Very large least-squares for parameter estimation: Algorithm and application.

Marc Baboulin

ICL - University of Tennessee & Cerfacs

SciDAC workshop on libraries and algorithms - 08/30/07

Outline

1. Incremental linear least-squares

Outline

1. Incremental linear least-squares
2. Parallel implementation of QR updating

Outline

1. Incremental linear least-squares
2. Parallel implementation of QR updating
3. LS sensitivity analysis

Outline

1. Incremental linear least-squares
2. Parallel implementation of QR updating
3. LS sensitivity analysis
4. Application to space geodesy

Outline

1. Incremental linear least-squares
2. Parallel implementation of QR updating
3. LS sensitivity analysis
4. Application to space geodesy
5. Conclusion

Outline

1. Incremental linear least-squares
2. Parallel implementation of QR updating
3. LS sensitivity analysis
4. Application to space geodesy
5. Conclusion

General framework

- parameter estimation problem
  LLSP $\min_{x \in \mathbb{R}^n} \|Ax - b\|_2, \; A \in \mathbb{R}^{m \times n}, \; b \in \mathbb{R}^m$ where each row of A and b corresponds to one observation, these observations being collected periodically,
- accumulate observations and/or use regularization techniques
  until A is full column rank,
- problem too large to be solved using only one factorization.

Incremental linear least-squares
Parallel implementation of QR updating
Parallel tools for LS sensitivity analysis
Application to space geodesy
Conclusion

Update of normal equations

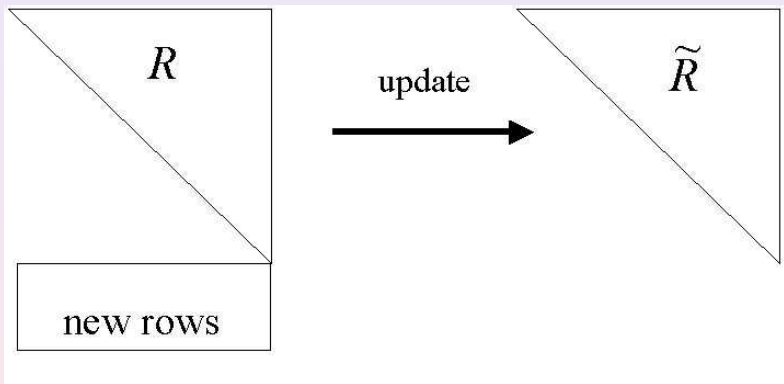
- $A^T A \leftarrow A^T A + (\text{new rows})^T (\text{new rows})$
- regularization (special case of Tikhonov) :
  add to $A^T A$ a diagonal matrix $D = diag(0, \cdots, 0, \alpha, \cdots, \alpha)$
- least-squares solver based on NE already implemented (using MPI and BLAS3) in packed storage [Baboulin et al.,05] .

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 $\begin{pmatrix} R \\ \text{new rows} \end{pmatrix}$
- regularization: QR factorization of $\begin{pmatrix} R \\ D \end{pmatrix}$

Outline

QR Updating



Updating of the R factor in an incremental QR factorization.

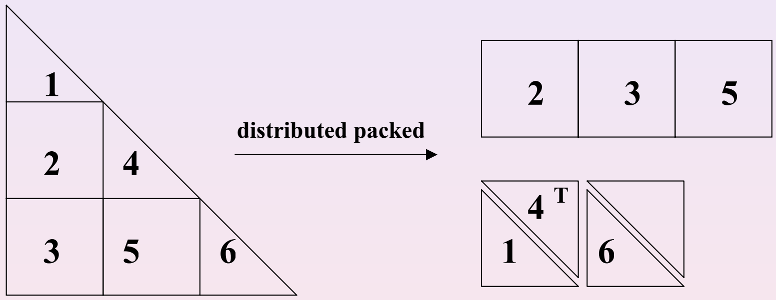
out-of-core algorithm in [Gunter et al.,05]

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

Parallel packed storage

- ScaLAPACK 2-D block cyclic distribution
  block size s, $p \times q$ process grid
- **distributed packed format** [Baboulin et al.,07] :
  R factor partitioned into square blocks of size S
  $S \propto lcm(p,q) \times s$
  $$\begin{pmatrix} B_1 & B_2 & B_3 \\ 0 & B_4 & B_5 \\ 0 & 0 & B_6 \end{pmatrix} \rightarrow \begin{bmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 \end{bmatrix}$$
- use of PBLAS and/or ScaLAPACK kernels
  here: PDGEQRF (QR factorization) and PDORMQR
  (multiplication by Q^T)

