Journal of Scientific Computing (2024) 98:55
https://doi.org/10.1007/s10915-023-02447-4

Double Precision is not Necessary for LSQR for Solving Discrete Linear Ill-Posed Problems

Haibo Li[1]

Received: 15 April 2023 / Revised: 1 November 2023 / Accepted: 21 December 2023

Abstract
The growing availability and usage of low precision floating point formats attracts many interests of developing lower or mixed precision algorithms for scientific computing problems. In this paper we investigate the possibility of exploiting mixed precision computing in LSQR for solving discrete linear ill-posed problems. Based on the commonly used regularization model for linear inverse problems, we analyze the choice of proper computing precision in the two main parts of LSQR, including the construction of Krylov subspace and updating procedure of iterative solutions. We show that, under some mild conditions, the Lanczos vectors can be computed using single precision without loss of any accuracy of the final regularized solution as long as the noise level is not extremely small. We also show that the most time consuming part for updating iterative solutions can be performed using single precision without sacrificing any accuracy. The results indicate that several highly time consuming parts of the algorithm can be implemented using lower precisions, and provide a theoretical guideline for implementing a robust and efficient mixed precision variant of LSQR for solving discrete linear ill-posed problems. Numerical experiments are made to test two mixed precision variants of LSQR and confirming our results.

Keywords Mixed precision · Linear ill-posed problem · Regularization · LSQR · Roundoff unit · Semi-convergence

Mathematics Subject Classification 65F22 · 65F10 · 65G50

1 Introduction

Although for most traditional scientific computing problems, computations are carried out with double precision (64-bit) rather than lower precisions such as single (32-bit) or half

This work was supported in part by the National Natural Science Foundation of China under Grant No. 3192270206.

✉ Haibo Li
haibolee1729@gmail.com

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Published online: 01 February 2024

(16-bit) precision, on modern computing architectures, the performance of 32-bit operations is often at least twice as fast as that of 64-bit operations [1], which stimulates the trial of using lower precision floating point formats in an algorithm. To exploit this computation power without sacrificing accuracy of the final result, numerical algorithms have to be designed that use lower or mixed precision formats. By using a combination of 64-bit and 32-bit (even 16-bit) floating point arithmetic, the performance of many numerical algorithms can be significantly enhanced while maintaining the 64-bit accuracy of the resulting solution. New mixed precision variants of many numerical linear algebra algorithms have been recently proposed, such as matrix multiplications [2, 6], LU and QR matrix factorizations [32, 44], Krylov solvers [8, 11, 17, 18] and many others [3, 25].

In this paper, we investigate how to exploit mixed precision computing for solving discrete linear ill-posed problems. This type of problems typically arise from the numerical solution of inverse problems that appear in various applications of science and engineering, such as image deblurring, geophysics, computerized tomography and many others; see e.g., [23, 28, 35, 40]. A basic linear inverse problem leads to a discrete linear system of the form

$$Ax = b, \quad b = Ax_{ex} + e, \tag{1.1}$$

where the matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ without loss of generality, and the right-hand side b is a perturbed version of the unknown exact observation $b_{ex} = Ax_{ex}$. In this paper we suppose e is a Gaussian white noise. The problem is ill-posed in the sense that A is extremely ill-conditioned with its singular values decaying gradually towards zero without any noticeable gap, which leads to that the naive solution $x_{nai} = A^{\dagger}b$ of 1.1 is a poor approximation to the exact solution $x_{ex} = A^{\dagger}b_{ex}$, where "$\dagger$" denotes the Moore-Penrose inverse of a matrix. Therefore, some forms of regularization must be used to deal with the noise e in order to extract a good approximation to x_{ex}.

One of the popular regularization techniques is the Tikhonov regularization [41], in which a quadratic penalty is added to the objective function:

$$x_\lambda = \arg\min_{x\in\mathbb{R}^n} \{\|Ax - b\|^2 + \lambda\|Lx\|^2\},$$

where $\lambda > 0$ is the regularization parameter and $L \in \mathbb{R}^{p\times n}$ is the regularization matrix. Throughout the rest of the paper $\|\cdot\|$ always denotes either the vector or matrix 2-norm. The proper choice of L depends on the particular application, which should be chosen to yield a regularized solution with some known desired features of x_{ex}. A suitable value of λ should have a good balance between the data fidelity term $\|Ax - b\|$ and the regularization term $\|Lx\|$, only in which case we can get a regularized solution that is a good approximation to x_{ex}. Although many types of regularization parameter choice rules have been proposed, such as discrepancy principle (DP) [34], unbiased predictive risk estimator [42], generalized cross validation [16] and L-curve criterion [19], it is often computationally very expensive to choose a suitable λ for large scale problems, since many different values of λ must be tried to get x_λ.

For large scale ill-posed problems, iterative regularization method is the soundest choice. For standard-form regularization with $L = I_n$, Krylov subspace based methods such as LSQR [4, 14, 37] are the most commonly used. The methods project (1.1) onto a sequence of lower-dimensional Krylov subspaces, and then solves the projected small scale problems, where the iteration number plays the role of regularization parameter [20, 22]. This approach usually exhibits semi-convergence: as the iteration proceeds, the iterative solution first approximates x_{ex} while afterwards the noise e starts to deteriorate the solution so that it gradually diverge from x_{ex} and instead converges to x_{nai}. Therefore, the iteration must be stopped early properly

by using a regularization parameter choice rule [7, 38]. The semi-convergence behavior can be mitigated by using a hybrid method, which applies a standard regularization technique, such as Tikhonov regularization or truncated SVD, to the projected problem at each iteration [10, 29, 39].

In this paper, we focus on the LSQR algorithm for iteratively solving large scale discrete linear ill-posed problems that is based on the Lanczos bidiagonalization [37]. The motivation for this work is to answer whether lower precisions can be used in some parts of the algorithm while maintaining the 64-bit accuracy of the regularized solutions. The LSQR for linear ill-posed problems mainly includes two parts, that is the construction of Krylov subspace by Lanczos bidiagonalization and updating procedure of iterative solutions. In addition, a proper iteration number should be estimated to stop the iteration near the semi-convergence point. Since b is contaminated by the noise e, we can never get a regularized solution with error as small as $\|e\|$ [12, 20], and this error is usually much bigger than the roundoff unit of double precision with the value 2^{-53}. This fact inspires us that double precision may be not necessary for LSQR to compute a regularized solution with the same accuracy as the best regularized one. To the best of our knowledge, however, there is still no theoretical analysis about how to choose proper lower computing precision in LSQR for linear ill-posed problems. This issue is crucial for deciding which lower precision format should be used in each part of the algorithm to get a more efficient mixed precision implementation.

