Fortran- and C-codes for higher order and degree geopotential and derivatives computation

Helio Koiti Kuga [1]
Valdemir Carrara [1]

[1] Instituto Nacional de Pesquisas Espaciais - INPE
Caixa Postal 515 - 12227-010 - São José dos Campos - SP, Brasil
hkk@dem..inpe.br, val@dem.inpe.br

Abstract. Geopotential computation in general did not present numerical problems in older models where order and degree were as low as 30. However with the appearance of much higher order and degree the older conventional algorithms could not be used anymore. The kernel of such algorithms is the recursive computation of the Legendre polynomials. They must be computed in their normalized recursions and variants to avoid numerical flaws at a cost of more expensive processing times mainly due to the need of square roots computations. This paper outlines the main implementation aspects of the geopotential computation and derivatives to higher order and degree. The codes were tested up to 2159 order and 2190 degree without any noticeable flaw. It is expected some numerical degradation near the poles, however to the 0.000001° of proximity, i.e. ±89.999999° latitude, no problems are reported. Computer codes were developed in double precision and are freely available in Fortran-77 and ANSI-C languages.

Keywords: Legendre polynomial recursions, high order and degree geopotential, geopotential derivatives

1. Introduction

Older models of harmonics coefficients did not need analysis or optimization for the numerical computation of geopotential. They were mainly devoted to ways of avoiding singularities and algebraic modifications of the original geopotential formula (Cunningham, 1970; Pines, 1973). The coefficients barely arrived at higher order and degree, e.g. 30 for GEM9 and 10 models (Lerch et al., 1979), to cause computational problems in geopotential computation. Such computation requires calculation of the Legendre polynomials, which should be recursively evaluated for high order and degree. One of the works concerned with numerical problems when computing the Legendre polynomials were from Lundberg and Schutz (1988) whom analyzed several recursions for the Legendre polynomials in fully normalized form up to order/degree 180. Nowadays the gravitational models easily start from order/degree 360 like EGM96 model (Lemoine et al., 1998) up to more than order/degree 2100 like EGM2008 (Pavlis et al., 2008). As such many different algorithms overcoming numerical hindrances had to be drawn, see e.g. Fantino and Casotto (2009). Herein one describes the standard-forward-column implementation proposed by Holmes and Featherstone (2002) which is believed to be numerically one of the most efficient algorithms. They are quite suited for geopotential evaluation and Earth orbit computations. Some representative results are also drawn to show the capabilities of the codes implemented. Sources codes as well as test codes are provided in both Fortran-77 and ANSI-C languages (Kuga and Carrara, 2012).

2. Geopotential

A material point (body) subject to attraction by a non-central gravitational field (Earth) suffers disturbances due to non-spherical and non-symmetrical distribution of its mass. This uneven distribution of mass is expressed by the so-called coefficients of spherical harmonics. Then the potential of a body relative to the Earth is calculated in a generic form by:

$$V = \frac{GM}{r} \sum_{n=0}^{\infty} \sum_{m=o}^{n} \left(\frac{a}{r}\right)^n [C_{nm} \cos m\lambda + S_{nm} \sin m\lambda] P_{nm}(\sin\psi) \tag{1}$$

where *V* is the potential, *G* is the universal gravitational constant, *M* is the Earth mass, *r* is the distance to body from the Earth center, *a* is the Earth equatorial radius, λ is the longitude of body, Ψ is the geocentric latitude of body, P_{nm} are the Legendre polynomials of order *n* and degree *m* , C_{nm}, S_{nm} are the spherical harmonics coefficients.

2.1 Spherical Harmonics Coefficients

According to Equation 1, the coefficients should represent faithfully the irregular shape of the distribution of the Earth's mass, so that the model of potential is represented properly. The determination of these coefficients is obtained experimentally by reduction and data analysis of artificial satellites and also from gravimetric methods.

Currently all the harmonics coefficients are listed in its fully normalized form, therefore the relation with the non-normalized ones is:

$$\begin{pmatrix} \overline{C}_{nm} \\ \overline{S}_{nm} \end{pmatrix} = \left[\frac{1}{\varepsilon_m (2n+1)} \frac{(n+m)!}{(n-m)!} \right]^{1/2} \begin{pmatrix} C_{nm} \\ S_{nm} \end{pmatrix} \tag{2}$$

where *n* is the degree, *m* is the order, $\varepsilon_m = \begin{cases} 1 & \text{if} \quad m = 0 \\ 2 & \text{if} \quad m \geq 0 \end{cases}$, and $\overline{C}$, $\overline{S}$ are the fully normalized coefficients. The *C* coefficients with *m=0* are called zonal coefficients, and the *S* with *m=0* are null, i.e. $S_{n0} = 0$. The remaining coefficients *C* and *S* (those for $m \geq 1$) are called the sectoral coefficients, when $m = n$, and tesseral coefficients when $m \neq n$. Therefore, in this case $m \geq 1$, and Equation 2 can be deployed as:

$$\begin{pmatrix} \overline{C}_{nm} \\ \overline{S}_{nm} \end{pmatrix} = \left[\frac{1}{4n+2} \frac{(n+m)!}{(n-m)!} \right]^{1/2} \begin{pmatrix} C_{nm} \\ S_{nm} \end{pmatrix}, \tag{3}$$

where $\overline{C}_{nm}$ and $\overline{S}_{nm}$ correspond to the full normalization of the coefficients C_{nm} and S_{nm}.

