



On a Fortran procedure for rotating spherical-harmonic coefficients

R. H. Gooding, C. A. Wagner

► To cite this version:

R. H. Gooding, C. A. Wagner. On a Fortran procedure for rotating spherical-harmonic coefficients. *Celestial Mechanics and Dynamical Astronomy*, 2010, 108 (1), pp.95-106. 10.1007/s10569-010-9293-3 . hal-00568376

HAL Id: hal-00568376

<https://hal.science/hal-00568376>

Submitted on 23 Feb 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Running Head: Procedure for Rotating the Gravitational Potential

On a Fortran Procedure for Rotating Spherical-Harmonic Coefficients

R. H. Gooding · C. A. Wagner

Submitted to Celest Mech Dyn Astr: 7 November 2009

Abstract The authors describe a Fortran subroutine that rotates the coefficients of a given spherical-harmonic model (in particular the geopotential). It is based on a paper by T. Risbo, working with the d -functions fundamental to axis rotations in quantum mechanics, his approach being equally applicable to the inclination functions of satellite geodesy. The subroutine applies Risbo's approach to evaluate, for a given inclination, stably and accurately, the necessary d -functions up to a specified degree, whilst at the same time 'rotating' (for each degree in turn) the values of the harmonic coefficients.

We follow Risbo's helpful example by supplying a listing of the new subroutine.

Keywords Rotation Matrices · Inclination functions · Rotation of spherical harmonics

Electronic supplementary material The online version of this article (doi ...) contains supplementary material, which is available to registered users. ~~[Editor: It was handled this way in our 2008 paper, so assume the same now (though we would originally have preferred to stay with Appendices A and B as here.)]~~

R. H. Gooding

Centre of Satellite Engineering Research, University of Surrey, Guildford, Surrey, England, UK

Present Address

R. H. Gooding (**envelope icon**) I assume that your actual icon will be placed here

19 Abbot's Ride, Farnham, Surrey, GU9 8HY, England, UK

e-mail: family.gooding@virgin.net

C. A. Wagner

National Oceanic and Atmospheric Administration, Laboratory for Satellite Altimetry,
Silver Spring, MD, USA

e-mail: carl.wagner@noaa.gov)

1 Introduction

From the advent of Earth satellites and the study of their orbital perturbations, it was foreseen how vital this study would become to a vast increase in accurate knowledge of the geopotential. The outcome has included a flood of papers on the role and computation of the so-called *inclination functions*, these being fundamental to the expression of the geopotential in an arbitrary axis system - in practice obtained by three successive Euler rotations from the Earth's 'natural' axis system (defined by the equator and Greenwich meridian) to a system based on the current orbital plane of a satellite of interest. The effective role of the functions is to 'rotate' the familiar spherical harmonics that model the geopotential, from the natural system to the required one, and one of the recent papers on these functions has been our own (Gooding and Wagner, 2008), which for brevity will from now on be referred to as G&W.

In spite of the considerable literature on the subject, there has been a striking absence of papers on a complementary one, the rotation of *all* the spherical-harmonic coefficients (now referred to as *CS*-coefficients) in a given geopotential model, as opposed to the harmonic functions themselves. The first author was reminded of this absence, after reading the paper by Risbo (1996), which expounded an approach that would be far more appropriate to tackling the complementary task than the ones we were familiar with.

The result has been a new Fortran subroutine that is based on one of Risbo's own, but extends it in two ways. First, the new subroutine does not (for each degree) merely compute the *d*-functions that are so crucial for Euler-based axis rotations in quantum mechanics and other fields - in addition it uses them (in parallel) to obtain the necessary quantities closely related to the inclination (or *F*-) functions. Second, these quantities are then used to 'rotate' the *CS*-coefficients.

Section 2 of the paper adds substantially more to this introductory background, whilst Section 3 (in sub-sections for the main components of the procedure) effectively provides a manual for using the subroutine. Section 4 summarizes results via a pair of graphs; Section 5 is devoted to how the subroutine can be modified to provide the actual *F*-functions; and Section 6 concludes the paper. The Fortran code itself **is listed in supplementary material**.

2 Further background

The seminal textbook on 'satellite geodesy' was by Kaula (1966). *Inter alia*, he introduced a version of the inclination functions as essential to the task of rotating the geopotential to an axis system based on a satellite's orbital plane. In retrospect, there were weaknesses in Kaula's approach to the functions' definition and notation, which are worth summarizing. First, the formula for getting each *F* from the relevant spherical-harmonic functions involved three series-summations, where, as anticipated by Allan (1965), a single series would suffice.

The second and third weaknesses were that Kaula defined the functions asymmetrically and with unhelpful signs that caused other authors to make unnecessary appeal to the so-called *entier* function to express them. The asymmetry weakness arose from Kaula's notation, $F_{lm}(I)$, where l and m are the degree and order of the coefficients C_{lm} and S_{lm} , and I is the inclination of the assumed satellite orbit. That left us with ' p ', a third integer, taking all values from 0 to l . The first of the present authors, having observed the asymmetry, which will be referred to again later, suggested (Gooding, 1966) the switch to a different 'mate' for m , to be denoted by k and such that its relation to Kaula's p is implied by

$$2p = l - k ; \quad (1)$$

k , like p (and m), takes $l + 1$ possible values, but the conventionally *legitimate* ones (to keep p integral) must have the same parity as l .

