HAL
open science

Efficient computation of condition estimates for linear least squares problems

Marc Baboulin, Serge Gratton, Rémi Lacroix, Alan J. Laub

To cite this version:

Marc Baboulin, Serge Gratton, Rémi Lacroix, Alan J. Laub. Efficient computation of condition estimates for linear least squares problems. [Research Report] RR-8065, 2012, pp.15. hal-00731136v1

HAL Id: hal-00731136

https://inria.hal.science/hal-00731136v1

Submitted on 12 Sep 2012 (v1), last revised 13 Sep 2012 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Efficient computation of condition estimates for linear least squares problems

Marc Baboulin, Serge Gratton, Rémi Lacroix, Alan J. Laub

RESEARCH REPORT
N° 8065
Septembre 2012
Project-Team Grand-Large

ISSN 0249-6399 ISRN INRIA/RR--8065--FR+ENG

Efficient computation of condition estimates for linear least squares problems

Marc Baboulin*, Serge Gratton†, Rémi Lacroix‡, Alan J. Laub§

Project-Team Grand-Large

Research Report n° 8065 — Septembre 2012 — 15 pages

Abstract: Linear least squares (LLS) is a classical linear algebra problem in scientific computing, arising for instance in many parameter estimation problems. In addition to computing efficiently LLS solutions, an important issue is to assess the numerical quality of the computed solution. The notion of conditioning provides a theoretical framework that can be used to measure the numerical sensitivity of a problem solution to perturbations in its data. We recall some results for least squares conditioning and we derive a statistical estimate for the conditioning of an LLS solution. We present numerical experiments to compare exact values and statistical estimates. We also propose performance results using new routines on top of the multicore-GPU library MAGMA. This set of routines is based on an efficient computation of the variance-covariance matrix for which, to our knowledge, there is no implementation in current public domain libraries LAPACK and ScaLAPACK.

Key-words: linear least squares, condition number, statistical condition estimation, variance-covariance, GPU computing, MAGMA library.

* Inria and Université Paris-Sud, France (marc.baboulin@inria.fr).
† ENSEEIHT-IRIT and CERFACS, France (serge.gratton@enseeiht.fr).
‡ Inria and Université Paris-Sud, France (remi.lacroix@u-psud.fr).
§ University of California Los Angeles, USA (laub@ats.ucla.edu).

RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE

1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

Calcul efficace d'estimateurs de conditionnement pour problèmes de moindres carrés linéaires

Résumé : Les moindres carrés linéaires sont des problèmes classiques en algèbre linéaire dans le calcul scientifique, intervenant par exemple dans les problèmes d'estimation de paramètres. En plus de résoudre efficacement ces problèmes, il est important de garantir la qualité numérique de la solution calculée. La notion de conditionnement fournit un cadre théorique pour mesurer la sensibilité numérique de la solution d'un problème à des perturbations sur les données. Nous rappelons certains résultats sur le conditionnement des moindres carrés et nous dérivons un estimateur statistique pour le conditionnement de la solution. Nous présentons des expériences numériques permettant de comparer les valeurs exactes et les estimateurs statistiques. Nous proposons aussi des analyses de performance en utilisant une nouvelle implémentation à partir de la bibliothèque logicielle pour architectures multicœurs+GPU MAGMA. Cet ensemble de programmes repose sur un calcul efficace de la matrice de variance-covariance pour laquelle, à notre connaissance, il n'y a pas d'implémentation dans les biliothèques logicielles du domaine public LAPACK et ScaLAPACK.

Mots-clés : moindres carrés linéaires, conditionnement, estimateur statistique de conditionnement, variance-covariance, calculs sur GPU, bibliothèque MAGMA.

1 Introduction

We consider the overdetermined linear least squares (LLS) problem

$$\min_{x\in\mathbb{R}^n} \|Ax-b\|_2, \tag{1}$$

with $A \in \mathbb{R}^{m\times n}, m \geq n$ and $b \in \mathbb{R}^m$. Assuming that A is full column rank, Equation (1) has a unique solution $x = A^+b$ where A^+ is the Moore-Penrose pseudoinverse of the matrix A, expressed as $A^+ = (A^TA)^{-1}A^T$. We can find for instance in [6, 13, 19] a comprehensive survey of the methods that can be used for solving efficiently and accurately LLS problems.

The condition number is a measure of the sensitivity of a mapping to perturbations. It was initially defined in [25] as the maximum amplification factor between a small perturbation in the data and the resulting change in the problem solution. The perturbations are measured using metrics, for example norms. Namely, if the solution x of a given problem can be expressed as a function $g(y)$ of a data y, then if g is differentiable (which is the case for many linear algebra problems), the condition number of g at y can be defined as (see e.g. [12])

$$\kappa(y) = \max_{z\neq 0} \frac{\|g'(y).z\|}{\|z\|}. \tag{2}$$

From this definition, $\kappa(y)$ is a quantity that, for a given perturbation size on the data y, allows us to predict *to first order* the perturbation size on the solution x. Associated with a backward error [29], condition numbers are useful to assess the numerical quality of a computed solution. Indeed numerical algorithms are always subject to errors although their sensitivity to errors may vary. These errors can have various origins like for instance data uncertainty due to instrumental measurements or rounding and truncation errors inherent to finite precision arithmetic.

LLS can be very sensitive to perturbations in data in particular when the right-hand side is too far from the column space (see [20, p. 98]). It is then crucial to be able to assess the quality of the solution in practical applications. It was shown in [14] that the 2-norm condition number $\text{cond}(A)$ of the matrix A plays a significant role in LLS sensitivity analysis. It was later proved in [28] that the sensitivity of LLS problems is proportional to $\text{cond}(A)$ when the residual vector is small and to $\text{cond}(A)^2$ otherwise. Then [12] provided a closed formula for the condition number of LLS problems, using the Frobenius norm to measure the perturbations of A. Since then many results on normwise LLS condition numbers have been published (see e.g. [2, 6, 11, 15, 16]).

