Parallel Gibbs Sampling for Computing and Propagating Large Covariance Matrices

Karl-Rudolf Koch, Jürgen Kusche, Christian Boxhammer, Brigitte Gundlich

Summary
The Gibbs sampler for inverting large matrices of normal equations and for propagating the resulting covariance matrices is modified for parallel computing. The matrix times vector multiplications for generating the error vectors of the unknown parameters are replaced by matrix times matrix multiplications so that the computationally efficient BLAS and ATLAS subroutines can be applied. To judge the accuracy of the computed covariance matrix a criterion is derived which gives information about the significant digits obtained during the process of the parallel computing. Finally, based on the blocking technique for reducing correlations between the generated error vectors a storage scheme is used which allows the inversion of matrices of normal equations of any size. The parallel Gibbs sampler has been applied for the inversion of normal equations as they result from the data analysis for the new satellite gravity missions. With a parallel computer consisting of 15 nodes the speed of the computations could be increased by a factor of 7.4 in comparison to a single computer.

Zusammenfassung
Das Gibbs-Verfahren zur Inversion großer Normalgleichungsmatrizen und zur Fehlerfortpflanzung mittels der resultierenden Kovarianzmatrizen wird für paralleles Rechnen modifiziert. Die Multiplikation von Matrizen mit Vektoren zur Generierung der Fehlervektoren für die unbekannten Parameter wird durch die Multiplikation von Matrizen ersetzt, so dass die rechentechnisch effizienten BLAS und ATLAS Unterprogramme benutzt werden können. Um die Genauigkeit der berechneten Kovarianzmatrix zu beurteilen, wird ein Kriterium abgeleitet, das während des parallelen Rechnens die Information über die erzielten signifikanten Stellen verschafft. Schließlich wird noch mit Hilfe der Zerlegung der Matrizen in Teilmatrizen, die zur Verringerung der Korrelationen zwischen generierten Fehlervektoren dient, ein Speicherschema für die Matrizen entwickelt, das die Inversion von Normalgleichungen beliebiger Größe erlaubt. Das parallelisierte Gibbs-Verfahren wird auf Normalgleichungen angewendet, wie sie aus der Datenanalyse der neuen Satellitenmissionen für Schwerefeldbestimmungen resultieren. Mit einem parallelen Rechner, der aus 15 Knoten besteht, konnte bei den Berechnungen die Rechengeschwindigkeit um den Faktor von 7.4 im Vergleich zu einem einzigen Rechner erhöht werden.

1 Introduction

The new satellite missions for determining the gravity field of the earth and its temporal variations necessitate the estimation of several ten thousands of unknown parameters describing the gravity field. The gradiometer data to be collected by the GOCE satellite, for instance, should allow the determination of the spherical harmonic coefficients of the gravity field up to degree and order 240 which results in about 60 000 unknown parameters (ESA 1999, p. 85). Computing this large number of unknown parameters either by solving the system of normal equations or, if an assembly of normal equations is avoided, by iterative methods will be computationally quite demanding. In addition, for the quality control of the solutions not only the estimates but also the variances and covariances of the unknown parameters are needed.

Furthermore, quantities will be derived from the solutions for the gravity field like geoid undulations or satellite orbits. The variances and covariances of these quantities are also needed. They are obtained from error propagation either in a linear or in a nonlinear form. These computations together with obtaining the covariance matrix for the unknown parameters of the gravity field are much more demanding than computing the estimates of the unknown parameters.

Even without supercomputers these computational tasks can be solved if ordinary computers like workstations or PCs are combined for parallel computing in clusters or grids, see for instance Plank (2002). Ideally, the time for the computations can be reduced by a scalability factor which equals the number of nodes in the cluster. Of course, this factor cannot be reached because of the necessary transfer of data between the nodes which is slow in comparison to the transfer of data within a node. In addition, not all computations can be done in parallel. Nevertheless, the speed of the computations can be considerably increased by parallel computing.

Parallel computing has been simplified during the last years by specifying the message-passing interface (MPI). This interface enables the communication between the computing nodes of a parallel computer formed by a cluster of computers. MPI allows messages between the nodes to be sent and received, see for instance Gropp et al. (1999a; 1999b). In a simple setup MPI provides the means to establish a master which distributes the data to the clients which are formed by the rest of the nodes of the parallel computer. The clients are then able to perform special tasks and send the results back to the master. A complete implementation of the MPI specifications is the LAM/MPI programming environment by The LAM/MPI Team (2003).

For statistical applications the LAPACK software (Anderson et al. 1999) of linear algebra is very helpful be-

cause it provides for instance the solution of linear equations, the Cholesky factorization of symmetric positive definite matrices, and their inversions. To run these programs on parallel computers the ScaLAPACK library has been developed (Blackford et al. 1997). Because of the large number of unknown parameters describing the earth's gravity field the core storage of a cluster might not be large enough to keep the coefficient matrices or the matrices of normal equations. However, for out-of-core matrices only solvers based on factorizations are provided by ScaLAPACK, but no inversions.

Monte Carlo algorithms, on the other hand, are well suited for parallel computing. This is quite understandable because Monte Carlo methods allow to draw independent samples from a given distribution which are subsequently subject to statistical inference. Instead of one computer many computers can generate these samples, the results have only to be collected at the end of the computations.

For computing and propagating large covariance matrices, Gundlich, Koch and Kusche (2003) suggested Markov Chain Monte Carlo methods, that is the Gibbs sampler. Error vectors are generated for the unknown parameters, for instance the harmonic coefficients of the gravity field of the earth, from which the covariance matrix is estimated. The error vectors can also be applied for error propagations so that the covariance matrix is not needed at all for that purpose. This is especially important, since test computations suggest that the required number of error vectors may be less than one tenth of the number of unknown parameters to compute the covariance matrix and the error propagations with sufficient accuracy. For quick-look computations, even a few error vectors may be sufficient. Thus, storing and handling the covariance matrix which is huge for many unknown parameters can be avoided for error propagations. Another advantage, already mentioned above, is the fact that the Gibbs sampler can be easily applied in parallel computing. In fact, that was the motivation for developing the Markov Chain Monte Carlo method for computing and propagating large covariance matrices.

In the following chapter, the main formulas of Gundlich, Koch and Kusche (2003) are collected for easier reference. The third chapter introduces modifications for the parallel computing. It also gives a method to obtain the accuracy of the covariance matrix during the process of generating the samples. Chapter 4 reports numerical experiments. The conclusions finally follow in the last chapter.

2 Gibbs Sampler for Generating Covariance Matrices

Random errors are generated for the unknown parameters describing the gravity field of the earth so that the unknown parameters are also considered as random quantities. This is the concept of Bayesian statistics, which will be applied in the following. It will be assumed that the unknown parameters are estimated in a linear model. Let $\boldsymbol{X}$ be the $n \times u$ coefficient matrix of the linear model with full column rank, $\boldsymbol{\beta}$ the $u \times 1$ vector of unknown parameters, $\boldsymbol{y}$ the $n \times 1$ vector of observations which will be assumed as normally distributed, and $\boldsymbol{P}$ its positive definite $n \times n$ weight matrix, the posterior distribution for the vector $\boldsymbol{\beta}$ is then given by the normal distribution, see for instance Koch (2000, p. 90)

$$\boldsymbol{\beta}|\boldsymbol{y} \sim N(\hat{\boldsymbol{\beta}}, D(\boldsymbol{\beta}|\boldsymbol{y})) \tag{2.1}$$

where $\hat{\boldsymbol{\beta}}$ is the estimate of $\boldsymbol{\beta}$ given by the well known formula of least squares adjustment

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{P}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{P}\boldsymbol{y} \tag{2.2}$$

and $D(\boldsymbol{\beta}|\boldsymbol{y})$ the covariance matrix of $\boldsymbol{\beta}$

$$D(\boldsymbol{\beta}|\boldsymbol{y}) = (\boldsymbol{X}'\boldsymbol{P}\boldsymbol{X})^{-1} = \boldsymbol{N}^{-1} = \boldsymbol{V} \tag{2.3}$$

with $\boldsymbol{N} = \boldsymbol{X}'\boldsymbol{P}\boldsymbol{X}$ being the matrix of normal equations for $\boldsymbol{\beta}$. The inverse $\boldsymbol{V}$ of $\boldsymbol{N}$, i. e. the covariance matrix $D(\boldsymbol{\beta}|\boldsymbol{y})$, needs to be determined. The $u \times 1$ vector $\boldsymbol{e}$ of errors of the unknown parameters $\boldsymbol{\beta}$ is introduced by

$$\boldsymbol{e} = \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \tag{2.4}$$

so that the distribution follows because of (2.1), see for instance Koch (1999, p. 122)

$$\boldsymbol{e} \sim N(\boldsymbol{0}, \boldsymbol{V}) \tag{2.5}$$

with expectation $E(\boldsymbol{e}) = \boldsymbol{0}$ and covariance matrix

$$D(\boldsymbol{e}) = D(\boldsymbol{\beta}|\boldsymbol{y}) = \boldsymbol{V}\,. \tag{2.6}$$

