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

Marc Baboulin

CERFACS
Toulouse, France
joint work with Centre National d'Etudes Spatiales, France
in collaboration with Julien Langou (Univ. Tennessee)

PARA'06 - Umeå
June 19, 2006

Outline

1. Gravity field computation: GOCE

Outline

1. Gravity field computation: GOCE
2. Parallel implementation of QR updating

Outline

1. Gravity field computation: GOCE
2. Parallel implementation of QR updating
3. Experimental results

Outline

1. Gravity field computation: GOCE
2. Parallel implementation of QR updating
3. Experimental results
4. Conclusion

Outline

GOCE mission

GOCE: ESA project (Gravity field and steady-state Ocean Circulation Explorer)

- satellite scheduled for launch in 2007
- will provide a model of the Earth's gravity field and of the geoid with an unprecedented accuracy
- follows the CHAMP (2000) and GRACE (2002) missions

GOCE mission

- gravitational potential of the Earth (spherical coordinates) [Balmino et al.,82] :

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

- G = gravitational constant, M = Earth's mass, R = Earth's reference radius, $l_{max} \simeq 300$.
- objective: determine $\overline{C}_{lm}$ and $\overline{S}_{lm}$ as accurately as possible number of unknowns $n = (l_{max} + 1)^2 \simeq 90,000$

numerical and computational challenge

Gravity coefficients computation

1. dynamics: $\ddot{r} = f(r, \dot{r}, \gamma, t), \quad r(t_0) = r_0, \quad \dot{r}(t_0) = r_0'$
2. measurements: $Q_j = h(r, \dot{r}, \gamma, t_j) + \varepsilon_j = h_j(\gamma) + \varepsilon_j$
3. nonlinear LSP: $\min_\gamma \sum_{j=1}^{m} \|\widetilde{Q_j} - h_j(\gamma)\|_2^2$
4. solved by Gauss-Newton algorithm and computation of

$$A = h'(\gamma) = \begin{pmatrix} \frac{\partial Q_1}{\partial \gamma_1} & \cdots & \frac{\partial Q_1}{\partial \gamma_n} \\ \vdots & & \vdots \\ \frac{\partial Q_m}{\partial \gamma_1} & \cdots & \frac{\partial Q_m}{\partial \gamma_n} \end{pmatrix}, \quad b = \begin{pmatrix} \widetilde{Q_1} - h_1(\gamma) \\ \vdots \\ \widetilde{Q_m} - h_m(\gamma) \end{pmatrix}$$

5. LLSP $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2$ where $x = \Delta\gamma$.

Numerical methods for GOCE

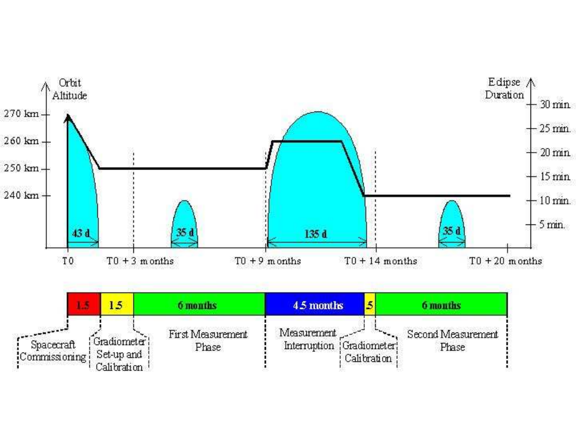
After accumulating a sufficient number of observations and/or by using regularization techniques, A is full column rank (incremental LLSP).

A is several $10^6 \times 90,000$ dense (needs 6 months of measurements)

- iterative methods: slow convergence, accuracy ?
- direct methods
  - normal equations method (e.g CNES)
  - orthogonal transformations (e.g out-of-core QR, GRACE)

computational cost, better accuracy

Incremental LLSP



GOCE mission profile (early 2007 - end 2008).

Update of normal equations

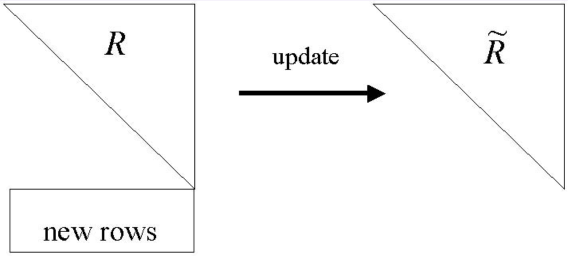
- $A^TA \leftarrow A^TA + (\text{new rows})^T(\text{new rows})$
- Kaula regularization:
  add to A^TA a diagonal matrix $D = diag(0, \cdots, 0, \alpha, \cdots, \alpha)$
  where $\alpha \propto 10^{-5}/l_{max}^2$
- already implemented in packed storage and currently used at CNES