It was observed in [18] that normwise condition numbers can lead to a loss of information since they consolidate all sensitivity information into a single number. Indeed in some cases this sensitivity can vary significantly among the different solution components (some examples for LLS are presented in [2, 22]). To overcome this issue, it was proposed the notion of "componentwise" condition numbers or condition numbers for the solution components [9]. Note that this approach must be distinguished from the componentwise metric also applied to LLS for instance in [4, 10]. This approach was generalized by the notion of *partial* or *subspace* condition numbers where we study the conditioning of L^Tx with $L \in \mathbb{R}^{n\times k}, k \leq n$, proposed for instance in [2, 5] for least squares and total least squares, or [8] for linear systems. When L is a canonical vector e_i, it is equivalent to the condition number of the ith component, while when L is the identity matrix, it is the same as the classical condition number mentioned above. The motivation for computing the conditioning of L^Tx can be found for instance in [2, 3] for normwise LLS condition numbers.

Even though condition numbers provide interesting information about the quality of the computed solution, they are expected to be calculated in an acceptable time compared to the cost for the solution itself. Computing the exact (subspace or not) condition number requires $\mathcal{O}(n^3)$ flops when the LLS solution x has been aready computed (e.g., using a QR factorization)

and can be reused to compute the conditioning [2, 3]. This cost is affordable when compared to the cost for solving the problem ($\mathcal{O}(2mn^2)$ flops when $m \gg n$). However statistical estimates can reduce this cost to $\mathcal{O}(n^2)$ [17, 21]. The theoretical quality of the statistical estimates can be formally measured by the probability to give an estimate in a certain range around the exact value.

This paper is organized as follows. In Section 2 we first summarize some existing results for the condition numbers of the LLS solution or its components. For each of these quantities, we propose practical algorithms and evaluate the computational cost. More specifically in Section 2.1 we derive a new expression for the statistical estimate of the conditioning of x. Then in Section 3 we present numerical experiments to compare the LLS conditioning with their corresponding statistical estimates. We also propose performance results for the computation of these quantities using new routines on top of the MAGMA [27] parallel library. For the exact values, these routines are based on the computation of the variance-covariance for which, to our knowledge, there is no routine in the public domain libraries LAPACK [1] and ScaLAPACK [7], contrary to the NAG [26] library. Our implementation takes advantage of the current hybrid multicore-GPU architectures and aims at being integrated into MAGMA.

Notations $A \in \mathbb{R}_r^{m\times n}$ means that A is a m-by-n matrix of rank r. The notation $\|\cdot\|_2$ applied to a matrix (resp. a vector) refers to the spectral norm (resp. the Euclidean norm) and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. The matrix I is the identity matrix and e_i is the ith canonical vector. The uniform continuous distribution between a and b is abbreviated $\mathcal{U}(a,b)$ and the normal distribution of mean μ and variance σ^2 is abbreviated $\mathcal{N}(\mu,\sigma^2)$. $\mathrm{cond}(A)$ denotes the 2-norm condition number of a matrix A, defined as $\mathrm{cond}(A) = \|A\|_2\|A^+\|_2$. The notation $|\cdot|$ applied to a matrix or a vector holds componentwise.

2 Condition estimation for linear least squares

In Section 2.1 we are concerned in calculating the condition number of the LLS solution x and in Section 2.2 we compute or estimate the conditioning of the components of x. We suppose that the LLS problem has already been solved using a QR factorization (the normal equations method is also possible but the condition number is then proportional to $\mathrm{cond}(A)^2$ [6, p. 49]). Then the solution x, the residual r, and the factor $R \in \mathbb{R}^{n\times n}$ of the QR factorization of A are readily available (we recall that the Cholesky factor of the normal equations is, in exact arithmetic, equal to R up to some signs). We also make the assumption that both A and b can be perturbed, these perturbations being measured using the weighted product norm $\|(\Delta A, \Delta b)\|_F = \sqrt{\|\Delta A\|_F^2 + \|\Delta b\|_2^2}$. In addition to providing us with simplified formulas, this product norm has the advantage, mentioned in [15], to be appropriate for estimating the forward error obtained when the LLS problem is solved via normal equations.

2.1 Conditioning of the least squares solution

Exact formula We can obtain from [3] a closed formula for the condition number of the LLS solution as

$$\kappa_{LS} = \|R^{-1}\|_2 \left(\|R^{-1}\|_2^2\|r\|_2^2 + \|x\|_2^2 + 1\right)^{\frac{1}{2}}. \tag{3}$$

This equation requires mainly to compute the minimum singular value of the matrix A (or R), which can be done using iterative procedures like the inverse power iteration on R, or more expensively with the full SVD of R ($\mathcal{O}(n^3)$ flops). Note that $\|R^{-T}\|_2$ can be approximated by other matrix norms (see [19, p. 293]).

