precise, the term "appropriate" means that the approximation admits a fast evaluation of F in combination with a sufficient accuracy. Let $a_1, \dots, a_N \in \mathbb{R}$ be the coefficients of Eq. (1) corresponding to the nodes $\eta_1, \dots, \eta_N \in \Omega$. Using the partition of the sphere introduced in Sect. 2.1, the function F can be written as follows:

$$F(\xi) = \sum_{k=1}^{N_l} \sum_{\eta_i \in \Delta_k^l} a_i K(\xi \cdot \eta_i), \quad \xi \in \Omega \tag{3}$$

for each subdivision level $l = 1, \dots, L$, where L denotes the finest level, N_l denotes the number of triangles in level l and Δ_l^k is the kth triangle in the lth level. The second sum in Eq. (3) is understood to be taken over all indices i for which the point η_i belongs to the triangle Δ_l^k. Separation of the triangles into near- and far-field panels leads to the following representation:

$$F(\xi) = F_{NF}(\xi) + F_{FF}(\xi) \ , \tag{4}$$

where $F_{NF}(\xi)$ contains the contribution of the near-field of ξ, and $F_{FF}(\xi)$ represents the far-field contribution of ξ.

Our strategy is as follows. First, a technique to approximate the particular contributions of any triangle is developed. Then we have to decide which triangles belong to the far- and the near-field. Finally, the approximations of the far-field panels and the contribution of the exact near-field are put together to an approximation of the function F.

Our procedure starts with a definition.

Definition 2.1. Let F be given as in Eq. (1) with $K : t \mapsto K(t)$, $t \in [-1, +1]$, a continuous function. Then

$$F(\xi; \Delta_l^k) = \sum_{\eta_i \in \Delta_l^k} a_i K(\xi \cdot \eta_i) \tag{5}$$

is called the *contribution of the triangle* Δ_l^k to F at $\xi \in \Omega$.

Now we consider the problem of approximating a spherical radial basis function by spherical harmonics. To this end, remember that $K(\xi \cdot \eta)$ only depends on the spherical distance, i.e. the inner product of ξ and η, and can therefore be treated as a one-dimensional function over the interval $[-1, +1]$. A relation between functions on $[-1, +1]$ (here Legendre polynomials) and functions on the unit sphere Ω (here spherical harmonics) is guaranteed via the addition theorem. Hence, an approximation of $F(\xi; \Delta_l^k)$ by polynomial interpolation with Legendre polynomials P_n in one dimension is suggested. As it is well known, the error occurring in polynomial interpolation due to the choice of the nodes is minimized by the zeros of the Tschebyscheff polynomials (cf. e.g. Überhuber 1995). Motivated by this result we only use these Tschebyscheff nodes. Therefore $G(\xi; \Delta_l^k)$ defined by

$$G(\xi; \Delta_l^k) = \sum_{\eta_i \in \Delta_k^l} a_i \sum_{n=0}^{M} b_n P_n(\xi \cdot \eta_i) \ , \quad \xi \in \Omega \tag{6}$$

represents an approximation of $F(\xi; \Delta_l^k)$, where $t \mapsto \sum_{n=0}^{M} b_n P_n(t)$, $t \in [-1, 1]$ is a polynomial of degree M and the coefficients $b_0, \dots, b_M$ are obtained by solving the linear interpolating equations

$$\sum_{n=0}^{M} b_n P_n(t_j) = K(t_j), \quad j = 0, \dots, M \tag{7}$$

with the Tschebyscheff nodes $t_0, \dots, t_M$ corresponding to the interval $[-1, t^*]$. Due to the fact that all nodes η_i satisfying $t^* \leq \xi \cdot \eta_i \leq 1$ belong to the near-field of ξ and we only want to approximate the far-field, $K(t)$ has to be interpolated just in the interval $-1 \leq t < t^*$. Figure 8 shows the relation between the distance on the sphere and the radial basis function on the interval $[-1, +1]$. The counterpart of a spherical cap of extension arccos (t^*) is the interval $[t^*, +1]$.

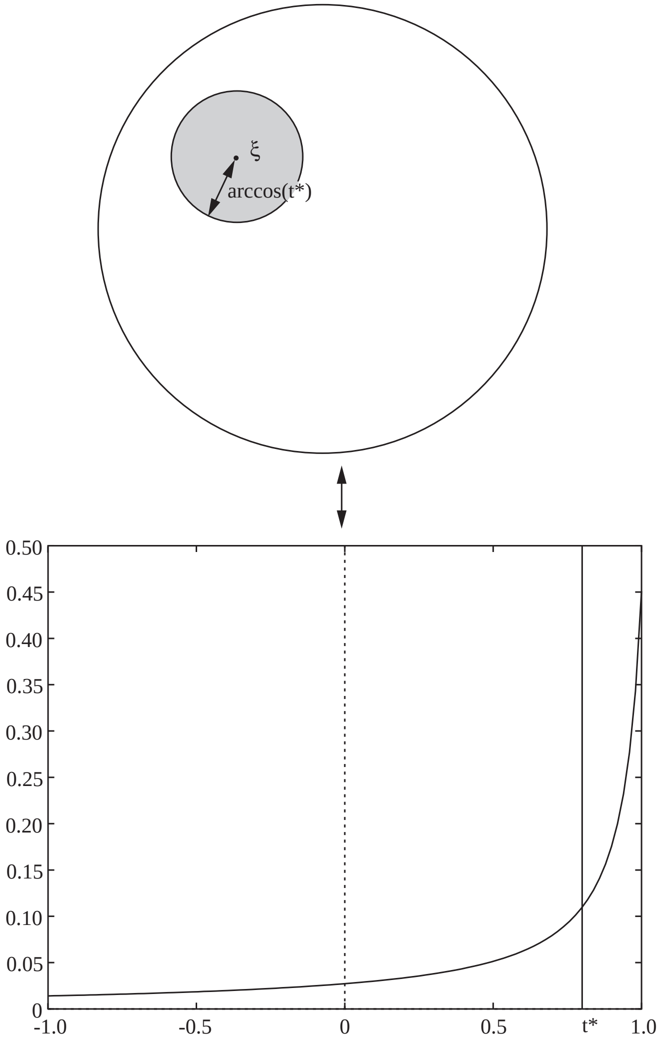

Fig. 8. Relation between spherical distance and one-dimensional level

Applying the addition theorem to Eq. (6) yields

$$G(\xi;\Delta_l^k) = \sum_{n=0}^{M} b_n \frac{4\pi}{2n+1} \sum_{j=1}^{2n+1} \sum_{\eta_i \in \Delta_k^l} a_i Y_{n,j}(\eta_i) Y_{n,j}(\xi) \tag{8}$$

$\xi \in \Omega$, where $\{Y_{n,j}\}_{j=1,\ldots,2n+1}$ forms an orthonormal basis of the class $Harm_n(\Omega)$ of spherical harmonics of order n and $\{Y_{n,j}\}_{\substack{n=0,\ldots,M \\ j=1,\ldots,2n+1}}$ is an orthonormal basis of the set $Harm_{0,\ldots,M}$ of spherical harmonics of orders $\le M$. Setting

$$V_{n,j}(\Delta_l^k) = b_n \frac{4\pi}{2n+1} \sum_{\eta_i \in \Delta_k^l} a_i Y_{n,j}(\eta_i) \tag{9}$$

we obtain for Eq. (8)

$$G(\xi;\Delta_l^k) = \sum_{n=0}^{M} \sum_{j=1}^{2n+1} V_{n,j}(\Delta_l^k) Y_{n,j}(\xi) \ . \tag{10}$$

The quantities $V_{n,j}(\Delta_l^k)$ in Eq. (10), henceforth called the *far-field coefficients* of the triangle Δ_l^k, are calculated explicitly and stored for any triangle in the finest level L. The hierarchical subdivision scheme enables us to calculate the far-field coefficients of triangles in coarser levels recursively from the coefficients of the daughter triangles in the following way. If Δ_l^k, $k = 1, \ldots, 4$, are the daughter panels of Δ_{l-1}^m, the far-field coefficients $V_{n,j}(\Delta_{l-1}^m)$ can be obtained by

$$V_{n,j}(\Delta_{l-1}^m) = \sum_{k=1}^{4} V_{n,j}(\Delta_l^k) \ , \tag{11}$$

$n = 0, \ldots, M, j = 1, \ldots, 2n+1$. This recursive method is illustrated in Fig. 9. While the subdivision process is performed forwardly from the parent to the daughter triangles (from level 0 to level L), we determine the far-field coefficients in the opposite direction (from level L to level 0).