If $\boldsymbol{x}$ is a random vector with expectation $E(\boldsymbol{x})$, its covariance matrix is defined by

$$D(\boldsymbol{x}) = \int_{\mathcal{X}} (\boldsymbol{x} - E(\boldsymbol{x}))(\boldsymbol{x} - E(\boldsymbol{x}))' p(\boldsymbol{x}) d\boldsymbol{x} \tag{2.7}$$

where $\mathcal{X}$ denotes the domain of the integration and $p(\boldsymbol{x})$ the probability density function for $\boldsymbol{x}$. This integral can be evaluated numerically by the Monte Carlo integration. Thus, by drawing M samples $\boldsymbol{x}^{(k)}$ with $k \in \{1, \ldots, M\}$ from $p(\boldsymbol{x})$, the estimate $\hat{D}(\boldsymbol{x})$ of $D(\boldsymbol{x})$ is obtained by

$$\hat{D}(\boldsymbol{x}) = \frac{1}{M} \sum_{k=1}^{M} (\boldsymbol{x}^{(k)} - E(\boldsymbol{x}))(\boldsymbol{x}^{(k)} - E(\boldsymbol{x}))'\,. \tag{2.8}$$

If we generate M samples $\boldsymbol{e}^{(k)}$ with the distribution (2.5), the estimate $\hat{\boldsymbol{V}}$ of the inverse $\boldsymbol{V}$ of $\boldsymbol{N}$, that is the estimate $\hat{D}(\boldsymbol{\beta}|\boldsymbol{y})$ of the covariance matrix of $\boldsymbol{\beta}$, is found accordingly by

$$\hat{D}(\boldsymbol{\beta}|\boldsymbol{y}) = \hat{D}(\boldsymbol{e}) = \hat{\boldsymbol{V}} = \frac{1}{M} \sum_{k=1}^{M} \boldsymbol{e}^{(k)} \boldsymbol{e}^{(k)'}\,. \tag{2.9}$$

This is the basic idea of computing covariance matrices by Monte Carlo methods. If for error propagations the covariance matrix $D(f(\boldsymbol{\beta})|\boldsymbol{y})$ of the transformation $f(\boldsymbol{\beta})$ of the unknown parameters $\boldsymbol{\beta}$ is needed, we obtain its estimate

$$\hat{D}(f(\boldsymbol{\beta})|\boldsymbol{y}) = \hat{D}(f(\boldsymbol{e})) = \frac{1}{M}\sum_{k=1}^{M}\Big(f(\boldsymbol{e}^{(k)}) - \hat{E}(f(\boldsymbol{e}))\Big)\Big(f(\boldsymbol{e}^{(k)}) - \hat{E}(f(\boldsymbol{e}))\Big)' \tag{2.10}$$

with

$$\hat{E}(f(\boldsymbol{e})) = \frac{1}{M}\sum_{k=1}^{M} f(\boldsymbol{e}^{(k)})$$

where $f(\boldsymbol{\beta})$ or $f(\boldsymbol{e})$ might be a linear or nonlinear function of $\boldsymbol{\beta}$. In contrast to the law of error propagation $\boldsymbol{FVF}'$ for the linear function $\boldsymbol{F\beta}$, the inverse $\boldsymbol{V}$ of the matrix $\boldsymbol{N}$ of normal equations is not needed in (2.10).

Drawing samples $\boldsymbol{e}^{(k)}$ from (2.5) requires the inverse $\boldsymbol{V}$ to be known. However, this can be avoided (Harville 1999) if we apply the Gibbs sampler (Geman and Geman 1984), a special Markov Chain Monte Carlo method, see also Koch (2000, p. 204) or Smith and Roberts (1993), because the Gibbs sampler uses conditional distributions. We divide the error vector $\boldsymbol{e}$ into r subvectors $\boldsymbol{e}_i$ with $i \in \{1,\ldots,r\}$ and the matrix $\boldsymbol{N}$ of normal equations and its inverse $\boldsymbol{V}$ accordingly into $r \times r$ subblocks

$$\boldsymbol{e} = \begin{vmatrix} \boldsymbol{e}_1 \\ \vdots \\ \boldsymbol{e}_r \end{vmatrix}, \quad \boldsymbol{N} = \begin{vmatrix} \boldsymbol{N}_{11} & \ldots & \boldsymbol{N}_{1r} \\ \vdots & \ddots & \vdots \\ \boldsymbol{N}_{r1} & \ldots & \boldsymbol{N}_{rr} \end{vmatrix},$$

$$\boldsymbol{V} = \begin{vmatrix} \boldsymbol{V}_{11} & \ldots & \boldsymbol{V}_{1r} \\ \vdots & \ddots & \vdots \\ \boldsymbol{V}_{r1} & \ldots & \boldsymbol{V}_{rr} \end{vmatrix}. \tag{2.11}$$

The Gibbs sampler is iteratively applied

$$\begin{aligned}
&\texttt{do } k = 1, M \\
&\texttt{draw} \quad \boldsymbol{e}_1^{(k)} \quad \texttt{from} \quad p(\boldsymbol{e}_1|\boldsymbol{e}_2^{(k-1)},\ldots,\boldsymbol{e}_r^{(k-1)}) , \\
&\quad \vdots \\
&\texttt{draw} \quad \boldsymbol{e}_l^{(k)} \quad \texttt{from} \quad p(\boldsymbol{e}_l|\boldsymbol{e}_1^{(k)},\ldots,\boldsymbol{e}_{l-1}^{(k)}, \boldsymbol{e}_{l+1}^{(k-1)},\ldots,\boldsymbol{e}_r^{(k-1)}) , \\
&\quad \vdots \\
&\texttt{draw} \quad \boldsymbol{e}_r^{(k)} \quad \texttt{from} \quad p(\boldsymbol{e}_r|\boldsymbol{e}_1^{(k)},\ldots,\boldsymbol{e}_{r-1}^{(k)}) , \\
&\texttt{next } k.
\end{aligned} \tag{2.12}$$

The starting values are $\boldsymbol{e}_l = \boldsymbol{0}$ because of $E(\boldsymbol{e}_l) = \boldsymbol{0}$. The probability density functions in (2.12) are obtained from the conditional normal distribution, see Gundlich, Koch and Kusche (2003)

$$\boldsymbol{e}_l^{(k)}|\boldsymbol{e}_1^{(k)},\ldots,\boldsymbol{e}_{l-1}^{(k)},\boldsymbol{e}_{l+1}^{(k-1)},\ldots,\boldsymbol{e}_r^{(k-1)} \sim N\Big(-\boldsymbol{N}_{ll}^{-1}\Big(\sum_{j<l}\boldsymbol{N}_{lj}\boldsymbol{e}_j^{(k)} + \sum_{j>l}\boldsymbol{N}_{lj}\boldsymbol{e}_j^{(k-1)}\Big), \boldsymbol{N}_{ll}^{-1}\Big) . \tag{2.13}$$

If the vector $\boldsymbol{z}_l^{(k)}$ contains independently generated random numbers with normal distributions $N(0,1)$, random numbers for $\boldsymbol{e}_l^{(k)}$ with the conditional distribution (2.13) are computed by the transformation, see for instance Koch (2000, p. 187)

$$\boldsymbol{e}_l^{(k)} = \boldsymbol{G}_l\boldsymbol{z}_l^{(k)} - \boldsymbol{N}_{ll}^{-1}\Big(\sum_{j<l}\boldsymbol{N}_{lj}\boldsymbol{e}_j^{(k)} + \sum_{j>l}\boldsymbol{N}_{lj}\boldsymbol{e}_j^{(k-1)}\Big) \tag{2.14}$$

with $\boldsymbol{G}_l$ being a lower triangular matrix, the Cholesky factor of $\boldsymbol{N}_{ll}^{-1}$,

$$\boldsymbol{N}_{ll}^{-1} = \boldsymbol{G}_l\boldsymbol{G}_l' . \tag{2.15}$$

For $M \to \infty$, the joint distribution for $\boldsymbol{e}^{(k)} = |\boldsymbol{e}_1^{(k)'},\ldots,\boldsymbol{e}_r^{(k)'}|'$ converges to the normal distribution (2.5), see Geman and Geman (1984).