2.2 Fully normalized Legendre Polynomial recursion

When one considers fully normalized harmonic coefficients, the associated Legendre polynomials P_{nm} should also be fully normalized, so that Equation 1 modeling the geopotential is compatible. Thus it takes the following form:

$$V = \frac{GM}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \left(\frac{a_e}{r} \right)^n \left[\overline{C}_{nm} \cos m\lambda + \overline{S}_{nm} \sin m\lambda \right] \overline{P}_{nm} (\sin \psi) \tag{4}$$

For the computation of fully normalized Legendre polynomials it was used the forward columns method described in Holmes and Featherstone (2002). This recursion is the most used for computing $\overline{P}_{nm}(\theta)$ and is described below:

$$\overline{P}_{nm}(\theta) = a_{nm}\, t\, \overline{P}_{n-1,m}(\theta) - b_{nm}\, \overline{P}_{n-2,m}(\theta), \qquad \forall n > m \tag{5}$$

where

$$t = \cos\theta \tag{6}$$

$$a_{nm} = \sqrt{\frac{(2n-1)(2n+1)}{(n-m)(n+m)}} \tag{7}$$

$$b_{nm} = \sqrt{\frac{(2n+1)(n+m-1)(n-m-1)}{(n-m)(n+m)(2n-3)}} \tag{8}$$

For the sake of notational simplicity the dependency of *P* on $\cos\theta$ is represented by $P(\theta)$, where symbol θ is the colatitude, i.e., $\theta = 90^o - \psi$. When $n = m$, the recursion described by

Equation 9 is used. The initial values are $\overline{P}_{0,0}(\theta)=1$, $\overline{P}_{1,0}(\theta)=\sqrt{3}\,t$ and $\overline{P}_{1,1}(\theta)=\sqrt{3}\,u$**,** where $u=\sin\theta$**.**

$$\overline{P}_{mm}(\theta)=u\sqrt{\frac{2m+1}{2m}}\overline{P}_{m-1,m-1}(\theta),\qquad \forall m>1\,\textbf{.} \tag{9}$$

Computationally this transformation provides better numerical accuracy since, after normalization, $\overline{C}$, $\overline{S}$ and $\overline{P}$ have better numerically conditioned values to perform calculations and do not introduce factorials. Such calculations can produce values that makes computational representation troublesome (large positive or negative exponents), prone to overflows/underflows, and therefore susceptible to numerical errors.

2.3 Trigonometric recursion of cos(*mλ*) and sin(*mλ*)

One also applies recursion formulas for trigonometric functions, so as to avoid the explicit evaluation of cos(*mλ*) and sin(*mλ*), such as:

$$\cos(m\lambda)=\cos(m-1)\lambda\cos\lambda-\sin(m-1)\lambda\sin\lambda\,, \tag{10}$$

$$\sin(m\lambda)=\sin(m-1)\lambda\cos\lambda+\cos(m-1)\lambda\sin\lambda\,. \tag{11}$$

It only needs $\sin\lambda$ and $\cos\lambda$ to start the recursion for $m=1,\ldots,M$.

2.4 Forward column recursion implementation

To implement the algorithm, one prefers to reverse the order of computation of the summation, where the outer loop in *m* is first computed. One will follow closely the approach described by Holmes and Featherstone (2002). Let us rewrite the geopotential summation as:

$$V=\frac{GM}{r}+\frac{GM}{r}\sum_{m=0}^{M}\left[\cos(m\lambda)\sum_{n=\mu}^{M}\left(\frac{a}{r}\right)^{n}\overline{C}_{nm}\overline{P}_{nm}(\theta)+\sin(m\lambda)\sum_{n=\mu}^{M}\left(\frac{a}{r}\right)^{n}\overline{S}_{nm}\overline{P}_{nm}(\theta)\right]. \tag{12}$$

