HPC tools for solving accurately the large dense linear least squares problems arising in gravity field computations.

Marc Baboulin*

Abstract

Our work is related to the GOCE[1] mission that strives for a very accurate model of the Earth's gravity field and of the Geoid. These model parameters are estimated via an incremental linear least squares problem that involves a huge quantity of data ($90,000$ parameters and several million observations). We present a parallel distributed solver based on PBLAS [7] and ScaLAPACK [6] kernel routines that enables us to update the solution with new GOCE observations by using a QR factorization algorithm. It uses a recently defined distributed packed format [2] that minimizes the storage. This solver also implements a condition number estimate and a Kaula [11] regularization technique that makes the computation more stable. We provide performance analysis and first results.

Keywords: gravity field computation, scientific computing, dense linear algebra, parallel distributed algorithms, ScaLAPACK, QR factorization.

1 Physical problem

The GOCE satellite [3, 12] is scheduled for launch in 2007 and will provide a model of the Earth's gravity field and of the Geoid with an unprecedented accuracy. It will have applications in many scientific areas such as solid-Earth physics, geodesy, oceanography, glaciology and climate change. The model of the gravitational potential will be represented by about $90,000$ spherical harmonic coefficients up to a degree of 300. In terms of accuracy, the GOCE mission objectives are

- to determine the gravity-field anomalies with an accuracy of 1 mGal (where $1mGal = 10^{-5}m/s^2$),
- to determine the geoid with an accuracy of 1-2 cm,
- to achieve the above at a spatial resolution better than 100 km.

The gravity gradients are measured by satellite gravity gradiometry, combined with satellite-to-satellite tracking using GPS (SST-hl). The estimation of the Earth's gravity field using GOCE observations is a numerical and computational challenge. The numerical difficulty comes from the observation noise inherent in instruments used for measurement (gradiometers). Numerical instability may also result from missing observations at the poles, due to the non-polar orbit of the satellite.
The computational task is quite challenging because of the huge quantity of daily accumulated data (about $90,000$ parameters and several million observations) and because of the coupling of the parameters resulting in completely dense matrices.
Following [5], the Earth's gravitational potential V is expressed in spherical coordinates (r, θ, λ) by:

$$V(r,\theta,\lambda) = \frac{GM}{R} \sum_{l=0}^{l_{max}} \left(\frac{R}{r}\right)^{l+1} \sum_{m=0}^{l} \overline{P}_{lm}(\cos\theta) \left[\overline{C}_{lm}\cos m\lambda + \overline{S}_{lm}\sin m\lambda\right] \tag{1}$$

*CERFACS, 42 av. Gaspard Coriolis, 31057 Toulouse Cedex, France. Email : baboulin@cerfacs.fr

[1]Gravity field and steady-state Ocean Circulation Explorer - European Space Agency

where G is the gravitational constant, M is the Earth's mass, R is the Earth's reference radius, the $\overline{P}_{lm}$ represent the fully normalized Legendre functions of degree l and order m and $\overline{C}_{lm}$,$\overline{S}_{lm}$ are the corresponding normalized harmonic coefficients. In the above expression, we have $|m| \leq l \leq l_{max}$ with $l_{max} \simeq 300$ (about $90,000$ unknowns). For the previous missions CHAMP[2] and GRACE[3], we had respectively $l_{max} \simeq 120$ (about $15,000$ unknowns) and $l_{max} \simeq 150$ (about $23,000$ unknowns). We point out that the number of unknown parameters is expressed by

$$n = (l_{max} + 1)^2.$$

We have to compute the harmonic coefficients $\overline{C}_{lm}$ and $\overline{S}_{lm}$ as accurately as possible.

2 Numerical method for gravity field calculations

The gravity field parameters are computed at CNES[4] using the orbit determination software GINS [4]. Measurements which are a function of a satellite position and/or velocity are taken into account. These observations are obtained via ground stations (Laser, Doppler) or other satellites (GPS). Then we aim to minimize the difference between the measurements and the corresponding quantities evaluated from the computed orbit by adjusting given parameters (here the gravity field coefficients). This yields to a nonlinear least squares problem that can be solved via a Gauss-Newton algorithm [10] by solving successively linear least squares problems (LLSP) of the form