We study the lower precision computing for the two main parts of LSQR, including the construction of Krylov subspace and updating iterative solutions. In finite precision arithmetic, we implement the Lanczos bidiagonalization in LSQR with full reorthogonalization of Lanczos vector, which is a frequently used strategy to avoid slowing down and irregular convergence of iterative solutions [26, 30]. First, under an ideal model describing the linear ill-posed problem (1.1), our result estimates an upper bound on the proper value of $\mathbf{u}$ corresponding to the used computing precision for constructing Lanczos vectors with full reorthogonalization, and it indicates that for not extremely small noise levels we can exploit single precision for this part without loss of accuracy of the final regularized solution. Second, for the updating procedure part, we theoretically show that, under a condition which can be almost always satisfied, the updated regularized solutions can be computed using single precision without sacrificing any accuracy. We also investigate the L-curve and discrepancy principle methods for estimating the optimal early stopping iteration combined with the mixed precision implementation of LSQR. Overall, the results theoretically show that several highly time consuming parts of LSQR for solving discrete linear ill-posed problems can be implemented using lower precisions, which has a great potential of defeating the double precision implementation in computation efficiency. The results can guide us towards a mixed precision implementation that is both robust and efficient, and should be considered by application developers for practical problems.

The paper is organized as follows. We start in Sect. 2 with a brief review of regularization theory and algorithm of linear ill-posed inverse problems, and the IEEE 754 floating point standard. Next, in Sect. 3, we analyze the proper choice of $\mathbf{u}$ corresponding to the used computing precision for constructing Lanczos vectors with full reorthogonalization and give an upper bound. In Sect. 4, we analyze the possibility of using lower precision for updating iterative solutions and give a mixed precision variant of LSQR. We also discuss the estimation of optimal early stopping iteration for the mixed precision LSQR. Some numerical experimental results are presented in Sect. 5 and we conclude the paper in Sect. 6.

2 Preliminaries

In this section, we review some basic knowledge of regularization theory of linear ill-posed inverse problems and the LSQR regularization algorithm, we also review finite precision computing based on the IEEE 754 Standard floating point number system.

2.1 Regularization of Linear Ill-Posed Problems and LSQR

Suppose the singular value decomposition (SVD) of A is

$$A = U \begin{pmatrix} \Sigma \\ \mathbf{0} \end{pmatrix} V^T, \tag{2.1}$$

where $U = (u_1, \ldots, u_m) \in \mathbb{R}^{m\times m}$ and $V = (v_1, \ldots, v_n) \in \mathbb{R}^{n\times n}$ are orthogonal, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{n\times n}$ with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$. The naive solution to (1.1) is

$$x_{nai} = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i = \sum_{i=1}^{n} \frac{u_i^T b_{ex}}{\sigma_i} v_i + \sum_{i=1}^{n} \frac{u_i^T e}{\sigma_i} v_i = x_{ex} + \sum_{i=1}^{n} \frac{u_i^T e}{\sigma_i} v_i.$$

Note that the second term in x_{nai} is extremely large since σ_i decay to zero, making x_{nai} a meaningless solution. A direct regularized method is the truncated SVD (TSVD) method, which forms x_k^{tsvd} by truncating the first k components of x_{nai} corresponding to large singular values: $x_k^{tsvd} = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i} v_i$. Regularization theory of linear inverse problems can be used to investigate the accuracy of the regularized solution to (1.1), among which the *discrete Picard condition(DPC)* plays a central role. We give a brief review in the following.

The DPC for the exact right-hand side b_{ex} can be written in the following popular simplifying model [20, 22]:

$$|u_i^T b_{ex}| = \rho_0 \sigma_i^{1+\beta}, \quad \beta > 0, \ i = 1, 2, \ldots, n, \tag{2.2}$$

where β is a model parameter that controls the decay rates of $|u_i^T b_{ex}|$,[1]. The DPC implies that the noisy coefficients $u_i^T b$ gradually decay in average and are larger than $|u_i^T e|$ for $i = 1, 2 \ldots$ until the noise dominates. Suppose that the noise in $|u_i^T b|$ starts to dominate at $k = k^* + 1$, i.e., k^* is the transition point such that [22, §3.5.1]

$$|u_{k^*}^T b| \approx |u_{k^*}^T b_{ex}| > |u_{k^*}^T e|, \quad |u_{k^*+1}^T b_{ex}| \approx |u_{k^*+1}^T e|. \tag{2.3}$$

Under the assumption that $e \in \mathbb{R}^m$ is a Gaussian white noise, it is shown that $|u_i^T e| \approx m^{-1/2}\|e\|$ [22, §3.5.1]. Thus the DPC for noisy b can be written in the following form:

$$|u_i^T b| = |u_i^T b_{ex} + u_i^T e| \approx \begin{cases} \rho_0 \sigma_i^{1+\beta}, & 1 \leq i \leq k^*; \\ m^{-1/2}\|e\|, & i \geq k^* + 1. \end{cases} \tag{2.4}$$

The accuracy of the best regularized solution to (1.1) is closely connected with the DPC (2.4), which can be reflected by the *effective resolution limit* denoted by η_{res}. The effective resolution limit of (1.1) denotes the smallest coefficient $|v_i^T x_{ex}|$ that can be recovered from the given A and noisy b [20, §4.1], and it is shown in [20, §4.5] that $\eta_{res} \approx (m^{-1/2}\|e\|)^{\frac{\beta}{1+\beta}}$.

[1] In [22, §4.6] the corresponding model is $|u_i^T b_{ex}| = \sigma_i^{1+\beta}$, which does not include the constant ρ_0. In fact, Hansen [22, p.68] points out that "while this is, indeed, a crude model, it reflects the overall behavior often found in real problems".