The next step in the setup stage is to establish a criterion for the membership of a panel in the near- or far-field. At this point it should be mentioned that we specify near- and far-field for any panel in the finest level L instead of specifying them for any evaluation point ξ. This has the advantages that it can be performed completely in the setup stage and the matrix $C = (K(\eta_i \cdot \eta_j))_{i,j=1,\ldots,N}$ retains its symmetry. Later we just have to locate the finest level panel containing ξ. For the decision as to which field a panel belongs to, the following criteria apply. *A triangle Δ_l^m, regardless of what level, is said to be in the far-field of a triangle Δ_L^k of level L, if the spherical distance between these two triangles is greater than* arccos(t^*) *for a prescribed* t^* (see Fig. 8). If we find that a parent triangle is far away, then the daughter triangles are no longer considered and do not belong to the far-field. Then all nodes in Δ_l^m are far away from any point in Δ_L^k. *A triangle Δ_L^m is said to be in the near-field of Δ_L^k, if it is of the finest level L and the spherical distance between Δ_L^k and Δ_L^m is equal to or less than* arccos(t^*). Consequently the near-field only consists of finest-level panels, while the far-field may contain panels of any level. In practice, the determination of near- and far-field for a finest-level triangle Δ_L^k works as follows. For each triangle Δ_0^i in the coarsest level, set $\Delta = \Delta_0^i$ and goto (i)

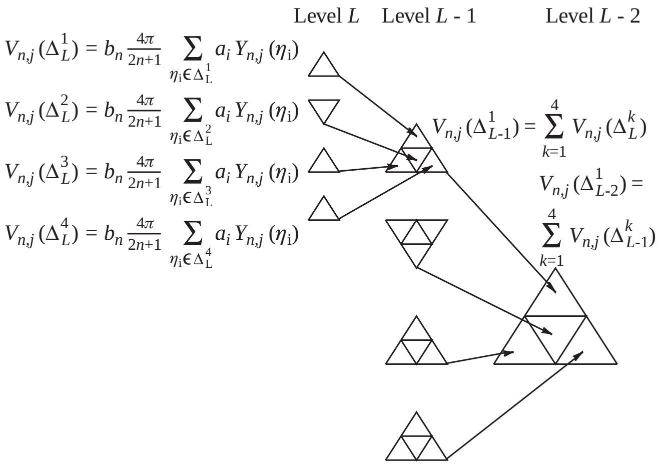
Level L
Level L - 1
Level L - 2
$V_{n,j}(\Delta_L^1) = b_n \frac{4\pi}{2n+1} \sum_{\eta_i \in \Delta_L^1} a_i Y_{n,j}(\eta_i)$
$V_{n,j}(\Delta_L^2) = b_n \frac{4\pi}{2n+1} \sum_{\eta_i \in \Delta_L^2} a_i Y_{n,j}(\eta_i)$
$V_{n,j}(\Delta_L^3) = b_n \frac{4\pi}{2n+1} \sum_{\eta_i \in \Delta_L^3} a_i Y_{n,j}(\eta_i)$
$V_{n,j}(\Delta_L^4) = b_n \frac{4\pi}{2n+1} \sum_{\eta_i \in \Delta_L^4} a_i Y_{n,j}(\eta_i)$
$V_{n,j}(\Delta_{L-1}^1) = \sum_{k=1}^{4} V_{n,j}(\Delta_L^k)$
$V_{n,j}(\Delta_{L-2}^1) = \sum_{k=1}^{4} V_{n,j}(\Delta_{L-1}^k)$

Fig. 9. Recursive calculation of the far-field coefficients

(i) check the spherical distance between Δ_L^k and Δ

(ii) if the distance is greater than $\arccos(t^*) \Rightarrow \Delta$ belongs to the far-field of Δ_L^k
else if the distance is equal to or less than $\arccos(t^*)$ and Δ is in the finest level $\Rightarrow$ triangle belongs to the near-field of Δ_L^k
else for any daughter triangle Δ_{-1} of Δ set $\Delta = \Delta_{-1}$ and goto (i).

Such a field construction with maximum subdivision level three is presented in Fig. 10. The far-field of the black, finest level triangle is shaded, while the near-field, consisting only of panels in the third level, is left blank. It is obvious that the near-field is somewhat larger than the spherical cap indicated by the circle.

In conclusion, it is not necessary to sum up the far-field coefficients of all triangles in the finest level, but in most cases it is possible to deal with lower-level triangles, leading to a reduction in computational costs.

For convenience we introduce the following sets.

$NF(\Delta_L^m)$ is the set of all triangles Δ_L^k, $k = 1, \ldots, N_L$ such that Δ_L^k is in the near-field on Δ_L^m.

$FF(\Delta_L^m)$ is the set of all triangles Δ_l^k, $l = 0, \ldots, L$, $k = 1, \ldots, N_l$, such that Δ_l^k is in the far-field of Δ_L^m.

Furthermore, let

$$\mathscr{N}_L^m = \{i \mid \eta_i \in \Delta_L^k, \quad \Delta_L^k \in NF(\Delta_L^m)\}$$

$$\mathscr{F}_L^m = \{i \mid \eta_i \in \Delta_l^k, \quad \Delta_l^k \in FF(\Delta_L^m)\}$$

for $m = 1, \ldots, N_L$. Thus, the following relations are valid for $m = 1, \ldots, N_L$:

$$\bigcup_{\Delta_L^k \in NF(\Delta_L^m)} \Delta_L^k \cup \bigcup_{\Delta_l^k \in FF(\Delta_L^m)} \Delta_l^k = \Omega$$

$$\Delta_l^k \cap \Delta_p^q = \emptyset, \quad \text{if } l \neq p \text{ or } k \neq q$$

$$\Delta_l^k,\ \Delta_p^q \in NF(\Delta_L^m) \cup FF(\Delta_L^m)$$

An approximation $G_{FF}(\xi)$ of the far-field term $F_{FF}(\xi)$ in Eq. (4) can be derived from the considerations above for each evaluation point $\xi \in \Delta_L^m \subset \Omega$:

$$\begin{aligned} G_{FF}(\xi) &= \sum_{\Delta_l^k \in FF(\Delta_L^m)} G(\xi; \Delta_l^k) \\ &= \sum_{\Delta_l^k \in FF(\Delta_L^m)} \sum_{n=0}^{M} \sum_{j=1}^{2n+1} V_{n,j}(\Delta_l^k) Y_{n,j}(\xi) \end{aligned}$$

Changing the order of summations leads to

$$G_{FF}(\xi) = \sum_{n=0}^{M} \sum_{j=1}^{2n+1} \left(\sum_{\Delta_l^k \in FF(\Delta_L^m)} V_{n,j}(\Delta_l^k) \right) Y_{n,j}(\xi)$$

The third summation, over the triangles, depends only on the panel Δ_L^m containing ξ and is independent of the particular point $\xi \in \Delta_L^k$. This enables us to calculate the magnitudes

$$W_{n,j}(\Delta_L^m) = \sum_{\Delta_l^k \in FF(\Delta_L^m)} Y_{n,j}(\Delta_l^k) \qquad (12)$$

$n = 0, \ldots, M$, $j = 1, \ldots, 2n+1$, in the setup stage for any panel in the finest level and to store them. The coefficients $V_{n,j}(\Delta_l^k)$ are no longer needed, so they can be deleted at this point. After locating the panel Δ_L^m containing ξ, the far-field contribution can be calculated with a computational cost of $\sum_{n=0}^{M} 2n + 1 = (M+1)^2$ multiplications and additions.