Next define the inner summation components by:

$$X_{mC}\equiv\sum_{n=\mu}^{M}\left(\frac{a}{r}\right)^{n}\overline{C}_{nm}\overline{P}_{nm}(\theta)\,,\qquad X_{mS}\equiv\sum_{n=\mu}^{M}\left(\frac{a}{r}\right)^{n}\overline{S}_{nm}\overline{P}_{nm}(\theta) \tag{13}$$

$$\Omega_m\equiv\cos(m\lambda)X_{mC}+\sin(m\lambda)X_{mS}\quad , \tag{14}$$

where μ is an integer that depends on *m*. Therefore

$$V=\frac{GM}{r}+\frac{GM}{r}\sum_{m=0}^{M}\Omega_m\,\textbf{.} \tag{15}$$

The outer loop starts increasing $m=0,\ldots,M$. For each *m*, all the dependencies of $n=m,\ldots,M$ are evaluated. For $m=n$ the recursion uses Equation 9 to start the polynomial recursion. Then recursion of Equation 5 is used up to $n=M$, and the inner sums are evaluated by Equations 13 and 14. Figure 1 shows the mechanization of the algorithm.

An additional computational saving is to pre-compute and store $\overline{P}_{mm}$ and q_m for $m=0,\ldots,M$ using both Equations 9 and 16:

$$q\equiv\left(\frac{a}{r}\right)\,,\qquad q_m=q\,q_{m-1} \tag{16}$$

starting with $q_o=1$. In the subroutine (or procedure) it is performed before starting the main *m* and *n* loops.

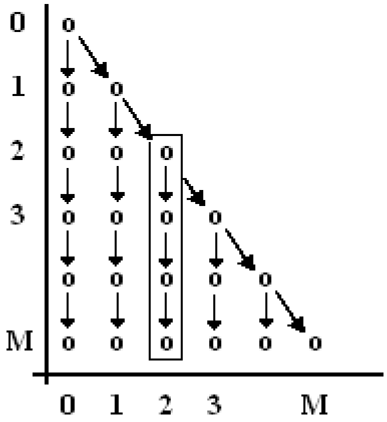

Figure. 1. Forward recursion schematic (source: Holmes and Featherstone, 2002)

3. Geopotential gradient and acceleration computation

Here one describes the mathematical details of the recursions implemented for computing the geopotential gradient. For that one needs the recursion of the first derivative $\overline{P}^1_{nm}$ of the Legendre polynomial:

$$u\,\overline{P}^1_{nm}(\theta) = -n\,t\,\overline{P}_{nm}(\theta) + f_{nm}\,\overline{P}_{n-1,m}(\theta), \quad \forall n \ge m, \tag{17}$$

with f_{nm} given by:

$$f_{nm} = \sqrt{\frac{(n^2 - m^2)(2n+1)}{2n-1}} = \frac{2n+1}{a_{nm}}. \tag{18}$$

One first evaluates the gradient of the potential in terms of spherical coordinates λ, θ, r. For λ it may be computed by:

$$V_\lambda \equiv \frac{\partial V}{\partial \lambda} = -\frac{GM}{r} \sum_{m=0}^{M} m\left[\cos(m\lambda) X_{mC} + \sin(m\lambda) X_{mS}\right] \tag{19}$$

with X_{mC}, X_{mS} given by Equations 13-14. For θ one needs the first derivative of the Legendre polynomial, Equation 17:

$$\begin{aligned} X^\theta_{mC} &\equiv \sum_{n=m}^{M} \left(\frac{a}{r}\right)^n \overline{C}_{nm} \overline{P}^1_{nm}(\theta) \\ X^\theta_{mS} &\equiv \sum_{n=m}^{M} \left(\frac{a}{r}\right)^n \overline{S}_{nm} \overline{P}^1_{nm}(\theta) \end{aligned} , \tag{20}$$

$$V_\theta \equiv \frac{\partial V}{\partial \theta} = \frac{GM}{r} \sum_{m=0}^{M} \left[\cos(m\lambda) X^\theta_{mC} + \sin(m\lambda) X^\theta_{mS}\right]. \tag{21}$$

For r one also needs the first derivative of the Legendre polynomial, Equation 17:

$$\begin{aligned} X^r_{mC} &\equiv \sum_{n=m}^{M} \left(\frac{a}{r}\right)^n (n+1) \overline{C}_{nm} \overline{P}^1_{nm}(\theta) \\ X^r_{mS} &\equiv \sum_{n=m}^{M} \left(\frac{a}{r}\right)^n (n+1) \overline{S}_{nm} \overline{P}^1_{nm}(\theta) \end{aligned} \tag{22}$$

$$V_r \equiv \frac{\partial V}{\partial r} = -\frac{GM}{r^2} \sum_{m=0}^{M} \left[\cos(m\lambda) X^r_{mC} + \sin(m\lambda) X^r_{mS}\right] \tag{23}$$

Finally, the transformation from spherical to cartesian coordinates takes into account the partial derivatives relating them and is achieved by:

$$\begin{aligned}
\ddot{x} &= u\cos\lambda\, V_r - \frac{t}{r}\cos\lambda\, V_\theta - \frac{\sin\lambda}{u\,r}V_\lambda \\
\ddot{y} &= u\sin\lambda\, V_r - \frac{t}{r}\sin\lambda\, V_\theta + \frac{\cos\lambda}{u\,r}V_\lambda \\
\ddot{z} &= t\,V_r + \frac{u}{r}V_\theta
\end{aligned} \tag{24}$$

4. Results

Herein one shows some outcomes from the source code implemented according to the formulations described. The full EGM 2008 geopotential model (Pavlis et al., 2008) was used to generate the results.