In general, subsequent samples for $\boldsymbol{e}^{(k)}$ will be correlated which affects the convergence rate of the Gibbs sampler. Thus, after an initial burn-in phase where all samples are discarded, only each s-th sample is used in the Monte Carlo integration. The size of s depends on the correlation between subsequent samples for $\boldsymbol{e}^{(k)}$ which can be computed if the inverse $\boldsymbol{V}$ is known as shown by Gundlich, Koch and Kusche (2003). Highly correlated unknown parameters cause correlations between the samples $\boldsymbol{e}^{(k)}$ of their generated error vectors. By applying a blocking technique (Liu 2001, p. 131), i. e. by clustering errors of correlated unknown parameters in subvectors $\boldsymbol{e}_l$, the correlation can be considerably reduced. This is the reason for dividing the error vector $\boldsymbol{e}$ into the subvectors $\boldsymbol{e}_l$ according to (2.11).

There is another reason for dividing the error vector $\boldsymbol{e}$ into subvectors $\boldsymbol{e}_l$. It allows the estimation by conditioning (Harville 1999) which reduces the variance of the estimate (2.9) because a certain part of the integral of (2.7) is solved analytically by using the expected values of conditional density functions. The estimation by conditioning gives instead of (2.9) the subblocks in (2.11) of the covariance matrix $\boldsymbol{V}$ of the unknown parameters $\boldsymbol{\beta}$ by, see Gundlich, Koch and Kusche (2003)

$$\hat{\boldsymbol{V}}_{ll} = \boldsymbol{N}_{ll}^{-1} + \frac{1}{M}\sum_{k=1}^{M}\boldsymbol{\mu}_l^{(k)}\boldsymbol{\mu}_l^{(k)'}$$

and

$$\hat{\boldsymbol{V}}_{lj} = \frac{1}{M}\sum_{k=1}^{M}\boldsymbol{\mu}_l^{(k)}\boldsymbol{e}_j^{(k)'} ,$$

$$\text{for} \quad j,l \in \{1,\ldots,r\} , \; j \neq l \tag{2.16}$$

with

$$\boldsymbol{\mu}_l^{(k)} = -\boldsymbol{N}_{ll}^{-1}\Big(\sum_{j<l}\boldsymbol{N}_{lj}\boldsymbol{e}_j^{(k)} + \sum_{j>l}\boldsymbol{N}_{lj}\boldsymbol{e}_j^{(k-1)}\Big)\,. \tag{2.17}$$

As can be seen from (2.16), the estimation by conditioning starts off a covariance matrix given by the inverse blocks $\boldsymbol{N}_{ll}^{-1}$ on the diagonal. Because of the symmetry, only the elements on the diagonal and above the diagonal are computed. The samples $\boldsymbol{e}_j^{(k)}$ are obtained with the Gibbs sampler (2.12) and the transformation (2.14) by taking after a burn-in phase each s-th sample of the iteratively generated samples.

It should be mentioned, finally, that for normal equation systems in gravity modelling very efficient ordering schemes for the unknown parameters exist. This means, from satellite mission characteristics one knows a priori which parameters will be highly correlated.

3 Modifications for Parallel Computing

Generating the error vectors by the Gibbs sampler (2.12) requires the multiplication of matrices and vectors, that is the multiplication of rows and columns. Because of the many transfers of elements between the cache and the memory of a processor, these multiplications cannot be as efficiently organized as the multiplication of matrices. Block algorithms can then be applied which operate on rectangular subblocks of matrices and keep the subblocks as long as possible in the cache of a processor. To optimize matrix times matrix and also matrix times vector multiplications, basic linear algebra subroutines (BLAS) have been developed, see for instance Demmel (1997, p. 66). For an efficient implementation on an individual computer, the automatically tuned linear algebra software (ATLAS) is available (Whaley et al. 2000).

Because of the efficient multiplication of matrices by BLAS and ATLAS subroutines, the error vectors $\boldsymbol{e}^{(k)}$ in (2.9) should not be generated in one single run of the Gibbs sampler (2.12), but in several parallel runs. Thus, samples will not be generated for only one error vector leading by the s-th sample to the sample $\boldsymbol{e}^{(k)}$ in (2.9), but for p error vectors, giving as s-th samples $\boldsymbol{e}_{(1)}^{(k)}, \boldsymbol{e}_{(2)}^{(k)}, \ldots, \boldsymbol{e}_{(p)}^{(k)}$. Thus, instead of one burn-in phase there are p burn-in phases. We collect the samples $\boldsymbol{e}_{(m)}^{(k)}$ with $m \in \{1,\ldots,p\}$ in the $u \times p$ matrix $\boldsymbol{E}^{(k)}$

$$\boldsymbol{E}^{(k)} = \Big|\boldsymbol{e}_{(1)}^{(k)}, \boldsymbol{e}_{(2)}^{(k)}, \ldots, \boldsymbol{e}_{(p)}^{(k)}\Big| \tag{3.1}$$

and obtain

$$\boldsymbol{E}^{(k)}\boldsymbol{E}^{(k)\prime} = \sum_{m=1}^{p}\boldsymbol{e}_{(m)}^{(k)}\boldsymbol{e}_{(m)}^{(k)\prime}\,. \tag{3.2}$$

The number p of parallel runs of the Gibbs sampler is selected such that the number M of samples used for estimating the covariance matrix divided by p gives an integer. Thus, instead of (2.9) one gets

$$\begin{aligned}\hat{\boldsymbol{V}} &= \frac{1}{M}\sum_{k=1}^{M/p}\boldsymbol{E}^{(k)}\boldsymbol{E}^{(k)\prime}\\ &= \frac{1}{M}\Big(\sum_{m=1}^{p}\boldsymbol{e}_{(m)}^{(1)}\boldsymbol{e}_{(m)}^{(1)\prime} + \ldots + \sum_{m=1}^{p}\boldsymbol{e}_{(m)}^{(M/p)}\boldsymbol{e}_{(m)}^{(M/p)\prime}\Big)\,.\end{aligned} \tag{3.3}$$

The generated error vector $\boldsymbol{e}_{(m)}^{(k)}$ in (3.3) has to be subdivided according to (2.11) leading to

$$\boldsymbol{e}_{(m)}^{(k)} = \begin{vmatrix}\boldsymbol{e}_{1(m)}^{(k)}\\ \vdots\\ \boldsymbol{e}_{r(m)}^{(k)}\end{vmatrix}\,. \tag{3.4}$$

The vectors $\boldsymbol{e}_{l(m)}^{(k)}$ for $l \in \{1,\ldots,r\}$ are collected in the matrix $\boldsymbol{E}_l^{(k)}$

$$\boldsymbol{E}_l^{(k)} = \Big|\boldsymbol{e}_{l(1)}^{(k)}, \boldsymbol{e}_{l(2)}^{(k)}, \ldots, \boldsymbol{e}_{l(p)}^{(k)}\Big|\,. \tag{3.5}$$

Before generating the samples, the inverses $\boldsymbol{N}_{ll}^{-1}$ in (2.14) and their Cholesky factors $\boldsymbol{G}_l$ are computed by LAPACK routines which also apply the BLAS subprograms. The matrices $\boldsymbol{N}_{ll}^{-1}$ and $\boldsymbol{N}_{lj}$ in (2.14) and (2.17) are then multiplied to obtain the matrix $\boldsymbol{R}_{lj}$

$$\boldsymbol{R}_{lj} = -\boldsymbol{N}_{ll}^{-1}\boldsymbol{N}_{lj} \quad \text{for} \quad j,l \in \{1,\ldots,r\}\,. \tag{3.6}$$

By replacing the vectors $\boldsymbol{e}_j^{(k)}$ and $\boldsymbol{e}_j^{(k-1)}$ by the matrices $\boldsymbol{E}_j^{(k)}$ and $\boldsymbol{E}_j^{(k-1)}$ from (3.5), we obtain instead of the vector $\boldsymbol{\mu}_l^{(k)}$ in (2.17) the matrix $\boldsymbol{U}_l^{(k)}$

$$\boldsymbol{U}_l^{(k)} = \sum_{j<l}\boldsymbol{R}_{lj}\boldsymbol{E}_j^{(k)} + \sum_{j>l}\boldsymbol{R}_{lj}\boldsymbol{E}_j^{(k-1)}\,. \tag{3.7}$$