Parallel packed storage

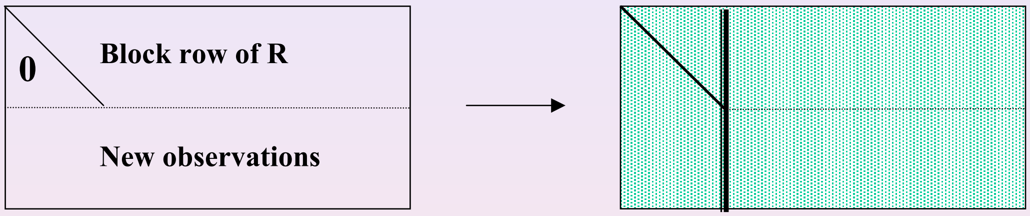


- 2 ScaLAPACK arrays
- diagonal blocks stored using RFP storage [Gustavson,04]

General algorithm

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

Implementation of R factor updating



PDGEQRF partial

- QR factorization that stops after the first S columns
- cost: $\sim\ 3Sn^2$ (if $n \gg S$)
- 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.

Incremental linear least-squares
Parallel implementation of QR updating
Parallel tools for LS sensitivity analysis
Application to space geodesy
Conclusion

Packed algorithm for QR updating

read new data in $L_{1:N}$ ($N = n/S$); $\widetilde{L}_{1:N} \leftarrow L_{1:N}$

for $i = 1 : N$

$j = INDGET(i, i)$ (indirect adressing $B_j = A_{ii}$)

$$C \leftarrow \begin{bmatrix} B_{j:j+N-i} \\ \widetilde{L}_{i:N} \end{bmatrix}$$

$\widetilde{C} = qr(C)$ stopped after the S first columns have been factored

$$\rightarrow \widetilde{C} = \begin{bmatrix} \widetilde{B}_{j:j+N-i} & \\ * & \bar{L}_{i+2:N} \end{bmatrix}$$ **(PDGEQRF_partial)**

$B_{j:j+N-i} \leftarrow \widetilde{B}_{j:j+N-i}$

end (i-loop)

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 × 25600 R factor by 51200 new observations (1 × 4 procs)
IBM pSeries 690.

Other works

- Choslesky factorization on distributed memory using packed storage (Fred Gustavson)
- tiled QR algorithm for multicore processors (A. Buttari et al.)
- new QR algorithms for distributed memory (J. Demmel, J. Langou)

Outline

Least squares conditioning

- parallel packed implementation of triangular solve gives $R^{-1}x$
  similarly we get Rx, R^Tx and $R^{-T}x$ in parallel packed format
  then we get $K(A) = \sqrt{K(R^TR)}$ (power method or Lanczos).
- but: $K(A)$ does not give sufficient information
- need tools to assess the quality of the solution

Least squares conditioning

- If we consider

$$\begin{array}{cccc} g : & \mathbb{R}^{m\times n} \times \mathbb{R}^m & \longrightarrow & \mathbb{R}^k \\ & A, b & \longmapsto & g(A,b) = L^Tx = L^T(A^TA)^{-1}A^Tb, \end{array}$$

  then the condition number of g at (A, b) is the norm of the derivative of g
- We are interested here in the case where $L = e_i$ or $L = I$
- Example of applications: all kind of parameter estimation problems (e.g determination of positions using GPS, determination of gravity field coefficients...)

Theoretical results

- Metric on data
  $\|(A,b)\|_{\mathrm{F\ or\ 2}} = \sqrt{\alpha^2 \|A\|^2_{\mathrm{F\ or\ 2}} + \beta^2 \|b\|_2^2},\ \alpha, \beta > 0$
  (perturbations on A and b can be monitored with α and β).
- We have general expressions for the condition numbers of x and each x_i in Frobenius or spectral norm and we show that the corresponding condition numbers lie within a factor $\sqrt{6}$

We obtain from [baboulin et al., 07]

- $\kappa_i(A,b) = \left(\left\| R^{-1}(R^{-T}e_i) \right\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + \left\| R^{-T}e_i \right\|_2^2 \left(\frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \right)^{\frac{1}{2}}$
- $\kappa_{LS}(A,b) = \left\| R^{-1} \right\|_2 \sqrt{\frac{\|R^{-1}\|_2^2 \|r\|_2^2 + \|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2}}$

Link with statistics

- Stat. model $Ax = b + \epsilon$ with $E(\epsilon) = 0$ and $var(\epsilon) = \sigma^2 I$
- Variance-covariance matrix $C = \sigma^2 (A^T A)^{-1} = \sigma^2 R^{-1} R^{-T}$