4.1 Geopotential results

To show the numerical behavior at the singularities near the poles the geopotential was computed to order/degree *n* = *m* = 90. It is claimed that EGM 2008 captures most of the Grace geopotential model, considered the best one generated from satellite data, when it is truncated at least to 90 order and degree.

Table 1 shows the perturbed geopotential, i.e. the geopotential minus the *GM/r* term (around 62494813.9631322 m^2/s^2) which is the dominant value, near the poles (north and south). This way it is possible to analyze the variation at pole proximity in terms of changes of significant figures. Three longitudes were also taken into account (0°, 120°, 240°) although longitude values are nearly irrelevant close to the poles (undefined longitude). It is seen that no major numerical problem arose up to 0.000001° of closeness to either north or south pole, and the 5 to 6-th significant figure is affected, meaning that the geopotential value is affected in the digit right after comma. Logically the dependency with longitude becomes negligible when approaching the poles.

Figure 2 shows the convergence of the perturbed geopotential value (geopotential value minus *GM/r* term of each case) with increasing *n*, *m* at mid latitudes (±45°), for altitude 0m. The longitude was arbitrarily fixed in -40°. The upper graphic shows the geopotential values to increasing *n*, *m* at 45° of north latitude. The lower graphic shows the same but to -45° of south latitude. It is seen that depending on the required accuracy it is neither wise nor beneficial to compute the geopotential using the full EGM 2008 up to 2159 order and 2190 degree. The price to pay is that the CPU time increases unwieldy without gaining much more accuracy, making cost (CPU) benefit (accuracy) ratio unfavorable.

Table 1 – Perturbed geopotential value for *n*=*m*=90 and altitude 0m, near the poles

Longitude (°)	θ = 0.01°	θ = 0.0001°	θ = 0.000001°	θ = 179.99°	θ = 179.9999°	θ= 179.999999°
0	-67364.80815	-67364.98690	-67364.98866	-67785.42128	-67785.45193	-67785.45221
120	-67365.15018	-67364.99032	-67364.98869	-67785.44560	-67785.45217	-67785.45221
240	-67364.99839	-67364.98881	-67364.98868	-67785.47909	-67785.45252	-67785.45221

4.2 Short orbit integration

For depicting the performance of the codes for geopotential spherical derivatives and acceleration, short orbit integration was exercised. A fixed-step Runge-Kutta 4 (RK4) was used as baseline numerical integrator. It evaluates ODEs (Ordinary Differential Equations) by calling the derivatives 4 times per integration step. A 10 seconds step-size was fixed and 100 minutes of integration was carried out. This is typically one period of orbit of LEO (Low Earth Orbit) satellites (around 700-800km altitude). Explicitly the following set of 6 EDOs was solved:

$$\begin{aligned}\dot{\mathbf{r}} &= \mathbf{v}\\ \dot{\mathbf{v}} &= \mathbf{a}_g\,,\end{aligned} \qquad (26)$$

where $\mathbf{a}_g$ is the acceleration due to the geopotential in the inertial frame.