The accuracy of the best regularized solution x_{opt} is dependent on η_{res}, that is, one can only hope that x_{opt} reaches an accuracy corresponding to the effective resolution limit, which means

$$\frac{\|x_{opt} - x_{ex}\|}{\|x_{ex}\|} \geq C_1 \varepsilon^{\frac{\beta}{1+\beta}} \tag{2.5}$$

with a moderate constant C_1, where $\varepsilon = \|e\|/\|b_{ex}\| < 1$ is the noise level; see e.g., [12, 20] for more details. Note that $\|e\| < \|b_{ex}\|$; otherwise all information from b_{ex} is lost in b. In particular, it is known from [12, 20] that $x_{k^*}^{tsvd}$ is a 2-norm filtering best possible regularized solution of (1.1) when only deterministic 2-norm filtering regularization methods are taken into account.

For large scale ill-posed problems, the LSQR algorithm with early stopping is the most common used iterative regularization method, which is based on Lanczos bidiagonalization of A with starting vector b, as described in Algorithm 1.

Algorithm 1 k-step Lanczos bidiagonalization

Let $\beta_1 = \|b\|,\ p_1 = b/\beta_1$
$\alpha_1 = \|A^T p_1\|,\ q_1 = A^T p_1/\alpha_1$
for $j = 1, 2, \ldots, k$ **do**
  $s_j = Aq_j - \alpha_j p_j$
  $\beta_{j+1} = \|s_j\|,\ p_{j+1} = s_j/\beta_{j+1}$
  $r_j = A^T p_{j+1} - \beta_{j+1} q_j$
  $\alpha_{j+1} = \|r_j\|,\ q_{j+1} = r_j/\alpha_{j+1}$
end for

In exact arithmetic, the k-step Lanczos bidiagonalization produces a lower bidiagonal matrix

$$B_k = \begin{pmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \beta_3 & \ddots & \\ & & \ddots & \alpha_k \\ & & & \beta_{k+1} \end{pmatrix} \in \mathbb{R}^{(k+1)\times k},$$

and two groups of Lanczos vectors $\{p_1, \ldots, p_{k+1}\}$ and $\{q_1, \ldots, q_{k+1}\}$ that are orthonormal bases of Krylov subspaces $\mathcal{K}_{k+1}(AA^T, b)$ and $\mathcal{K}_{k+1}(A^T A, A^T b)$, respectively. The k-step Lanczos bidiagonalization can be written in the matrix form

$$P_{k+1}(\beta_1 e_1^{(k+1)}) = b, \tag{2.6}$$

$$AQ_k = P_{k+1} B_k, \tag{2.7}$$

$$A^T P_{k+1} = Q_k B_k^T + \alpha_{k+1} q_{k+1} (e_{k+1}^{(k+1)})^T, \tag{2.8}$$

where $e_i^{(l)}$ denotes the i-th canonical basis vector of $\mathbb{R}^l$, and $P_{k+1} = (p_1, \ldots, p_{k+1})$ and $Q_k = (q_1, \ldots, q_{k+1})$ are two orthonormal matrices. The LSQR for (1.1) is mathematically equivalent to the conjugate gradient (CG) method applied to the normal equation of $\min_{x\in\mathbb{R}^n} \|Ax - b\|$, i.e. $A^T Ax = A^T b$, which seeks approximations to x_{ex} from the k dimensional Krylov subspace $\mathcal{K}_k(A^T A, A^T b) = \mathcal{R}(V_k)$ starting with $k = 1$ onwards, and

the iteration should be terminated at a proper step near the semi-convergence point to get a good regularized solution [4]. At the k-th step, by (2.6) and (2.7) we have

$$\min_{x=Q_k y} \|Ax - b\| = \min_{y\in\mathbb{R}^k} \|B_k y - \beta_1 e_1^{(k+1)}\|,$$

and thus the k-step LSQR solution is

$$x_k = Q_k y_k, \quad y_k = \arg\min_{y\in\mathbb{R}^k} \|B_k y - \beta_1 e_1^{(k+1)}\| = B_k^{\dagger}(\beta_1 e_1^{(k+1)}). \tag{2.9}$$

We note that if α_{k+1} or β_{k+1} is zero, then the iteration terminates and $x_k = x_{nai}$ [37]. This case rarely happens in real computations and we assume that the iteration does not terminate.

From the above description, the computation of x_k can be divided into two parts. The first part is the Lanczos bidiagonalization that generates two orthonormal bases of Krylov subspaces $\mathcal{K}_{k+1}(AA^T, b)$ and $\mathcal{K}_{k+1}(A^TA, A^Tb)$, respectively, while the second part is solving the projected problem (2.9) to obtain x_k. In the practical implementation, there is a recursive formula to update x_{k+1} from x_k without solving the projected least squares problems at each iteration. This updating procedure will be investigated in Sect. 4.

2.2 Finite Precision Computing

In practical computational tasks, the accuracy of a computed result and the time needed to complete the algorithm both heavily depend on the floating point format used for storage and arithmetic operations. Here we review the IEEE 754 Standard floating point number format, which is composed of a sign bit, an exponent η, and a significand t:

$$x = \pm\mu \times 2^{\eta - t},$$

where μ is any integer in $[0, 2^t - 1]$ and η is an integer in $[\eta_{\min}, \eta_{\max}]$. Roughly speaking, the length of the exponent determines the value range of a floating point format, and the length of the significand determines the relative accuracy of the format in that range. A short analysis of floating point operations [24, Theorem 2.2] shows that the relative error is controlled by the roundoff unit $\mathbf{u} := \frac{1}{2} \cdot 2^{1-t}$. Table 1 shows main parameters of the three different floating point number formats.

In finite precision arithmetic, the Lanczos vectors u_i and v_i computed by the Lanczos bidiagonalization gradually lose their orthogonality, and it may slow down the convergence of iterative solutions and make the propagation of noise during the iterations rather irregular [26, 33]. A frequently used strategy is implementing the Lanczos bidiagonalization with full reorthogonalization (LBFRO)[2] to maintain stability of convergence. In the rest of the paper, we investigate the *LSQR implemented with full reorthogonalization of Lanczos vectors* in finite precision. From now on, notations such as P_k, B_k, α_k, etc. denote the computed quantities in finite precision computing.