The estimates (2.16) of the subblocks of the covariance matrix then follow with (3.3) by

$$\hat{\boldsymbol{V}}_{ll} = \boldsymbol{N}_{ll}^{-1} + \frac{1}{M}\sum_{k=1}^{M/p}\boldsymbol{U}_l^{(k)}\boldsymbol{U}_l^{(k)\prime}$$

and

$$\hat{\boldsymbol{V}}_{lj} = \frac{1}{M}\sum_{k=1}^{M/p}\boldsymbol{U}_l^{(k)}\boldsymbol{E}_j^{(k)\prime}$$

$$\text{for} \quad j,l \in \{1,\ldots,r\}\,,\ j \neq l\,. \tag{3.8}$$

The matrix $\boldsymbol{E}_l^{(k)}$ in (3.5) is obtained from (2.14) by

$$\boldsymbol{E}_l^{(k)} = \boldsymbol{G}_l\boldsymbol{Z}_l^{(k)} + \boldsymbol{U}_l^{(k)} \tag{3.9}$$

with

$$\boldsymbol{Z}_l^{(k)} = \Big|\boldsymbol{z}_{l(1)}^{(k)}, \boldsymbol{z}_{l(2)}^{(k)}, \ldots, \boldsymbol{z}_{l(p)}^{(k)}\Big| \tag{3.10}$$

where the vectors $z_{l(m)}^{(k)}$ for $m \in \{1,\ldots,p\}$ contain independently generated random numbers with normal distributions $N(0,1)$. Instead of the matrix times vector multiplications in (2.14), (2.16), and (2.17), we now have matrix times matrix multiplications for generating the random realizations by (3.9) with (3.7) and for estimating the covariance matrix by (3.8).

There is a third reason besides reducing the correlation between samples and estimating by conditioning for subdividing the error vector $\boldsymbol{e}$ and the matrix of normal equations and its inverse according to (2.11). The matrices of normal equations obtained for high-resolution modelling of the gravity field are too huge to keep them in the core storage of an ordinary computer. The matrices have to be divided into subblocks according to (2.11) and the subblocks have to be stored on disks. For the computations discussed here, the size of the blocks are selected such that the multiplication (3.6) of $\boldsymbol{N}_{ll}^{-1}\boldsymbol{N}_{lj}$, which is performed before generating the samples, fits into the core storage. For generating the samples, the matrices $\boldsymbol{E}_l^{(k)}$ in (3.5) and $\boldsymbol{U}_l^{(k)}$ in (3.7) use afterwards the core storage of $\boldsymbol{R}_{lj}$. It is thus determined how many samples can be generated at most in parallel runs, which gives the maximum number for p in (3.1).

It is obvious from (3.7) to (3.9) that the matrices $\boldsymbol{E}_l^{(k)}$ of error vectors may be generated not only by one computer, but in parallel by several ones. Let us assume that we have z computers in a cluster, each one with core storage and disks. We then obtain the matrix $\boldsymbol{E}_{lz}^{(k)}$ of error vectors, where the matrices $\boldsymbol{E}_{l(o)}^{(k)}$ from (3.5) with $o \in \{1,\ldots,z\}$ of error vectors of each of the z computers of the cluster are collected

$$\boldsymbol{E}_{lz}^{(k)} = \left|\boldsymbol{E}_{l(1)}^{(k)}, \boldsymbol{E}_{l(2)}^{(k)}, \ldots, \boldsymbol{E}_{l(z)}^{(k)}\right| . \tag{3.11}$$

Accordingly, we get instead of (3.7)

$$\boldsymbol{U}_{lz}^{(k)} = \left|\boldsymbol{U}_{l(1)}^{(k)}, \boldsymbol{U}_{l(2)}^{(k)}, \ldots, \boldsymbol{U}_{l(z)}^{(k)}\right| . \tag{3.12}$$

If we select the number p of parallel runs of the Gibbs sampler such that $M/(pz)$ gives an integer, we find instead of (3.8) the estimate by conditioning

$$\hat{\boldsymbol{V}}_{ll} = \boldsymbol{N}_{ll}^{-1} + \frac{1}{M}\sum_{k=1}^{M/(pz)} \boldsymbol{U}_{lz}^{(k)}\boldsymbol{U}_{lz}^{(k)'}$$

and

$$\hat{\boldsymbol{V}}_{lj} = \frac{1}{M}\sum_{k=1}^{M/(pz)} \boldsymbol{U}_{lz}^{(k)}\boldsymbol{E}_{jz}^{(k)'}$$

$$\text{for} \quad j,l \in \{1,\ldots,r\}\ ,\ j \neq l\ . \tag{3.13}$$

Given the number M of samples for estimating the covariance matrix, each computer of the cluster needs to generate in case of parallel processing according to (3.13) only $M/(pz)$ samples while a single computer would have to generate according to (3.8) M/p samples.

When estimating the covariance matrix $\boldsymbol{V}$ of the unknown parameters $\boldsymbol{\beta}$, it is necessary to get an idea of the accuracy of the estimate in order to determine the number M of samples. Gundlich, Koch and Kusche (2003) used, among other criteria, the scaled Frobenius norm d

$$d = \left(\frac{1}{u^2\max(v_{ii})^2}\sum_{i=1}^{u}\sum_{j=1}^{u}(\hat{v}_{ij} - v_{ij})^2\right)^{1/2} \tag{3.14}$$

where u denotes the number of unknown parameters, $\hat{v}_{ij}$ and v_{ij} the elements of $\hat{\boldsymbol{V}}$ and $\boldsymbol{V}$ and where $\boldsymbol{V} = \boldsymbol{N}^{-1}$ from (2.3) is obtained by inverting the matrix $\boldsymbol{N}$ of normal equations. The quantity d gives approximately the variance of the estimate $\hat{v}_{ij}$ averaged over all elements of $\hat{\boldsymbol{V}}$ and scaled by the maximum variance $\max(v_{ii})$ because of $|v_{ij}| \leq \max(v_{ii})$. By taking the square root, the scaled averaged standard deviation is obtained. Thus, d indicates the number of significant digits with respect to the maximum variance, i. e. the number of digits not distorted by errors of the estimate. For instance, $d = 1 \times 10^{-3}$ means on the average three significant digits in the estimated elements $\hat{v}_{ij}$ of $\hat{\boldsymbol{V}}$.

For computing d from (3.14), the inverse $\boldsymbol{V} = \boldsymbol{N}^{-1}$ is needed. It can only be assumed as given for test computations because it is the aim to determine $\boldsymbol{V}$ or to generate error samples for an error propagation. However, the estimated covariance matrix $\hat{\boldsymbol{V}}$ follows from (3.13) as the mean value of the samples. The variance of the mean can be easily obtained, if the samples are independent. This may be approximately assumed because of the applied blocking technique and the fact that only each s-th generated sample enters the estimation (3.13). The variances of the elements of $\hat{\boldsymbol{V}}$ from (3.13) are therefore computed as the variance of a mean value.