2.3 Fast evaluation in terms of radial basis functions

Let us summarize the results of the previous sections. The near-field term $F_{NF}(\xi)$ in Eq. (4) can be written as

$$F_{NF}(\xi) = G_{NF}(\xi) = \sum_{i \in \mathscr{N}_L^m} a_i K(\xi \cdot \eta_i), \quad \xi \in \Delta_L^m \subset \Omega \qquad (13)$$

and $G_{FF}(\xi)$ takes the following form:

$$G_{FF}(\xi) = \sum_{n=0}^{M} \sum_{j=1}^{2n+1} W_{n,j}(\Delta_L^m) Y_{n,j}(\xi), \quad \xi \in \Delta_L^m \subset \Omega \qquad (14)$$

so we obtain for $G(\xi)$ as an approximation of $F(\xi)$

$$\begin{aligned} G(\xi) &= G_{NF}(\xi) + G_{FF}(\xi) \\ &= \sum_{i \in \mathscr{N}_L^m} a_i K(\xi \cdot \eta_i) + \sum_{n=0}^{M} \sum_{j=1}^{2n+1} W_{n,j}(\Delta_L^m) Y_{n,j}(\xi) \end{aligned} \qquad (15)$$

if $\xi \in \Delta_L^m \subset \Omega$.

Now we are able to summarize our algorithm for the evaluation of functions of the type of Eq. (1).

Algorithm 2.2

Setup stage

Step 1. Performance of the hierarchical subdivision of the sphere introduced in Sect. 2.1 down to a certain level L and distribution of the nodes into the triangles.

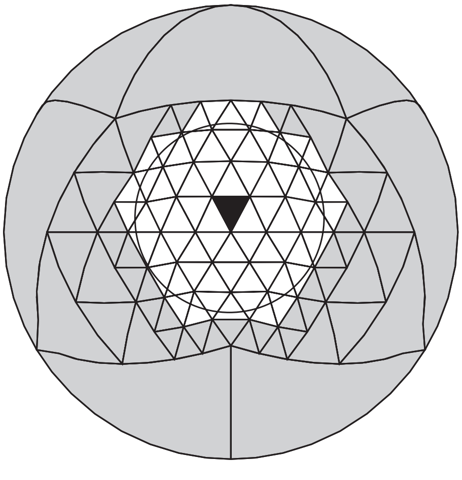

Fig. 10. Far- (shaded) and near-field (unshaded) panels

Step 2. Approximation of the radial basis function K by polynomial interpolation of a certain degree M in one variable.
Step 3. Determination of near- and far-field panels for each finest-level panel.
Step 4. Calculation of the far-field coefficients $V_{n,j}(\Delta_L^k)$ for every finest-level triangle Δ_L^k via Eq. (9).
Step 5. Recursive calculation of the far-field coefficients $V_{n,j}(\Delta_l^k)$ for levels $L-1,\dots,0$ via Eq. (11).
Step 6. Summation of the far-field coefficients to obtain the values $W_{n,j}(\Delta_L^k)$ for all triangles in level L via Eq. (12) and deletion of the far-field coefficients $V_{n,j}(\Delta_l^k)$.

Evaluation stage

For any evaluation point ξ we have to perform two steps:

Step 1. Location of the finest level panel containing ξ.
Step 2. Evaluation of $G(\xi)$ as an approximation of $F(\xi)$ by direct calculation of the near-field of ξ via Eq. (13) and using Eq. (14) to approximate the far-field contribution of ξ.

Since the reduction of computational costs is the main subject of the algorithm, ranges for the number of operations are now given. One operation consists of one multiplication and one summation. Before we start treating the particular steps, some abbreviations are introduced. The number of operations required by a single angle-sum test (see Appendix) is denoted by c_{ip}; the number of operations needed for the evaluation of an orthonormal system of spherical harmonics $\{Y_{n,j}\}_{\substack{n=0,\dots,M\\ j=1,\dots,2n+1}}$ at a single point $\xi \in \Omega$ is denoted by $c_{Y_{n,j}}$; and $c_{\Delta\Delta}$ denotes the number of operations to check the distance between two triangles. With these constants we can now start our considerations.

Setup stage

Step 1. The complexity of the tree construction only depends on the depth of the subdivision scheme. The sorting requires for any nodal point minimum one angle-sum test at each level and maximum N_0 tests at the root level plus maximum four tests at each other level. This yields for the number of operations $\sharp\,\mathrm{Op}_{S1}$ for step 1

$$(L+1)Nc_{ip} \le \sharp\mathrm{Op}_{S1} \le (N_0+4L)Nc_{ip}$$

i.e. it is of order $\mathcal{O}(N)$.

Step 2. Direct solution of a system of M linear equations requires $\mathcal{O}(M^3)(\sharp\mathrm{Op}_{S2} \hat{=} c_{int})$ operations and is therefore independent of N. Since M can be chosen to be small, this step is indeed negligible.

Step 3. For every finest-level panel we have to check the distance to maximum $\sum_{l=0}^{L} N_l$ triangles, if the far-field consists only of triangles in the finest level. $\sharp\mathrm{Op}_{S3}$ is independent of N and can be estimated by

$$\sharp\mathrm{Op}_{S3} \le \sum_{l=0}^{L} N_l c_{\Delta\Delta}$$

Step 4. Every node corresponds to one triangle in the finest level. The number of operations in step 4 is

$$\sharp\mathrm{Op}_{S4} = Nc_{Y_{n,j}} + N(M+1)^2$$

and is therefore $\mathcal{O}(N)$.

Step 5. The overall number of panels in levels $0,\dots,L-1$ is

$$\sum_{l=0}^{L-1} N_l = \sum_{l=0}^{L-1}(N_0 4^l), \quad N_0 \in \{8,20\}$$

and we have to sum up the far-field coefficients $\{V_{n,j}\}_{\substack{n=0,\dots,M\\ j=1,\dots,2n+1}}$ of four daughter triangles for each panel in levels $0,\dots,L-1$. Consequently, the number of summations in step 5 holds

$$\sharp\mathrm{Sum}_{S5} = \left(\sum_{l=0}^{L-1} N_l\right) 4(M+1)^2$$

Step 6. The number of triangles in the far-field is uniformly bounded to $\sharp\mathrm{FF}$, and we have to apply Eq. (12) to N_L triangles, so that $\sharp\mathrm{Sum}_{S6}$ fulfils

$$\sharp\mathrm{Sum}_{S6} \le N_L(M+1)^2 \sharp\mathrm{FF}$$

Evaluation stage

Step 1. Locating the finest-level triangle containing the evaluation point $\xi \in \Omega$ takes the same amount of numerical complexity as sorting one node in step 1 of the setup stage.

$$(L+1)c_{ip} \le \sharp\mathrm{Op}_{E1} \le (N_0+4L)c_{ip}$$

Step 2. Let $\sharp N_{NF}$ be the maximum number of nodes in the near-field and c_K the number of operations to evaluate $K(\xi\,\cdot)$ at a single point. Then the evaluation of Eq. (14) requires

$$c_{Y_{n,j}} + (M+1)^2$$

operations and Eq. (13) requires less than

$$\sharp N_{NF} c_K + \sharp N_{NF}$$

operations.

Summarizing these results, we obtain that

$$\begin{aligned}\sharp\mathrm{Op}_{\mathrm{Setup}} &= \sharp\mathrm{Op}_{S1} + \sharp\mathrm{Op}_{S2} + \sharp\mathrm{Op}_{S3} + \sharp\mathrm{Op}_{S4} \\ &\le ((N_0+4L)c_{ip} + c_{Y_{n,j}} \\ &\quad + (M+1)^2)N \\ &\quad + c_{int} + \left(\sum_{l=0}^{L} N_l\right) c_{\Delta\Delta}\end{aligned}$$

operations, and additionally

$$\begin{aligned}\sharp\mathrm{Sum}_{\mathrm{Setup}} &= \sharp\mathrm{Sum}_{S5} + \sharp\mathrm{Sum}_{S6}\\ &\leq (M+1)^2\left(4\sum_{l=0}^{L-1} N_l + N_L \sharp FF\right)\end{aligned}$$

summations have to be done in the setup stage. In the evaluation stage we have to perform an overall number of

$$\begin{aligned}\sharp\mathrm{Op}_{\mathrm{Eval}} &= \sharp\mathrm{Op}_{E1} + \sharp\mathrm{Op}_{E2}\\ &\leq (N_0 + 4L)c_{ip} + c_{Y_{n,j}} + (M+1)^2\\ &\quad + \sharp N_{NF}(c_K + 1)\end{aligned}$$

operations for the evaluation at a single point $\xi \in \Omega$.