Allan's signs were an improvement on Kaula's, but not ideal, as the *entier* function could only be eliminated at the expense of making half the F 's pure imaginary and, unfortunately, this was defined by the parity of $l-m$, not that of $l-k$. Resolution of this (third) weakness only came when Allan (1973), by introducing the appropriate power of $i = \sqrt{-1}$, arrived at an optimal definition for the F 's. It was actually a return to the original definition by Izsak (1964), who anticipated Allan's adoption of the single-series summation (though he referred to the now universal " F 's" as " K 's"). It only remained for the present authors to recommend in G&W that (to make the notation entirely unambiguous) the index k should become a superscript, rather than a subscript. The relation of Kaula's original F 's to these possibly final ones can now be expressed via

$$F_{lm}(I) = (-1)^{E[\frac{1}{2}(l-m+1)]} F_{lm}^k(I) , \quad (2)$$

where the *entier* function, $E(x)$, supplies the integer part of x . We can illustrate the symmetry introduced by the use of k , via the relation connecting F 's for opposite values of k (Eq. 10 of G&W):

$$F_{lm}^k(I) = (-1)^{l-m} F_{lm}^{-k}(\pi - I) . \quad (3)$$

There remain two Kaula weaknesses to be remarked upon, though it is important to stress that this term is not meant to imply disdain - his book remains a *tour de force*! The first is that our journey through the " F -world" had effectively been taken much earlier when, starting with Wigner (1959, from 1931), the quantum physicists journeyed through the " d -world".¹ Our F 's are little more than a tailored subset of the d 's for a given degree l , where

¹ It is not easy to find a simple definition of the d -functions in the literature. Though they were introduced in Wigner's book, he did so (at his Eq. 15.E.1) in a rather oblique way. A somewhat clearer definition may be found in other books, with Appendix V of Brink and Satchler (1993) as good a source as any; and Risbo

summarizes the origins of the functions. Alternatively, Eq. 7 in the present paper may be regarded as providing a definition by series.

the complete set of d 's constitute the elements of an orthogonal matrix; the rows and columns of this matrix are effectively labelled by m and k , both running from $-l$ to $+l$. The 'tailoring' amounts to a simple relation between the conventionally *normalized* F 's and the relevant d 's (which can be regarded as being already normalized). This may be expressed, as in Eq. 16 of G&W, by

$$\bar{F}_{lm}^k(I) = i^{k-l} d_{km}^l(I) \bar{P}_{lk}(0) \{(2 - \delta_{0m})/(2 - \delta_{0k})\}^{1/2}, \quad (4)$$

where $\bar{P}_{lk}(0)$ is the usual *associated Legendre function*, evaluated for $I = 0$ (that is, on the equator) and normalized on the same basis as $\bar{F}_{lm}^k(I)$. The functions of the two Kronecker-deltas in Eq. 4 were explained in G&W, Eq. 16 there being the same as Eq.4 here.

But it is the effect of the initial i^{k-l} that is of most interest in Eq. 4: it leads to a real quantity only when l and k are of the same parity; but, since $\bar{P}_{lk}(0) = 0$ when l and k are of opposite parity, we have the gratifying fact that $\bar{F}_{lm}^k(I)$, as well as $d_{km}^l(I)$, is confirmed as *always* real!

If, in addition, we move the factor i^{k-l} from its present position in Eq. 4 and define an academic function by combining it with $\bar{P}_{lk}(0)$, this new function will always be positive in the 'same-parity' case, and the 'opposite-parity' zeroes can be replaced by smoothly-interpolated positive values. The \bar{F} values in the former case are what G&W described as 'legitimate', whilst the others were described (with irony) as 'forbidden'.

Kaula's final omission arose from confining his attention to the transformation of the harmonic functions themselves, ignoring their coefficients. This neglect of the transformation of the CS -values has been widespread, though Goldstein (1984) provides a notable exception (at the cost of having to extend the three Euler rotations to five).

3 Subroutine's basic components

We are describing a single subroutine, which we therefore refer to as "the SR", called *RisboBasedCSrotation* and largely based on the subroutine *pecurs* of Risbo. We explain its code via 'components', avoiding the words 'phase' and 'stage' because the components are (necessarily) intermingled, to produce an integrated subroutine.

3.1 In/out components

The SR's four dummy arguments are xi , $ambda0$ and $ambdapr0$ (three Euler angles, all in

degrees), and L_{\max} : λ_0 is the inclination I of the *primed* plane $Ox'y'$ (often the orbital plane of an Earth satellite) to the *unprimed* initial plane Oxy (in practice the Earth's equator with Ox towards Greenwich); λ_0 specifies λ_0 , the (unprimed) 'longitude' of the primed plane's ascending node; and λ'_0 is λ'_0 , the 'primed' longitude-equivalent of that node (now seen as *descending*). Finally, L_{\max} is the degree, l_{\max} , to which the geopotential model is being taken. In Section 4, we explain how and why a small adjustment is effectively made to the input value of I .

The geopotential model is read from unit 3 as a formatted file, with *GeopotentialModel.txt* its default name. The file is immediately *rewound* after being opened, so that the SR can be entered more than once in the same computing session.

The only *output* (apart from monitor prints and test output to unit 2) is to unit 4, for the formatted file *PostRotationModel.txt*. This output consists of l, k, C'_{lk}, S'_{lk} , for each l and every relevant k in turn, where k is the order of the specified (primed) harmonic pair. Since the standard notation, C_{lm} and S_{lm} , is used for the original model, one might have expected the adopted notation (relative to the primed plane) to be m' rather than k ; but, not only would this be cumbersome, the whole point of using k is that it ties in with the notations $\bar{F}_{lm}^k(I)$ and $d_{mk}^l(I)$ (or $d_{km}^l(I)$).

Apropos the preceding line, textbook inconsistency in notation between d_{mk}^l and d_{km}^l can be confusing, as is noted in G&W, just before its Eq. 5; in the listing of the present SR, it should be borne in mind that Risbo uses 'k' for the second argument of the array 'd', whereas the additional code here effectively uses 'k' for the first argument. That remark is relevant to the expression of 'd' by series in Section 3.3. Note: there is no real need for unnormalized versions of C and S , so the unbarred notation is used (throughout the paper) for normalized coefficients.

