Journal of Geodetic Science

• 3(3) • 2013 • 219-223 DOI: 10.2478/jogs-2013-0025 •

Multi-processing least squares collocation: Applications to gravity field analysis

Research Article

E. Kaas[1], B. Sørensen[1], C. C. Tscherning[1] *, M. Veicherts[2]

1 *Niels Bohr Institute, University of Copenhagen, Copenhagen, Denmark.*
2 *Geological Survey of Denmark and Greenland, Copenhagen, Denmark.*

Abstract:

Least Squares Collocation (LSC) is used for the modeling of the gravity field, including prediction and error estimation of various quantities. The method requires that as many unknowns as number of data and parameters are solved for. Cholesky reduction must be used in a non-standard form due to missing positive-definiteness of the equation system. Furthermore the error estimation produces a rectangular or triangular matrix which must be Cholesky reduced in the non-standard manner.

LSC have the possibility to add new sets of data without processing previously reduced parts of the equation system. Due to these factors standard Cholesky reduction programs using multi-processing cannot easily be applied. We has therefore implemented Fortran Open Multi-Processing (OpenMP) to the non-standard Cholesky reduction. In the computation of matrix elements (covariances) as well as the evaluation spherical harmonic series used in the remove/restore setting we also take advantage of multi-processing.

We describe the implementation using quadratic blocks, which aids in reducing the data transport overhead. Timing results for different block sizes and number of equations are presented. OpenMP scales favorably so that e.g. the prediction and error estimation of grids from GOCE TRF vertical gradient data and ground gravity data can be done in the less than two hours for a 25° by 25° area with data selected close to 0.125° nodes.

Keywords:

Gravity • least-squares collocation • multiprocessing

Received 25-07-2013; accepted 02-08-2013

1. Introduction.

Least-Squares Collocation (LSC), (Moritz, 1980, Torge and Müller, 2012, Sansó and Sideris, 2013) has been and is used for gravity field modeling. This includes the determination of parameters and the estimation of errors. The quantity to be modeled is the so-called anomalous potential T, which is a harmonic function outside the masses.

The method requires that a number of equations equal to the number of data and of contingent parameters are solved for. This has forced the development of "fast" methods (e.g. Sansó and Tscherning, 2002) which require that data are interpolated to the nodes of a grid. However, the use of multiprocessing has changed the picture, so that it has become possible to use LSC using large (200000+) sets of data.

The use of the method requires that a function space with a reproducing kernel is selected, so that the kernel approximates an empirical covariance function of T. This function is generally determined from gravity anomalies and approximated with a corresponding reproducing kernel. This assures the positive-definiteness of a part of the LSC equation system (see Appendix A). We will not discuss this here, but explain how multiprocessing has permitted us to overcome the limits posed by the number of equations to be solved.

However, multiprocessing may also aid in other parts of the LSC process as discussed in the following.

The results presented are obtained using a Dual Processor Intel(R) Xeon(R) CPU at 2.40GHz with a total of 24 threads.

*E-mail: cct@gfy.ku.dk, Tel: +4535320582

2. Remove-restore procedure

In order to statistically homogenize T, to de-correlate data and to permit the use of spherical approximation in regional applications, the effect of a global Earth gravity model (EGM) may be subtracted and subsequently added. The expression for such a model can be written as

$$T = (\varphi, \lambda, r) = \frac{GM}{r} \sum_{n=0}^{N} (a/r)^n \sum_{m=-n} nP_{nm}(\sin\overline{\varphi}) C_{nm} \times \begin{pmatrix} \cos(m\lambda), & m \geq 0 \\ \sin(m\lambda), & m < 0 \end{pmatrix}$$
$$= \frac{GM}{r} \sum_{m=0}^{N} \left(\sum_{n=m}^{N} \left(\frac{a}{r} \right)^n P_{nm}(\sin\overline{\varphi}) C_{nm} \right) \times \begin{pmatrix} \cos(m\lambda), & m \geq 0 \\ \sin(m\lambda), & m < 0 \end{pmatrix}. \quad (1)$$

Then the "inner" bracket can be evaluated separately for each n, and can hence be evaluated by several processors simultaneously. The outer bracket may be evaluated by FFT. For the summation the Clenshaw method is used so that neither the associated Legendre functions (P_{nm}) (or their derivatives) need to be evaluated explicitly, (Tscherning and Poder, 1982). These features have been implemented in the Gravsoft (Forsberg and Tscherning, 2008) program geocol several years ago.