It should be noted that most steps (steps 2, 3, 5 and 6 of the setup stage and step 1 of the evaluation stage) are independent of N. Thus, the larger the number of the nodes, the more efficient are these steps in comparison to the ordinary method of evaluation. Besides, the complete setup stage has to be performed only once in the algorithm. Indeed, the number of nodes in the near-field depends on N, but it can be kept small in comparison to the overall number of nodes. In the case of the matrix–vector product, the numerical effort of the evaluation stage is reduced to

$$\begin{aligned}\sharp\mathrm{Op}_{\mathrm{Eval}} &= N\sharp\mathrm{Op}_{E2}\\ &\leq N((M+1)^2 + c_{Y_{n,j}} + \sharp N_{NF})\end{aligned}$$

operations for the computation of the vector q, since evaluation is only required at the nodes and therefore the location is already done in step 1 of the setup stage. Furthermore, the value of $K(\eta_i \cdot \eta_j)$ can be stored, provided that it belongs to the near-field.

As mentioned previously, the panel clustering method can be applied to an iterative solution of a system of linear equations fitted by interpolation. An algorithm for this task is presented in Sect. 2.4.

2.4 Iterative solution of the spline interpolation problem

If we consider the spherical spline interpolation problem described in Sect. 2, the system of linear Eq. (2) is to be solved to obtain the coefficients $a_1, \ldots, a_N \in \mathbb{R}$. Direct methods for the solution are impractical for a large N since they require at least $O(N^3)$ operations and $O(N^2)$ storage. The error estimates for the spline interpolation on the sphere contain terms depending on the nodal width and the maximum value of the function. For example, the nodal width of a "Reuter grid"(see Fig. 11) with 12 500 grid points is about 200 km and for 50 000 grid points the width is still about 100 km if we assume a spherical earth with radius $R = 6378.137$ km. Due to this fact, a large number of grid points is required to obtain tolerable results. The limits of today's computers in storage and computational resources are rapidly reached.

Observing the shape of a spherical radial basis function, one might consider setting all entries $K(\eta_i \cdot \eta_j)$ equal to zero for η_j with $-1 \leq (\eta_i \cdot \eta_j) < t^*$ for $i = 1, \ldots, N$ in order to obtain a sparse band matrix (cf. Schneider 1996). Then the linear equations could be solved iteratively; however, only localizing kernel functions K being "numerically zero" outside a certain cap allow this treatment. Iterative methods like the conjugate gradient method require one matrix–vector product in each iteration step, which is similar to the N-times evaluation of the considered functions. We take up the idea of cutting off the matrix at a certain t^*, but we approximate the entries for the case $-1 \leq (\eta_i \cdot \eta_j) < t^*$, $i, j = 1, \ldots, N$, instead of setting them equal to zero. To be more specific, consider a matrix–vector multiplication

$$q_j = \sum_{i=1}^{N} p_i K(\eta_i \cdot \eta_j), \quad j = 1, \ldots, N \tag{16}$$

occurring in any iteration step. For the calculation of q in Eq. (16), we use Algorithm 2.2 with some modifications:

- Steps 1, 2, and 3 of the setup stage must be performed only once before the iterative procedure starts.
- Since the coefficients p_i are changed in each iteration step, we have to perform steps 4, 5 and 6 in any iteration step.
- Evaluation is only required at the nodes $\eta_1, \ldots, \eta_N$. Hence, we are able to identify the finest-level panel containing η_i, $i = 1, \ldots, N$, in the setup procedure and, in addition, we may store $K(\eta_i \cdot \eta_j)$ if it is a near-field term.

The conjugate gradient method to solve the system of linear Eq. (2) corresponding to the prescribed data set $\{(\eta_1, y_1), \ldots, (\eta_N, y_N)\}$ has the following form.

Algorithm 2.3

Setup stage

Step 1. Hierarchical subdivision of the sphere (Sect. 2.1) down to a certain level L and distribution of the nodes into the triangles (includes location of the finest level triangles containing $\eta_1, \ldots, \eta_N$).

Step 2. Approximation of the kernel function $K : t \mapsto K(t)$ by interpolating it with a univariate polynomial of a certain degree M.

Step 3. Determination of far- and near-field panels for any finest-level panel.

Conjugate gradient method

```
• Choose $a^0 \in \mathbb{R}^N$ as starting vector:
  set $p^0 = r^0 = y - Ca^0 \qquad I = 0, \quad \varepsilon > 0$
• Iteration:
  while $(|r| > \varepsilon)$
    – Calculation of the far-field coefficients $V_{n,j}(\Delta_L^k)$ for
      each finest-level triangle $\Delta_L^k$ via formula (9) with
      the current coefficients $p_1^I, \ldots, p_N^I$.
    – Recursive computation of the far-field coefficients
      $V_{n,j}(\Delta_l^k)$ for coarser-level triangles via Eq. (11).
```

- Calculation of the values $W_{n,j}(\Delta_L^k)$ for any triangle in level L via Eq. (12) and deletion of the far-field coefficients $V_{n,j}(\Delta_l^k)$.
- Performance of the matrix–vector product $q^I = Cp^I$: use Eq. (13) to compute the near-field and Eq. (14) to compute the far-field of $q_i = G(\eta_i)$ with the current coefficients $p_1^I, \ldots, p_N^I$ for every η_i, $i = 1, \ldots, N$.
- Set

$$\begin{aligned} \alpha &= (r^I, r^I)/(p^I, q^I) \\ a^{I+1} &= a^I + \alpha p^I \\ r^{I+1} &= r^I - \alpha q^I \\ \beta &= (r^{I+1}, r^{I+1})/(r^I, r^I) \\ p^{I+1} &= r^{I+1} + \beta p^I \\ I &= I + 1 \end{aligned}$$

Finally the iteration is left with an approximate solution $a_1, \ldots, a_N$ of $Ca = y$. A perturbation analysis is presented by Glockner (1997).

3 Numerical tests

All the calculations in this section are based on the OSU91A model (cf. Rapp et al. 1991) restricted to expansion coefficients $U^\wedge(n,j)$ from $n = 3$ to $n = 180$, i.e.

$$U(x) = \frac{GM}{R} \sum_{n=3}^{180} \sum_{j=1}^{2n+1} U^\wedge(n,j) \left(\frac{R}{|x|}\right)^{n+1} Y_{n,j}\left(\frac{x}{|x|}\right)$$

$x = |x|\xi$, $|x| > R$, where $GM = 0.39860044 \cdot 10^{15}\,\mathrm{m}^3\,\mathrm{s}^{-2}$ is the product of the gravitational constant and the mass of the earth and $R = 6378.137$ km is the spherical radius of the earth. In Sect. 3.1 the CPU times required by Algorithm 2.2 compared with those taken by an explicit evaluation are shown via the problem of upward continuation of the gravitational potential U. In Sect. 3.2, Algorithm 2.3 is applied to spherical spline interpolation; Algorithm 2.2 is used for the reconstruction of U by spherical wavelets in Sect. 3.3.

3.1 Numerical integration (upward continuation)

If the gravitational potential U is assumed to be known on the Earth's surface (i.e. $|x| = R$), upward continuation means nothing other than convolution of U against the Abel–Poisson kernel

$$Q_h(\xi \cdot \eta) = \frac{1}{4\pi} \frac{1 - h^2}{(1 - 2h(\xi \cdot \eta) + h^2)^{3/2}}$$

To be more specific

$$U(x) = \int_\Omega Q_{\frac{R}{r}}(\xi \cdot \eta) U(R\eta)\, \mathrm{d}\omega(\eta), \quad x = r\xi$$

is the gravitational potential at height $r = |x|$. The low discrepancy method (cf. e.g. Cui and Freeden 1997) enables us to express $U(x)$ approximately by the sum

$$U_N(x) = \frac{4\pi}{N} \sum_{i=1}^{N} U(R\eta_i) Q_{\frac{R}{r}}(\xi \cdot \eta_i), \quad x = r\xi$$

This is an expression of the type of Eq. (11) with $a_i = (4\pi/N)U(R\eta_i)$, $i = 1, \ldots, N$.