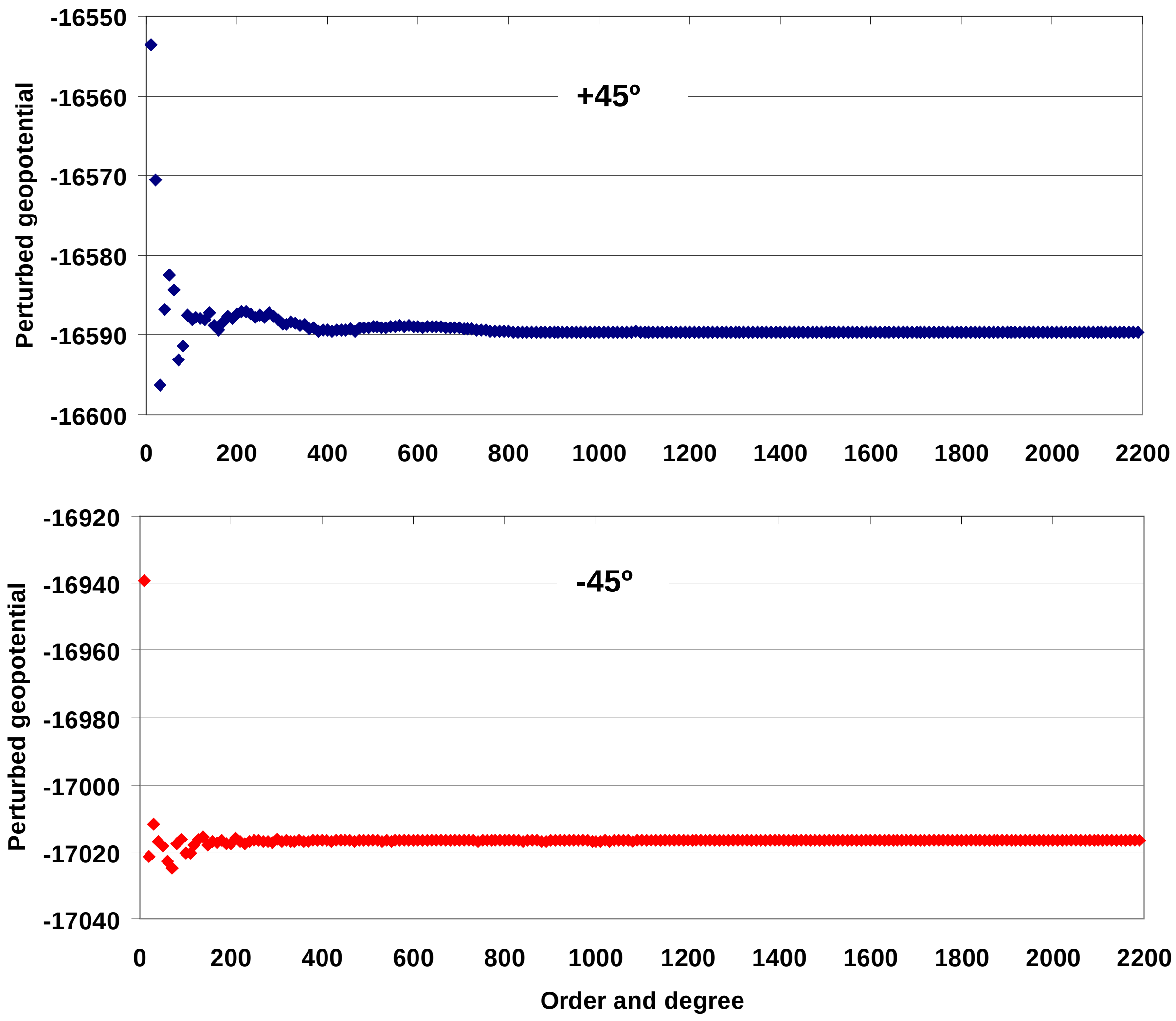

Figure 2. Perturbed geopotential for mid latitudes (±45°) vs increasing *n,m*

That means that the right side of Equation 26 was in total evaluated 4×600 times (4 times per step with 600 steps), using the RK4 integrator. Because the geopotential acceleration must be evaluated in the body fixed ITRF (International Terrestrial Reference Frame) system, a simple conversion of inertial position to ITRF position was used, accounting only for the sidereal rotation around the *z*-axis. The initial sidereal angle was computed at the beginning and then, inside the derivative function of EDOs, it was updated using the sidereal rotation rate. The ITRF position is then used to evaluate the geopotential acceleration through Equation 24. Finally the ITRF acceleration is converted back to inertial using the same sidereal rotation. For computational evaluation such rough approach is considered sufficient for the sake of comparison between the orbit integration discrepancies to several truncation levels of the geopotential recursions. Two types of LEO orbits were also considered as representatives of most of the LEO orbit integration applications:

- Nearly polar sunsynchronous orbit of CBERS (China Brazil Earth Resources Satellite), 780km altitude
- Low inclination (25º) circular orbit of SCD (Brazilian Data Collection Satellite), 750km altitude

A desktop PC with Intel Core2 QUAD CPU Q6600 at 2.4GHz and 3.25GB of RAM memory was used, under Microsoft Windows XP Operating System. The Fortran code was compiled with Compaq Visual Fortran Professional Edition 6, without any optimization option.

Figure 3 shows the CPU times increase with increasing levels of order and degree truncation. The ordinate is in log scale so that it is clearly seen the exponential growth of the CPU times with order/degree. For one orbit integration of both types of orbits it took around 1000s (16.7 minutes) of CPU time when using the full model (order 2159, degree 2190).