In Equation (1) $\overline{\varphi}$ is the geocentric latitude, r is the distance from the origin, λ is the longitude, a a constant close to the semi-major axis of the Earth, GM the product of the gravitational constant and the mass of the Earth and C_{ij} the spherical harmonic coefficients. The upper limit N may be many thousands. In Table 1 are shown the results of tests using different number of processors and different values of the maximal summation order, N. The benefit of using multiple processors is clearly seen.

3. Covariance computation

A major part of the LSC computations is the evaluation of the covariance-vectors, which both are present in Eq. (A2) - (A5). It has been rather complicated to implement multiprocessing, since the type of the functionals L_i may change as a function of the index i. The size (N) of the vectors may require that the variables used to compute the covariances such as quarternions (defining the attitude of the measuring instrument) may require paging to disk. Furthermore the vectors have to be transferred into the so-called chunks, see Fig. 1, of the (upper triangular part of) the N × N matrix $\overline{C}$ in Eq. (A2) and the matrix of R*(N+M) quantities used when computing R error estimates. M is the number of parameters to be solved for. In Table 2 is shown timing results for different values of N, different number of processors and different block-size k. We see the expected scaling due to the use of more processors and also the effect of the overhead used to transport the coefficients to back-ground memory (disk). The covariances computed are covariances of GOCE vertical gravity gradients in the terrestrial reference frame (TRF).

Table 1. Summation times for EGM2008 (Pavlis et al. 2012) for 26 points (at different latitude) including read overhead of coefficients.

N – max. degree	Number of processors	Time (s)
400	1	0.8
	4	0.39
	24	0.53
1000	1	54.0
	4	24.6
	24	5.8
2190	1	293.0
	4	116.8
	24	26.1

Table 2. Computation of $N*(N+1)/2$ covariances. Time in seconds as a function of number of processors and of block-size $k*k$. For a small block-size most of the processor time is spent on the transfer to and from disk.

N	37971	22464	22464	22464
Processors	22	22	4	4
Blocksize, k				
	s	s	s	s
05		8486	32044	31341
10	3737	1101	3703	4784
15	4623	1268	3159	4430
20	3547	895	2847	3851
25	3600	1047	2974	3694
30	3621	1031	3101	3804

4. Reduction (factorization), solution of equations and error-estimation

The LSC method has been applied since the 1960'ties on computers with very small central processing units (CPU). Cholesky-reduction was used to reduce (not invert!) and solve the systems of equations. However often it was possible to store elements of equation systems on external media like a disk or drum. A simple implementation is described in (Poder and Tscherning, 1973) and used in the LSC implementation (Tscherning, 1974). Here the equations were divided in rectangular blocks, filling in as many columns as possible, and storing the blocks on an external media. This implementation is favorable in case sparse matrices have to be reduced. In LSC the matrices are in general full, see however (Moreau

et al., 1999) where the use of finite covariances is discussed.

However this implementation is not suitable for multiprocessing, where each row may be processed individually using Cholesky-reduction, see Appendix B. We therefore implemented a division of the matrices in square blocks called "chunks" like shown in Fig. 1. Cholesky reduction is used in Eq. (A2) with details in Appendix B. This is followed by what is known as back-substitution and multiplication with the Cholesky-reduced in the other two equations, for details see (Tscherning, 1980). However, the concatenated set of matrices (Eq. B7) is not positive definite, due to the "minus" occurring in Eq. (A2). This is however by-passed by using positive accumulation instead of negative accumulation in the Cholesky-algorithm, see Eq. (B4) and (B5). But for the use of multiprocessing this causes difficulties, so that it is only used to reduce the chunks which do not contain a part of the A matrix, which holds contingent contributions from parameters. This is also the primary reason for that standard LAPAC (http://www.netlib.org/lapack/), or similar equation solvers can't be used. Another reason is that we want to be able to take advantage of the so-called permanency property (Freden, 1982), where new columns can be added to an already reduced system of equations, and the reduction then continued from the first chunk, which contains unreduced elements.