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2 \tag{2}$$

where $b \in \mathbb{R}^m$ is the observation vector and $A \in \mathbb{R}^{m \times n}$ is the data matrix (Jacobian of a nonlinear least squares problem). Each row of A and b corresponds to one observation, these observations being collected periodically. The LLSP is currently solved at CNES by the normal equations method where we solve the linear system of equations $A^T Ax = A^T b$.
A parallel distributed solver that performs the assembly of the normal equations and computes a solution using a Cholesky factorization is described in [1]. We propose here a more reliable way of solving (2) that consists of using a QR factorization that utilizes Householder transformations. We can find in [9] a comprehensive presentation of the algorithms available for computing and updating a QR factorization.
It is appropriate to update the previous QR factorization or at least the R factor with the newly collected data rather than computing a whole QR factorization involving original data combined with new data. Such an incremental algorithm is more efficient in terms of computational cost than performing the QR factorization on the whole matrix.

Let $\mathcal{A}_N$ and $\mathcal{B}_N$ be respectively the cumulated parameter matrix and observation vector up to date N. Let A_N and b_N be respectively the data matrix and observation vector that has been collected at date N. If we denote by R_N the R-factor obtained at date N, then we observe that the QR factorization of $\mathcal{A}_{N+1} = \begin{pmatrix} \mathcal{A}_N \\ A_{N+1} \end{pmatrix}$ produces the same upper triangular factor as does the factorization of $\begin{pmatrix} R_N \\ A_{N+1} \end{pmatrix}$ i.e R_{N+1}.
Furthermore, the storage of the Householder vectors can be avoided by appending the observation vector b_N to the matrix to be factorized and overwriting this vector with the $(n+1)$-th column of the so-obtained triangular factor.
The result is that the updating of the R-factor at date $N+1$ is done by performing the QR factorization of $\begin{pmatrix} R_N & \tilde{\mathcal{B}}_N \\ A_{N+1} & b_{N+1} \end{pmatrix}$ where $\tilde{\mathcal{B}}_N$ contains the updated values of $\mathcal{B}_K$ $(K \leq N)$ resulting

[2]CHAllenging Minisatellite Payload for Geophysical Research an Application, GFZ, launched July 2000
[3]Gravity Recovery and Climate Experiment, NASA, launched March 2002
[4]Centre National d'Etudes Spatiales, Toulouse, France

from the N previous QR factorizations. This enables us to obtain the upper triangular matrix $\begin{pmatrix} R_{N+1} & \tilde{\mathcal{B}}_{N+1} \end{pmatrix}$ and the solution x_{N+1} is computed by solving $R_{N+1}x_{N+1} = \mathcal{Z}_{N+1}$ where $\mathcal{Z}_{N+1}$ contains the first n elements of $\tilde{\mathcal{B}}_{N+1}$.

3 A parallel distributed solver for GOCE calculations

3.1 Parallel implementation

An efficient out-of core implementation for updating a QR factorization is described in [9]. Here, for faster computation of a partial solution or of the covariance, we deliberately choose to keep the R factor **in-core** by storing R compactly.

We suppose that the R factor is partitioned into distributed blocks $\begin{pmatrix} B_1 & B_2 & B_3 \\ 0 & B_4 & B_5 \\ 0 & 0 & B_6 \end{pmatrix}$. We denote by b the size of the distributed blocks by N_b the number of block rows in R. Then we store R using the distributed packed format defined in [2] as $\begin{bmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 \end{bmatrix}$ that we also denote by $B_{1:6}$.
The new observations are stored in a block matrix $L_{1:N_b}$ that contains $N_b.b$ columns and we first assume that L contains b rows. The updating of R is obtained by successively performing the QR factorization of each block row of R with L, as described below. At the first step, we factor:

$$\begin{array}{|c|}\hline B_{1:3} \\ \hline L_{1:3} \\ \hline \end{array} \longrightarrow \begin{array}{|c|}\hline \widetilde{B}_{1:3} \\ \hline \widetilde{L}_{1:3} \\ \hline \end{array}$$

and we advance the updating of the R factor as follows:

$$\begin{array}{|c|}\hline B_{4:5} \\ \hline \widetilde{L}_{2:3} \\ \hline \end{array} \longrightarrow \begin{array}{|c|}\hline \widetilde{B}_{4:5} \\ \hline \overline{L}_{2:3} \\ \hline \end{array}$$

and so on until completion.
Let now consider the work array $C = \begin{bmatrix} B_{j:j+N_b-i} \\ \widetilde{L}_{i:N_b} \end{bmatrix}$ where j is the index of the i-th diagonal block in the packed structure. C contains $2b$ rows and $(N_b - i + 1).b$ columns.

