USE OF MASSIVE PARALLEL COMPUTING LIBRARIES IN THE CONTEXT OF GLOBAL GRAVITY FIELD DETERMINATION FROM SATELLITE DATA

Jan Martin Brockmann and Wolf-Dieter Schuh

Institute of Geodesy and Geoinformation, Department of Theoretical Geodesy, University of Bonn, 53115 Bonn, Germany, {brockmann, schuh}@geod.uni-bonn.de

ABSTRACT

The estimation of the global Earth's gravity field parametrized as a finite spherical harmonic series is computationally demanding. The computational effort depends on the one hand on the maximal resolution of the spherical harmonic expansion (i.e. the number of parameters to be estimated) and on the other hand on the number of observations (which are several millions for e.g. observations from the GOCE satellite missions). To circumvent these restrictions, a massive parallel software based on high-performance computing (HPC) libraries as ScaLAPACK, PBLAS and BLACS was designed in the context of GOCE HPF WP6000 and the GOCO consortium. A prerequisite for the use of these libraries is that all matrices are block-cyclic distributed on a processor grid comprised by a large number of (distributed memory) computers. Using this set of standard HPC libraries has the benefit that once the matrices are distributed across the computer cluster, a huge set of efficient and highly scalable linear algebra operations can be used.

Key words: combined global gravity field determination; block-cyclic matrix distribution; SCALAPACK; MPI.

1. INTRODUCTION

Estimating of both a satellite-only and a combined global Earth's gravity field model as a spherical harmonic expansion is computationally demanding, as a large number of unknown spherical harmonic coefficients (more than 70 000 for satellite-only models at spherical harmonic degree and order (d/o) 250) is determined from a huge number of observations (hundreds of millions for GOCE). For satellite-only models, the resulting normal equations consume about 30 GB in main memory of a computer. This memory requirement even increases incorporating terrestrial data (with a higher signal content) in global gravity field determination.

Within this contribution we will concentrate on satellite-only models. The key tasks are assemblnig and decorrelation of the observation equations and the assembling of the normal equation matrices. To obtain an optimal satellite-only model these steps are repeated for different satellite mission observations as for example performed in the GOCO consortium for CHAMP, GRACE, GOCE and SLR observations ([3]). An optimal combined model can be determined by a weighted sum of the independent normal equations (which are of different resolution) and the solution of the resulting combined system of equations. The individual normal equations are normally given in a different numbering scheme, i.e. the sequence of the parameters differs in the individually assembled normal equations $\mathbf{N}_i, \mathbf{n}_i$. Thus they need to be reordered before computing the joint normal equation via

$$\mathbf{N} = \sum_{i=1}^{I} w_i \mathbf{N}_i, \tag{1}$$

$$\mathbf{n} = \sum_{i=1}^{I} w_i \mathbf{n}_i. \tag{2}$$

The combined normal equation is then solved for the unknown spherical harmonic coefficients $\hat{\mathbf{x}}$, e.g. via a Cholesky decomposition and forward and backward substitution. The estimated covariance matrix $\boldsymbol{\Sigma}_{\hat{\mathbf{x}}\hat{\mathbf{x}}}$ can be determined via the inverse of the combined normal equation matrix

$$\hat{\mathbf{x}} = \text{solve}(\mathbf{N}, \mathbf{n}) \tag{3}$$

$$\boldsymbol{\Sigma}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = \mathbf{N}^{-1}. \tag{4}$$

All these briefly summarized operations are performed on the matrices with a size of at least 30 GB. To handle these matrices (w.r.t. memory requirements and computing time), distributed memory computer clusters can be used. Thus, the matrix can be stored distributed using the joint memory of all compute nodes. To perform the necessary linear algebra operations, high performance computing (HPC) libraries as PBLAS and SCALAPACK ([1]) exist, providing matrix-matrix operations acting on matrices distributed along a set of computers. Using this software libraries, computations can be easily performed using the memory and the computing power of all computing nodes in common, once the matrices are distributed as specified by the libraries. The only prerequisite to use

Proc. of '4th International GOCE User Workshop', Munich, Germany 31 March – 1 April 2011 (ESA SP-696, July 2011)

these libraries is that the matrices are distributed block-cyclically across the computer cluster. This concept will be summarized within the next section.

2. PREREQUISITS FOR THE USE OF HIGH PERFORMANCE COMPUTING LIBRARIES

As mentioned above, if the routines for linear algebra operations from the massive parallel HPC libraries should be used, the matrices need to be block-cyclically distributed along the computer cluster. This concept is illustrated in the following, using a small example matrix, to visualize the local view on the matrix of each computing node.