We obtain

- $\kappa_i(A,b) = \frac{1}{\sigma} \left(\left\| \frac{C_i}{\sigma} \right\|_2^2 \frac{\|r\|_2^2}{\alpha^2} + c_{ii} \left(\frac{\|x\|_2^2}{\alpha^2} + \frac{1}{\beta^2} \right) \right)^{\frac{1}{2}}$
  where C_i = i-th column and c_{ii} = i-th diagonal element of C
- When only b is perturbed (common case), we get $\kappa_i(A,b) = \frac{\sqrt{c_{ii}}}{\sigma}$
- $\kappa_{LS}(A,b) \simeq \left(\frac{\mathrm{tr}(C)}{\sigma^2} \left(\frac{\mathrm{tr}(C)\|r\|_2^2 + \sigma^2 \|x\|_2^2}{\sigma^2 \alpha^2} + \frac{1}{\beta^2} \right) \right)^{\frac{1}{2}}$

Computation with (Sca)LAPACK

- Computation of least squares conditioning with (Sca)LAPACK

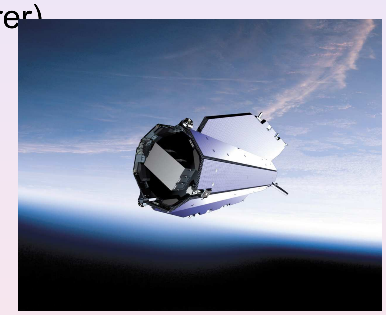
condition number	linear algebra operation	LAPACK routines	flops
$\kappa_i(A,b)$	$R^T y = e_i$ and $Rz = y$	2 calls to (P)DTRSV	$2n^2$
all $\kappa_i(A,b),\ i=1,n$	$RY = I$ and $ZR^T = Y$	(P)DPOTRI	$2n^3/3$
$\kappa_{LS}(A,b)$	estimate $\lVert R^{-1}\rVert_{1 \text{ or } \infty}$ compute $\lVert R^{-1}\rVert_F$	(P)DTRCON (P)DTRTRI	$\mathcal{O}(n^2)$ $n^3/3$

- There is currently no routine in (Sca)LAPACK for computing covariance, and we propose fragment codes to do this, similarly to the NAG library (routine F04YAF)

Outline

GOCE mission

GOCE: European Space Agency project (Gravity field and steady-state Ocean Circulation Explorer)



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

GOCE mission

- gravitational potential of the Earth (spherical coordinates)

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

- 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

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

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

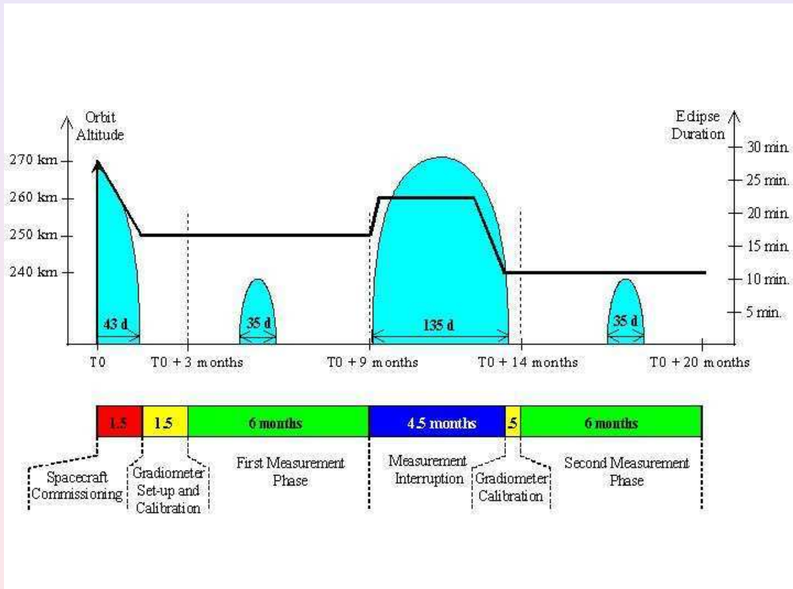
Numerical methods for GOCE

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

- iterative methods: (CG, FFT, multipole, spherical wavelets) 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 (end 2007 - end 2008).

Experimental results

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

User interface

Init packed storage and ScaLAPACK parameters

```
CALL INIT_PACKED( NBOBS, NBPAR, NPROW, NPCOL, MB,
                  S, N_ELEM)
```