Define the orthogonality level of Lanczos vectors $\{p_1, \ldots, p_k\}$ and $\{q_1, \ldots, q_k\}$ as

$$\mu_k = \|\mathbf{SUT}(I_k - P_k^T P_k)\|, \quad \nu_k = \|\mathbf{SUT}(I_k - Q_k^T Q_k)\|,$$

where $\mathbf{SUT}(\cdot)$ denotes the strictly upper triangular part of a matrix. The following result has been established for the k-step Lanczos bidiagonalization with reorthogonalization (not necessarily full reorthogonalization) [31].

[2] In full reorthogonalization, u_k and v_k are reorthogonalized against all previous vectors $\{u_1, \ldots, u_{k-1}\}$ and $\{v_1, \ldots, v_{k-1}\}$ as soon as they have been computed. This adds an arithmetic cost of about $4(m+n)k^2$ flops, which is affordable if $k \ll \min\{m, n\}$.

Table 1 Parameters for various floating-point formats. "Range" denotes the order of magnitude of the smallest positive (subnormal) x_{min}^s and smallest and largest positive normalized floating-point numbers

Type	Size (bits)	Range x_{min}^s	x_{min}	x_{max}	Roundoff unit $\mathbf{u}$
Half precision	16	5.96×10^{-8}	6.10×10^{-5}	6.55×10^{4}	4.88×10^{-4}
Single precision	32	1.40×10^{-45}	1.18×10^{-38}	3.40×10^{38}	5.96×10^{-8}
Double precision	64	4.94×10^{-324}	2.22×10^{-308}	1.80×10^{308}	1.11×10^{-16}

Theorem 2.1 *For the k-step Lanczos bidiagonalization with reorthogonalization, if* $\nu_{k+1} < 1/2$ *and* $\mu_{k+1} < 1/2$, *then there exist two orthornormal matrices* $\bar{P}_{k+1} = (\bar{p}_1, \ldots, \bar{p}_{k+1}) \in \mathbb{R}^{m\times(k+1)}$ *and* $\bar{Q}_{k+1} = (\bar{q}_1, \ldots, \bar{q}_{k+1}) \in \mathbb{R}^{n\times(k+1)}$ *such that*

$$\bar{P}_{k+1}(\beta_1 e_1^{(k+1)}) = b + \delta_b, \tag{2.10}$$

$$(A + E)\bar{Q}_k = \bar{P}_{k+1}B_k, \tag{2.11}$$

$$(A + E)^T \bar{P}_{k+1} = \bar{Q}_k B_k^T + \alpha_{k+1}\bar{q}_{k+1}(e_{k+1}^{(k+1)})^T, \tag{2.12}$$

where E *and* δ_b *are perturbation matrix and vector, respectively. We have error bounds*

$$\|\bar{P}_{k+1} - P_{k+1}\| \le 2\mu_{k+1} + \mathcal{O}(\mu_{k+1}^2), \quad \|\bar{Q}_{k+1} - Q_{k+1}\| \le \nu_{k+1} + \mathcal{O}(\nu_{k+1}^2),$$

and

$$\|E\| = \mathcal{O}(c(n,k)\|A\|(\mathbf{u} + \nu_{k+1} + \mu_{k+1})), \quad \|\delta_b\| = O(\|b\|\mathbf{u}).$$

where $c(n,k)$ *is a moderately growing constant depends on* n *and* k.

For the LBFRO, the orthogonality levels of u_i and v_i are kept around $\mathcal{O}(\mathbf{u})$, thus by Theorem 2.1 we have

$$\|\bar{P}_{k+1} - P_{k+1}\| = \mathcal{O}(\mathbf{u}), \quad \|\bar{Q}_{k+1} - Q_{k+1}\| = \mathcal{O}(\mathbf{u}), \tag{2.13}$$

$$\|E\| = \mathcal{O}(c(n,k)\|A\|\mathbf{u}), \quad \|\delta_b\| = \mathcal{O}(\|b\|\mathbf{u}). \tag{2.14}$$

This result will be used in the next section to estimate upper bound on the proper value of roundoff unit $\mathbf{u}$ corresponding to the used computing precision.

3 Choice of Computing Precision for the Construction of Krylov Subspace

In this section, we investigate which lower precision format should be used for computing Lanczos vectors with full reorthogonalization. To this end, by assuming the LBFRO is implemented in finite precision computing with roundoff unit $\mathbf{u}$, we will give an upper bound on $\mathbf{u}$ such that the best computed regularized solution can achieve the same accuracy as the best regularized LSQR solution to (1.1) obtained in exact arithmetic. In this section, the updating procedure or (2.9) is assumed to be implemented in exact arithmetic.

Theorem 3.1 *Suppose that the LBFRO in LSQR is implemented in finite precision with roundoff unit* $\mathbf{u}$. *If* (2.9) *is solved exactly, then the computed* x_k *satisfies*

$$\frac{\|x_k - \bar{x}_k\|}{\|\bar{x}_k\|} = \mathcal{O}(\mathbf{u}), \tag{3.1}$$

where $\bar{x}_k$ is the exact k-th LSQR solution to the perturbed problem

$$\min_{x\in\mathbb{R}^n} \|(A+E)x-(b+\delta_b)\|. \tag{3.2}$$

Proof Since (2.9) is solved exactly, by Theorem 2.1 we have

$$\begin{aligned} y_k &= \arg\min_{y\in\mathbb{R}^k} \|B_k y-\beta_1 e_1^{(k+1)}\| = \arg\min_{y\in\mathbb{R}^k} \|\bar{P}_{k+1}B_k y-\bar{P}_{k+1}\beta_1 e_1^{(k+1)}\| \\ &= \arg\min_{y\in\mathbb{R}^k} \|(A+E)\bar{Q}_k y-(b+\delta_b)\|. \end{aligned}$$