We go back to (2.16), which is identical with (3.13), where the difference

$$\hat{\boldsymbol{V}}_{ll} = \hat{\boldsymbol{V}}_{ll} - \boldsymbol{N}_{ll}^{-1} = \frac{1}{M}\sum_{k=1}^{M}\boldsymbol{\mu}_l^{(k)}\boldsymbol{\mu}_l^{(k)'}$$

and

$$\hat{\boldsymbol{V}}_{lj} = \frac{1}{M}\sum_{k=1}^{M}\boldsymbol{\mu}_l^{(k)}\boldsymbol{e}_j^{(k)'}$$

$$\text{for} \quad j,l \in \{1,\ldots,r\}\ ,\ j \neq l \tag{3.15}$$

are estimated as mean values. Let $\hat{\hat{v}}_{ij}$ be an element of $\hat{\hat{\boldsymbol{V}}}_{ll}$ for $l \in \{1,\ldots,r\}$. It is obtained from (3.15) by

$$\hat{\hat{v}}_{ij} = \frac{1}{M}\sum_{k=1}^{M}\mu_{li}^{(k)}\mu_{lj}^{(k)} \tag{3.16}$$

if $\mu_{li}^{(k)}$ denotes the i-th element in the vector $\boldsymbol{\mu}_l^{(k)}$. The variance $V(\hat{\hat{v}}_{ij})$ of the mean value $\hat{\hat{v}}_{ij}$ is given by, see for instance Koch (1999, p. 164)

$$V(\hat{\hat{v}}_{ij}) = \frac{1}{M(M-1)}\sum_{k=1}^{M}\left(\hat{\hat{v}}_{ij} - \mu_{li}^{(k)}\mu_{lj}^{(k)}\right)^2 . \tag{3.17}$$

Let now $\hat{v}_{mn}$ be an element of $\hat{\boldsymbol{V}}_{lj}$. Its variance follows accordingly from (3.15) by

$$V(\hat{v}_{mn}) = \frac{1}{M(M-1)} \sum_{k=1}^{M} \left(\hat{v}_{mn} - \mu_{lm}^{(k)} e_{jn}^{(k)}\right)^2 . \qquad (3.18)$$

The average variance is obtained by summing over the variances of all $u(u+1)/2$ estimated elements in $\hat{\boldsymbol{V}}$. If we scale in addition the average variance by the maximum value $\max(\hat{v}_{ii})$ of the variances of $\hat{\boldsymbol{V}}$ and take the square root, we find

$$\bar{d} = \left(\frac{2}{u(u+1)M(M-1)\max(\hat{v}_{ii})^2} \left[\sum_{i,j} \sum_{k=1}^{M} (\hat{\hat{v}}_{ij} - \mu_{li}^{(k)} \mu_{lj}^{(k)})^2 + \sum_{m,n} \sum_{k=1}^{M} (\hat{v}_{mn} - \mu_{lm}^{(k)} e_{jn}^{(k)})^2\right]\right)^{1/2} . \qquad (3.19)$$

This measure $\bar{d}$ of accuracy for estimating the covariance matrix approximates, according to its derivation, the Frobenius norm d in (3.14). The measure $\bar{d}$ gives like d on the average the number of significant digits of the elements of $\hat{\boldsymbol{V}}$ not distorted by errors of the estimate.

As explained for the estimate (2.16), there is a computationally more efficient way than determining $\bar{d}$ by (3.19). To see this, we rewrite the first sum of squares of residuals in (3.19) by (3.16)

$$\begin{aligned} &\sum_{k=1}^{M} \left(\hat{\hat{v}}_{ij} - \mu_{li}^{(k)} \mu_{lj}^{(k)}\right)^2 \\ &= M\hat{\hat{v}}_{ij}^2 - 2\hat{\hat{v}}_{ij} \sum_{k=1}^{M} \mu_{li}^{(k)} \mu_{lj}^{(k)} + \sum_{k=1}^{M} \mu_{li}^{(k)^2} \mu_{lj}^{(k)^2} \\ &= -\frac{1}{M}\left(\sum_{k=1}^{M} \mu_{li}^{(k)} \mu_{lj}^{(k)}\right)^2 + \sum_{k=1}^{M} \mu_{li}^{(k)^2} \mu_{lj}^{(k)^2} \end{aligned} \qquad (3.20)$$

and accordingly the second sum of squares of residuals in (3.19)

$$\begin{aligned} &\sum_{k=1}^{M} \left(\hat{v}_{mn} - \mu_{lm}^{(k)} e_{jn}^{(k)}\right)^2 \\ &= -\frac{1}{M}\left(\sum_{k=1}^{M} \mu_{lm}^{(k)} e_{jn}^{(k)}\right)^2 + \sum_{k=1}^{M} \mu_{lm}^{(k)^2} e_{jn}^{(k)^2} . \end{aligned} \qquad (3.21)$$

The first sum on the right-hand side of (3.20) is the square of the element with indices i, j in the sum of the first matrix product in (3.13) and the first sum of (3.21) the square of the elements with m, n in the sum for the second matrix product in (3.13), thus

$$\left(\sum_{k=1}^{M} \mu_{li}^{(k)} \mu_{lj}^{(k)}\right)^2 = \left(\sum_{k=1}^{M/(pz)} \boldsymbol{U}_{lz}^{(k)} \boldsymbol{U}_{lz}^{(k)'}\right)_{i,j}^2 \qquad (3.22)$$

$$\left(\sum_{k=1}^{M} \mu_{lm}^{(k)} e_{jn}^{(k)}\right)^2 = \left(\sum_{k=1}^{M/(pz)} \boldsymbol{U}_{lz}^{(k)} \boldsymbol{E}_{lz}^{(k)'}\right)_{m,n}^2 . \qquad (3.23)$$

If we replace the elements of the matrices $\boldsymbol{U}_{lz}^{(k)}$ and $\boldsymbol{E}_{lz}^{(k)}$ by their squares, which will be denoted by $\boldsymbol{U}_{lz}^{(k)^2}$ and $\boldsymbol{E}_{lz}^{(k)^2}$, we find the second sums in (3.20) and (3.21) as in (3.22) and (3.23) by

$$\sum_{k=1}^{M} \mu_{li}^{(k)^2} \mu_{lj}^{(k)^2} = \left(\sum_{k=1}^{M/(pz)} \boldsymbol{U}_{lz}^{(k)^2} \boldsymbol{U}_{lz}^{(k)^{2'}}\right)_{i,j} \qquad (3.24)$$

$$\sum_{k=1}^{M} \mu_{lm}^{(k)^2} e_{jn}^{(k)^2} = \left(\sum_{k=1}^{M/(pz)} \boldsymbol{U}_{lz}^{(k)^2} \boldsymbol{E}_{lz}^{(k)^{2'}}\right)_{m,n} . \qquad (3.25)$$

By summing over the $u(u+1)/2$ elements on the diagonal and above the diagonal of the matrices on the right-hand sides of (3.22) to (3.25), we obtain the sum of squares in (3.19). Thus, the sum of squares is efficiently computed by the sum of elements of matrices obtained by matrix multiplications.

Already during the process of generating the samples, one would like to get information about the accuracy of the estimates in order to adjust M if necessary. The measure $\bar{d}$ in (3.19) is therefore not only computed when all M samples have been generated, but already when in the p parallel runs of the Gibbs sampler in each processor of the cluster the s-th sample is obtained, that is for $k = 1, 2, \ldots, M/(pz)$. The sum of squares of each computation is saved and added so that with M samples $\bar{d}$ in (3.19) is obtained. However, this will be only an approximate measure $\bar{d}_a$ because for $k = 1, k = 2$ and so on only approximate values $\hat{\hat{v}}_{ija}$ and $\hat{v}_{mna}$ for the estimates $\hat{v}_{ij}$ and $\hat{v}_{mn}$ in (3.19) are available so that approximate sums of squares are computed and added, thus

$$\bar{d}_a = \left(\frac{2}{u(u+1)M(M-1)\max(\hat{v}_{iia})^2} \left[\sum_{i,j} \sum_{k=1}^{M} (\hat{\hat{v}}_{ija} - \mu_{li}^{(k)} \mu_{lj}^{(k)})^2 + \sum_{m,n} \sum_{k=1}^{M} (\hat{v}_{mna} - \mu_{lm}^{(k)} e_{jn}^{(k)})^2\right]\right)^{1/2} . \qquad (3.26)$$

Computations of examples have shown that nevertheless a good agreement exists between $\bar{d}_a$ from (3.26) and the Frobenius norm d from (3.14).

4 Test Computations

Examples have been computed by a cluster of 15 PCs of the Institute for Theoretical Geodesy of the University of Bonn. Each PC contains a Pentium 4 processor with 2.2 GHz, 1 GB of core storage and a 40 GB disk. One computer, the master, is connected with the network of the Institute and with the local network of the cluster consisting of the master and the 14 clients. The communication between master and clients is realized by a 100 MBit Ethernet.

For estimating the coefficients of a spherical harmonic expansion of the earth's gravity field, two matrices of normal equations were generated and used for the test computations, a CHAMP matrix and a GRACE matrix. The CHAMP matrix originates from an expansion of the harmonic coefficients up to degree and order 90 so that 8 277 unknown harmonic coefficients have to be estimated. The GRACE matrix is based on an expansion up to degree and order 140 so that 19 877 unknown parameters result.