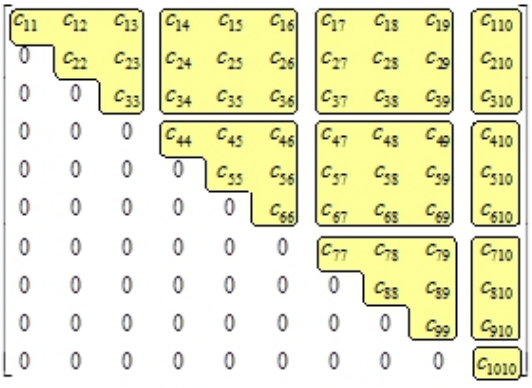

Figure 1. Chunk-wise division of the upper-triangular part of the $\overline{C}$ matrix concatenated contingently with the parameter matrix A. Each element c_{ij} is a sub-matrix of size $k*k$. In the figure a chunk consist of $3*3$ blocks.

In Table 3 are shown timing results for the reduction of covariance matrices $\overline{C}$ cf. Eq. (A2).

The dependency of the number of processors and the chunk row number is clearly seen in Table 3. The time used decreases when some processors are not in use, i.e. after the chunk row modulo number of processors has become zero. However the time for transfer of a chunk to and from the CPU clearly also plays a role. The pattern can easily be reproduced in a simulation, which a user should perform before embarking into a major computation with many unknowns.

We see clearly that the smaller the chunks the faster the processing. There is however some system limits which must be taken into account. Each chunk will be a separate file, and there will generally be an upper limit for the number of files, which may be open at the same time.

Let N be the number of rows, and M the chunk-size, e.g. 400*400 as used in Fig. 2. For N = 40000 we will have NC=N/M, or 100 rows of chunks, i.e. (NC*(NC+1))/2 files in total. As may be easily shown, the maximal number of open files is reached for chunk row number NC/2 and is equal to $NC^2/2$, i.e. in this case 5000 open files.

Table 3. Time in seconds as a function of number of processors, number of data (N) and block-size using OpenMP., http://gcc.gnu.org/onlinedocs/gfortran/OpenMP.html

N	37971	22464	22464	22464
Processors	22	22	4	4
Blocksize, k	s	s	s	s
05		369	5157	1698
10	440	136	421	639
15	966	208	419	612
20	1013	238	391	591
25	1307	346	538	755
30	1542	411	668	1060

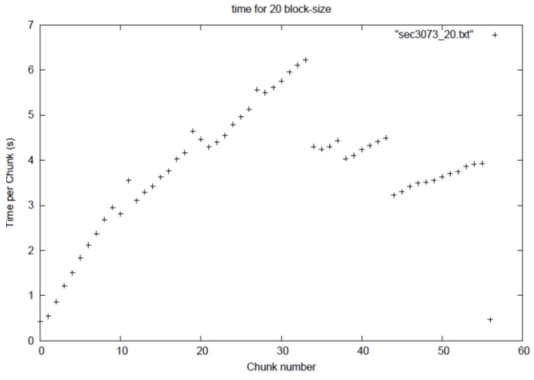

Figure 2. Time for Cholesky reduction using 20 column block-size and 20 columns of blocks with 22 processors. Total number of observations 22464 with $22464/(20*20)+1$ chunks.

5. Conclusion and outlook

In the textbook (Torge and Müller, 2012) it is stated that the use of LSC is restricted due to the effort needed to establish and solve the large number of equations which arise when using many data. In this paper we have shown that this restriction has been overcome. The use of multiprocessing has enabled the use of LSC for large sets of observations. The benefits are clearly demonstrated when (1) the contribution of a spherical harmonic model is removed and restored, (2) when establishing a column of covari-

ances and (3) when factorizing the equation system. The use of Chunks is primarily due to a planned the implementation of MPI (http://www.open-mpi.org/), using multiple computers in all 3 situations. But when this is done the plan is to use up to 1 million data simultaneously.

Acknowledgement

A part of this study has been financed by the ESA GOCE mission HPF project. Thanks to Jacob Trier Frederiksen, Niels Bohr Institute, for substantial assistance with the implementation of FORTRAN95 facilities.

References

Arabelos D. and C.C. Tscherning, 2008, Error-covariances of the estimates of spherical harmonic coefficients computed by LSC, using second-order radial derivative functionals associated with realistic GOCE orbits. J.Geod., DOI: 10.1007/s00190-008-0250-9.