From Theorem 2.1 we know that $\bar{Q}_k$ is the right orthonormal matrix generated by the Lanczos biagonalization of $A+E$ with starting vector $b+\delta_b$ in exact arithmetic, which means $\mathcal{R}(\bar{V}_k)=\mathcal{K}_k((A+E)^T(A+E),(A+E)^T(b+\delta_b))$. Let $\bar{x}_k=\bar{Q}_k y_k$. Then $\bar{x}_k$ is the k-th LSQR solution to (3.2). Since $x_k=Q_k y_k$, by (2.13) we have

$$\|\bar{x}_k-x_k\| \le \|Q_k-\bar{Q}_k\|\|y_k\| = \mathcal{O}(\|y_k\|\mathbf{u})).$$

Using $\|y_k\|=\|\bar{x}_k\|$, we obtain (3.1). □

This result indicates that $x_k \approx \bar{x}_k$ within $\mathcal{O}(\mathbf{u})$. If we rewrite (3.2) as

$$(A+E)x=(b_{ex}+Ex_{ex})+(e-Ex_{ex}+\delta_b), \tag{3.3}$$

then x_{ex} is the exact solution to $(A+E)x_{ex}=b_{ex}+Ex_{ex}$ and $e-Ex_{ex}+\delta_b$ is the noise term. Notice that $\|E\|/\|A\|=\mathcal{O}(c(n,k)\mathbf{u})$ and thus for a not big k the singular values of $A+E$ decrease monotonically without noticeable gap until they tend to settle at a level of $\mathcal{O}(\|A\|\mathbf{u})$. Therefore, the linear system (3.3) inherits the ill-posedness of (1.1). Moreover, if $\|-Ex_{ex}+\delta_b\| \ll \|e\|$, then $e-Ex_{ex}+\delta_b$ can be treated as a Gaussian noise. For a sufficiently small $\mathbf{u}$, we can hope that the best LSQR regularized solution to (3.3) has the same accuracy as that to (1.1).

Suppose that the best LSQR regularized solution to (3.3) is $\bar{x}_{k_0}$. Let the i-th largest singular values of $A+E$ be $\bar{\sigma}_i$ and the corresponding left singular vector be $\bar{u}_i$. Applying the regularizaton theory and DPC to (1.1) and (3.3), the accuracy of $\bar{x}_{k_0}$ will be the same as that of x_{opt} if:

- the DPC of (3.3) inherits properties of the DPC of (1.1), i.e., the DPC of (3.3) before the noise dominates should satisfies

$$|\bar{u}_i^T b| \approx \rho_0 \bar{\sigma}_i^{1+\beta}, \quad 1\le i\le k^*; \tag{3.4}$$

- the effective resolution limit of (3.3) has the same accuracy as that of (1.1),[3].

Note that Theorem 3.1 implies that $\|x_{k_0}-\bar{x}_{k_0}/\|\bar{x}_{k_0}\|=\mathcal{O}(\mathbf{u})$. Therefore, for the above $\bar{x}_{k_0}$, the corresponding x_{k_0} will have the same accuracy as x_{opt} if

$$\mathbf{u} \ll C_1 \varepsilon^{\frac{\beta}{1+\beta}}, \tag{3.5}$$

which implies that the best computed regularized solution among x_k can achieve the same accuracy as x_{opt}.

In the following we make some analysis about the above assertions.

[3] The assertion implicitly uses the order optimal property of the LSQR or its mathematical equivalent CG algorithm for linear inverse problems [12, §7.3] which means that the best LSQR regularized solution can achieve the same accuracy as that of x_{opt}. Although plenty of numerical results confirm it without any exception, this property has not been rigorously proved for discrete linear ill-posed problems; see [20, §6.3]and [9] for more discussions.

Lemma 3.1 *For the ill-posed problem* (3.2) *or its equivalence* (3.3)*, if* (3.4) *holds and the following two conditions are satisfied:* (**1**). $\|-Ex_{ex}+\delta_b\| \ll \|e\|$*;* (**2**). $\|E\| \ll \sigma_{k^*}$*, then the effective resolution limit of* (3.3) *has the same accuracy as that of* (1.1)*.*

Proof Since the noise in problem (3.3) is $e - Ex_{ex} + \delta_b$, by Condition (**1**), the noise e dominates, thus we can regard this noise as Gaussian. Condition (**2**) implies that $\bar{\sigma}_i \approx \sigma_i$ until the noise in $|u_i^T b|$ starts to dominates, and thus the errors in $A + E$ starts to dominates at a point $\bar{k}^* > k^*$. Therefore, it follows from [20, §4.5], if (3.4) holds, that

$$\bar{\eta}_{res} \approx (m^{-1/2}\|e - Ex_{ex} + \delta_b\|)^{\frac{\beta}{1+\beta}} \approx (m^{-1/2}\|e\|)^{\frac{\beta}{1+\beta}},$$

where $\bar{\eta}_{res}$ is the effective resolution limit of (3.3). □

Lemma 3.2 *For the iteration number k not very big, if* $\mathbf{u}$ *satisfies*

$$\mathbf{u} \ll \min\{\varepsilon, (m^{-1/2}\varepsilon)^{\frac{1}{1+\beta}}\}, \tag{3.6}$$

then the relation (3.5) *and Conditions* (**1**) *and* (**2**) *hold.*

Proof Condition (**1**) holds if $\|Ex_{ex}\| \ll \|e\|$ and $\|\delta_b\| \ll \|e\|$. By (2.14), these two equalities can be satisfied if $\mathbf{u} \ll \|e\|/\|b\|$ and $\|E\| \ll \|e\|/\|x_{ex}\|$. Since

$$\frac{\|e\|}{\|b\|} \geq \frac{\|e\|}{\|b_{ex}\| + \|e\|} = \frac{\varepsilon}{1+\varepsilon} > \frac{\varepsilon}{2},$$

we have $\mathbf{u} \ll \|e\|/\|b\|$ if $\mathbf{u} \ll \varepsilon$. Note that

$$\frac{\|e\|}{\|x_{ex}\|} \leq \frac{\|A\|\|e\|}{\|b_{ex}\|} = \varepsilon\|A\|,$$

and the value of $\|e\|/\|x_{ex}\|$ should not deviate too far from $\varepsilon\|A\|$ since $\|x_{ex}\|$ is usually a moderate quantity. Using $\|E\| = \mathcal{O}(c(n,k)\|A\|\mathbf{u})$, we have $\|E\| \ll \|e\|/\|x_{ex}\|$ if $\mathbf{u} \ll \varepsilon$ since $c(n,k)$ is a moderate quantity.