Statistical estimate Similarly to [8] for linear systems, we can estimate the condition number of the LLS solution using the method called *small-sample theory* [21] that provides statistical condition estimates for matrix functions. By Taylor's theorem, the forward error Δx on the solution $x(A,b)$ can be expressed as

$$\Delta x = x'(A,b).(\Delta A, \Delta b) + \mathcal{O}(\|(\Delta A, \Delta b)\|_F^2). \tag{4}$$

The notation $x(A,b)$ means that x is a function of the data A and b and $x'(A,b)$ is the derivative of this function. $x'(A,b).(\Delta A, \Delta b)$ denotes the image of $(\Delta A, \Delta b)$ by the linear function $x'(A,b)$. Then, as mentioned in Equation (2), the condition number of x corresponds to the operator norm of $x'(A,b)$, which is a bound to first order on the sensitivity of x at (A,b). We now use [21] to estimate $\|\Delta x\|_2$ by

$$\xi(q) = \frac{\omega_q}{\omega_n}\sqrt{|z_1^T \Delta x|^2 + \cdots + |z_q^T \Delta x|^2}, \tag{5}$$

where $z_1, \cdots, z_q$ are random orthogonal vectors selected uniformly and randomly from the unit sphere in n dimensions, and ω_q is the Wallis factor defined by

$$\begin{aligned}
\omega_1 &= 1, \\
\omega_q &= \frac{1 \cdot 3 \cdot 5 \cdots (q-2)}{2 \cdot 4 \cdot 6 \cdots (q-1)} \text{ for } q \text{ odd}, \\
\omega_q &= \frac{2}{\pi}\,\frac{2 \cdot 4 \cdot 6 \cdots (q-2)}{1 \cdot 3 \cdot 5 \cdots (q-1)} \text{ for } q \text{ even}.
\end{aligned}$$

ω_q can be approximated by $\sqrt{\frac{2}{\pi(q-\frac{1}{2})}}$.

It comes from [21] that if for instance we have $q = 3$, then the probability that $\xi(q)$ lies within a factor α of $\|\Delta x\|_2$ is

$$Pr(\frac{\|\Delta x\|_2}{\alpha} \leq \xi(q) \leq \alpha \|\Delta x\|_2) \approx 1 - \frac{32}{3\pi^2\alpha^3}. \tag{6}$$

For $\alpha = 10$, we obtain a probability of 99.9%.

For each $i \in \{1, \cdots, q\}$, using Equation (2) we have the first-order bound

$$|z_i^T \Delta x| \leq \kappa_i \|(\Delta A, \Delta b)\|_F, \tag{7}$$

where κ_i denotes the condition number of the function $z_i^T x(A,b)$. Then using (5) and (7) we get

$$\xi(q) \leq \frac{\omega_q}{\omega_n}\left(\sum_{i=1}^{q} \kappa_i^2\right)^{\frac{1}{2}} \|(\Delta A, \Delta b)\|_F.$$

Since on the other hand we have

$$\|\Delta x\|_2 \leq \kappa_{LS} \|(\Delta A, \Delta b)\|_F,$$

then we will consider that

$$\bar{\kappa}_{LS} = \frac{\omega_q}{\omega_n}\left(\sum_{i=1}^{q} {\kappa_i}^2\right)^{\frac{1}{2}} \tag{8}$$

is an estimate for κ_{LS}.

We point out that $\bar{\kappa}_{LS}$ is a scalar quantity that must be distinguished from the estimate given in [22] which is a vector. Indeed the small-sample theory is used here to derive an estimate of the condition number of x whereas it is used in [22] to derive estimates of the condition numbers of the components of x (see Section 2.2). Now we can derive Algorithm 2.1 that computes $\bar{\kappa}_{LS}$ as expressed in Equation (8) and using the condition numbers of $z_i^T x$. The vectors $z_1, \cdots, z_q$ are obtained for instance via a QR factorization of a random matrix $Z \in \mathbb{R}^{n\times q}$. The condition number of $z_i^T x$ can be computed using the expression given in [3]) as

$$\kappa_i = \left(\|R^{-1}R^{-T}z_i\|_2^2\|r\|_2^2 + \|R^{-T}z_i\|_2^2(\|x\|_2^2+1)\right)^{\frac{1}{2}}. \tag{9}$$

The accuracy of the estimate can be tweaked by modifying the number q of considered random samples. The computation of $\bar{\kappa}_{LS}$ requires computing the QR factorization of an $n\times q$ matrix for $\mathcal{O}(nq^2)$ flops. It also involves solving q times two $n\times n$ triangular linear systems, each triangular system being solved in $\mathcal{O}(n^2)$ flops. The resulting computational cost is $\mathcal{O}(2qn^2)$ flops (if $n \gg q$).

Algorithm 2.1 Statistical condition estimation for linear least squares solution (SCE_LLS)

Require: $q \geq 1$, the number of samples
  Generate q vectors $z_1, z_2, ..., z_q \in \mathbb{R}^n$ with entries in $\mathcal{U}(0,1)$
  Orthonormalize the vectors z_i using a QR factorization
  for $j = 1$ to q **do**
    Compute $\kappa_j = \left(\|R^{-1}R^{-T}z_j\|_2^2\|r\|_2^2 + \|R^{-T}z_i\|_2^2(\|x\|_2^2+1)\right)^{\frac{1}{2}}$
  end for
  Compute $\bar{\kappa}_{LS} = \frac{\omega_q}{\omega_n}\sqrt{\sum_{j=1}^{q}\kappa_j^2}$ with $\omega_q = \sqrt{\frac{2}{\pi(q-\frac{1}{2})}}$

2.2 Componentwise condition estimates

In this section, we focus on calculating the condition number for each component of the LLS solution x. The first one is based on the results from [3] and enables us to compute the exact value of the condition numbers for the ith component of x. The other is a statistical estimate from [22].

Exact formula By considering in Equation (9) the special case where $z_i = e_i$, we can express in Equation (10) the condition number of the component $x_i = e_i^T x$ and then calculate a vector $\kappa_{CW} \in \mathbb{R}^n$ with components κ_i being the exact condition number for the ith component expressed by

$$\kappa_i = \left(\|R^{-1}R^{-T}e_i\|_2^2\|r\|_2^2 + \|R^{-T}e_i\|_2^2(\|x\|_2^2+1)\right)^{\frac{1}{2}}. \tag{10}$$

The computation of one κ_i requires two triangular solves ($R^T y = e_i$ and $Rz = y$) corresponding to $2n^2$ flops. When we want to compute all κ_i, it is more efficient to solve $RY = I$ and then compute YY^T, which requires about $2n^3/3$ flops.