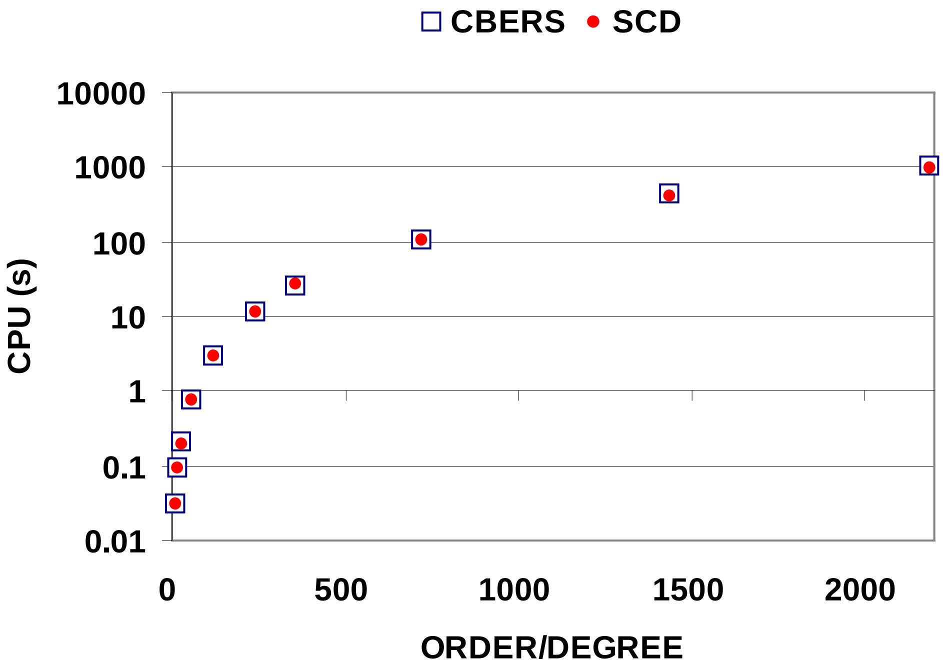

Figure 3. CPU times (s) for integration of one orbit of CBERS and SCD-like satellites versus order and degree

For orbit accuracy analysis purposes, the full model (order 2159 degree 2190) was used as reference orbit. Figure 4 shows the discrepancies along the time for different levels of truncation compared to the full model, in terms of radial, normal, and along-track (RNT) components, for the near polar orbit (CBERS). The left side plots shows up to 25 m difference in along-track direction for truncation on *n*=*m*=10. The right side shows differences for truncation from *n*=*m*=60 to 240. For CBERS orbit types (near polar circular) it is seen that to keep millimeter level accuracy the *n*, *m* truncation should be somewhere between 120 and 240.

Figure 5 shows the discrepancies along one orbit for several truncations compared to the full model, in terms of RNT components, for the near equatorial orbit (SCD). The left side plots shows up to 21 m difference in along-track direction for truncation on n=m=10, Of course the discrepancies decrease with increase of order and degree. The right side shows differences for truncation from n=m=60 to 240. For SCD orbit types (nearly equatorial circular) the conclusion is similar, that is, to millimeter level accuracy the n, m truncation should be somewhere between 120 and 240.