loop on block rows ELEM of observations

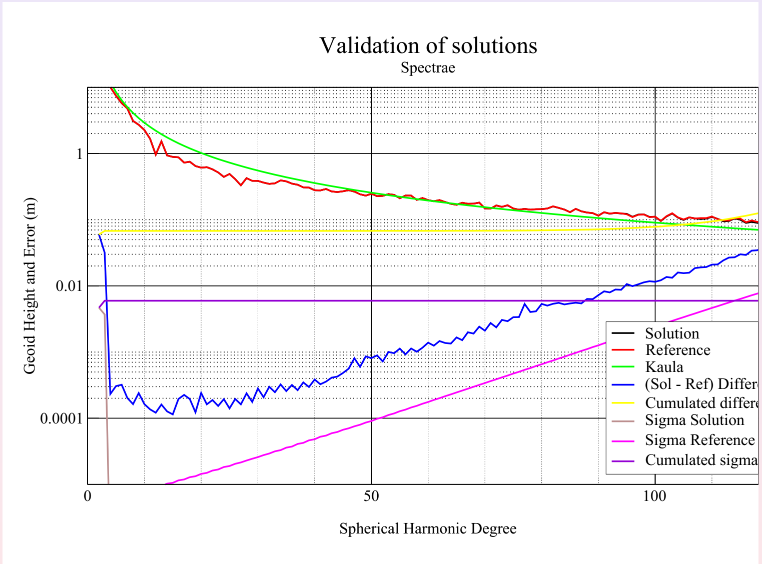
```
for I = 1:N_ELEM
    CALL READ_ELEM( I, S, ELEM)
    CALL QR_UPDATE( I, ELEM)
end
```


Experimental results

Machine	Power5 1.9 GHz
DGEMM (Gflops)	6
Init. R (Gflops)	4.4
Update R (Gflops)	4.3
Total time	4 h 10 min

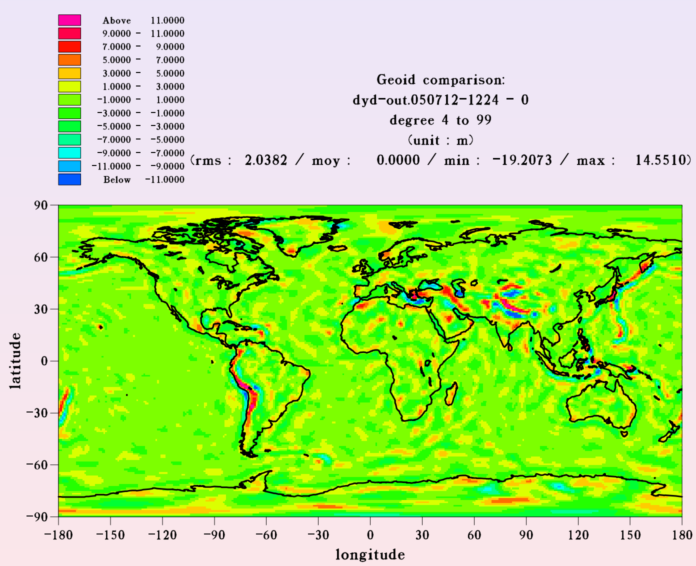
Performance for gravity field computation on 4 procs (IBM Power5).
($m = 165,960$ and $n = 22,801$).

Experimental results



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

Experimental results



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

Remark on accuracy

- $K(A) = \left(\|R^T R\| \cdot \|(R^T R)^{-1}\|\right)^{1/2} = 5 \cdot 10^6$.
- Householder QR
  $\frac{\|x-\tilde{x}\|}{\|x\|} \leq K(A)\left(1 + K(A)\frac{\|r\|_2}{\|A\|_2\|x\|_2}\right) u = 6 \cdot 10^{-10}$
- influence of measurement errors:
  measurement noise $10^{-9} m/s^2$
  Let perturb b with $b_i = b_i + mod(i, 10) \cdot \|b\| \cdot 10^{-8}$
  then $\frac{\|\tilde{x}-\hat{x}\|}{\|\tilde{x}\|} = 1.2 \cdot 10^{-6}$

Outline

Conclusion

- incremental algorithm that makes current gravity field computations affordable using direct methods
- trade-off between memory and performance
- next simulations will require more computational capabilities
- ongoing experiments:
  -GOCE computations (300 degrees, 90,000 parameters) on lower orbit, 60 days of observations, 43,200 obs./day
  $\rightarrow$ 2.6 millions observations (2 Tbytes)
  Teraflops computer $\rightarrow$ 28h + I/O
  -computation of conditioning of the solution components (via the covariance)

References

[1] M. Baboulin, M. Arioli, S. Gratton,
A partial condition number for linear least-squares problems.
SIAM J. Matrix Analysis and Applications, Vol. 29, No 2, pp. 413-433 (2007).

[2] M. Baboulin, L. Giraud, S. Gratton, J. Langou,
A distributed packed storage for large dense in-core parallel calculations.
Concurrency and Computation: Practice and Experience., Vol. 19, No4, pp. 483-502 (2007).

[3] M. Baboulin, L. Giraud, S. Gratton, J. Langou,
HPC tools for solving incremental least-squares problems.
LAPACK Working Note 179 (2006).

[4] M. Baboulin, L. Giraud, S. Gratton,
A parallel distributed solver for large and dense symmetric systems: application to geodesy and electromagnetic problems.
Int J. of High Performance Computing Applications, Vol. 19 No 4, pp. 353-363 (2005).