The CHAMP and GRACE satellites are equipped with on-board GPS receivers of geodetic quality, allowing very precise orbit determination (POD). The GRACE satellite pair, in addition, measures the intersatellite range-rate using a high-precision K-band microwave instrumentation. Whereas a number of preliminary gravity models have already been published from CHAMP data, including error estimates, at the time of writing only very few results are available from GRACE. For our purpose, which is demonstrating the feasibility of our method under realistic conditions, we introduce some simplifications. This is allowed since we do not process actual data but are interested in error propagation. The normal equations for CHAMP have been assembled from about 80 days in 2002, assuming a truncation of the spherical harmonic series at $L = 90$. We have used CHAMP orbits from GFZ and assume that the energy conservation approach is applied to convert the POD information, which are cartesian x, y, z coordinates in a pseudo-inertial system, and corresponding velocities, into in-orbit residual potential values. The matrix of normal equations contains in this case products of associated Legendre functions evaluated in the satellite's positions. For building this normal equation matrix, it is not relevant what type of orbit, whether kinematic, reduced-dynamic or dynamic is used. For interpreting the corresponding covariance matrix one should bear in mind that we assume a standard deviation of 1 m^2/s^2 for the residual potential observations. This would correspond to a somewhat pessimistic estimate of what can be obtained from differentiating kinematic orbits. For the GRACE normal equation matrix we have used a simulated 30 days orbit (Ilk et al. 2003). The degree of truncation was set to $L = 140$ in this case. It is assumed that only the intersatellite range-rate is used as observation. According to the energy conservation principle, applied to a pair of satellites, this range-rate is proportional to the potential difference between the two satellite positions. Again, the normal equation matrix can be built up from products of associated Legendre functions. It is known that many corrections have to be made to use this approach for actual gravity recovery purposes, but for error propagation it is sufficient to retain the functional model in its simple form. For interpretation of the corresponding covariance matrix, it is important to know that we assume a standard deviation of 0.01 m^2/s^2 for the residual potential difference observations, corresponding to 1 μm/s for the range-rate observation. However, we have compared the error degree variance from our simulation with those from the first published GRACE gravity model (Reigber et al. 2003) and found agreement of better than one order of magnitude for the whole range of the model.

The harmonic coefficients are arranged as usual in a parameter estimation in satellite geodesy first by increasing order and then by increasing degree. This provides for most satellite observation techniques a block-dominant structure of the covariance matrix of the harmonic coefficients. This structure is well suited for the estimate (3.13) by conditioning which starts from a block diagonal covariance matrix. The blocking technique which was applied therefore collects the harmonic coefficients order by order. The size of the blocks is determined as described in the previous chapter by the size of the core storage of the master and the clients of the cluster. Given their core storage the maximum number of unknown parameters in one block turns out to be about 6 000.

The parallel computations in the cluster are organized as follows: the master reads the blocks of the normal equations defined by (2.11) on the diagonal and above the diagonal with a number of unknown harmonic coefficients not exceeding about 6 000. The master stores these blocks and sends them to the clients where they are also stored. To prepare for generating the samples, the matrices $\boldsymbol{N}_{ll}^{-1}\boldsymbol{N}_{lj}$ in (3.6) and the Cholesky factors $\boldsymbol{G}_l$ in (2.15) are computed by the master and the clients and stored on disks. These computations could also be organized in a parallel computing. However, it would necessitate much data transfer between the master and the clients which is slow in comparison to the computation in a node so that computer time would only be saved for a large number of blocks of normal equations.

The master then generates and distributes random values to each client to start the parallel runs of the Gibbs sampler at the master and the clients. Whenever p parallel runs have been finished, each computer in the cluster computes by (3.26) the measure $\bar{d}_a$ of accuracy of the estimation. For error propagations, whenever the clients finish the p runs, they send the error vectors used to estimate the covariance matrix to the master where they are stored. When M samples have been generated, the master collects the sum of the squares of the residuals of the clients to compute by (3.26) the measure $\bar{d}_a$ of accuracy based on all M samples. If the whole covariance matrix is needed, the sums in (3.13) of each client are sent to the master where they are added to give the final estimate of the covariance matrix. If the variances of the unknown parameters have to be known, only the sums for the blocks on the diagonal are given to the master. As test computations have shown, the parallel computing reduces the time of the computations of a single computer by a factor of about 1/7.4 if only the error vectors are stored. The factor 1/7.4 does not lie closer to 1/15 because of the computations discussed above to prepare according to (3.6) generating the samples. These computations are performed by the master and all clients so that during this time no computer time is saved by the parallel

processing. The same holds true for the time needed by the master to distribute data to the clients and to collect the results.

The CHAMP matrix was divided into four blocks with the first block $\boldsymbol{N}_{11}$ containing 6 021 unknown harmonic coefficients for the orders $m = 0$ to $m = 43$. The last block $\boldsymbol{N}_{22}$ contains the rest of the 8 277 unknown parameters for the orders up to 90. Gundlich, Koch and Kusche (2003) found out that the number M of samples does not have to exceed one tenth of the number of unknown parameters to ensure a sufficient accuracy for the estimation which was considered to be three significant digits. This was confirmed for the CHAMP matrix. With $M = 825$ samples, the approximate measure $\bar{d}_a$ of accuracy from (3.26) was computed to be $\bar{d}_a = 1.4 \times 10^{-3}$. This is an excellent approximation of the Frobenius norm $d = 1.5 \times 10^{-3}$ from (3.14) with the inverse $\boldsymbol{N}^{-1}$ for the 8 277 unknown parameters obtained by LAPACK routines.

The $M = 825$ samples are obtained from the 15 computers in the cluster by 55 parallel runs of the Gibbs sampler. With these 55 samples, $\bar{d}_a$ from (3.26) is determined by each computer to be approximately $\bar{d}_a = 5.0 \times 10^{-3}$. This result gives a good indication what can be expected with 825 samples because of $5.0 \times 10^{-3}/(825/55)^{1/2} = 1.3 \times 10^{-3}$. Every fifth generated sample was taken to estimate the covariance matrix according to (3.13) because the maximum correlation between consecutive samples equals 0.668 and drops to 0.141 for samples which are five samples apart. Accordingly, a burn-in phase of ten samples was chosen at the beginning of each of the 55 parallel runs of the Gibbs sampler.

The 825 error vectors applied to estimate the inverse of the CHAMP matrix from (3.13) were also used for an error propagation according to (2.10). They represent random realizations of the errors of spherical harmonic coefficients, as they can be derived from CHAMP data by applying the energy conservation principle to pre-processed satellite orbits, as described earlier. It is common to propagate these errors into those of geoid heights or gravity anomalies in a gridded representation in the space domain. We have $f(\boldsymbol{e}) = \boldsymbol{F}\boldsymbol{e}$ and $E(f(\boldsymbol{e})) = \boldsymbol{0}$ in (2.10) with the matrix $\boldsymbol{F}$ containing in case of geoid heights as elements $R\cos(m\lambda)P_{lm}(\sin\phi)$ and $R\sin(m\lambda)P_{lm}(\sin\phi)$ evaluated on the λ, ϕ-grid with λ being the longitude and ϕ the latitude. For gravity anomalies, the entries of $\boldsymbol{F}$ are $\frac{GM}{R^2}\cos(m\lambda)P_{lm}(\sin\phi)$ and $\frac{GM}{R^2}\sin(m\lambda)P_{lm}(\sin\phi)$, where R is the earth's mean radius, GM the gravitational constant times the earth's mass, and P_{lm} are the fully normalized Legendre functions. Fig. 1 shows on the left-hand side the standard deviations of geoid heights in [m], computed on a $1^\circ \times 1^\circ$ grid, from 10, 196, and 825 error vectors. On the right-hand side, the corresponding standard deviations for gravity anomalies in [mgal] are given. We propagate only the committed errors up to a spherical harmonic truncation degree of 70, whereas the solution itself was computed to harmonic degree of 90. This is because the first real CHAMP solutions confirm that almost no signal power is included above this degree. It is obvious that already after about 200 samples, that is when the ratio of samples to unknown parameters appears roughly as 2%, we obtain a good picture of the propagated standard deviations in the space domain. One should note that the stripe pattern in the figures is to a large extend caused by the groundtrack distribution of the satellite and therefore does not represent an artefact. For 825 samples, the accuracy of the standard deviations surpasses 1 cm for the geoid undulations and 1 mgal for the gravity anomalies, which means three significant digits.