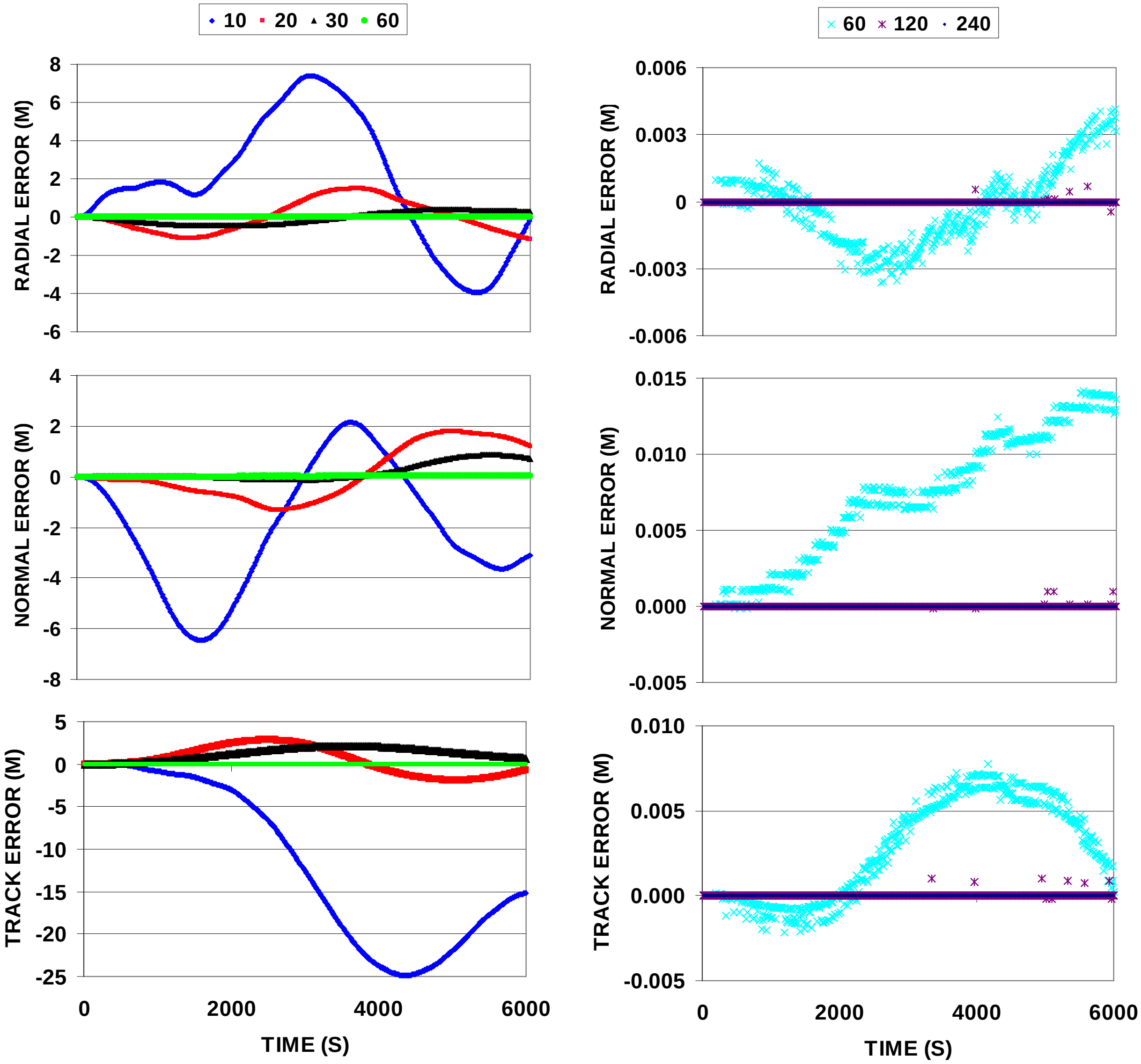

Figure 4. Discrepancies in RNT components for one CBERS satellite orbit with different levels of order and degree truncation

A discrepancy with periodic (sinusoidal shape) trend is observed in CBERS orbit integration on the along-track component. However this fact hardly can be considered conclusive because the same shape does not manifest in the SCD orbit. Probably longer integration times should be carefully analyzed in terms of precise orbit prediction. For now it is considered out of scope of this work, and it is postponed to follow-up works.

5. Final considerations

This report shows the mathematical details of the implemented geopotential related computation. Considerable effort was put in optimizing the code so as to save computational costs. Some extra CPU saving is possible if the user, knowing the intended application, pre-stores some arrays. Also some memory saving is possible if the user stores the matrix of geopotential harmonic coefficients in vectorized arrays (single dimension). This is not implemented in the Fortran version as it impairs the understanding and cleanliness of code, but is as easily implemented by experienced users. Some care was also taken with respect to the mix between integer and double precision numbers, as well as arithmetic operation sequences to avoid premature over- under-flows. Some illustrative results showed the power and stability of the codes implemented. Sources codes as well as test codes are freely available in both Fortran-77 and ANSI-C programming languages (Kuga and Carrara, 2012). It should be remarked that the Fortran source code is the baseline, and the C-code is a transcription of Fortran into ANSI-C.