2.1. Grids of distributed memory computers (computer cluster): Naming conventions

Assume a set of computers, which are connectet via a fast network. These computers form a distributed memory computer cluster. Each of them may contain several CPUs or cores. Every individual core of a computer is called computing node within this context. These computing nodes may be virtually arranged in a rectangular $P_p \times Q_p$ computing grid (cf. Fig. 1). A row of this grid is called node row and the set of nodes in the same column is called node column. These nodes are uniquely addressed via their 2-dimensional Cartesian coordinates (p_r, p_c) in the grid, or by their 1-dimesional rank r_p (counted row-wise, cf. Fig. 1).

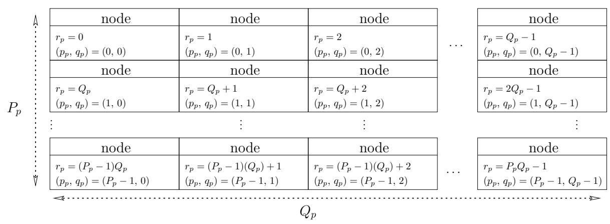


Figure 1. Computing nodes & their adresses in a rectangular computing grid.

2.2. Block-cyclic matrix distribution for the use in HPC libraries: An Example

A prerequisite for the use of HPC libraries such as PBLAS or ScaLAPACK is that the matrix to be operated on is block-cyclically distributed over the computing grid. To distribute a general matrix block-cyclically along the grid of computing nodes (cf. Fig. 2) the matrix is divided into patches of a given block size $b_r \times b_c$. These patches are cyclically distributed along a row of the node grid and cyclically along the columns of the grid ([1]).

To visualize the concept of the block-cyclic matrix distribution, a general matrix of size 8×9 (cf. Fig. 2) should

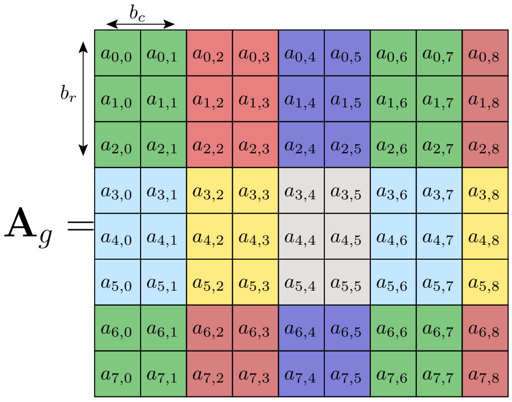


Figure 2. Example Matrix of Dimension 8×9 which is block-cyclically distributed along a grid of 2×3 nodes with block size of 3×2.

be block-cyclically distributed on a 2×3 computing grid. The block size of the distribution is chosen as 3×2. The colors in Fig. 2 represent the computing nodes of the grid in Fig. 3 to which the matrix patch is distributed to. The matrix patches are concatenated on each node patch by patch (row- and column-wise) into the local matrix $\mathbf{A}_l$ as sown for the example in Fig. 4.

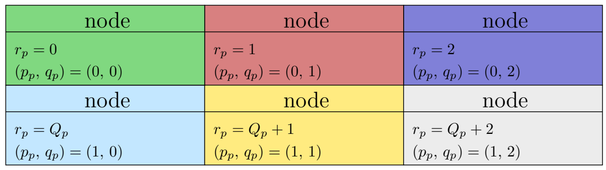


Figure 3. Example computing grid of 6 computing nodes set up as a rectangular 2×3 computing grid.

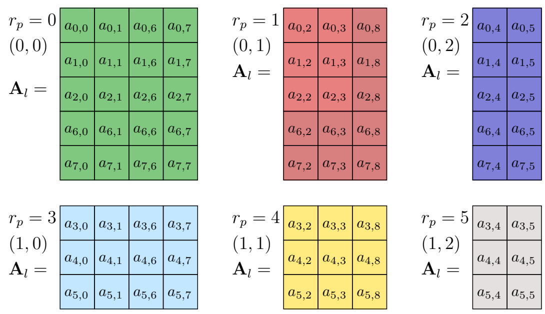


Figure 4. Local matrices on each of the nodes (p_r, p_c) for the example.