We compare accuracy and CPU time (the calculations have been performed on a HP 9000 Series 700 workstation) for examples where the grid points $\eta_1, \ldots, \eta_N$ are located at different circles of latitude. More precisely, in terms of polar coordinates with a given $\gamma \in \mathcal{N}$, we let

$$\begin{aligned} &\vartheta_0 = 0, \quad \varphi_{01} = 0 \text{ (North Pole)} \\ &\vartheta_i = i\Delta\vartheta, \quad 1 \le i \le \gamma - 1, \quad \Delta\vartheta = \pi/\gamma \\ &\gamma_i = (2\pi/\arccos((\cos\Delta\vartheta - \cos^2\vartheta_i)/\sin^2\vartheta_i)), \\ &\qquad 1 \le i \le \gamma - 1 \\ &\varphi_{ij} = (j - 1/2)\frac{2\pi}{\gamma_i}, \quad 1 \le j \le \gamma_i \\ &\vartheta_\gamma = \quad \pi, \quad \varphi_{\gamma 1} = 0 \text{ (South Pole)} \end{aligned}$$

Consequently, every point $\eta_1, \ldots, \eta_N \in \Omega$ is determined by $(\vartheta_i, \varphi_{ij})$, $0 \le i \le \gamma$, $1 \le j \le \gamma_i$, and N is controlled by γ via $N(\gamma) \le 2 + 4/\pi\gamma^2$. A graphical impression of this lattice, known as a *Reuter grid*, is shown in Fig. 11.

Tables 1 and 2 show the first results for Algorithm 2.2 concerning accuracy and CPU times. The second and the third columns show the maximum and the mean errors for variational values of the degree in the polynomial interpolation. As expected, we observe smaller errors with an increasing degree M. Furthermore, we

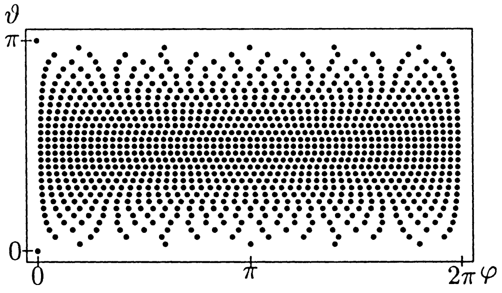

Fig. 11. Reuter grid for $\gamma = 30, N = 1129$

Table 1. Parameter studies for $\gamma = 150$, $N(\gamma) = 28\,568, K = Q_h$, $h = 0.99, L = 3, t^* = 0.95 \Rightarrow$ averaged 1486 points in the near-field

M	Max error ($\mathrm{m}^2\,\mathrm{s}^{-2}$)	Mean error ($\mathrm{m}^2\,\mathrm{s}^{-2}$)	Time far field coeff (s)	Total time (s)
10	$1.871 \cdot 10^{-1}$	$3.916 \cdot 10^{-2}$	7.92	290.09
15	$2.327 \cdot 10^{-2}$	$6.050 \cdot 10^{-3}$	17.04	301.05
20	$5.125 \cdot 10^{-3}$	$8.936 \cdot 10^{-4}$	29.78	315.57
25	$6.799 \cdot 10^{-4}$	$1.399 \cdot 10^{-4}$	46.40	338.33
30	$1.117 \cdot 10^{-4}$	$2.209 \cdot 10^{-5}$	66.26	370.71
35	$2.478 \cdot 10^{-5}$	$3.612 \cdot 10^{-6}$	89.51	398.70
40	$3.474 \cdot 10^{-6}$	$6.142 \cdot 10^{-7}$	117.24	431.67

Table 2. Parameter studies for $\gamma = 150$, $N(\gamma) = 28\,568$, $K = Q_h$, $h = 0.99$, $L = 4$, $t^* = 0.98 \Rightarrow$ averaged 522 points in the near-field

M	Max error ($\mathrm{m^2\,s^{-2}}$)	Mean error ($\mathrm{m^2\,s^{-2}}$)	Time far field coeff (s)	Total time (s)
10	$2.041 \cdot 10^{0}$	$6.153 \cdot 10^{-1}$	24.56	278.71
15	$4.853 \cdot 10^{-1}$	$1.156 \cdot 10^{-1}$	53.80	309.41
20	$1.126 \cdot 10^{-1}$	$2.133 \cdot 10^{-1}$	95.54	352.55
25	$4.277 \cdot 10^{-2}$	$1.170 \cdot 10^{-2}$	148.47	408.59
30	$1.337 \cdot 10^{-2}$	$3.018 \cdot 10^{-3}$	214.39	478.11
35	$2.986 \cdot 10^{-3}$	$6.463 \cdot 10^{-4}$	291.25	557.83
40	$1.222 \cdot 10^{-3}$	$2.372 \cdot 10^{-4}$	381.61	653.50

only present the CPU times for the calculation of the far-field coefficients in columns 4 and the total time requirement of the algorithm in columns 5, because the influence of a varying degree is negligible for the remaining steps. In the current example, the error between an explicit calculation of $U_N(x)$, $|x| = r$ and an approximation of $U_N(x)$ made by the panel clustering algorithm is evaluated at 11 400 points. The CPU times in Tables 1 and 2 are given for an 11 400-times evaluation via Algorithm 2.2. An explicit calculation of U_N at these points takes about 3911 s. Thus by the application of our method we obtain an excellent reduction of the computational costs, even for such small numbers of points.

Now we deal with the more interesting cases of increasing numbers of nodes N and evaluation points. Table 3 shows our considerations for different numbers of nodes and a fixed number (11 400) of evaluation points, while Table 4 presents some results for the inverse case. The given ratios point out that we obtain a better efficiency of the algorithm if we have to deal with large numbers of nodes or evaluation points. While the factor of time reduction is just 4.32 for 12 684 nodes, it reaches a value of nearly 18 for 114 444 nodes, which means that the new method is 18 times faster than an explicit calculation. A similar behaviour is recognizable if we increase the number of evaluation points. It develops from a factor of 11.88 for 11 400 points to 31.98 for 125 500 points.

Table 3. Time requirements and time ratios for different N with $K = Q_h$, $h = 0.99$, $M = 30$, $L = 4$, $t^* = 0.98$

N	Time expl calc (s)	Time alg (s)	$\frac{time_{alg}}{time_{expl}}$	$\frac{time_{expl}}{time_{alg}}$
12 684	1733.89	401.13	0.231	4.32
28 568	3911.53	478.31	0.122	8.18
50 832	6959.91	603.89	0.084	11.88
114 444	15669.66	875.92	0.056	17.89

3.2 Spline interpolation

On the basis of global data sets we investigate the behaviour of spline interpolation via Algorithm 2.3. The data sets contain the global geoid undulations according to the OSU91A model. The locations of the interpolation nodes are again given by the Reuter grid (Fig. 11). The geoid undulations as references for the calculations of this section are visualized in Fig. 12. The picture is given in the ranges of latitude 90°N to 90°S and longitude 170°W to 190°E. As trial function for interpolation we use the Abel– Poisson kernel Q_h.