Update of QR factorization

- QR factorization of $\begin{pmatrix} A \\ \text{new rows} \end{pmatrix}$ produces the same upper triangular factor as does the factorization of $\begin{pmatrix} R \\ \text{new rows} \end{pmatrix}$
- Kaula: QR factorization of $\begin{pmatrix} R \\ D \end{pmatrix}$

Outline

QR Updating



Updating of the R factor in an incremental QR factorization.

out-of-core algorithm in [Gunter et al.,05]

here we want the R factor to be kept in-core

Packed storage scheme

- ScaLAPACK 2-D block cyclic distribution
  block size s, $p \times q$ process grid
- **distributed packed format**:
  R factor partitioned into square blocks of size b
  $b \propto lcm(p,q) \times s$

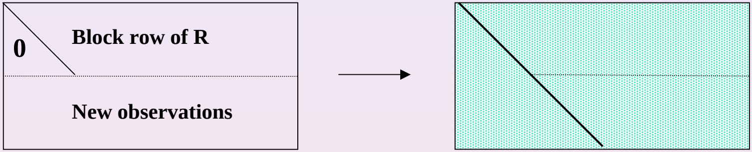
$$\begin{pmatrix} B_1 & B_2 & B_3 \\ 0 & B_4 & B_5 \\ 0 & 0 & B_6 \end{pmatrix} \rightarrow \begin{bmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 \end{bmatrix}$$

- a unique ScaLAPACK array for the packed structure (possibly a specific array for storing diagonal triangles)
- use of ScaLAPACK routines PDGEQRF (QR factorization) and PDORMQR (multiplication by Q^T)

General algorithm

- new observations are stored in a block matrix L that contains n columns and b rows
- first we factor $\begin{bmatrix} B_{1:3} \\ \hline L_{1:3} \end{bmatrix} \longrightarrow \begin{bmatrix} \widetilde{B}_{1:3} \\ \hline \widetilde{L}_{1:3} \end{bmatrix}$
- then $\begin{bmatrix} B_{4:5} \\ \hline \widetilde{L}_{2:3} \end{bmatrix} \longrightarrow \begin{bmatrix} \widetilde{B}_{4:5} \\ \hline \overline{L}_{2:3} \end{bmatrix}$,
- ...until completion.

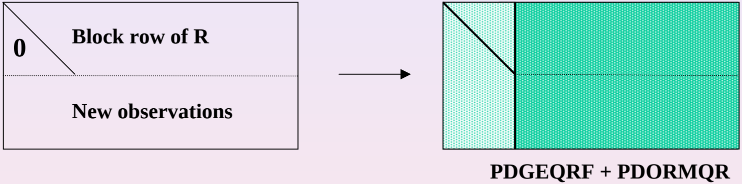
Implementation of R factor updating



PDGEQRF

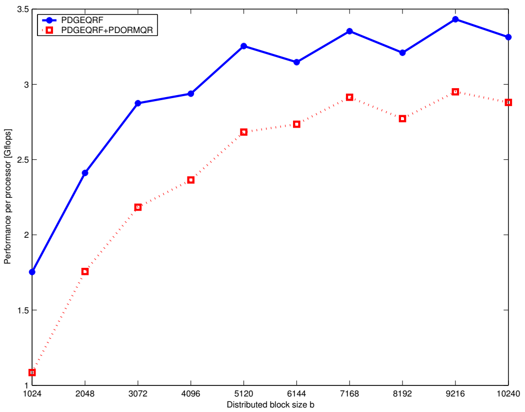
- QR factorization applied to the whole matrix
- cost: $\sim\ 4bn^2$ (if $n \gg b$)
- too many operations

Implementation of R factor updating



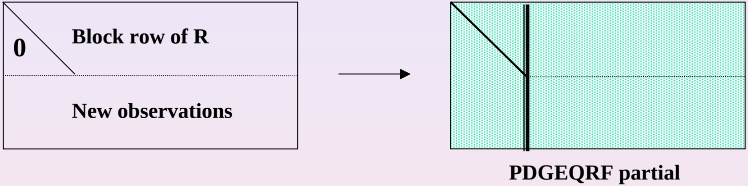
- QR factorization of the first b columns followed by the multiply by Q^T
- cost: $\sim\ 3bn^2$ (if $n \gg b$)
- poor performance of the PDORMQR routine

Tuning



QR factorization of a $b \times 2b$ matrix (4 processors of IBM pSeries 690).