3.2 Risbo component

Risbo (1996) lists two subroutines, but only one (*pecurs*) was required here and we have modified it substantially as a component of our own SR. From our viewpoint its function is just to develop all the matrices of d -functions (up to l_{\max}) recursively. Thus the functions $d_{mk}^l(I)$, stored as the Fortran-array $d(0:2l, 0:2l)$ of dimension $(2l+1) \times (2l+1)$ with m and k ranging from $-l$ to l , are built from the functions $d_{mk}^{l-1/2}(I)$. But this cannot be done *in situ*, due to unavoidable interference, so Risbo builds an independent array, dd , only overwriting the existing d -array (of degree $l-1/2$) when dd (of degree l) is ready to replace it.

The procedure has to be initialized, and Risbo starts with $l = 1/2$, involving a 2×2 array for d . The Fortran assignments for the four array elements are $d(0,0) = d(1,1) = c$ and $d(0,1) = -d(1,0) = s$, where $c = \cos(1/2I)$ and $s = \sin(1/2I)$; the Fortran-array subscripts 0 and 1 there correspond to subscripts $-1/2$ and $+1/2$ in the d -matrix. As d evolves, the orthogonality of the matrix persists, with the aid of square roots where required.

The concept of *matrix convolution* is crucial, but Risbo does not explain it and it may be as unfamiliar to most mathematicians as it was to us. The concept may be defined via the

general formula for the convolution product of a pair of (arbitrary) matrices, say A and B , of dimensions $M \times N$ and $P \times Q$, respectively, where all four (array) subscripts start from zero and the maximum row-subscript of A (for example) is $M-1$. Then if the convolution product of A and B is to be C , with its subscripts again starting from zero, the dimension of C has to be $(M+P-1) \times (N+Q-1)$ and the formula for the general element is $c_{ij} = \sum a_{mn} b_{pq}$, where the summation is over all m and p that add to i , and all n and q that add to j .

If matrix B is of dimension 2×2 , the rows and columns of A can be thought of as being increased in length by 1 as a result of evolution to C ; and the new multiplication (unlike the familiar matrix multiplication) is automatically commutative. Applying this to the present context, starting from the specific 2×2 d -matrix above, we can multiply it indefinitely by itself to generate matrices of dimension 3×3 , 4×4 , 5×5 , etc, in turn. They correspond to $l = 1$, $1\frac{1}{2}$, 2 , etc, with general l (integer or half-integer) corresponding to dimension $(2l+1) \times (2l+1)$; moreover, by multiplying each element by a simple (and appropriate) factor, we can preserve orthogonality. The matrices generated in this way are precisely the d -matrices, with l going from the initialized $\frac{1}{2}$ up through 1 , $1\frac{1}{2}$, 2 , etc.

Risbo covered the procedure by four lines of code inside three do-loops. Their mathematical equivalents, as equations for increments (Δ) of the relevant quantities, are

$$\Delta dd(i, k) = (2l - i)^{\frac{1}{2}} (c/2l) (2l - k)^{\frac{1}{2}} d(i, k) , \quad (5a)$$

$$\Delta dd(i+1, k) = -(i+1)^{\frac{1}{2}} (s/2l) (2l - k)^{\frac{1}{2}} d(i, k) , \quad (5b)$$

$$\Delta dd(i, k+1) = (2l - i)^{\frac{1}{2}} (s/2l) (k+1)^{\frac{1}{2}} d(i, k) , \quad (5c)$$

$$\Delta dd(i+1, k+1) = (i+1)^{\frac{1}{2}} (c/2l) (k+1)^{\frac{1}{2}} d(i, k) . \quad (5d)$$

We present Eqs. 5 in this way, with pairs of square roots separated and Risbo's code modified accordingly, to minimize the danger of overflow when l is very large. Another modification was made while the paper was being reviewed. It takes advantage (as Risbo did not), of some simple symmetries (with 'anti-symmetries' covered) in the dd 's; they are given by Sneeuw (1992), for example. The advantage emanates from considerable saving in time (for large l) when more than 50% of the dd -computation can be obtained with Eqs. 5 bypassed.

3.3 Unsophisticated proof of Risbo's tacit recurrence relation

We can validate Eqs. 5 by reformulating them as a recurrence relation for d_{mn}^l . In generating the required single equation, we decrease l by $\frac{1}{2}$ and replace the computer-array 'subscripts' i and k by the mathematical subscripts $m = i - l$ and $n = k - l$, running from $-l$ to $+l$ instead of 0

to $2l$. (It was to avoid a clash with Risbo's use of k as in Eqs.5 that we are temporarily using n as the second mathematical subscript of d .) Then our new equation (with implicit arguments I for the d -functions) is

$$d_{mn}^l = A_{++} d_{m+\frac{1}{2}, n+\frac{1}{2}}^{l-\frac{1}{2}} + A_{-+} d_{m-\frac{1}{2}, n+\frac{1}{2}}^{l-\frac{1}{2}} + A_{+-} d_{m+\frac{1}{2}, n-\frac{1}{2}}^{l-\frac{1}{2}} + A_{--} d_{m-\frac{1}{2}, n-\frac{1}{2}}^{l-\frac{1}{2}}, \quad (6)$$

where for brevity we omit the actual expressions for the A 's. If either of a pair of d -subscripts on the right-hand side (RHS) of Eq. 6 numerically exceeds $l-\frac{1}{2}$, that d is itself then zero, so in particular cases there are only two or three non-zero terms on the RHS. These cases correspond (in the computer's d -array) to elements on the boundary of the putative " d^l array" and outside the square of the " $d^{l-\frac{1}{2}}$ array".