The GRACE matrix was split up into 16 blocks. The first block $\boldsymbol{N}_{11}$ on the diagonal contains 6 071 harmonic coefficients from order $m = 0$ up to order $m = 23$. The next block $\boldsymbol{N}_{22}$ on the diagonal extends from order $m = 24$ to $m = 52$ with the unknown parameters 6 072 to 12 045 if numbered consecutively. The third block $\boldsymbol{N}_{33}$ collects the harmonic coefficients from order $m = 53$ to $m = 98$ with the parameters 12 046 to 18 071, and the last block $\boldsymbol{N}_{44}$ has the parameters for $m = 99$ to $m = 140$ from 18 072 to 19 877. With $M = 1995$ samples, whose number equals about one tenth of the number of unknown parameters, the approximate measure $\bar{d}_a$ of accuracy from (3.26) was $\bar{d}_a = 2.2 \times 10^{-5}$, which means more than four significant digits for the estimates. The $M = 1995$ samples were obtained by the 15 computers in the cluster through 133 parallel runs of the Gibbs sampler. With these 133 samples, $\bar{d}_a$ from (3.26) is determined by each computer to be approximately $\bar{d}_a = 7.7 \times 10^{-5}$. This result indicates what can be expected with $M = 1995$ samples because of $7.7 \times 10^{-5}/(1995/133)^{1/2} = 2.0 \times 10^{-5}$.

The Frobenius norm d in (3.14) and the correlation between samples could not be computed because the GRACE matrix did not fit into the core storage of the computers which were available so that the inverse could not be computed. As for the CHAMP matrix, every fifth generated sample was used to estimate the covariance matrix by (3.13) and a burn-in phase of ten samples was allowed at the beginning of each of the 133 parallel runs of the Gibbs sampler.

The 1995 error vectors used for estimating the inverse of the GRACE matrix by (3.13) were also applied for an error propagation according to (2.10). These error vectors represent random realizations of the errors of spherical harmonic coefficients, as they were derived from GRACE data following our set-up described earlier in this chapter. Again these errors are propagated into corrresponding errors of geoid heights and gravity anomalies on a global grid. We have $f(\boldsymbol{e}) = \boldsymbol{F}\boldsymbol{e}$ and $E(f(\boldsymbol{e})) = \boldsymbol{0}$ in (2.10) with the matrices $\boldsymbol{F}$ containing the same elements as given above for the error propagation of the CHAMP data. Fig. 2 shows on the left-hand side the standard deviations of geoid heights in [m] computed on a $1^\circ \times 1^\circ$ grid from 10, 415, and 1995 error vectors. On the right-hand side, the corresponding standard deviations for gravity anomalies in [mgal] are given. Again, after about 400 samples,

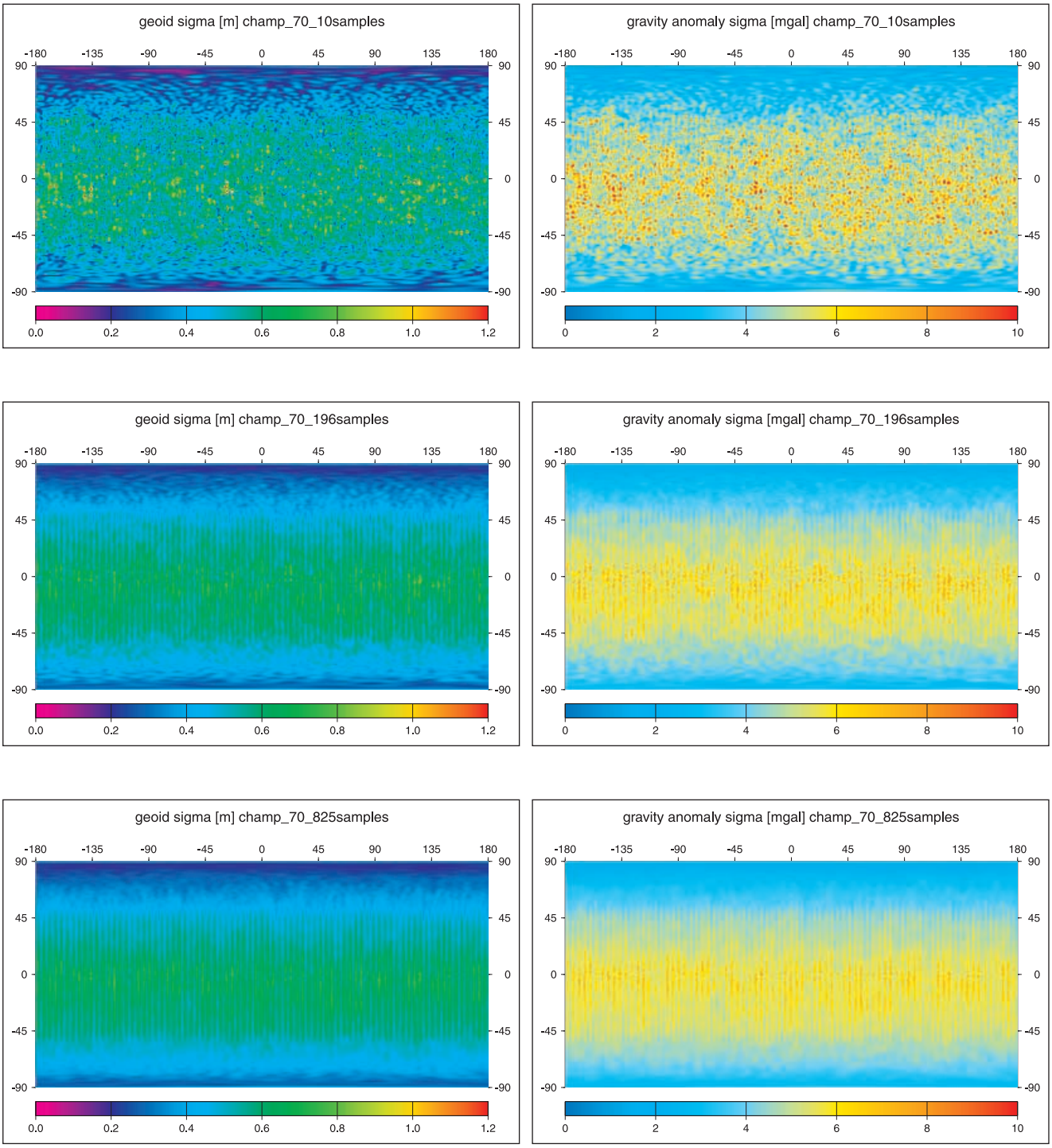

Fig. 1: Propagated standard deviations for geoid heights [m] (left) and gravity anomalies [mgal] (right) from 80 days of CHAMP data with L = 70 (solved 90). Upper row after 10 samples, middle row after 196 samples, lower row after 825 samples.