Statistical condition estimate We can find in [22] three different algorithms to compute statistical componentwise condition estimation for LLS problems. Algorithm 2.2 corresponds to the algorithm that uses unstructured perturbations and it can be compared with the exact value given in Equation (10). Algorithm 2.2 computes a vector $\bar{\kappa}_{CW} = (\bar{\kappa}_1, \cdots, \bar{\kappa}_n)^T$ containing the statistical estimate for the κ_i's. Depending on the needed accuracy for the statistical estimation, the number of random perturbations $q \geq 1$ applied to the input data in Algorithm 2.2 can be

adjusted. This algorithm involves two $n \times n$ triangular solves with q right-hand sides, which requires about qn^2 flops.

Algorithm 2.2 Componentwise statistical condition estimate for linear least squares (SCE_LLS_CW)

Require: $q \geq 1$, the number of perturbations of input data
  for $j = 1$ to q **do**
    Generate $S_j \in \mathbb{R}^{n\times n}$, $g_j \in \mathbb{R}^n$ and $h_j \in \mathbb{R}^n$ with entries in $\mathcal{N}(0,1)$
    Compute $u_j = R^{-1}(g_j - S_j x + \|Ax - b\|_2 R^{-T} h_j)$
  end for
  Let $p = m(n+1)$ and compute vector $\bar{\kappa}_{CW} = \frac{\sum_{i=1}^{q} |u_j|}{q\omega_p\sqrt{p}}$ with $\omega_q = \sqrt{\frac{2}{\pi(q-\frac{1}{2})}}$

3 Numerical experiments

In the following experiments, random LLS problems are generated using the method given in [24] for generating LLS test problems with known solution x and residual norm. Random problems are generated as $[A, x, r, b] = P(m, n, \rho, l)$ such that $A \in \mathbb{R}^{m\times n}$, $\|r\|_2 = \rho$ and $\text{cond}(A) = n^l$. The matrix A is generated using

$$A = Y \begin{pmatrix} D \\ 0 \end{pmatrix} Z^T,\ Y = I - 2yy^T,\ Z = I - 2zz^T$$

where $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ are random unit vectors and $D = n^{-l} diag(n^l, (n-1)^l, (n-2)^l, \cdots, 1)$. We have $x = (1, 2^2, ..., n^2)^T$, the residual vector is given by $r = Y \begin{pmatrix} 0 \\ v \end{pmatrix}$ where $v \in \mathbb{R}^{m-n}$ is a random vector of norm ρ and the right-hand side is given by $b = Y \begin{pmatrix} DZx \\ v \end{pmatrix}$. In Section 3.1, we will consider LLS problems of size $m \times n$ with $m = 9984$ and $n = 2496$.

3.1 Accuracy of statistical estimates

3.1.1 Conditioning of LLS solution

In this section we compare the statistical estimate $\overline{\kappa}_{LS}$ obtained via Algorithm 2.1 with the exact condition number κ_{LS} computed using Equation (3). In our experiments, the statistical estimate is computed using two samples ($q = 2$). For seven different values for $\text{cond}(A) = n^l$ (l ranging from 0 to 3, $n = 2496$) and several values of $\|r\|_2$, we report in Table 1 the ratio $\bar{\kappa}_{LS}/\kappa_{LS}$, which is the average of the ratios obtained for 100 random problems.

The results in Table 1 show the relevance of the statistical estimate presented in Section 2.1. For $n \geq \frac{1}{2}$ the averaged estimated values never differ from the exact value by more than one order of magnitude. We observe that when l tends to 0 (i.e., $\text{cond}(A)$ gets close to 1) the estimate becomes less accurate. This can be explained by the fact that the statistical estimate $\overline{\kappa}_{LS}$ is based on evaluating the Frobenius norm of the Jacobian matrix [17]. Actually some additional experiments showed that $\overline{\kappa}_{LS}/\kappa_{LS}$ evolves exactly like $\left\|R^{-1}\right\|_F^2 / \left\|R^{-1}\right\|_2^2$. In this particular LLS

Table 1: Ratio between exact and statistical condition numbers ($q = 2$)

cond(A)	n^0	$n^{\frac{1}{2}}$	n^1	$n^{\frac{3}{2}}$	n^2	$n^{\frac{5}{2}}$	n^3
$\|r\|_2 = 10^{-10}$	57.68	3.32	1.46	1.19	1.10	1.03	1.07
$\|r\|_2 = 10^{-5}$	57.68	3.33	1.45	1.18	1.07	1.09	1.05
$\|r\|_2 = 1$	57.68	3.36	1.45	1.19	1.19	1.05	1.15
$\|r\|_2 = 10^5$	57.68	3.33	1.24	1.04	1.05	1.05	1.02
$\|r\|_2 = 10^{10}$	57.68	1.44	1.07	1.09	1.00	1.01	1.07