We regard our proof of Eq. 6 as unsophisticated because we can validate it by use of a standard series for $d_{mn}^l(I)$, given, e.g, by Wigner (1959, from 1931) or by Brink and Satchler (1993). With c and s already defined, the series is

$$d_{mn}^l(I) = \sqrt{\frac{(l+n)!(l-n)!}{(l+m)!(l-m)!}} \sum_j (-1)^j \binom{l+m}{j} \binom{l-m}{l-n-j} c^{2l-n+m-2j} s^{n-m+2j}, \quad (7)$$

where the summation is effectively from $-\infty$ to $+\infty$, because the product of the binomial coefficients is always zero outside limits that need not be specified explicitly. (This, as an aside, is relevant to the remarks after Eq. 6.)

The crux of the proof of Eq. 6 is to work with the *general* summation index j when its left-hand side (LHS) is expanded by Eq. 7. From the corresponding expansions of the terms on the RHS of Eq. 6, we can infer the summation index needed for all to give the same powers of c and s . Though Eq.7 (with l replaced by $l-\frac{1}{2}$) gives powers only adding to $2l-1$, an extra c or s comes from Eqs. 5 via the A 's of Eq. 6. We can then simplify the five coefficients of $c^{2l-n+m-2j} s^{n-m+2j}$ we are seeking, by cancelling (from all of them) the overall (j -independent) square-root factor in Eq. 7. This leaves us with far simpler coefficients than the original A 's, in particular with just $(-1)^j [(l+m)!/j! (l+m-j)]$ for the LHS of Eq. 6, and this factor can then itself be cancelled from both sides of the reduced equation.

The final coefficients on the RHS of the reduced Eq. 6 are then $(l-n-j)/2l$, $j/2l$, $(n-m+j)/2l$ and $(l+m-j)/2l$, which combine to unity, the value we have forced on the LHS. This warrants a "*QED*" for the recurrence relation, since the starting point for induction is trivial, being Risbo's values for $l = \frac{1}{2}$, given earlier and easily confirmed via Eq. 7.

3.4 Getting-the-partials component

The 'partials' here are (for given I and l , and each m and k) the values of the four partial derivatives $\partial C'_{lk}/\partial C_{lm}$, $\partial C'_{lk}/\partial S_{lm}$, $\partial S'_{lk}/\partial C_{lm}$, $\partial S'_{lk}/\partial S_{lm}$, which are the linear multiples of the C_{lm} and S_{lm} , required to transform to C'_{lk} and S'_{lk} . They are derived from simple combinations of the d -functions for each pair, k and m , of indexes (l being assumed given).

To enlarge, the d -functions involve index values of both m and k that can be negative as often as they are positive. The \bar{F} 's, on the other hand, lack this symmetry, since only k is allowed to be negative. Further, the usual \bar{F} -application (to satellite-orbits) is limited to the primed plane itself, whereas in the present d -application there is no such limitation (with the rotated coefficients applying universally), though the values of both m and k are eventually restricted to being non-negative. The requirement is that to compute the partials for given l , m and k , we need two values of d , both involving the given k but one for the given m and the other for $-m$, the values being identical if $m = 0$. The two values are coded as d_+ and d_- , with division by $\sqrt{2}$ if either m or k is zero and by a 'full 2' if they are both zero. The sign of d_- must be reversed when m is odd, after which the SR sets the quantities C_{coef} and S_{coef} from $d_+ + d_-$ and $d_+ - d_-$ respectively.

Since the longitude of the desired node has been supplied to the SR as λ_0 , we might expect the first step in computing the partials to be a rotation by this angle, but two points arise here. First (and as noted in Section 1), the SR assumes a trio of rotations based on the conventional Euler angles, with the middle rotation about the y -axis, which then at once becomes the y' -axis; this means that the initial x -axis has only to be rotated through $\lambda_0 - 90^\circ$ for the y -axis to be automatically at the right place. Secondly, the longitude-factors of the spherical harmonic functions are based on the angle $m\lambda$, not just λ , with no rotation at all necessary for the 'zonals', so the angle really required (via its cosine and sine) is $m(\lambda_0 - 90^\circ)$. In the same way, the angle $k(\lambda'_0 - 90^\circ)$ is required in connection with the final rotation, but (because there is little need for longitude flexibility in the primed plane) a user may want to use the same value of λ'_0 on all occasions.

On the basis of the three Euler rotations, the required partials for each pair of m and k are computed via the quantities C_{coef} and S_{coef} referred to above.

3.5 Final component

This involves the accumulation of the values for each pair of rotated harmonics, C'_{lk} and S'_{lk} , via an *inner* loop corresponding to m , within an *outer* one corresponding to k . That happens in parallel with the 'partials component', with *everything* controlled by the *main* ' j -loop', where j (following Risbo in *pecurs*) is effectively $2l$ and hence integral even when l is not.

The scope of this final component has been limited in two obvious ways, due to the limited needs of the geopotential expansion. First, there is no need for non-integral l , so everything is now bypassed when j is odd. Second, the given harmonics are for $m \geq 0$ only, and the rotated harmonics are for $k \geq 0$ only; so the k -based and m -based loops do not cover negative values of either index.

4 Results

The SR has been tested for values of l_{max} up to 2000 for an artificial geopotential model

based on values of the (normalized) C_{lm} and S_{lm} compatible with Kaula's rule of thumb (as in Eq. 5.15 of Kaula (1966)) for these harmonics. Testing was based on the property for the harmonic coefficients that the sum of their squares is unchanged after any arbitrary rotation of the 3-axis reference system and specifically that, for each individual degree l ,

$$\sum_{k=0}^l (C'_{lk}{}^2 + S'_{lk}{}^2) = \sum_{m=0}^l (C_{lm}{}^2 + S_{lm}{}^2) . \quad (8)$$

So the quantity chosen as test criterion, for selected values of l , was the proportionate deficit of the LHS of Eq. 8 relative to its RHS, i.e. $(\text{RHS} - \text{LHS})/\text{RHS}$. (Assessment via relative *deficit*, rather than *excess*, was adopted on the naive basis, remarked upon in G&W, that if underflow was a major aspect of a computation, then it would be preferable to have most of any test results with a positive sign.)