We present two examples of spline interpolation. The first one is based on 12 684 nodes, the second one on 50 832 nodes. In order to illustrate the effect of a grid of greater density, we show an intersection curve at one latitude (viz. ϑ = const = 0.012272) and an error plot in both cases. The results show that the original signal can

Table 4. Time requirements and time ratios for different numbers of evaluation of points with $K = Q_h$, $h = 0.99$, $M = 30$, $L = 4$, $t^* = 0.98$, $\gamma = 200$, $N(\gamma) = 50832$

N_{eval}	Time expl calc (s)	Time alg (s)	$\frac{time_{alg}}{time_{expl}}$	$\frac{time_{expl}}{time_{alg}}$
11 400	6959.9	586.01	0.084	11.88
20 200	12332.5	742.11	0.060	16.62
45 300	27656.5	1136.35	0.041	24.34
125 500	76620.1	2396.04	0.031	31.98

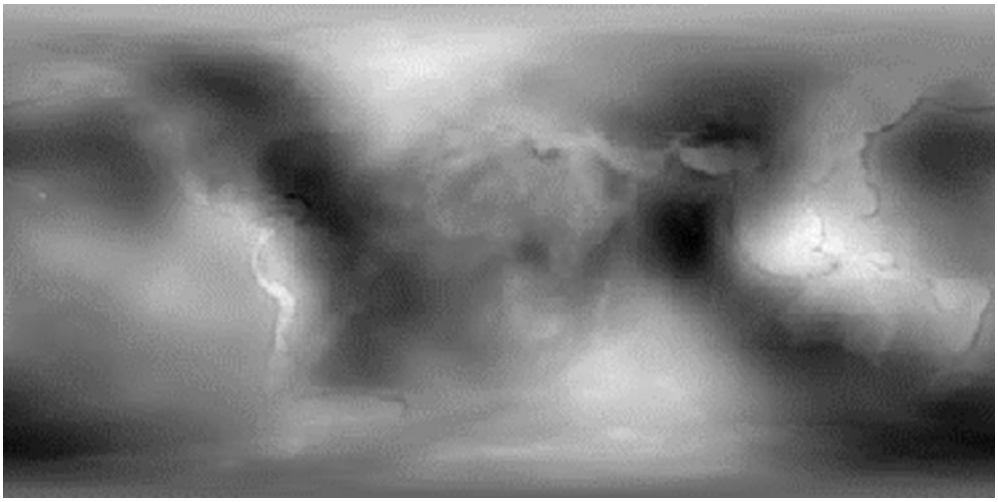

Fig. 12. Reference data: geoid undulations

be recovered only coarsely by a spline interpolant with 12 684 nodes (Fig. 13 and 14). Since it is known that the error occurring depends strongly on the nodal width of the given grid and convergence to the original function is reached if the nodal width tends to zero (cf. Freeden and Windheuser 1997; Freeden et al. 1998), we have to refine the nodal grid to obtain also the fine structures of the signal. As already mentioned, ordinary methods for the solution of systems of linear equations are inappropriate for this task, but our panel clustering method enables us to obtain better results by processing large sets of observations. This assumption is confirmed by the results we obtained from the interpolation of 50 832 nodes. In so doing we reach a strong improvement of the error (Fig. 16), so that even the fine structures of the function can be received. There is no visible difference between the signal and the interpolant in Fig. 15.

3.3 Wavelet reconstruction

According to Freeden and Schreiner (1998) and Glockner (1997), the gravitational potential U at the spherical Earth's surface admits a resolution in terms of spherical wavelets as follows:

$$U(R\xi) = \sum_{k=0}^{\infty} \mathscr{R}_k(U)(\xi), \quad \xi \in \Omega$$

where $\mathscr{R}_k(U)$ denotes the wavelet resolution of U at level k

$$\mathscr{R}_k(U)(\xi) = \sum_{i=0}^{A_k} \omega_i^k (WT)_{\Psi_0}(U)(k; \eta_i^k) \Psi_k(\xi \cdot \eta_i^k)$$

and $(WT)_{\Psi_0}(U)$ is the wavelet transform at scale k corresponding to the cubic polynomial wavelet (CP wavelet) given by

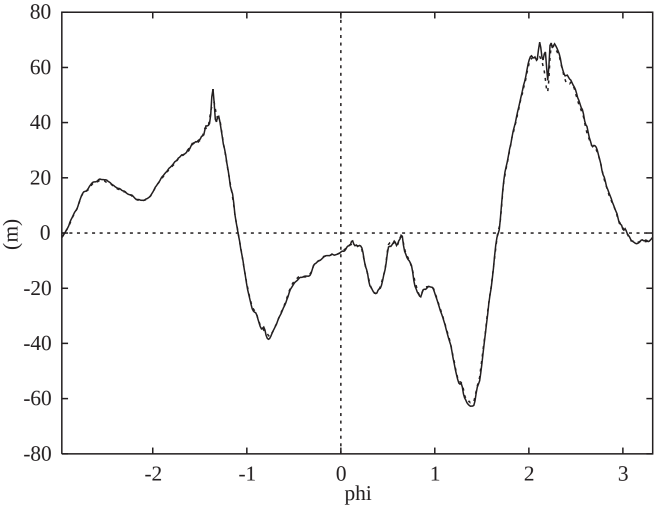

Fig. 13. Intersection of the signal (line) and the spline interpolant, Abel–Poisson kernel, $h = 0.97$ (broken line); 12 684 interpolation nodes at $\vartheta = 0.012272$

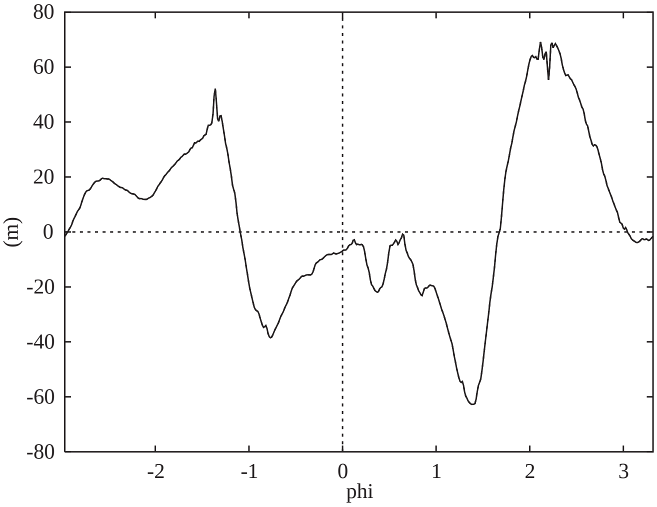

Fig. 15. Intersection of the signal (line) and the spline interpolant, Abel–Poisson kernel, $h = 0.96$ (broken line); 50 832 interpolation nodes at $\vartheta = 0.012272$

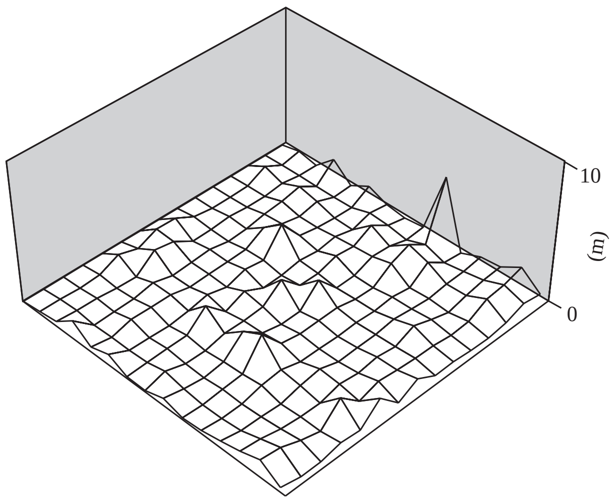

Fig. 14. Error surface plot of the spline interpolant, Abel–Poisson kernel $h = 0.97$, 12 684 interpolation nodes

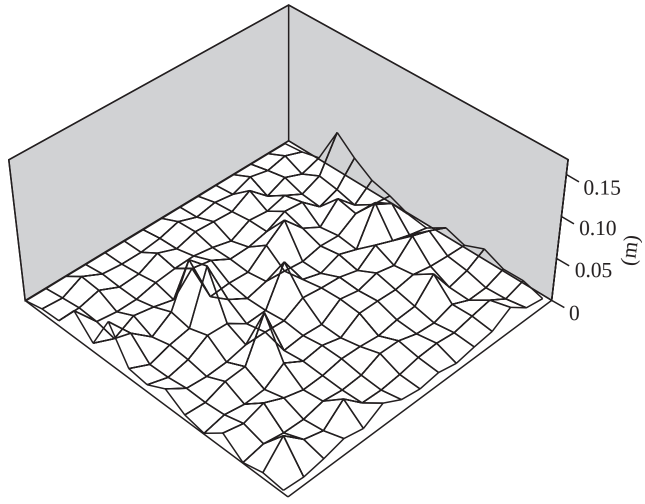

Fig. 16. Error surface plot of the spline interpolant, Abel–Poisson kernel, $h = 0.96$, 50 832 interpolation nodes

$$\Psi_k(\xi\cdot\eta)=\sum_{n=0}^{a_k}\frac{2n+1}{4\pi}\psi_k(n)P_n(\xi\cdot\eta),\quad (\xi,\eta)\in\Omega^2$$

with

$$(\psi_k(x))^2=(\varphi_{k+1}(x))^2-(\varphi_k(x))^2,\quad x\in[0,\infty)$$

and

$$\varphi_k(x)=\begin{cases}(1-2^{-k}x^2)(1+2^{-k+1}x), & x\in[0,2^k)\\ 0, & x\in[2^k,\infty)\end{cases}$$

$A_k=(2m_k+1)^2, a_k=2^{k+1}-1, k=0,1,\ldots.$ Figure 17 shows the CP wavelet Ψ_k for $k=1,\ldots,4$.