Implementation of R factor updating



- QR factorization that stops after the first b columns
- cost: $\sim 3bn^2$ (if $n \gg b$)
- exploits the good performance of PDGEQRF
- does not take into account the upper triangular structure of the block row of R. This can be compensated by storing more new rows in the work array to be factored.

Outline

Performance results

n procs	10240 1	14336 1×2	20480 1×4	28672 2×4	40960 2×8	61440 4×8	81920 4×16
Our solver	2.47	3.02	3.30	2.87	2.89	2.80	2.37
PDGEQRF	3.50	3.36	3.20	3.25	2.93	2.83	2.63

Performance of a complete QR factorization (Gflops)
IBM pSeries 690.

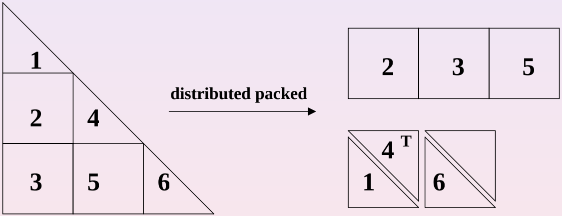
Performance results

Nb of new rows	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	1.50	1.31	1.22	1.16	1.14	1.14	1.13
Facto. time (sec)	7577	5824	5255	5077	5001	4894	4981
Gflops	3.33	3.61	3.59	3.47	3.44	3.50	3.40

Updating of a 25600 $\times$ 25600 R factor by 51200 new observations (1 $\times$ 4 procs)
IBM pSeries 690.

Remark on memory

If we store the entire diagonal blocks $\rightarrow$ we save $\sim$ 45%
Improvement:



- based on RFP storage defined in [Gustavson,04]
- 2 ScaLAPACK arrays

Experimental results

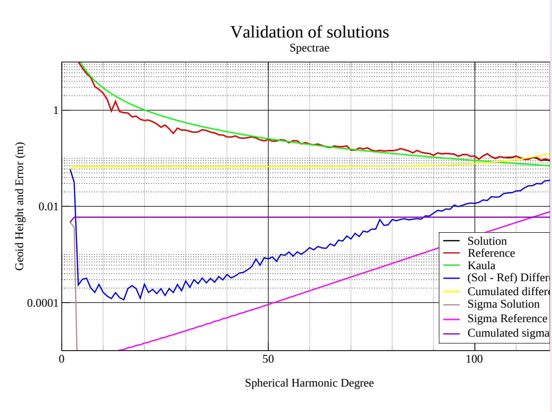
- 10 days of observations (GRACE measurements) $\Rightarrow m = 165,960$
- number of spherical harmonic coefficients $n = 22,801$
- we computed the 99 first degrees ($l_{max} = 99$)
- we compared with a reference solution

Experimental results

Machine	Power5 1.9 GHz
DGEMM (Gflops)	6
Init. R (Gflops)	4.4
Update R (Gflops)	4.3
Total time	4 h 40 min
Total Gflops	3.9

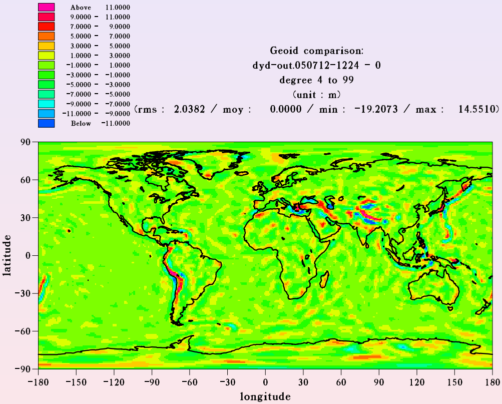
Performance for gravity field computation on 4 procs.
($m = 165,960$ and $n = 22,801$).

Experimental results



Gravity field computation for $m = 165,960$ and $n = 22,801$.

Experimental results



Geoid map ($4 \leq l \leq 99$).

Outline

Conclusion

- performance and storage make gravity field computations affordable on moderately parallel computers
- operational code based on QR factorization for gravity field computations (including CN estimate and regularization)
- trade-off between memory and performance
- further tests are scheduled in July on much bigger cases to assess the QR approach

References

[1] M. Baboulin, L. Giraud, S. Gratton,
A parallel distributed solver for large and dense symmetric systems: application to geodesy and electromagnetic problems.
CERFACS Tech. Rep. TR/PA/04/16, 2004.
Int J. of High Performance Computing Applications, volume 19 (No 4), pp. 353-363 (2005) .

[2] M. Baboulin, L. Giraud, S. Gratton, J. Langou
A distributed packed storage for large parallel calculations.
CERFACS Tech. Rep. TR/PA/05/30, 2005.

Submitted to *Concurrency and Computation: Practice and Experience.*