We can now explain what was referred to in the first paragraph of Section 3.1, which is that the accuracy of the SR can be significantly improved via an insignificantly small change to the input I , effectively to xi itself but in reality to the value of one of c and s after they have been initially set from $1/2I$ (following Risbo). With neither c nor s ever negative (and in general not equal), the SR replaces the *smaller* of them by the square root of $(1-r)(1+r)$, where r is the *larger* one. The goal here is the consistency of c and s by making $c^2 + s^2 = 1$ as accurately as possible. This goal is different from the one described in Section 10.1 of G&W, since there it was the cosine and sine of I itself that needed to be as consistent as possible. The least advantage is gained (in each case) when cosine and sine are numerically equal, with $I = 90^\circ$ here, and 45° or 135° in the G&W procedure.

Results for the relative deficit are summarized by two figures. Figure 1 displays the significant improvement in accuracy resulting from the effective adjustment of the input I (via c and s , as just explained). For a particular value of l , specified by $l_{\max} = 1000$, two curves are shown for the full range of I from 0° to 180° , the other input angles (λ_0 and λ'_0) being fixed (at 90°) along both curves - *fixed* because they are not critical in testing for large l_{\max} , and 90° because this corresponds to zero for the two longitudes of the primed polar direction (z' axis). The dashed curve shows relative deficit when no adjustment has been made, whilst the continuous curve shows the vast improvement that can follow when one has.

[space for .ps Fig somewhere here, with note on this at the very end of this .doc file]

Fig.1 Variation of relative deficit (R.D) with I , for degree $l = 1000$

Though the dashed curve's points in Figure 1 are, in principle, at 10° intervals in I , a few extra points were added at both ends, and we see that the terminal points (at 0° and 180°) both involve 'overshoots' of relative deficit to about 25×10^{-15} . The phenomenon disappears

with the other curve, so the explanation seems to arise from the 'trigonometric inconsistency' (when not adjusted) that has been described.

Figure 2 complements Figure 1 by plotting four curves of relative deficit against l , for somewhat arbitrary values of I . Each curve terminates (at $l = 1000$) consistently with the continuous curve of Figure 1.

[space for .ps Fig 2, etc, as for Fig 1 above]

Fig.2 Variation of relative deficit (R.D.) with l , for four values of I

The most significant feature of the curves in Figure 2 is that there is a common trend towards linearity as l increases. It is likely that this trend emanates from the intrinsic stability of the process, strongly indicated even for results (not shown) from before the consistency adjustment was introduced. This slow, linear increase in relative deficit seems the more remarkable when one recalls that the number of computing operations (to produce a complete set of C' and S' for a given l) increases quadratically (with l), not just linearly.

To sum up, we conclude that the accuracy and stability of the rotation process are excellent. In Section 5 we conclude that this is also true when the option is taken to output \bar{F} -functions.

5 Results from added subroutine option

The role of the \bar{F} -functions in the SR has effectively been as a link between the explicitly computed d -functions and the goal of rotating a set of spherical-harmonic coefficients. Recognizing Risbo's success in his accurate derivation of the d 's, we asked ourselves: is it not a simple matter to compute, directly from the d 's, the \bar{F} 's themselves?

The answer is *yes*, apart from the objection that there is rarely a demand for a 'complete' set of three-index \bar{F} 's in the same way as for the two-index CS -coefficients. What is more often required (for given I) is just a subset for a range of l -values and fixed values of m and k .

However, a small amount of additional software suffices to generate subsets of \bar{F} 's of better accuracy than was achieved in G&W, in particular with no longer any major error due to the onset of excessive underflow (or overflow), this being the bugbear of G&W. The core of our own additional code (within the usual subroutine *RisboBasedCSrotation*) is the optional statement "Fbar = FbarViaEqHarm(j/2,k-j/2,i-j/2,d(i,k))", which uses the current $d_{km}^l(I)$ to obtain the required value of $\bar{F}_{lm}^k(I)$ by the use of Eq. 4. Suitable code for the function *FbarViaEqHarm* is listed in [supplementary material](#), together with a further function, *EqHarm*, based on code by Wagner (1983), to provide $\bar{P}_k(0)$ as required in Eq. 4.

We include in the basic subroutine's code all the extra statements needed for the actual procedure that will now be described; they appear as comments, to be activated whenever the extension is wanted.

To validate the extra software, the results for a particular example were compared with those from Table 3 of G&W. The essence of that table had been the identity (or invariance) satisfied by the complete set of \bar{F} 's for a given degree, l , namely:

$$\sum_{k=-l}^l \sum_{m=0}^l [\bar{F}_{lm}^k(I)]^2 = 2l + 1 . \quad (9)$$

We used that identity (as Eq. 17 in G&W) before, to assess our "rapid stable procedure" by defining R.D (Eq. 34 in G&W) as a *relative deficit*, different from that of Section 4 here. The definition, based on Eq. 9 is

$$\text{R.D} = 1 - \sum_{k=-l}^l \sum_{m=0}^l [\bar{F}_{lm}^k(I)]^2 / (2l + 1) . \quad (10)$$

We had expected (in G&W) a preponderance of (very small) positive values for R.D, since a *major* source of *minor* discrepancy in the two sides of Eq.9 was underflow in $[\bar{F}_{lm}^k(I)]^2$.