By (2.2) and (2.3), we have $\rho_0\sigma_{k^*+1}^{1+\beta} \approx |u_{k^*+1}^T e| \approx m^{-1/2}\|e\|$, which implies that Condition (**2**) will hold if

$$\|E\| \ll \sigma_{k^*+1} \approx (m^{-1/2}\rho_0^{-1}\|e\|)^{\frac{1}{1+\beta}}. \tag{3.7}$$

By (2.2), we have $\rho_0 = |u_1^T b_{ex}|/\sigma_1^{1+\beta}$, and thus

$$(\rho_0^{-1}\|e\|)^{\frac{1}{1+\beta}} = \sigma_1\Big(\frac{\|e\|}{|u_1^T b_{ex}|}\Big)^{\frac{1}{1+\beta}} \geq \|A\|\Big(\frac{\|e\|}{\|b_{ex}\|}\Big)^{\frac{1}{1+\beta}} = \varepsilon^{\frac{1}{1+\beta}}\|A\|. \tag{3.8}$$

Therefore Condition (**2**) will hold if $\|E\| \ll (m^{-1/2}\varepsilon)^{\frac{1}{1+\beta}}\|A\|$, which can be satisfied if $\mathbf{u} \ll (m^{-1/2}\varepsilon)^{\frac{1}{1+\beta}}$. By the above derivations, one can check that (3.5) and Conditions (**1**) and (**2**) hold if $\mathbf{u}$ satisfies (3.6). □

In order to analyze (3.4), we adopt the following popular model describing the decay rates of σ_i for different types of ill-posedness [20]:

$$\sigma_i = \begin{cases} \zeta\rho^{-i}, & \rho > 1 \quad \text{severely ill-posed;} \\ \zeta i^{-\alpha}, & \alpha > 1 \quad \text{moderately ill-posed;} \\ \zeta i^{-\alpha}, & 1/2 < \alpha \leq 1 \quad \text{mildly ill-posed.} \end{cases} \tag{3.9}$$

Remark 3.1 Notice that the model (3.9) means that all the singular values of A are simple. For the case that A has multiple singular values, the model should be rewritten by the following modification; see [27] for using this modified model to analyze regularization effect of LSQR for the multiple singular values case. First rewrite the SVD of A as

$$A = \widehat{U}\begin{pmatrix}\Sigma\\ \mathbf{0}\end{pmatrix}\widehat{V}^T,$$

where $\widehat{U} = (\widehat{U}_1, \ldots, \widehat{U}_r, \widehat{U}_\perp)$ with $\widehat{U}_i \in \mathbb{R}^{m\times l_i}$ and $\widehat{V} = (\widehat{V}_1, \ldots, \widehat{V}_r)$ with $\widehat{V}_i \in \mathbb{R}^{n\times l_i}$ are column orthonormal, $\Sigma = \mathrm{diag}(\hat{\sigma}_1 I_{l_1}, \ldots, \hat{\sigma}_r I_{l_r})$ with the r distinct singular values $\hat{\sigma}_1 > \hat{\sigma}_2 > \cdots > \hat{\sigma}_r > 0$, each $\hat{\sigma}_i$ is l_i multiple and $l_1 + l_2 + \cdots + l_r = n$. Then the decay rate of $\hat{\sigma}_i$ can be written in the same form as (3.9). In this case, the DPC of (1.1) becomes

$$\|\widehat{U}_i^T b_{ex}\| = \rho_0 \hat{\sigma}_i^{1+\beta}, \quad i = 1, 2, \ldots, r, \tag{3.10}$$

which states that, on average the (generalized) Fourier coefficients $\|\widehat{U}_i^T b_{ex}\|$ decay faster than $\hat{\sigma}_i$.

Using the above model, we can give a sufficient condition under which the relation (3.4) holds. For notational simplicity, we also write $\hat{\sigma}_i$ as σ_i without causing confusions.

Lemma 3.3 *Suppose that the iteration number k is not very big and $\sigma_i - \sigma_{i+1} \gg \|E\|$ for $1 \le i \le k^*$. If (3.6) holds and*

$$\mathbf{u} \ll (m^{-1/2}\varepsilon)^{\frac{2+\beta}{1+\beta}} \left(\frac{\sigma_{k^*}}{\sigma_{k^*+1}} - 1\right), \tag{3.11}$$

then the relation (3.4) holds.

Proof There two cases needed to be proved.

Case 1. A has single singular values. Write the i-th left singular vector of $A + E$ as $\bar{u}_i = u_i + \delta_{u_i}$ where δ_{u_i} is an error vector. Since (3.6) holds, we have $\bar{\sigma}_i \approx \sigma_i$ for $1 \le i \le k^*$. Note (3.4) implies $|(u_i + \delta_{u_i})^T b| \approx \rho_0 \bar{\sigma}_i^{1+\beta} \approx \rho_0 \sigma_i^{1+\beta}$ for $1 \le i \le k^*$, which can be satisfied if

$$|\delta_{u_i}^T b| \ll \rho_0 \sigma_i^{1+\beta}, \quad 1 \le i \le k^*. \tag{3.12}$$

By the perturbation theorem of singular vectors [5, Theorem 1.2.8], we have the perturbation bound

$$|\sin\theta(u_i, \bar{u}_i)| \le \frac{\|E\|}{\sigma_i - \sigma_{i+1} - \|E\|} \approx \frac{\|E\|}{\sigma_i - \sigma_{i+1}}, \quad 1 \le i \le k^*$$

under the assumption that $\sigma_i - \sigma_{i+1} \gg \|E\|$ for $1 \le i \le k^*$, where $\theta(u_i, \bar{u}_i)$ is the angle between u_i and $\bar{u}_i$. Thus we have

$$\|\delta_{u_i}\| = 2|\sin(\theta(u_i, \bar{u}_i)/2)| \approx |\sin\theta(u_i, \bar{u}_i)| \lesssim \frac{\|E\|}{\sigma_i - \sigma_{i+1}}.$$