problem we have

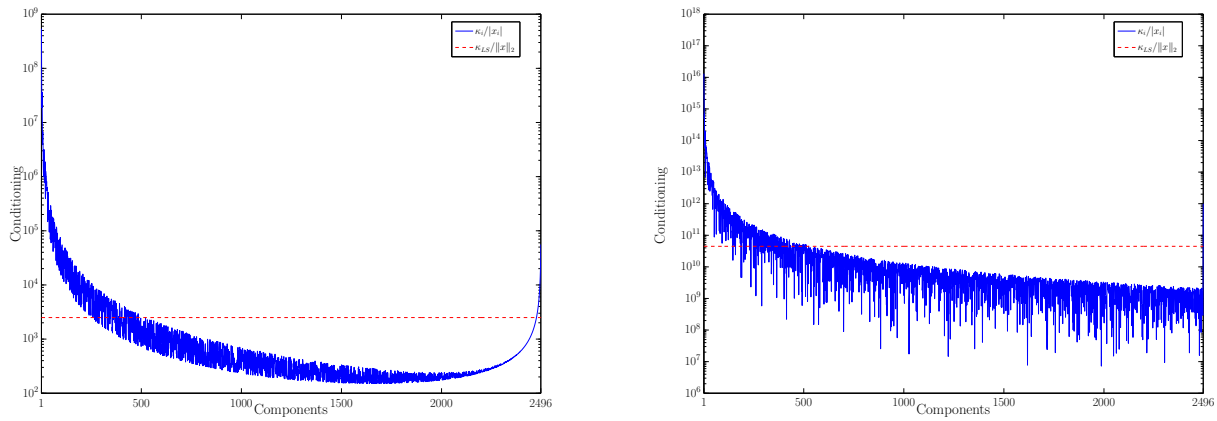
$$\begin{aligned} \left\|R^{-1}\right\|_F^2 / \left\|R^{-1}\right\|_2^2 &= \left(1 + (n/(n-1))^{2l} + (n/(n-2))^{2l} + \cdots + n^{2l}\right)/n^{2l} \\ &= \sum_{k=1}^{n} \frac{1}{k^{2l}}. \end{aligned}$$

Then when l tends towards 0, $\left\|R^{-1}\right\|_F / \left\|R^{-1}\right\|_2 \sim \sqrt{n}$, whereas this ratio gets closer to 1 when l increases. This is consistent with the well-known inequality $1 \leq \left\|R^{-1}\right\|_F / \left\|R^{-1}\right\|_2 \leq \sqrt{n}$. Note that the accuracy of the statistical estimate does not vary with the residual norm.

3.1.2 Componentwise condition estimation

Figure 1 depicts the conditioning for all LLS solution components, computed as $\kappa_i/|x_i|$ where the κ_i's are obtained using Equation (10). Figures 1(a) and 1(b) correspond to random LLS problems with respectively cond(A) $= 2.5 \cdot 10^3$ and cond(A) $= 2.5 \cdot 10^9$. These figures show the interest of the componentwise approach since the sensitivity to perturbations of each solution component varies significantly (from 10^2 to 10^8 for cond(A) $= 2.5 \cdot 10^3$, and from 10^7 to 10^{16} for cond(A) $= 2.5 \cdot 10^9$). The normalized condition number of the solution computed using Equation (3) is $\kappa_{LS}/\left\|x\right\|_2 = 2.5 \cdot 10^3$ for cond(A) $= 2.5 \cdot 10^3$ and $\kappa_{LS}/\left\|x\right\|_2 = 4.5 \cdot 10^{10}$ for cond(A) $= 2.5 \cdot 10^9$, which in both cases greatly overestimates or underestimates the conditioning of some components. Note that the LLS sensitivity is here well measured by cond(A) since $\|r\|_2$ is small compared to $\|A\|_2$ and $\|x\|_2$, as expected from [28] (otherwise it would be measured by cond(A)^2).

In Figure 2 we represent for each solution component, the ratio between the statistical condition estimate computed via Algorithm 2.2, considering two samples ($q = 2$), and the exact value computed using Equation (10). The ratio is computed as an average on 100 random problems. We observe that this ratio is lower than 1.2 for the case cond(A) $= 2.5 \cdot 10^3$ (Figure 2 (a)) and close to 1 for the case cond(A) $= 2.5 \cdot 10^9$ (Figure 2 (b)), which also confirms that, similarly to $\overline{\kappa}_{LS}$ in Section 3.1.1, the statistical condition estimate is more accurate for larger values of cond(A).



(a) $\kappa_i/|x_i|$ ($\text{cond}(A) = 2.5 \cdot 10^3$)

(b) $\kappa_i/|x_i|$ ($\text{cond}(A) = 2.5 \cdot 10^9$)

Figure 1: Componentwise condition numbers of LLS (problem size 9984×2496)

3.2 Computing least squares condition numbers on multicore-GPU architectures

3.2.1 Variance-covariance matrix

In many physical applications, LLS problems are considered using a statistical model often referred to as *linear statistical model* where we have to solve

$$b = Ax + \epsilon, \ A \in \mathbb{R}_n^{m \times n}, \ b \in \mathbb{R}^m,$$

where ϵ is a vector of random errors having expected value $E(\epsilon) = 0$ and variance-covariance $V(\epsilon) = \sigma_b^2 I$. In statistical language, the matrix A is called the regression matrix and the unknown vector x is called the vector of regression coefficients. Following the Gauss-Markov theorem [30], the least squares estimates $\hat{x}$ is the linear unbiased estimator of x satisfying

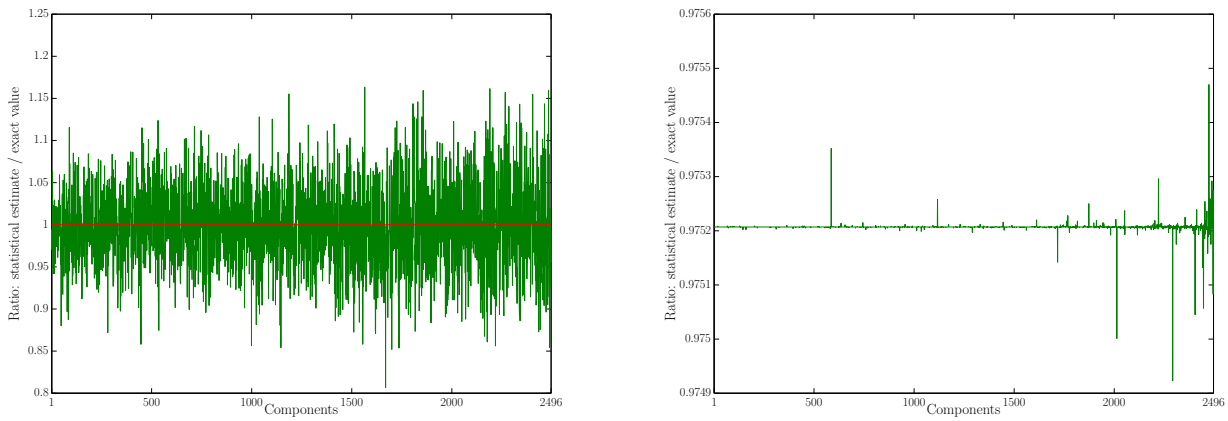
$$\hat{x} = arg \min_{x \in \mathbb{R}^n} \|Ax - b\|_2,$$