On each node, the two-dimensional matrix needs to be mapped to the linear main memory of the computer, which is done by column major order (CMO), storing the matrix column by column to a one-dimensional ordinary array $\_\mathbf{a}_l$. For the example matrix, the resulting local arrays on each node are shown in Fig. 5.

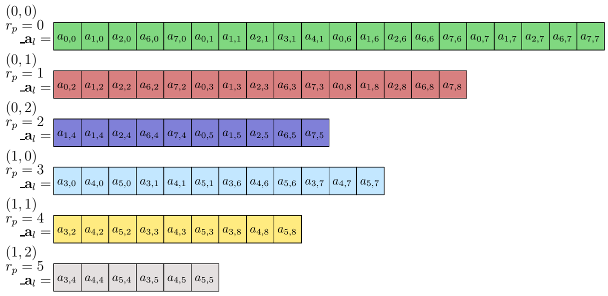


Figure 5. Local arrays $\_\mathbf{a}_l$ on each of the nodes $(p_r,\, p_c)$ for the choosen matrix] and distribution.

3. ESTIMATING COMBINED GRAVITY FIELDS FROM PREPROCESSED NORMAL EQUATIONS USING HPC LIBRARIES

Based on the massive parallel HPC libraries PBLAS and SCALAPACK a software package was designed in the context of the GOCO consortium to estimate combined gravity fields based on preprocessed normal equations of different satellite missions (e.g. CHAMP, GRACE and GOCE). Preprocessed normal equations of arbitrary size describing the satellite observations in terms of a spherical harmonic expansion can be combined performing the following steps (see Fig. 8 as an overview and details on the used routines):

1. Distribution: The matrices are directly read from a file into the block-cyclic scheme, thus no manual distribution of the matrix is required. The reading routine is implemented with the parallel file I/O concept of the MPI2.0 standard ([2]).
2. Conversion: The normal matrices of each individual group are converted to the same standards (e.g. constants GM, a, tide system).
3. Resolution: The individual normal matrices have different size, as different satellite missions recover another spectral range from the spherical harmonic coefficients, thus, the matrices are extended with zeros to the size of the largest normal equation system involved (the one with the maximal resolution).
4. Reordering: The normal matrices my differ in the numbering scheme of the parameters. The individual normal equations $\mathbf{N}_i$ are transferred to a common numbering scheme (column and row interchanges). Using adequate SCALAPACK routines, huge matrices can be reordered to arbitrary numbering schemes. Fig. 6 shows the performance for the reordering of a randomly shuffled matrices of different sizes (square matrices of size 200×200 – $100\,000 \times 100\,000$ (0–75GB), $b_r \times b_c$ fixed to 128×128, $P_p \times Q_p$ fixed to 6×5). The Timing for matrices corresponding to d/o 250 ($\hat{=}$ 30GB) gravity field normal matrices results in 50 $secs$ on a dedicated HPC computer cluster and to 780 $secs$ on a simple BEOWULF cluster for a “worst case“.
5. Combination: After the mentioned steps before, the combination can be computed via the weighted sum of matrices of same size.
6. Solution: The combined normal equations are solved using SCALAPACK operations. The covariance matrix is determined via the inversion of $\mathbf{N}$.
7. Analysis: In addition the normal equations can be analyzed estimating e.g. partial redundancies for the estimation of contribution, or the spectral behavior estimating eigenvalues and eigenvectors of the huge normal matrix, or propagating the variance/covariance information to gravity field functionals, ...

Currently, the software is in addition capable to assemble normal equations from GOCE gradiometry observations and from point-wise terrestrial gravity measurements.

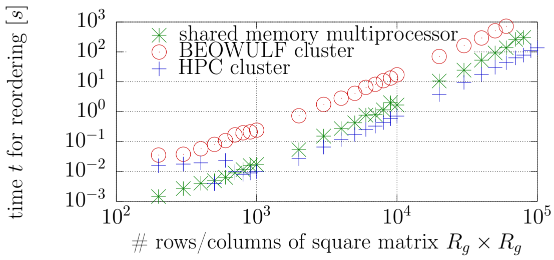


Figure 6. Runtime for square matrix reordering of different sizes.