Forsberg R. and C.C. Tscherning, 2008, An overview manual for the GRAVSOFT Geodetic Gravity Field Modelling Programs. 2.edition. Contract report for JUPEM. Available as: http://cct.gfy.ku.dk/publ_cct/cct1936.pdf

Freeden W., 1982, On the Permanence Property in Spherical Spline Interpolation. Reports of the Department of Geodetic Science and Surveying, No. 341, The Ohio State University, Columbus, Ohio.

Heiskanen W.A. and H. Moritz, 1967, Physical Geodesy. W.H. Freeman & Co, San Francisco, 1967.

Moritz H., 1980, Advanced Physical Geodesy. H.Wichmann Verlag, Karlsruhe.

Pavlis N.K., Holmes S.A., Kenyon S.C. and Factor J.K., 2012, The development and evaluation of the Earth Gravitational Model 2008 (EGM2008) Journal of Geophysical Research: Solid Earth (1978-2012) Volume 117, Issue B4, April.

Poder K. and C.C. Tscherning, 1973, Cholesky's Method on a Computer. The Danish Geodetic Institute Internal Report No. 8. (Available as http://cct.gfy.ku.dk/publ_cct/cct42.pdf)

Sanso F. and M.G. Sideris (Eds), 2013, Geoid Determination. Lecture Notes in Earth System Science 110, DOI: 10.1007/978-3-540-74700-0_7. Springer-Verlag, Berlin-Heidelberg.

Moreaux G., C.C. Tscherning and F. Sansó, 1999: Approximation of Harmonic Covariance Functions on the sphere by non Harmonic Locally Supported functions. Journal of Geodesy, Vol. 73, pp. 555 – 567.

Sansó F. and C.C. Tscherning, 2003, Fast Spherical Collocation - Theory and Examples. J. of Geodesy, Vol. 77, pp. 101-112, DOI 10.1007/s00190-002-0310-5.

Torge W. and J. Müller, 2012, Geodesy. 4th ed., de Gruyter.

Tscherning C.C.,1974, A FORTRAN IV Program for the Determination of the Anomalous Potential Using Stepwise Least Squares Collocation. Reports of the Department of Geodetic Science No. 212, The Ohio State University, Columbus, Ohio.

Tscherning C.C, 1978, A Users Guide to Geopotential Approximation by Stepwise Collocation on the RC 4000-Computer. Geodaetisk Institut Meddelelse No. 53.

Tscherning C.C. and K. Poder, 1982 Some Geodetic applications of Clenshaw Summation. Bolletino di Geodesia e Scienze Affini, Vol. XLI, no. 4,pp. 349-375.

Tscherning C.C. and M. Veicherts, 2008, Accelerating generalized Cholesky decomposition using multiple processors. Presentation, EGU 2008. Available as http://cct.gfy.ku.dk/publ_cct/accelerating08.ppt

Appendix A: BASIC EQUATIONS (CF. HEISKANEN AND MORITZ, 1967, MORITZ 1980 OR SANSŇ AND SIDERIS, 2013).

The gravity potential W, is the sum of the potential V due to the attraction of the masses and the centrifugal potential. The anomalous potential is the difference between W and the normal potential U. In space quantities related to V are measured, while at the surface of the Earth W is the important quantity. T is however the same everywhere, because the centrifugal part is eliminated.
T may be represented by a series in spherical harmonics, cf. Eq. (1).
The basic observation equation for LSC is

$$y_i = L_i(T_{LSC}) + e_i + A_i^T X, \qquad \text{(A1)}$$

where y_i is a vector of n observations, L_i is a vector of functionals associating T with the observarion, e_i is a vector of errors, X is a M-vector of parameters and A_i^T a N*M matrix relating the N observations and M parameters and T_{LSC} is the approximation to T which will be determined.
Here the contribution from a contingent datum-transformation and an Earth Gravity Model must have been subtracted.
The estimate of T_{LSC} is obtained by

$$\tilde{T}_{LSC}(P) = \{C_{Pi}\}^T \overline{C}^{-1} \left\{y - A^T X)\right\}, \qquad \text{(A2)}$$

where $\overline{C} = \{C_{ij} + \sigma_{ij}\}$, and σ_{ij} is the variance-covariances of the errors.
The estimate of the (M) parameters are obtained by

$$\tilde{X} = (A^T \overline{C}^{-1} A + W)^{-1} (A^T \overline{C}^{-1} y). \quad \text{(A3)}$$