with minimum variance-covariance equal to

$$C = \sigma_b^2 (A^T A)^{-1}. \tag{11}$$

The diagonal elements c_{ii} of C give the variance of each component $\hat{x}_i$. The off-diagonal elements $c_{ij}, \ i \neq j$ give the covariance between $\hat{x}_i$ and $\hat{x}_j$. Then instead of computing condition numbers (which are notions more commonly handled by numerical linear algebra practitioners) physicists often compute the variance-covariance matrix whose entries are intimately correlated with condition numbers κ_i and κ_{LS} mentioned previously.

When the variance-covariance matrix has been computed, the condition numbers described in Section 2 can be easily obtained. Indeed, we can use the fact that $\left\|R^{-1}\right\|_2^2 = \frac{\|C\|_2}{\sigma_b^2}$, $\|R^{-T} e_i\|_2^2 = \frac{c_{ii}}{\sigma_b^2}$, and $\|R^{-1} R^{-T} e_i\|_2 = \frac{\|C_i\|_2}{\sigma_b^2}$ where C_i and c_{ii} are respectively the ith column and the ith diagonal element of the matrix C. Then by replacing respectively in Equations (3) and (10) we



(a) ratio $\bar{\kappa}_i/\kappa_i$ ($\text{cond}_2(A) = 2.5 \cdot 10^3$)

(b) ratio $\bar{\kappa}_i/\kappa_i$ ($\text{cond}_2(A) = 2.5 \cdot 10^9$)

Figure 2: Comparison between componentwise exact and statistical condition numbers

get the formulas

$$\kappa_{LS} = \frac{\|C\|_2^{1/2}}{\sigma_b}((m-n)\|C\|_2 + \|x\|_2^2 + 1)^{1/2}, \tag{12}$$

and

$$\kappa_i = \frac{1}{\sigma_b}((m-n)\|C_i\|_2^2 + c_{ii}(\|x\|_2^2 + 1))^{1/2}. \tag{13}$$

Note that, when $m > n$, $\frac{1}{m-n}\|r\|_2^2$ is an unbiased estimate of σ_b^2 [6, p. 4].

3.2.2 Implementation details

To our knowledge, there is no existing routine in public domain libraries LAPACK [1], ScaLAPACK [7], PLASMA [23], MAGMA [27] to compute the variance-covariance matrix or condition estimates. We propose an implementation for the MAGMA library (release 1.2.1) which is a dense linear algebra library for heterogeneous multicore-GPU architectures with interface similar to LAPACK. We developped a set of routines that compute the following quantities:

- Variance-covariance matrix C.
- κ_{LS}, condition number of x.
- κ_{CW}, vector of the κ_i's, condition numbers of the solution components.
- $\bar{\kappa}_{LS}$, statistical estimate of κ_{LS}.
- $\bar{\kappa}_{CW}$, vector of the statistical estimates the κ_i's.

The variance-covariance computation requires inverting a triangular matrix and multiplying this triangular matrix by its transpose (similarly to the LAPACK routine DPOTRI [1, p. 26] that computes the inverse of a matrix from its Cholesky factorization). These operations use a block algorithm which, for the diagonal blocks, is performed recursively. The recursive part is performed by the CPU for sake of performance while the rest of the algorithm is executed on the GPU.

The computation of the exact condition number κ_{LS} from the variance-covariance using Equation (12) involves the computation of the spectral norm of C which is generally computed via an SVD. However, since A is a full rank matrix, C is symmetric positive definite and its singular values coincide with its eigenvalues. Then we use an eigenvalue decomposition of C which is faster than an SVD because it takes into account the symmetry of C. The tridiagonalization phase is performed on the GPU while the subsequent eigenvalue computation is performed on the CPU host.

The statistical estimates computed via Algorithms 2.1 and 2.2 require the generation and orthonormalization of random vectors followed by 2 triangular solves. The generation of the random vectors and the triangular solves are executed on the GPU. However the orthonormalization is applied to small matrices (due to the small number of samples) and thus is performed on the CPU because this procedure would not take advantage of the GPU.

3.2.3 Performance results

In this section we present performance results for computing the variance-covariance matrix and LLS condition numbers. The tests have been achieved on a multicore processor Intel Xeon E5645 (2 sockets $\times$ 6 cores) running at 2.4 GHz (the cache size per core is 12 MB and the size of the main memory is 48 GB). This system hosts two GPU NVIDIA Tesla C2075 running at 1.15 GHz with 6 GB memory each. MAGMA was linked with the libraries MKL 10.3.8 and CUDA 4.1, respectively, for multicore and GPU.