4. APPLICATION: ESTIMATED EIGENVALUES AND EIGENVECTORS FOR DIFFERENT GOCE BASED COMBINATION MODELS

To demonstrate the power of the concept, the software was used to estimate different GOCE based combined gravity field models. For the gradiometry processing the time-wise method ([3] in RL01 and RL02) was used. Eigenvalues and eigenvectors of the gradiometer-only solution of RL01 and RL02 are compared to combined models adding the GPS SST part and Kaula constrains (cf. Tab. 1) in Fig. 7 and 9. The eigenvalues demonstrate the effect of the data combination in terms of numerical stability. The eigenvectors belonging to the smallest eigenvalues show which of the coefficients are estimated instable. The results can be used e.g. to estimate a tailored regularization matrix.

5. SUMMARY AND CONCLUSIONS

Within this contribution the use of high performance computing libraries in the context of global gravity field determination was shown. The prerequisite to use libraries as PBLAS and SCALAPACK, the block-cyclic

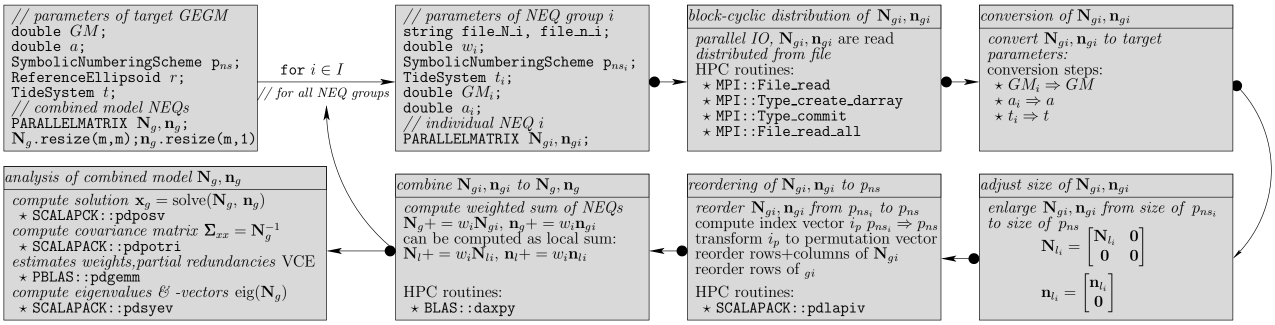
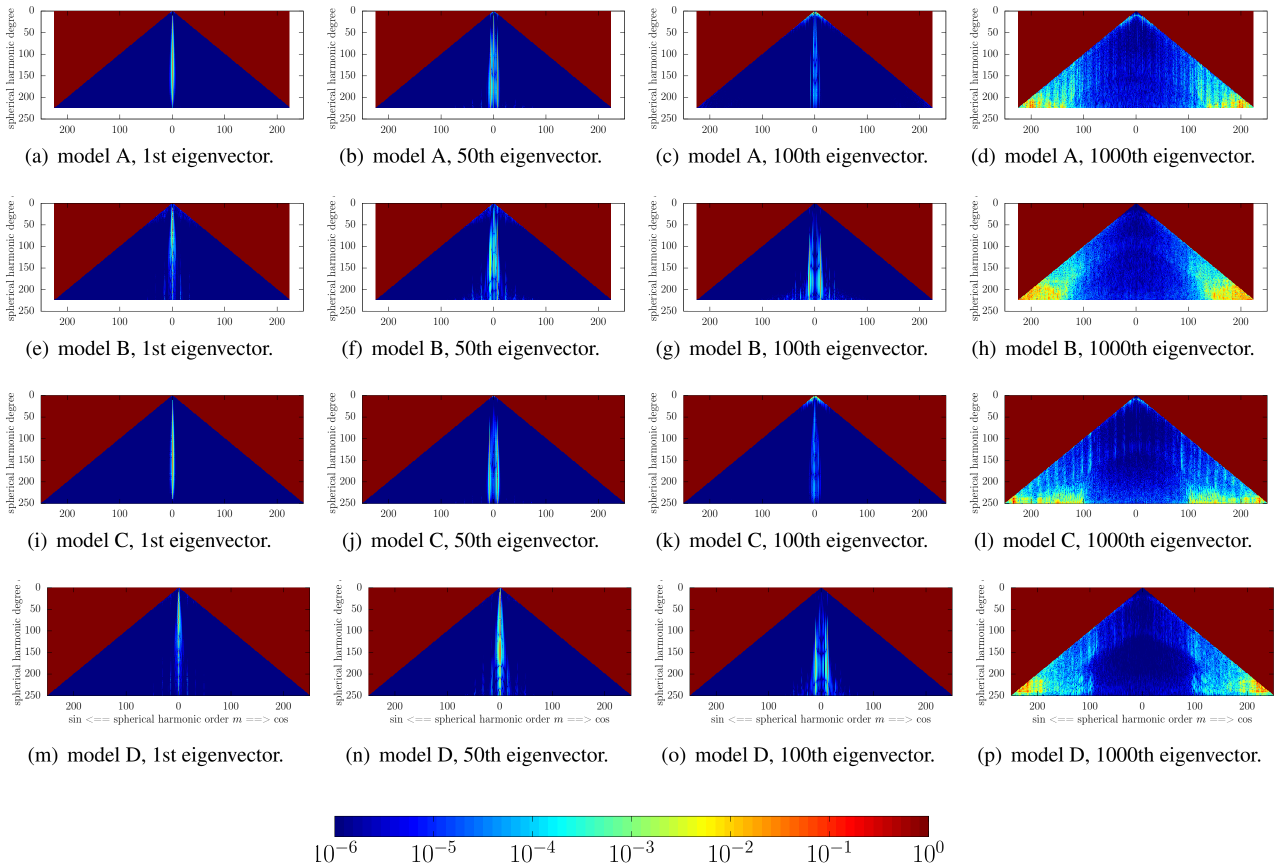