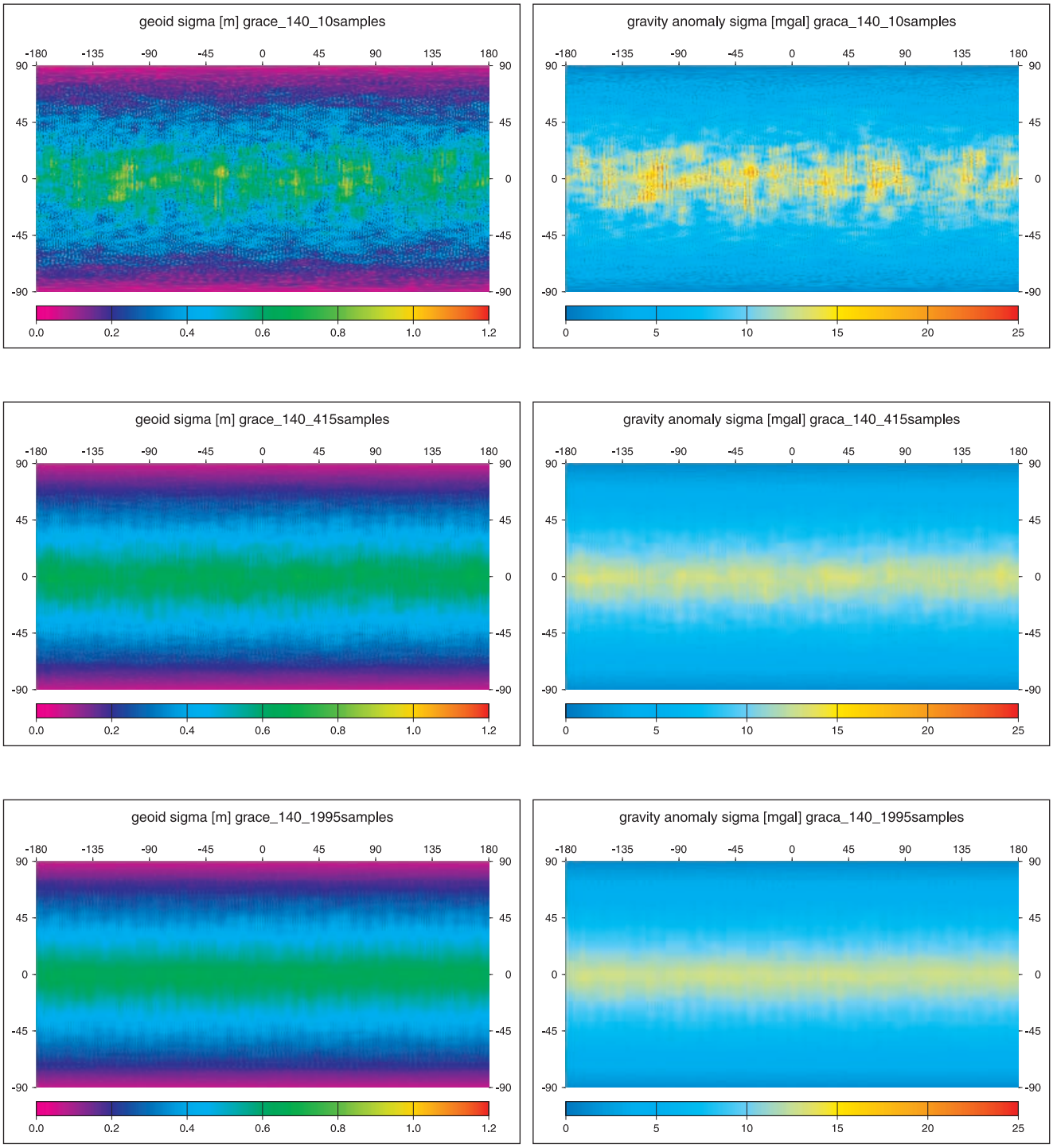

Fig. 2: Propagated standard deviations for geoid heights [m] (left) and gravity anomalies [mgal] (right) from 30 days of simulated GRACE data with L = 140. Upper row after 10 samples, middle row after 415 samples, lower row after 1995 samples.

that is when the ratio of samples to unknown parameters approaches roughly 2%, we get already a very good approximation of the propagated standard deviations in the space domain. As for the CHAMP results, the accuracy gained with 1995 error vectors are three significant digits. It should be emphasized that the figures represent error propagations from the full variance covariance matrix although this matrix was never stored nor completely built up.

The time needed to invert the CHAMP matrix for 8 277 unknown parameters and GRACE matrices for 12 045, 18 071 and 19 877 unknown harmonic coefficients and to store the error vectors applied for the inversion was used to estimate the increase of computer time as a function of the increase of the unknown parameters. If μ denotes the ratio of the increase of unknown parameters, one could expect that the computer time increases by a factor of about μ^3 because mostly matrix multiplications are needed for generating the samples. However, the covariance matrix is symmetric and only the elements on the diagonal and above the diagonal are estimated. In addition, the blocks N_{ll} on the diagonal, which need to be inverted, are symmetric. Our test computations have therefore shown that the factor by which the computer time increases is only about $0.7\mu^3$. Applying this factor, the time to invert a matrix of normal equations for 60 000 unknown harmonic coefficients and to store about 6 000 error vectors for the error propagation would take less than two days of parallel computing on the cluster of computers described above.

5 Conclusions

In our test computations we estimated covariance matrices for 8 277 harmonic coefficients of the earth's gravity field from CHAMP data and for 19 877 harmonic coefficients from generated GRACE data. In both cases it turned out that the number of generated error vectors for the harmonic coefficients does not have to exeed one tenth of the number of coefficients to ensure an accuracy of the elements of the estimated covariance matrix and of the errors of geoid undulations and gravity anomalies from the error propagation better than three significant figures. Thus, for error propagations, we do not have to handle the $u \times (u+1)/2$ elements of the covariance matrix, but only the $u \times u/10$ components of the error vectors, which is about one fifth. For the cluster of 15 computers applied in the test computations, inverting a matrix of normal equations for 60 000 unknown harmonic coefficients and storing 6 000 error vectors for error propagations will take less than two days. Investigations are under way to reformulate the parallel Gibbs sampler for problems where an explicit assembly of the matrices of normal equations is avoided, i. e. replaced by using the design matrix and its transpose.

References

Anderson E., Bai Z., Bischof C., Blackford S., Demmel J., Dongarra J., Du Croz J., Greenbaum A., Hammarling S., McKenney A. and Sorensen D.: LAPACK User's Guide. SIAM, Philadelphia, 1999.

Blackford L. S., Choi J., Cleary A., D'Azevedo E., Demmel J., Dhillon I., Dongarra J., Hammarling S., Henry G., Petitet A., Stanley K., Walker D. and Whaley R. C.: ScaLAPACK User's Guide. SIAM, Philadelphia, 1997.

Demmel J. W.: Applied Numerical Linear Algebra. SIAM, Philadelphia, 1997.

ESA: Gravity Field and Steady-State Ocean Circulation Mission. ESA Publications Division, ESA SP-1233(1), ESTEC, Noordwijk, 1999.

Geman S. and Geman D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans Pattern Anal Machine Intell, PAMI-6: 721–741, 1984.

Gropp W., Lusk E. and Skjellum A.: Using MPI. MIT Press, Cambridge, 1999a.

Gropp W., Lusk E. and Thakur R.: Using MPI-2. MIT Press, Cambridge, 1999b.

Gundlich B., Koch K. R. and Kusche J.: Gibbs sampler for computing and propagating large covariance matrices. J. Geodesy, 77: 514–528, 2003.

Harville D. A.: Use of the Gibbs sampler to invert large, possibly sparse, positive definite matrices. Linear Algebra and its Applications, 289: 203–224, 1999.

Ilk K. H., Visser P. and Kusche J.: Final Report, Special Commission 7, Satellite Gravity Field Missions. Travaux IAG, Vol. 32, General and technical reports 1999–2003, Sapporo, Japan, 2003.

Koch K. R.: Parameter Estimation and Hypothesis Testing in Linear Models, 2nd Ed. Springer, Berlin, 1999.

Koch K. R.: Einführung in die Bayes-Statistik. Springer, Berlin, 2000.

Liu J. S.: Monte Carlo Strategies in Scientific Computing. Springer, Berlin, 2001.

Plank G.: Implementation of the pcgma-package on massive parallel systems. In: ESA-Project »From Eötvös to mGal«, Final-Report. ESA/ESTEC Contract No. 14287/00/NL/DC, 183–216, 2002.

Reigber Ch., Schmidt R., Flechtner F., König R., Meyer U., Neumayer K. H., Schwintzer P. and Zhu S. Y.: First EIGEN gravity field model based on GRACE mission data only. Geophysical Research Letter, in preparation, 2003.

Smith A. F. M. and Roberts G. O.: Bayesian computation via the Gibbs sampler and related Markov Chain Monte Carlo methods. J Royal Statist Society, B 55: 3–23, 1993.

The LAM/MPI Team: LAM/MPI user's guide, version 7.0, http://www.lam-mpi.org, 2003.

Whaley R. C., Petitet A. and Dongarra J. J.: Automated empirical optimization of software and the ATLAS project. http://math-atlas.sourceforge.net, 2000.

Authors' adresses

Prof. Dr.-Ing. Dr.-Ing. E. h. mult. Karl-Rudolf Koch (em.)
Institute for Theoretical Geodesy, University of Bonn
Nussallee 17, 53115 Bonn, Germany
koch@theor.geod.uni-bonn.de

Ass. Prof. Dr.-Ing. Jürgen Kusche
Physical, Geometrical, and Space Geodesy, Delft University of Technology
Kluyverweg 1, P. O. Box 5058, 2600 GB Delft, The Netherlands
j.kusche@citg.tudelft.nl

Dipl.-Ing. Christian Boxhammer
Institute for Theoretical Geodesy, University of Bonn
Nussallee 17, 53115 Bonn, Germany
box@mail.geod.uni-bonn.de

Dr.-Ing. Brigitte Gundlich
Mathematical Geodesy and Positioning, Delft University of Technology
Kluyverweg 1, P. O. Box 5058, 2600 GB Delft, The Netherlands
b.gundlich@geo.tudelft.nl