We show in Figure 3 the CPU time to compute LLS solution and condition numbers using 12 threads and 1 GPU. We observe that the computation of the variance-covariance matrix and of the components conditioning κ_i's are significantly faster than the cost for solving the problem with respectively a time factor larger than 3 and 2, this factor increasing with the problem size. The κ_i's are computed with the variance-covariance matrix using Equation (13). The time overhead between the computation of the κ_i's and the variance-covariance computation comes from the computation of the norms of the columns (routine cublasDnrm2) which has a nonoptimal implementation. As expected, the routines that compute statistical condition estimates outperform the other routines. Note that we did not mention on this graph the performance for computing κ_{LS} using Equation (12). Indeed this involves an eigenvalue decomposition of the variance-covariance matrix (MAGMA routine magma_dsyevd_gpu), which turns out to be much slower than the LLS solution (MAGMA routine magma_dgels3_gpu) in spite of a smaller number of arithmetic operations. Even though the theoretical number of flops for computing κ_{LS} is much smaller than for computing x ($\mathcal{O}(n^3)$ vs $\mathcal{O}(mn^2)$), having an efficient implementation on the targetted architecture is essential to take advantage of the gain in flops.

We can illustrate this by comparing in Figure 4 the time for computing an LLS solution and its conditioning using LAPACK and MAGMA. We observe that MAGMA provides faster solution and condition number but, contrary to LAPACK, the computation of the condition number is slower than the time for the solution, in spite of a smaller flops count. This shows the need for improving the Gflop/s performance of eigensolvers or SVD solvers for GPUs but it also confirms the interest of considering statistical estimates on multicore-GPU architectures to get fast computations.

4 Conclusion

In this paper we studied the condition number of an LLS solution and of its components. We summarized the exact values and statistical estimates for these quantities. We also derived an expression for another statistical condition estimate for the LLS solution. In numerical experiments

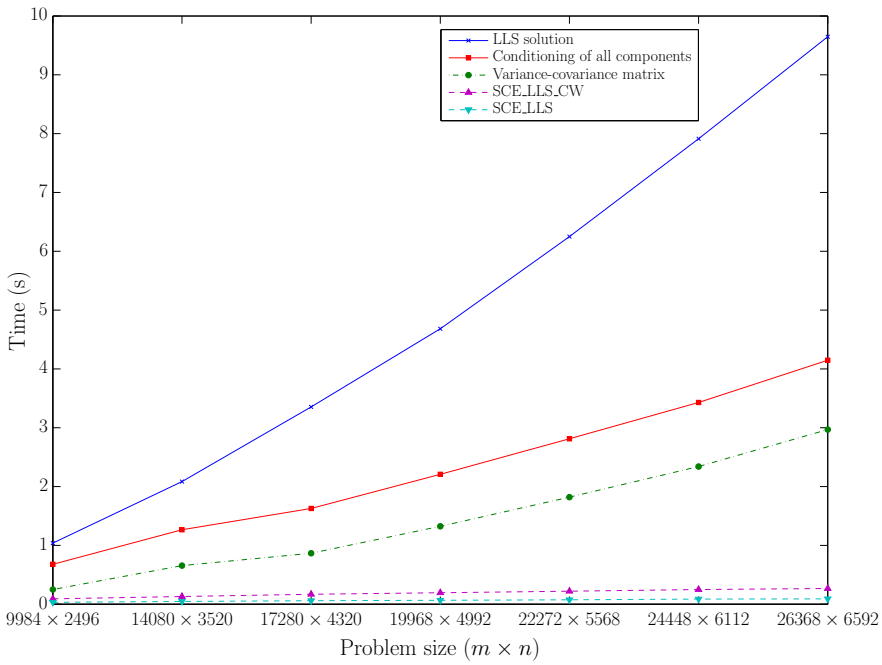


Figure 3: Performance for computing LLS condition numbers with MAGMA

we compared the statistical estimates with the exact values. We proposed a new implementation for computing the variance-covariance matrix and the condition numbers using the library MAGMA. The performance results that we obtained on a current multicore-GPU system confirm the interest of using statistical condition estimates. Subsequently to this work, new routines will be proposed in the next releases of LAPACK and MAGMA to compute the variance-covariance matrix after a linear regression.

Acknowledgements: We are grateful to Jack Dongarra and Stanimire Tomov (Innovative Computing Laboratory, University of Tennessee) for allowing us to perform experimentation on their multicore+GPU machines. We also want to thank Julien Langou (University of Colorado, Denver) for his collaboration on the LAPACK kernels for condition number computations.

References

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, 1999. Third edition.

[2] M. Arioli, M. Baboulin, and S. Gratton. A partial condition number for linear least-squares problems. *SIAM J. Matrix Analysis and Applications*, 29(2):413–433, 2007.