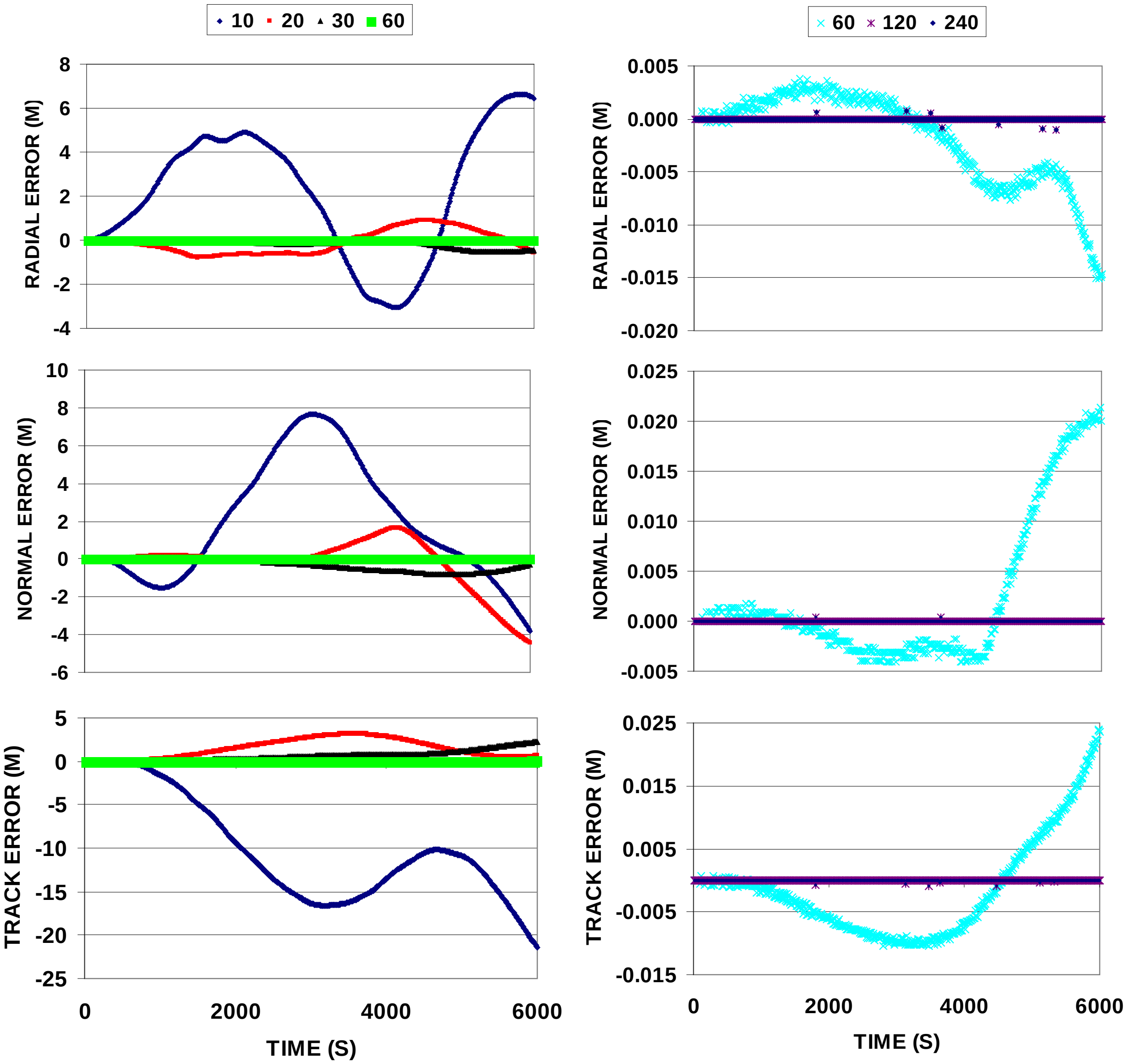

Figure 5. Discrepancies in RNT components for one SCD satellite orbit with different levels of order and degree truncation

References

Cunningham, L. E. On the computation of the spherical harmonic terms during the numerical integration of the orbital motion of an artificial satellite. **Celestial Mechanics**, V. 2(2), July 1970, pp. 207-216.

Fantino, E.; Casotto, S. Methods of harmonic synthesis for global geopotential models and their first-, second- and third-order gradients. **Journal of Geodesy**, 83, p. 595-619, 2009.

Holmes, S.A.; Featherstone, W.E. A unified approach to the Clenshaw summation and the recursive computation of very degree and order normalised associated Legendre functions. **Journal of Geodesy**, 76, p. 279-299, 2002.

Kuga, H. K.; Carrara, V. Fortran- and C-codes for higher order and degree geopotential and derivatives computation. http://www2.dem.inpe.br/hkk/software/high_geopot.htm, last access Nov. 2012.

Lemoine, F. G. ; Kenyon, S. C.; Factor, J. K.; Trimmer, R.G.; Pavlis, N. K.; Chinn, D. S.; Cox, C. M.; Klosko, S. M.; Luthcke, S. B.; Torrence, M. H.; Wang, Y. M.; Williamson, R. G.; Pavlis, E. C.; Rapp, R. H.; Olson, T. R. **The Development of the Joint NASA GSFC and NIMA Geopotential Model EGM96**, NASA Goddard Space Flight Center, Greenbelt, Maryland, July 1998. NASA/TP-1998-206861.

Lerch, J. F.; Klosko, S. M.; Laubscher, R. E.; Wagner, C. A. Gravity model improvement using Geos 3 (GEM 9 and 10). **Journal of Geophysical Research**, V. 84(B8), July 1979, pp. 3897-3916.

Lundberg, J. B.; Schutz, B. E. Recursion formulas of Legendre functions for use with nonsingular geopotential models. **Journal of Guidance and Control**, V. 11(1), Jan-Feb., 1988, p.31-38.

Pavlis, N.K.; Holmes, S.A.; Kenyon, S.C.; Factor, J.K. **An Earth Gravitational Model to Degree 2160: EGM2008.** Presented at the 2008 General Assembly of the European Geosciences Union, Vienna, Austria, April 13-18, 2008.

Pines, S. Uniform representation of the gravitational potential and its derivatives. **AIAA Journal**, V. 11(11), 1973, pp. 1508-1511.