At step i in the R updating, we apply to the matrix C the ScaLAPACK [6] routine PDGEQRF that has been modified in order to stop the factorization after the first b columns. From [8, p. 213 and 225], the computational cost is about $3bn^2$ (if $n \gg b$). Our algorithm does not take into account the upper triangular structure of $B_{j:j}$. As it will be confirmed on experiments, this can be compensated by storing more data into L and thus by considering a block matrix C containing more that $2b$ rows. As a result, the number of floating-point operations will also decrease.
The initialization of the R factor has been implemented by starting with $R = 0$ and then by successively updating the previous rows by a new one until we processed the N_b block rows of R.

3.2 Performance results

All the following experiments have been performed on the IBM pSeries 690 (2 nodes of 32 processors Power-4/1.7 GHz and 64 Gbytes memory per node). We used the PBLAS [7] and ScaLAPACK libraries provided by the vendor (in particular the Pessl library). Let n_L be the number of rows in the matrix L that contains the new observations for updating the QR factorization. In Table 1, we update a 25600×25600 matrix R by 51200 new observations and n_L varies from 512 to 25600. As expected, the number of operations decreases as n_L increases. This gain in operations is evaluated by computing the ratio between the operations involved in the updating of R and the operations required in a QR factorization of the 76800×25600 matrix containing the original data and the new observations. Then if n_L increases, the factorization time decreases but the performance is stable (close to the peak performance of the ScaLAPACK routine PDGEQRF). Choosing the best size for L corresponds to finding a compromise between performance and storage since large L demands more storage.

Number of rows in L	512	1024	2048	5120	10240	12800	25600
Storage (Gbytes)	0.72	0.75	0.80	0.96	1.22	1.35	2.00
Flops overhead (vs ScaLAPACK)	1.50	1.31	1.22	1.16	1.14	1.14	1.13
Factorization time (sec)	7577	5824	5255	5077	5001	4894	4981
Performance (Gflops)	3.33	3.61	3.59	3.47	3.44	3.50	3.40

Table 1: Updating of a 25600×25600 R factor by 51200 new observations (1×4 procs).

References

[1] M. Baboulin, L. Giraud, and S. Gratton, *A parallel distributed solver for large dense symmetric systems: applications to geodesy and electromagnetism problems*, Int. J. of High Performance Computing Applications **19** (2005), no. 4, 353–363.

[2] M. Baboulin, L. Giraud, S. Gratton, and J. Langou, *A distributed packed storage for large parallel calculations*, Technical Report TR/PA/05/30, CERFACS, Toulouse, France, 2005.

[3] G. Balmino, *The European GOCE Gravity Consortium (EGG-C)*, (April 2001), 7–12, Proceedings of the International GOCE User Workshop.

[4] G. Balmino, S. Bruinsma, and J-C. Marty, *Numerical simulation of the gravity field recovery from GOCE mission data*, (March 8-10, 2004), Proceedings of the Second International GOCE User Workshop “GOCE, The Geoid and Oceanography”, ESA-ESRIN, Frascati, Italy.

[5] G. Balmino, A. Cazenave, A. Comolet-Tirman, J. C. Husson, and M. Lefebvre, *Cours de géodésie dynamique et spatiale*, ENSTA, 1982.

[6] L. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. Whaley, *ScaLAPACK user's guide*, SIAM, 1997.

[7] J. Choi, J. Dongarra, L. Ostrouchov, A. Petitet, D. Walker, and R. Whaley, *A proposal for a set of parallel basic linear algebra subprograms*, Tech. report, 1995, LAPACK Working Note 100.

[8] G. H. Golub and C. F. van Loan, *Matrix computations*, The Johns Hopkins University Press, 1996, Third edition.

[9] B. Gunter and R. van de Geijn, *Parallel out-of-core computation and updating of the QR factorization*, ACM Trans. Math. Softw. **31** (2005), no. 1, 60–78.

[10] J. E. Dennis Jr. and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, SIAM, 1996.

[11] W. M. Kaula, *Theory of satellite geodesy*, Blaisdell Press, Waltham, Mass., 1966.

[12] H. Sünkel, *From Eötvös to milligal+, Final Report*, ESA/ESTEC Contract No. 13392/98/NL/GD, Graz University of Technology, 2000.