Figure 8. Scheme of massive parallel programm for the combination of preprocessed gravity field normal equations.



(a) model A, 1st eigenvector. (b) model A, 50th eigenvector. (c) model A, 100th eigenvector. (d) model A, 1000th eigenvector.

(e) model B, 1st eigenvector. (f) model B, 50th eigenvector. (g) model B, 100th eigenvector. (h) model B, 1000th eigenvector.

(i) model C, 1st eigenvector. (j) model C, 50th eigenvector. (k) model C, 100th eigenvector. (l) model C, 1000th eigenvector.

(m) model D, 1st eigenvector. (n) model D, 50th eigenvector. (o) model D, 100th eigenvector. (p) model D, 1000th eigenvector.

Figure 9. Logarithmic elements of i. normalized eigenvector in the coefficient triangle for models A–D

Table 1. Summary of the 4 computed models and the estiamted condition number of the normal equation system

id	data	κ
A	RL01: SGG	$1.3 \cdot 10^{12}$
B	RL01: SGG+SST+REG_{zonals}+REG_{higdeg}	$1.4 \cdot 10^{6}$
C	RL02: SGG	$4.8 \cdot 10^{13}$
D	RL02: SGG+SST+REG_{zonals}+REG_{higdeg}	$3.4 \cdot 10^{6}$

- SGG, SST $\hat{=}$ GOCE time-wise gradiometry & SST NEQ (RL01, 71days, [3], RL02, 225days)
- REG_{zonals} $\hat{=}$ Regularization of (near) zonal coefficients
- REG_{higdeg} $\hat{=}$ Regularization of high degree (RL01: > 170, RL02: > 180) coefficients

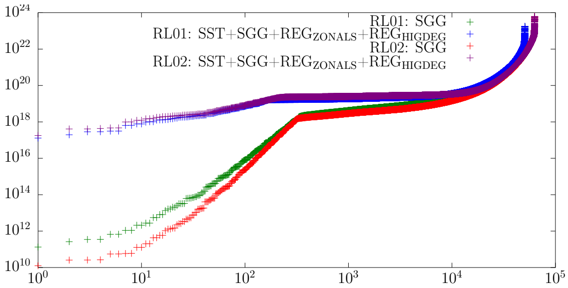


Figure 7. Eigenvalues for GOCE based gravity field models of RL01 and RL02.

matrix distribution, was recalled. To demonstrate the power of the libraries, the algorithmic complex operation of estimating eigenvalues and eigenvectors was performed on a 30 GB matrix of GOCE based gravity field normal equations.

ACKNOWLEDGMENTS

The authors acknowledge the European Space Agency for the provision of the GOCE data. The computations were performed on the JUROPA supercomputer at FZ Jülich, computing time was granted by John von Neumann Institute for Computing (project HBN15).

REFERENCES

[1] Blackford, L. S. et-al. (1997). *SCALAPACK Users Guide*. SIAM, Philadelphia, 2 edition.

[2] MPI-Forum (2009). MPI: A Message-Passing Interface Standard V2.2. `http://www.mpi-forum.org/docs/docs.html`.

[3] Pail, R. et-al. (2010b). GOCE gravity field model derived from orbit and gradiometry data applying the time-wise approach. In *ESA Living Planet Symposium Bergen, Proceedings*.