Therefore, (3.12) can be satisfied if

$$\frac{\|E\|\|b\|}{\sigma_i - \sigma_{i+1}} \ll \rho_0 \sigma_i^{1+\beta}, \quad 1 \le i \le k^*,$$

which is equivalent to

$$\|E\| \ll \frac{\rho_0 \sigma_i^{2+\beta}\left(1 - \frac{\sigma_{i+1}}{\sigma_i}\right)}{\|b\|}, \quad 1 \le i \le k^*. \tag{3.13}$$

Using the expression of σ_{k^*+1} in (3.7) and the model (3.9), the minimum of the right-hand term of the above inequality is achieved at $i = k^*$, which is

$$\begin{aligned}\frac{\rho_0\sigma_{k^*}^{2+\beta}\big(1-\frac{\sigma_{k^*+1}}{\sigma_{k^*}}\big)}{\|b\|} &= \frac{\rho_0\sigma_{k^*+1}^{2+\beta}\big(\frac{\sigma_{k^*}}{\sigma_{k^*+1}}\big)^{2+\beta}\big(1-\frac{\sigma_{k^*+1}}{\sigma_{k^*}}\big)}{\|b\|}\\ &\approx \frac{\rho_0(m^{-1/2}\rho_0^{-1}\|e\|)^{\frac{2+\beta}{1+\beta}}\big(\frac{\sigma_{k^*}}{\sigma_{k^*+1}}\big)^{1+\beta}\big(\frac{\sigma_{k^*}}{\sigma_{k^*+1}}-1\big)}{\|b_{ex}\|}\\ &\geq \frac{(m^{-1/2}\|e\|)^{\frac{2+\beta}{1+\beta}}\big(\frac{\sigma_{k^*}}{\sigma_{k^*+1}}-1\big)}{\|b_{ex}\|\rho_0^{\frac{1}{1+\beta}}}.\end{aligned}$$

By (3.8), we have

$$\begin{aligned}\frac{(m^{-1/2}\|e\|)^{\frac{2+\beta}{1+\beta}}}{\|b_{ex}\|\rho_0^{\frac{1}{1+\beta}}} &= (m^{-1/2})^{\frac{2+\beta}{1+\beta}}(\rho_0^{-1}\|e\|)^{\frac{1}{1+\beta}}\frac{\|e\|}{\|b_{ex}\|}\\ &\geq (m^{-1/2})^{\frac{2+\beta}{1+\beta}}\varepsilon^{\frac{1}{1+\beta}}\|A\|\varepsilon = (m^{-1/2}\varepsilon)^{\frac{2+\beta}{1+\beta}}\|A\|.\end{aligned}$$

Therefore, by (3.12) and (3.13) and using $\|E\| = \mathcal{O}(c(n,k)\|A\|\mathbf{u})$, we finally obtain the result.

Case 2. A has multiple singular values. Write the SVD of $A+E$ in a similar form as that of A, such that the left singular vectors can be written as $\bar{U} = (\bar{U}_1, \ldots, \bar{U}_r, \bar{U}_\perp)$. By the perturbation theorem of invariant singular subspaces [43], we have

$$\|\sin\Theta(\widehat{U}_i, \bar{U}_i)\| \leq \frac{\|E\|}{\hat{\sigma}_i - \hat{\sigma}_{i+1} - \|E\|} \approx \frac{\|E\|}{\hat{\sigma}_i - \hat{\sigma}_{i+1}}$$

where $\|\sin\Theta(\widehat{U}_i, \bar{U}_i)\| = \|\widehat{U}_i\widehat{U}_i^T - \bar{U}_i\bar{U}_i^T\|$ is the angle measure between subspaces spanned by $\widehat{U}_i$ and $\bar{U}_i$ [15, §2.5]. Notice that

$$\begin{aligned}|\|\bar{U}_i^T b\| - \|\widehat{U}_i^T b\|| &= |\|\bar{U}_i\bar{U}_i^T b\| - \|\widehat{U}_i\widehat{U}_i^T b\|| \leq \|(\bar{U}_i\bar{U}_i^T - \widehat{U}_i\widehat{U}_i^T)b\|\\ &\leq \|b\|\,\|\sin\Theta(\widehat{U}_i, \bar{U}_i)\|,\end{aligned}$$

where $|\|\bar{U}_i^T b\| - \|\widehat{U}_i^T b\||$ is the corresponding version of the left-hand term of (3.12). Using the same approach as that for analyzing (3.12), we can obtain the result for the multiple singular values case. □

Using model (3.9), the minimum of $\sigma_i - \sigma_{i+1}$ for $1 \leq i \leq k$ is achieved at $i = k^*$. Thus for $1 \leq i \leq k^*$, we have

$$\sigma_i - \sigma_{i+1} = \sigma_{i+1}\big(\frac{\sigma_i}{\sigma_{i+1}} - 1\big) \geq \begin{cases}\sigma_{k^*+1}(\rho-1) & \text{severely ill-posed;}\\ \sigma_{k^*+1}[(\frac{k^*+1}{k^*})^\alpha - 1] & \text{moderately/mildly ill-posed.}\end{cases}$$

By (3.7) and (3.8), we have $\sigma_{k^*+1} \gtrsim (m^{-1/2}\varepsilon)^{\frac{1}{1+\beta}}\|A\|$. Using these two inequalities, one can check that if (3.6) and (3.11) hold, then the assumption that $\sigma_i - \sigma_{i+1} \gg \|E\|$ for $1 \leq i \leq k^*$ can be satisfied.

Note that the semi-convergence point k_0 of LSQR for (3.3) is usually not big and thus $\|E\| = \mathcal{O}(c(n,k_0)\|A\|\mathbf{u})$ with $c(n,k_0)$ a moderate quantity. By Lemma 3.2 and Lemma 3.3, if $\mathbf{u}$ satisfies (3.6) and (3.11), then relations (3.4) and (3.5) holds and the effective resolution limit of (3.3) has the same accuracy as that of (1.1). Therefore by Theorem 3.1 $\bar{x}_{k_0}$ as well

as x_{k_0} will have the same accuracy as x_{opt}, which implies that the best computed regularized solution among x_k can achive the same accuracy as x_{opt}. The result is summarized in the following theorem.