Such underflows (for a few values of m and k) occurred, even for quite small values of l , being intrinsic to the recurrence formula (in l) we were using for (denormalized) F , in spite of which our procedure compared favourably with others in the field.

Table 1 here is a version of the G&W Table 3 with results added from the extended new subroutine, noting that the value of I (as chosen before) is 1° .

Table 1 Comparison of R.D and underflow counts (U.C) between the authors' previous study and the present one

[\[Space for this Table, at end of the file\]](#)

Table 1 has some striking features. The most obvious one reflects the main weakness in the earlier (G&W) algorithm, viz., the sudden catastrophic rise in R.D when l reaches 1024. As explained on pages 265-266 of G&W, this is the value of l at which (for the assumed precision) the l -independent factor in the initializing of the G&W recurrence relation happens to underflow when both m and k are also 1024 (only by coincidence a power of 2). In the new algorithm there are, overall for this value of l , 289 more underflows than before, but they are now all totally inconsequential, occurring in Risbo's implicit *recurrence* (not in its *initialization*). The single crucial underflow from the earlier algorithm is now avoided, the value of $\bar{F}_{1024,1024}^{1024}(1^\circ)$ being 7.861934, the square of which (61.80996...) is consistent with Eq.

9 and the earlier underflow.

A second striking (and surprising) observation from Table 1 is that, despite the entirely different approach from G&W's, the new underflow counts are almost the same. This is largely because inconsequential underflows in the d -computation are inevitable when either c or s is small (see Section 4), and here $s = \sin 0.5^\circ$. The important thing, in comparison with the first observation, is the difference between the two sources of underflow. Thus, the importance of I -consistency in the G&W procedure has given way to the importance of $\frac{1}{2}I$ -consistency in the present paper and it is possible that the virtual halving of $\sin \frac{1}{2}I$ (in comparison with $\sin I$) accounts for the slightly greater number of (inconsequential) underflows when using the current SR for such a low inclination.

The most gratifying observation from Table 1 is that (for this inclination) the new R.D is consistently so very much smaller than the old one from G&W. For higher inclinations, in particular for $I = 90^\circ$, where $c = s$, the position is likely to be reversed, with the new R.D much larger than was found for the Table 1 of G&W

6 Conclusion

The paper has described a subroutine for the rotation of the CS -coefficients of a given spherical-harmonic model (here an assumed geopotential). Thanks to access to Risbo's subroutine, which generates the elegant d -functions (usually associated with quantum mechanics but actually of far wider application), the new subroutine is extremely stable, accurate, fast, and essentially unlimited in regard to the degree and order of the coefficients to be rotated.

It was not the authors' original intention to add 'flexibility' to the subroutine's other virtues, but we soon saw how easy it was to introduce optional code to generate satellite-geodesy's (normalized) F -functions themselves, being already barely bypassed *en route* to the rotated coefficients. As another possible application for optional code, it might be required to rotate all the coefficients *of* a given degree, as opposed to *up to* a given degree.

In regard to the first point, readers will be aware that (despite the logic inherent in the authors' preference for a particular definition, there is still no unanimity as to how the F 's and \bar{F} 's should be defined. Fortunately, however, the rotation of C 's and S 's does not depend on those definitions, but on their own intrinsic definition, the coefficients being assumed to be normalized in accord with usual practice.

Acknowledgement

We are indebted to Prof. Sneeuw, currently at the Institute of Geodesy within the University of Stuttgart, who sent the first author a copy of Torben Risbo's key paper; and our overall thanks to Dr. Risbo hardly needs further emphasis! We also wish to thank the reviewer who, as part of the review,

made many detailed suggestions for the paper's improvement, both in the text and the code.

References [followed by Appendices, probably to become supplementary material]

- Allan, R. R.: On the motion of nearly synchronous satellites. *Proc. Roy. Soc. A* **288**, 60-68 (1965)
- Allan, R. R.: Change of inclination in passing through resonance. *Recent Adv. in Dyn. Astr.*, 333-348 (1973)
- Brink, D. M., Satchler, G.R.: *Angular Momentum*, 3rd Ed. Clarendon Press, Oxford (1993)
- Goldstein, J. D.: The effect of coordinate system rotation on spherical harmonic expansions: a numerical method. *J. Geophys. Res.* **89**, 4413-4418 (1984)
- Gooding, R. H.: Satellite motion in an axi-symmetric field, with an application to lunisolar perturbations. Royal Aircraft Estab. Tech. Report 66018 (1966)
- Gooding, R. H., Wagner, C. A.: On the inclination functions and a stable procedure for their evaluation together with derivatives, *Celest. Mech. Dyn. Astr.*, **101**, 247-272 (2008)
- Izsak, I. G.: Tesseral harmonics of the geopotential and corrections to station coordinates. *J. Geophys. Res.* **69**, 2621-2630 (1964)
- Kaula, W. M.: *Theory of Satellite Geodesy*, Blaisdell, Waltham (Massachusetts) (1966)
- Risbo, T.: Fourier transform summation of Legendre series and D-functions, *J. of Geodesy*, 70, 383-396 (1996)
- Sneeuw, N. J.: Representation Coefficients and their Use in Satellite Geodesy. *Manuscr. Geodaetica* **17**, 117-123 (1992)
- Wagner, C. A.: Direct determination of gravitational harmonics from low-low GRAVSAT data, *J. Geophys. Res.* **88** (B12), 10310-10321 (1983)
- Wigner, E.P.: *Group Theory and its Application to the Quantum Mechanics of Atomic Spectra*. Academic Press, New York (Translated from *Gruppentheorie*, Vieweg, Braunschweig, 1931) (1959)