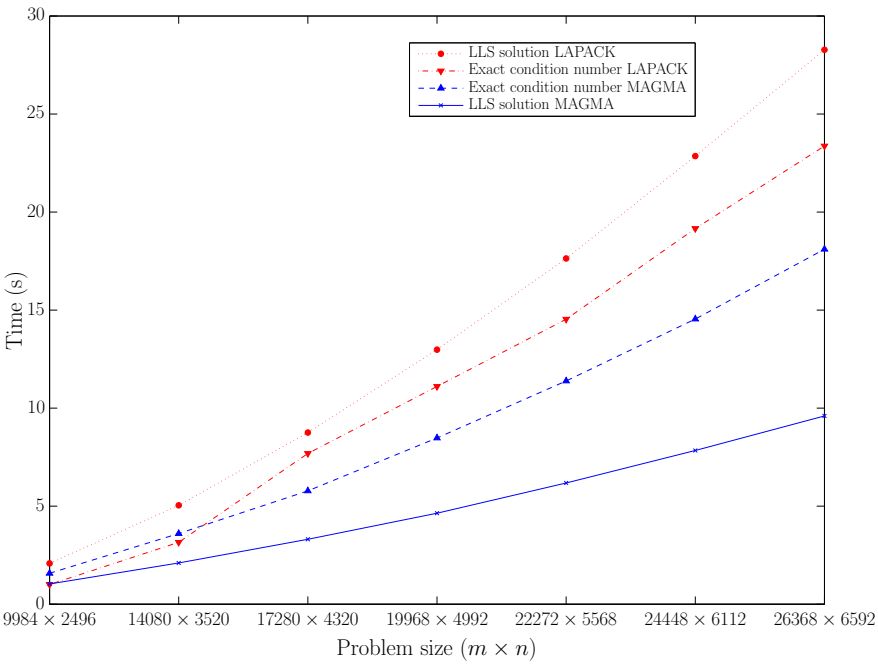


Figure 4: Time for LLS solution and condition number

[3] M. Baboulin, J. Dongarra, S. Gratton, and J. Langou. Computing the conditioning of the components of a linear least squares solution. *Numerical Linear Algebra with Applications*, 16(7):517–533, 2009.

[4] M. Baboulin and S. Gratton. Using dual techniques to derive componentwise and mixed condition numbers for a linear functional of a linear least squares solution. *BIT*, 49(1):3–19, 2009.

[5] M. Baboulin and S. Gratton. A contribution to the conditioning of the total least squares problem. *SIAM Journal on Matrix Analysis and Applications*, 32(3):685–699, 2011.

[6] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[7] 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 Users' Guide*. SIAM, Philadelphia, 1997.

[8] Y. Cao and L. Petzold. A subspace error estimate for linear systems. *SIAM J. Matrix Analysis and Applications*, 24:787–801, 2003.

[9] S. Chandrasekaran and I. C. F Ipsen. On the sensitivity of solution components in linear systems of equations. *Numerical Linear Algebra with Applications*, 2:271–286, 1995.

[10] F. Cucker, H. Diao, and Y. Wei. On mixed and componentwise condition numbers for Moore-Penrose inverse and linear least squares problems. *Mathematics of Computation*, 76(258):947–963, 2007.

[11] L. Eldén. Perturbation theory for the least squares problem with linear equality constraints. *SIAM J. Numerical Analysis*, 17:338–350, 1980.

[12] A. J. Geurts. A contribution to the theory of condition. *Numerische Mathematik*, 39:85–96, 1982.

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, 1996. Third edition.

[14] G. H. Golub and J. H. Wilkinson. Note on the iterative refinement of least squares solution. *Numerische Mathematik*, 9(2):139–148, 1966.

[15] S. Gratton. On the condition number of linear least squares problems in a weighted Frobenius norm. *BIT*, 36:523–530, 1996.

[16] J. F. Grcar. Adjoint formulas for condition numbers applied to linear and indefinite least squares. Technical Report LBNL-55221, Lawrence Berkeley National Laboratory, 2004.

[17] T. Gudmundsson, C. S. Kenney, and A. J. Laub. Small-sample statistical estimates for matrix norms. *SIAM J. Matrix Analysis and Applications*, 16(3):776–792, 1995.

[18] N. J. Higham. A survey of componentwise perturbation theory in numerical linear algebra. In W. Gautschi, editor, *Mathematics of Computation 1943–1993: A Half Century of Computational Mathematics*, volume 48 of *Proceedings of Symposia in Applied Mathematics*, pages 49–77. American Mathematical Society, Providence, 1994.

[19] N. J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, Philadelphia, 2002. Second edition.

[20] I. C. F. Ipsen. *Numerical Matrix Analysis: Linear Systems and Least Squares.* SIAM, Philadelphia, 2009.

[21] C. S. Kenney and A. J. Laub. Small-sample statistical condition estimates for general matrix functions. *SIAM J. Sci. Comput.*, 15(1):36–61, 1994.

[22] C. S. Kenney, A. J. Laub, and M. S. Reese. Statistical condition estimation for linear least squares. *SIAM J. Matrix Analysis and Applications*, 19(4):906–923, 1998.

[23] University of Tennessee. *PLASMA Users' Guide, Parallel Linear Algebra Software for Multicore Architectures, Version 2.3.* 2010.

[24] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8(1):43–71, 1982.

[25] J. Rice. A theory of condition. *SIAM J. Numerical Analysis*, 3:287–310, 1966.

[26] The Numerical Algorithms Group. *NAG Library Manual, Mark 21.* NAG, 2006.

[27] S. Tomov, J. Dongarra, and M. Baboulin. Towards dense linear algebra for hybrid GPU accelerated manycore systems. *Parallel Computing*, 36(5&6):232–240, 2010.

[28] P.-Å. Wedin. Perturbation theory for pseudo-inverses. *BIT*, 13:217–232, 1973.

[29] J. H. Wilkinson. *Rounding errors in algebraic processes*, volume 32. Her Majesty's Stationery Office, London, 1963.

[30] M. Zelen. Linear estimation and related topics. In J.Todd, editor, *Survey of numerical analysis*, pages 558–584. McGraw-Hill, New York, 1962.

Contents

1 Introduction **3**

2 Condition estimation for linear least squares **4**
2.1 Conditioning of the least squares solution 4
2.2 Componentwise condition estimates 6

3 Numerical experiments **7**
3.1 Accuracy of statistical estimates 7
3.1.1 Conditioning of LLS solution 7
3.1.2 Componentwise condition estimation 8
3.2 Computing least squares condition numbers on multicore-GPU architectures . . . 9
3.2.1 Variance-covariance matrix 9
3.2.2 Implementation details 10
3.2.3 Performance results 11

4 Conclusion **11**

RESEARCH CENTRE
SACLAY – ÎLE-DE-FRANCE

1 rue Honoré d'Estienne d'Orves
Bâtiment Alan Turing
Campus de l'École Polytechnique
91120 Palaiseau

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399