The weights $\omega_1^k,\ldots,\omega_{N_k}^k$ and the wavelet coefficients $(WT)_{\Psi_0}(F)(k:)$ are derived by the pyramid scheme described by Schreiner (1997). The lower scales are not of special interest here, because they require only a small number of nodes $\eta_1^k,\ldots,\eta_{N_k}^k\in\Omega$. We present results for the reconstruction of U at the scales 5, 6 and 7. If the sets X_j are chosen as by Discroll and Healy (1994), exact integrations require 16 384 nodes (scale 5), 65 536 nodes (scale 6) and 262 144 nodes (scale 7).

Figures 18–20 show polynomials of degree 35 approximating the CP wavelets $\Psi_5(t),\Psi_6(t)$ and $\Psi_7(t)$ in the interval $[-1,0.95]$. For each scale we finally show two figures. In Figs. 21–23, the figure on the left-hand side (a) contains the detailed behaviour of the signal obtained from an exact reconstruction and the right-hand picture (b) results from a reconstruction via Algorithm 2.2 in each case. It is remarkable that even the fine structures in the figures of the exact reconstruction can be recognized in the figures resulting from our algorithm.

Appendix

Angle sum test

During the execution of our algorithm, points must be related to spherical triangles several times, e.g. sorting

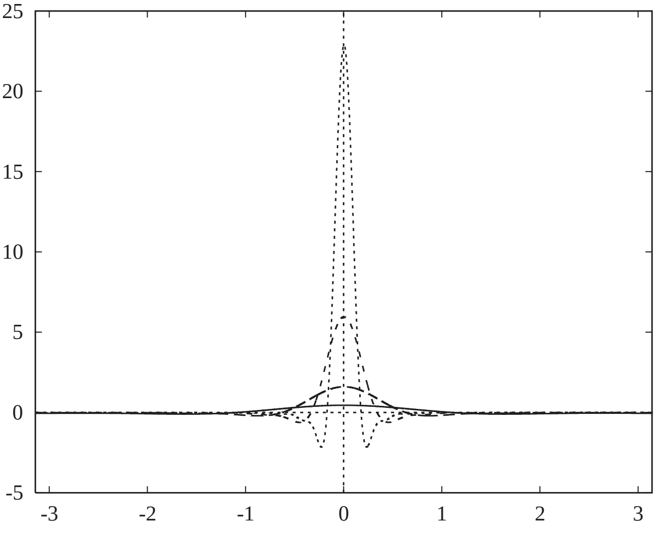

Fig. 17. CP wavelets $\Psi_k(\cos\vartheta)$ for $k=1,\ldots,4$

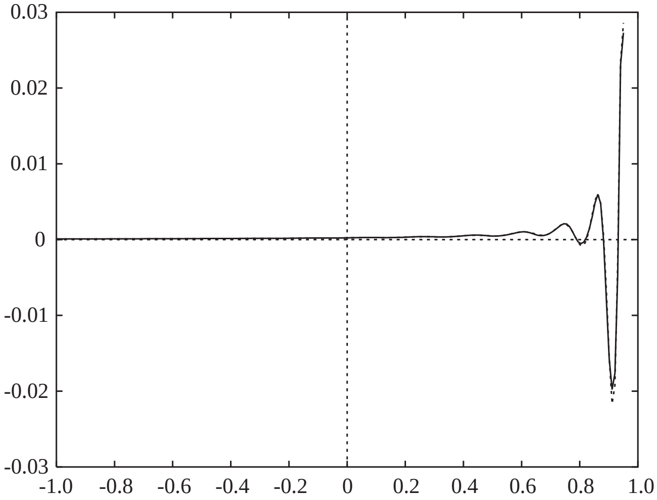

Fig. 18. Interpolation of the CP wavelet $\Psi_5(t)$ in the interval $[-1,0.95]$

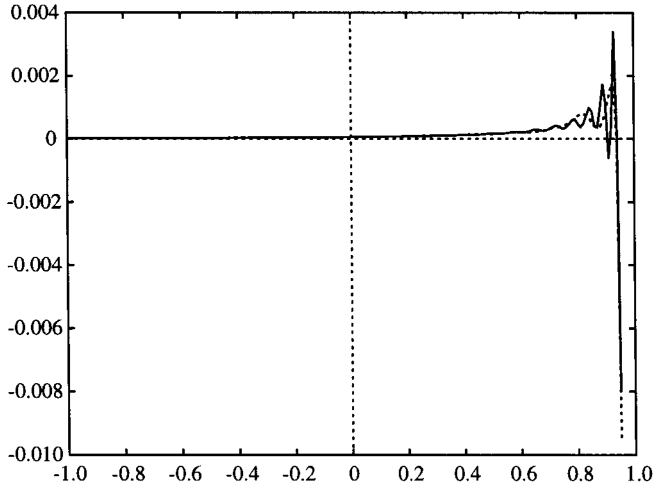

Fig. 19. Interpolation of the CP wavelet $\Psi_5(t)$ in the interval $[-1,0.95]$

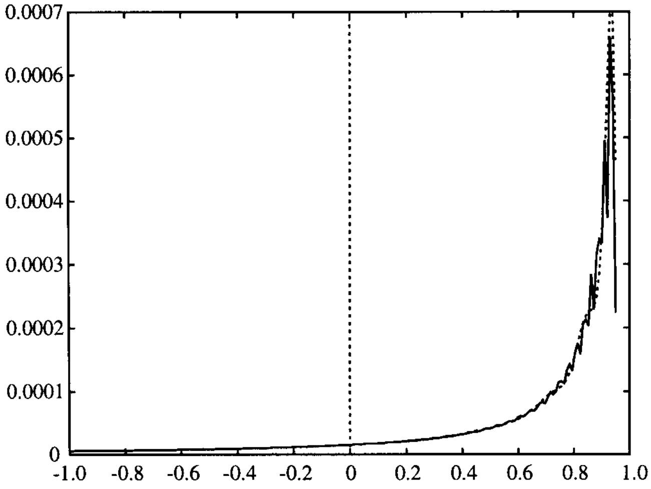

Fig. 20. Interpolation of the CP wavelet $\Psi(t)$ in the interval $[-1,0.95]$

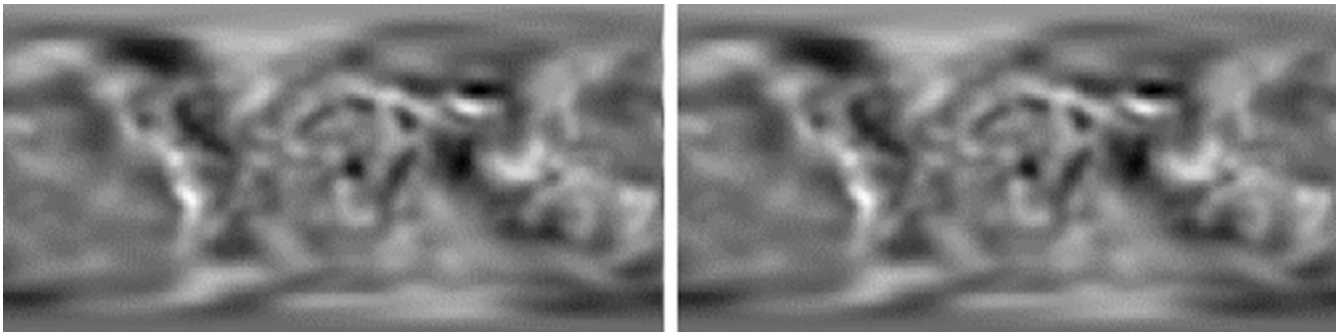

Fig. 21. $R_5(U)$, reconstructed (**a**) explicitly and (**b**) by Algorithm 2.2

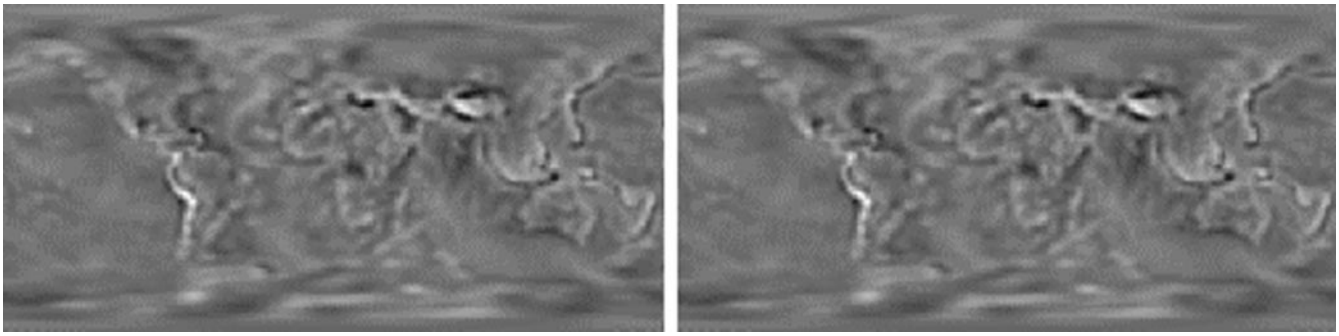

Fig. 22. $R_6(U)$, reconstructed (**a**) explicitly and (**b**) by Algorithm 2.2

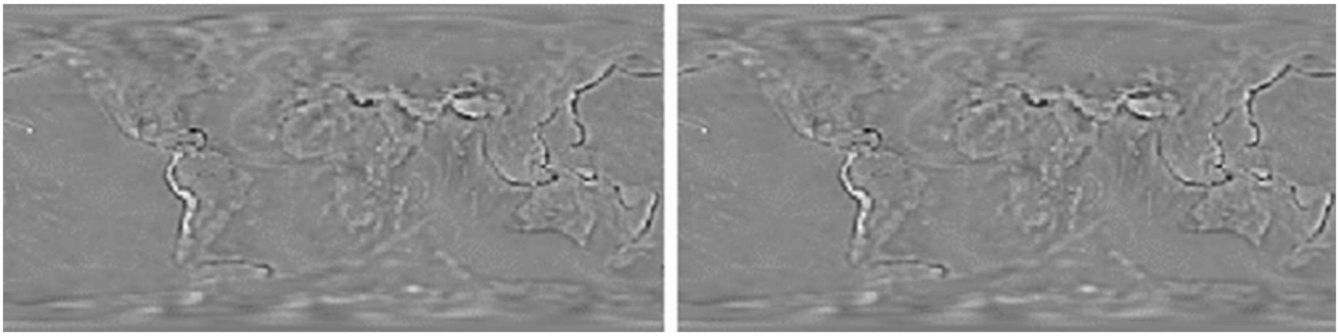

Fig. 23. $R_7(U)$, reconstructed (**a**) explicitly and (**b**) by Algorithm 2.2

the nodes in the beginning of the setup stage and location of the finest-level triangle containing the evaluation point in the evaluation stage. We decide whether or not a point is lying in a triangle via an angle sum test introduced here.

We are given a point P on the sphere and the three vertices V_1, V_2, V_3 of a spherical triangle. We want to check if the point P in Fig. 24 belongs to the triangle with the vertices V_1, V_2, V_3.

We start by setting

$$
\begin{aligned}
A &= (V_1 \cdot V_2)\\
B &= (V_2 \cdot V_3)\\
C &= (V_3 \cdot V_1)\\
a &= \arccos((V_1 \cdot P))\\
b &= \arccos((V_2 \cdot P))\\
c &= \arccos((V_3 \cdot P))
\end{aligned}
$$

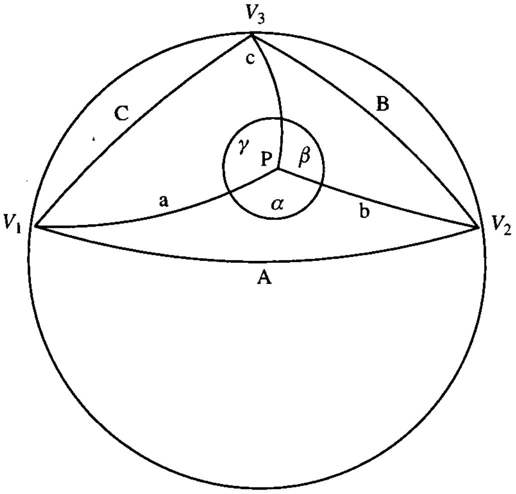

Fig. 24. Angle sum test