Appendix A Listing of Fortran Subroutine RisboBasedCSrotation

```
subroutine RisboBasedCSrotation (xi,ambda0,ambdapr0, Lmax)
!   First 3 (dummy) inputs are Euler angles, in unconventional order
!   beta, alpha and gamma (beta being inclination of primed plane, with
!   alpha & gamma the "longitudes" of the "beta-rotation node"); fourth
!   input is the final degree to which coefficient rotation is required.
!   File-input array is GeopotentialModel.txt, containing C,S pairs.
!   File-output array is PostRotationlModel.txt, getting C',S' pairs.
!   There is also some printing to Unit 2.
  implicit real*8 (A-H,O-Z)
  parameter (ndim=1000, sqrt2=1.4142135623731, radian=57.295779513082320, &
&   halfpi = 90./radian)
  dimension sqt(0:2*ndim+2),d(0:2*ndim+2,0:2*ndim+2),dd(0:2*ndim+3,0:2*ndim+3)
  dimension CEGM(1:ndim, 0:ndim), SEGM(1:ndim, 0:ndim)
!o logical lTest ! see later re the F-option of Section 5
  write (2,*) 'xi, Lmax, ambda0, ambdapr0 are ', xi, Lmax, ambda0, ambdapr0
  open(3,Form='Formatted', MODE='read', FILE='GeopotentialModel.txt')
  open(4,Form='Formatted', MODE='write', FILE='PostRotationModel.txt')
  rewind 3
  read (3,1) ((CEGM(i,j), SEGM(i,j), j=0,i), i=1,Lmax)
1 format (10X, 2E25.15)
  write (4,2) xi, Lmax, ambda0, ambdapr0
```

```

2 format ('Rotated C/S values for I, Lmax, ambda0, ambdap0 = ', F8.3, I5, &
& 2F8.3 / ' L k C_Lk S_Lk')
thet = xi/radian ! 1st of 3 lines converting Euler angles from degrees
amb0 = ambda0/radian
ambpr0 = ambdap0/radian
jmax = 2*abs(Lmax)
do i=0,jmax
  sqt(i) = sqrt(real(i))
end do
s = sin(thet/2.)
c = cos(thet/2.)
if (s*s.lt.0.5) then ! consistency for I/2 is set up in these 5 lines
  s = sqrt((1d0 - c)*(1d0 + c)) ! xImod from 2*asin(s)*radian if wanted
else
  c = sqrt((1d0 - s)*(1d0 + s)) ! xImod from 2*acos(c)*radian if wanted
end if
! --- initialize d-matrix degree 1/2 (Risbo comment)
d(0,0) = c
d(0,1) = s
d(1,0) = -s
d(1,1) = c
! --- MAIN LOOP: combination of Risbo (for key d-file) & Gooding (for rotation)
do j=2,jmax
  L = j/2
  fj = j
  do i=0,j !**23.5.10 1st of 2 lines with "j+1" repaced by "j"
    do k=0,j !**23.5.10 2nd, validity having been checked
      dd(i,k) = 0.
    enddo
  enddo
!o lTest = mod(j,2).eq.0 .and. mod(1,50).eq.0
! Example of selecting a subset of L values for printing in the F-option]
!o if (lTest) then ! "l" is used for comments stopping the F-option
!o SumFsq = 0.
!o numUdrs = 0
!o end if
! --- recursion in two directions from d to dd (Risbo comment)
do i= 0,j-1
  kp = max(0,i-1) ! 1st of 3 lines exploiting symmetries ignored by Risbo
  kq = j-1-kp
  do k=kp,kq
    dd(i,k) = dd(i,k) + sqt(j-i)/fj*sqt(j-k)*d(i,k)*c
    dd(i+1,k) = dd(i+1,k) - sqt(i+1)/fj*sqt(j-k)*d(i,k)*s
    dd(i,k+1) = dd(i,k+1) + sqt(j-i)/fj*sqt(k+1)*d(i,k)*s
    dd(i+1,k+1) = dd(i+1,k+1) + sqt(i+1)/fj*sqt(k+1)*d(i,k)*c
  enddo
enddo
do i=0,L ! next 10 lines complete the symmetry exploitation
  do k= i,j-i
    if (i.le.k.and.dd(i,k).ne.0.) then
      dd(j-k,j-i) = dd(i,k)
      dd(k,i) = dd(i,k)
      if (mod(k-i,2).eq.1) dd(k,i) = -dd(i,k)
      dd(j-i,j-k) = dd(k,i)
    end if
  enddo
enddo
do i=0,j
  do k=0,j
    d(i,k) = dd(i,k)
!o if (lTest.and.k.ge.j/2 .and. mod(j-i,2).eq.0) then