The error-estimates and error-covariances, ec_{kl} are found with:

$$H_k = \{COV(L_k, L_i)\}^T \bar{C}^{-1}, \text{ N} \times \text{M matrix}$$

$$m_x^2 = (A^T \overline{C}^{-1} A + W)^{-1} \quad \text{(A4)}$$

$$\{ec_{ki}\} = \{\sigma_{ki}\} - H\{\text{cov}(Lj, L_i)\} + H_k A M_x (H_i A)^T. \quad \text{(A5)}$$

Appendix B: CHOLESKY FACTORISATION, GENERALIZED

Cholesky factorization was originally a procedure to "take the square-root" of a symmetric positive-definite matrix. However, the equation system described in Appendix A is a combination of a positive definite matrix and a negative definite one due to the minus sign in Eq. (A2). The combined system is illustrated in Fig. 1:

$$\begin{Bmatrix} \overline{C} & A \\ A^T & W \end{Bmatrix} \begin{Bmatrix} x \\ X \end{Bmatrix} = \begin{Bmatrix} y \\ y' \end{Bmatrix}$$

where $\overline{C}$ has the dimension N × N, A the dimension N × M, x and y are N-vectors and X and y' are M-vectors.
The Cholesky factorization of $\overline{C}$ is derived by introduction of a lower-triangular matrix L, so that

$$\overline{C}x = LL^T x = y, \quad \text{(B1)}$$

$$L^{-1}(LL^T)x = L^{-1}y, \quad \text{(B2)}$$

$$L^T x = L^{-1} y. \quad \text{(B3)}$$

The final equation is an upper-triangular system, with the bottom element given rise to one equation with one unknown, which is substituted and the full x vector is found. L is determined by

$$L_{i,m+1} = \left(\overline{C}_{i,m+1} - \sum_{k=1}^{m} L_{ik} L_{m+1,k} \right) / L_{ii} \quad \text{(B4)}$$

$$L_{m+1,m+1} = \sqrt{\overline{C}_{m+1,m+1} - \sum_{k=1}^{m} L_{k,m+1}{}^2}. \quad \text{(B5)}$$

If the number of rows and columns in a chunk is equal to i, and $j = m/i + 1$, we may rewrite Eq. (B5)

$$L_{i,m+1} \left(\overline{C}_{i,m+1} - \sum_{p=1}^{j} \left(\sum_{q=i*(p-1)}^{p*i} (L_{iq} L_{m+1,q}) \right) \right) / L_{ii}. \quad \text{(B6)}$$

For each chunk column, the inner bracket may be evaluated independently of other chunks in the same row of chunks. This is what makes multiprocessing feasible. But it is also seen that for each chunk to be reduced, the chunks in the same column of chunks but with a smaller subscript (p in Eq. (B6)) is used in the product-sum calculation. This is what causes the reduction times in Fig. 2 to increase as a function of chunk row number until contingently all processors may be used in the reduction.

We can see that the primary part of the computations is used for the evaluation of Eq. (B4) - (B6) and contingently for the transport of a chunk of the original equation system to the CPU and the back after reduction.

Let now L be the generalized factorized element of the full system. For the first N columns (B5) and (B6) applies. For the following M+1 columns and N rows they also apply, but for the last N+1 to N+M+1 elements we must switch from negative accumulation to positive accumulation, see Tscherning(1978, Appendix 1).

The calculation of error-estimates for several quantities (R) may also take advantage of this. Let P be an (N+M) × R matrix of covariances between the observations and the R quantities to be predicted and Q an R × R matrix of covariances of the quantities to be predicted. Then we have an extended (still symmetric) system

$$\begin{Bmatrix} \overline{C} & A & P \\ A^T & W & P' \\ P^T & P'^T & Q \end{Bmatrix}. \quad \text{(B7)}$$

which may also be factorized by the generalized procedure. The result of (B4 and B5) will be the error-covariances and error-variances of the predicted quantities. This was used in Arabelos and Tscherning, 2008.

The system has the permanence property. So that if new columns are added, one may start the factorization from the first of these new columns. In this case multiprocessing should be re-started from the last fully reduced chunk.