From spherical trignometry we know theorems for the calculation of spherical triangles. From the so-called side-cosine theorem we find

$$\alpha = \arccos\left(\frac{A - \cos a \cos b}{\sin a \sin b}\right)$$

$$\beta = \arccos\left(\frac{B - \cos b \cos c}{\sin b \sin c}\right)$$

$$\gamma = \arccos\left(\frac{C - \cos a \cos c}{\sin a \sin c}\right)$$

Now we have a criterion for the desired decision. If one of the three angles α, β, γ is equal to 180°, then P lies on one side of the triangle. In addition, if the sum of the angles $\alpha + \beta + \gamma$ is equal to 360° then P is in the interior of the triangle. In both cases P belongs to the triangle. Otherwise, if $\alpha \neq 180°$, $\beta \neq 180°$, $\gamma \neq 180°$ and $\alpha + \beta + \gamma \neq 360°$, then P is outside the triangle.

Remark. More details, demonstrating the economy and efficiency of the spherical panel clustering method presented above, can be found in the diploma thesis of Glockner (1997).

Acknowledgement. Support from the Stiftung Rheinland-Pfalz für Innovation is greatfully acknowledged.

References

Beatson RK, Newsam GN (1992) Fast Evaluation of radial basis functions I. Comp Math Applic 24: 7–19

Cui J, Freeden W (1997) Equidistribution on the sphere. SIAM J Sci Stat Comput 18: 595–609

Driscoll JR, Healy RM (1994) Computing Fourier transforms and convolutions on the 2-sphere. Adv Appl Math 15: 202–250

Freeden W, Schreiner M (1998) Orthogonal and non-orthogonal multiresolution analysis, scale discrete and exact fully discrete wavelet transform on the sphere. Constr Approx (in press)

Freeden W, Windheuser U (1997) Combined spherical harmonic and wavelet expansion – a future concept in Earth's gravitational determination. Appl Comput Harm Anal (ACHA) 4: 1–37

Freeden W, Schreiner M, Franke R (1997) A survey on spherical spline approximation. Surv Math Ind 7: 29–85

Freeden W, Gervens T, Schreiner M (1998) Constructive approximation on the sphere (with applications to geomathematics). Oxford Science Publications, Oxford

Glockner O (1997) Spherical panel clustering and its numerical aspects in spline and wavelet computation. Diploma Thesis, Geomathematics Group, University of Kaiserslautern, Kaiserslautern

Heiskanen WA, Moritz H (1967) Physical geodesy. Freeman, San Francisco

Rapp RH, Wang M, Pavlis N (1991) The Ohio State 1991 geopotential and sea surface topography harmonic coefficient model. Dept Geod Sci Surv Rep 410, OSU, Columbus, Ohio

Schneider, F (1996) A hierarchical spherical spline interpolation method and its application to a given set of air pressure data. In: Neunzert H (ed) Proc 8th Conf European Consortium for Mathematics in Industry, ECMI 94, pp 219–233

Schreiner M (1997) A pyramid scheme for spherical wavelets. AGTM Rep 170, Geomathematics Group, University of Kaiserslautern, Kaiserslautern

Überhuber C (1995) Computer Numerik 1. Springer, Berlin Heidelberg New York