```

```

!o          Fbar = FbarViaEqHarm(j/2,k-j/2,i-j/2,d(i,k))
!o          if (Fbar.eq.0d0) numUdrs = numUdrs + 1
!o          SumFsq = SumFsq + Fbar**2
!o      end if
!o  enddo
!o  enddo
!o  if (lTest) then
!o      crit = j + 1d0          ! because j = 2L
!o      RelDef = 1d0 - SumFsq/crit
!o      write (2,4) j/2,RelDef,NumUdrs
!o  4      format ('L, RelDef, G&Wudrs = ', I4, F22.16, I9)
!o  end if
!--- do the rotation (Gooding), ignoring half-integer degrees
if (mod(j,2).eq.0) then
    Sos = 0.          ! initialize for Eq. 8
    SosPrime = 0.      ! ditto
    L= j/2
    do jj=0,L
        Sos = Sos + CEGM(L,jj)**2
        if (jj.ne.0) Sos = Sos + SEGM(L,jj)**2
    end do
    do k=0,L
        Crot = 0.
        Srot = 0.
        kK = k + L
        do m=0,L
            dplus = d(kK,L+m)
            dminus = d(kK,L-m)
            ambdam = (amb0 - halfpi)*m
            ambdak = (ambpr0 - halfpi)*k
            if (m.eq.0) then
                dplus = dplus/sqrt2
                dminus = dminus/sqrt2
            end if
            if (k.eq.0) then
                dplus = dplus/sqrt2
                dminus = dminus/sqrt2
            end if
            if (mod(m,2).eq.1) dminus = -dminus
            Ccoef = dplus + dminus
            Scoef = dplus - dminus
            Cm = cos(ambdam)
            Sm = sin(ambdam)
            Cprimed = cos(ambdak)
            Sprimed = sin(ambdak)
            cc = Cm*Ccoef
            cs = Sm*Ccoef
            sc = -Sm*Scoef
            ss = Cm*Scoef
            dCdC = Cprimed*cc - Sprimed*sc
            dCdS = Cprimed*cs - Sprimed*ss
            dSdC = Sprimed*cc + Cprimed*sc
            dSdS = Sprimed*cs + Cprimed*ss
            Crot = Crot + dCdC*CEGM(L,m) + dCdS*SEGM(L,m)
            Srot = Srot + dSdC*CEGM(L,m) + dSdS*SEGM(L,m)
        end do
        SosPrime = SosPrime + Crot**2 + Srot**2
        write (4,3) L, k, Crot, Srot
    3 format (2I5, 2E25.15)
    end do
    RelDef = 1. - SosPrime/Sos

```



```

        if (L.eq.Lmax) write (2,*) 'L, Sos, SosPrime & RelDef = ', &
&      L, Sos, SosPrime, RelDef      ! Delete "if (...)" to output for every L
    end if
enddo
return
end                                !of subroutine RisboBasedCSrotation

```

Appendix B Listing of two supplementary Fortran Functions

```

function FbarViaEqHarm(L,m,k,d)
!   First three (dummy) inputs are the usual indexes; the fourth is a d-matrix element.
!   The only output is FbarViaEqHarm itself, being F(bar) for I = 0 (equatorial).
implicit real*8 (a-h,o-z)
FbarViaEqHarm = EqHarm(L,k)*d ! Use 'equatorial harmonic'
if (k.eq.0) FbarViaEqHarm = FbarViaEqHarm*sqrt(2.)
if (m.eq.0) FbarViaEqHarm = FbarViaEqHarm/sqrt(2.)
return
end                                ! of function FbarViaEqHarm(L,m,k,d)

function EqHarm(L,k) ! (From SR 'pnm' of Wagner, & relevant to "Fbar's via d's")
!   The two (dummy) inputs are the usual indexes
!   The only output is FbarViaEqHarm itself, being the (fully normalized) associated
! Legendre functions for I = 0 or zero when L and m are of opposite parity
implicit real*8 (a-h,o-z)
m = abs(k)                                ! as when used for P_Lk(0), k may be negative
if (mod(L-m,2).eq.1) then
    EqHarm = 0d0                            ! need extra code for non-zero 'forbidden whites'
    return
end if
!   Set starter for (P_mm)sq by recurrence relation
EqHsq = 1d0
do i=1,m
    EqHsq = EqHsq*(2d0*i+1d0)/2d0/dbl(i)
    if (i.eq.1) EqHsq = 2d0*EqHsq
end do
!   Get desired even-parity result via the recursion of Wagner (1983)
do i=m+2,L,2                                ! using coefficient of CAW's Eq'n (6)
    EqHsq=EqHsq*(i-m-1d0)/dbl(i-m)*(i+m-1d0)/dbl(i+m)*(2d0*i+1d0)/(2d0*i-3d0)
end do
EqHarm = dsqrt(EqHsq)
if (mod(L-m,4).eq.2) EqHarm = -EqHarm
if (k.lt.0.and.mod(m,2).eq.1) EqHarm = -EqHarm
return
end                                ! of function EqHarm(L,k)

```

Legends for Figures (Actual .ps figures in separate attachments ?)

Figs 1 & 2 were originally submitted in .pdf format (against CelM instructions), because our attempts to include them in .ps failed. I (RHG) have the .ps files from my co-author {CAW}, who has the software to read them (as Figs!) but I don't. For me to insert them (VERY long files in pure txt format) now seems unlikely to be acceptable, so the best idea may be for CAW to send his .ps files directly, attached to a separate email (to Deepan Selvaraj). One reviewer stressed that the .pdf quality is not up to the standard of .ps (whence the formal CELE instructions no doubt). But in this version

Fig.1 Variation of relative deficit (R.D) with I , for degree $l = 1000$ [space for .ps above]

Fig.2 Variation of relative deficit (R.D.) with l , for four values of I [space for .ps above]

Table 1

Table 1 Comparison of R.D and underflow counts (U.C) between the authors' previous study and the present one

l	<u>Results from previous study</u>		<u>Results from present study</u>	
	R.D $\times 10^{16}$	U.C	R.D $\times 10^{16}$	U.C
100	284	286	0	287
200	943	10793	0	10791
300	204	38201	32	38199
400	486	83347	17	83381
500	732	146641	28	146696
550	-87	185170	4	185239
600	-442	228323	0	228376
650	-624	276107	0	276201
700	-253	328558	32	328636
750	-564	385664	79	385764
800	-1006	447445	62	447575
850	-1636	513945	58	514049
900	-1086	585096	98	585278
950	-355	660994	109	661149
1000	-484	741521	77	741775
1023	-1159	780167	92	780448
1024	301659391441197	781866	115	782155
1025	301466320492436	783564	109	783870
1026	375107328406188	785264	182	785583
1027	Fails due to Overflow!		24	787295
1100	So no more values available		144	917223
1200	etc	etc	148	1111593
1300			225	1325006
1400			168	1557422
1500			129	1808922
1600			163	2079471
1700			223	2369157
1800			98	2678003
1900			212	3006022
2000			298	3353224