Theorem 3.2 *Suppose that the LBFRO in LSQR is implemented in finite precision with round-off unit* $\mathbf{u}$ *and* (2.9) *is solved exactly. If* $\mathbf{u}$ *satisfies*

$$\mathbf{u} \ll \varrho(m^{-1/2}\varepsilon)^{\frac{2+\beta}{1+\beta}} \tag{3.14}$$

where

$$\varrho = \begin{cases} \min\{1, \rho - 1\} & \text{severely ill-posed;} \\ \min\{1, (\frac{k^*+1}{k^*})^{\alpha} - 1\} & \text{moderately/mildly ill-posed,} \end{cases} \tag{3.15}$$

then the best computed regularized solution among x_k *can achieve the same accuracy as the best regularized LSQR solution to* (1.1) *obtained in exact arithmetic.*

In Theorem 3.2, the parameters α, β and ρ are unknown in practical computations. In fact, these parameters are ideal for simplifying singular value decaying and DPC models, and they are closely related to properties of a given ill-posed problem. However, it is instructive from them to get insight into a practical choice of $\mathbf{u}$. For severely ill-posed problems, $\rho - 1$ is usually a constant not very small, while for moderately/mildly ill-posed problems, if ε is very small and α is not big that means the singular values of A decaying very slowly, then k^* will be big and thus $(\frac{k^*+1}{k^*})^{\alpha} - 1 \approx \alpha/k^*$ will be very small. Therefore, for moderately/mildly ill-posed problems, if ε is very small and α is not big, then (3.14) may give a too small upper bound on $\mathbf{u}$.

Theorem 3.2 implies that for noisy level ε not very small, we can exploit lower precision for constructing Lanczos vectors in the LSQR for solving (1.1) without loss of any accuracy of final regularized solutions. We will use numerical examples to show that single precision is enough for the three types of linear ill-posed problem. We need to stress a special practical case that $k^* = n$ for a too small ε and α, which may be encountered in some image deblurring problems. In this case the noise amplification is tolerable even without regularization, which makes x_{nai} a good approximation to x_{ex}, and the LSQR solves (1.1) in their standard manners as if they solved an ordinary other than ill-posed problem. Thus our result can not be applied to this case.

4 Updating x_k Using Lower Precision

In this section, we discuss how to use lower precision for updating x_k step by step. Suppose that the k-step LBFRO is implemented using the computing precision chosen as in Theorem 3.2. We first review the procedure for updating x_k from $x_0 = \mathbf{0}$ proposed in [37]. First, the QR factorization

$$\hat{Q}_k \left(B_k \ \beta_1 e_1^{(k+1)} \right) = \begin{pmatrix} R_k & f_k \\ & \bar{\phi}_{k+1} \end{pmatrix} = \left(\begin{array}{ccccc:c} \rho_1 & \theta_2 & & & & \phi_1 \\ & \rho_2 & \theta_3 & & & \phi_2 \\ & & \ddots & \ddots & & \vdots \\ & & & \rho_{k-1} & \theta_k & \phi_{k-1} \\ & & & & \rho_k & \phi_k \\ \hdashline & & & & & \bar{\phi}_{k+1} \end{array} \right) \tag{4.1}$$

is performed using a series of Givens rotations, where at the i-th step the Givens rotation is chosen to zero out β_{i+1}:

$$\begin{pmatrix} c_i & s_i \\ s_i & -c_i \end{pmatrix} \begin{pmatrix} \bar{\rho}_i & 0 & \bar{\phi}_i \\ \beta_{i+1} & \alpha_{i+1} & 0 \end{pmatrix} \begin{pmatrix} \rho_i & \theta_{i+1} & \phi_i \\ 0 & \bar{\rho}_{i+1} & \bar{\phi}_{i+1} \end{pmatrix},$$

and the orthogonal matrix Q_k is the product of these Givens rotation matrices. Since

$$\|B_k y - \beta_1 e_1^{(k+1)}\|^2 = \left\| \hat{Q}_k \begin{pmatrix} B_k & \beta_1 e_1^{(k+1)} \end{pmatrix} \begin{pmatrix} y \\ -1 \end{pmatrix} \right\|^2 = \|R_k y - f_k\|^2 + |\bar{\phi}_{k+1}|^2, \quad (4.2)$$

the solution to $\min_{y \in \mathbb{R}^k} \|B_k y - \beta_1 e_1^{(k)}\|$ is $y_k = R_k^{-1} f_k$. Factorize R_k as

$$R_k = D_k \widehat{R}_k, \quad D_k = \begin{pmatrix} \rho_1 & & & \\ & \rho_2 & & \\ & & \ddots & \\ & & & \rho_k \end{pmatrix}, \quad \widehat{R}_k = \begin{pmatrix} 1 & \theta_2/\rho_1 & & \\ & 1 & \theta_3/\rho_2 & \\ & & \ddots & \theta_k/\rho_{k-1} \\ & & & 1 \end{pmatrix},$$

then we get

$$x_k = Q_k y_k = Q_k R_k^{-1} f_k = (Q_k \widehat{R}_k^{-1})(D_k^{-1} f_k). \quad (4.3)$$

Let $W_k = Q_k \widehat{R}_k^{-1} = (w_1, \ldots, w_k)$. By using back substitution for solving $W_k \widehat{R}_k = Q_k$ we obtain the updating procedure for x_i and w_i:

$$x_i = x_{i-1} + (\phi_i/\rho_i) w_i, \quad w_{i+1} = q_{i+1} - (\theta_{i+1}/\rho_i) w_i, \quad (4.4)$$

which is described in Algorithm 2.

Algorithm 2 Updating procedure

Let $x_0 = \mathbf{0}$, $w_1 = q_1$, $\bar{\phi}_1 = \beta_1$, $\bar{\rho}_1 = \alpha_1$
for $i = 1, 2, \ldots, k$, **do**
  $\rho_i = (\bar{\rho}_i^2 + \beta_{i+1}^2)^{1/2}$
  $c_i = \bar{\rho}_i/\rho_i$, $s_i = \beta_{i+1}/\rho_i$
  $\theta_{i+1} = s_i \alpha_{i+1}$, $\bar{\rho}_{i+1} = -c_i \alpha_{i+1}$
  $\phi_i = c_i \bar{\phi}_i$, $\bar{\phi}_{i+1} = s_i \bar{\phi}_i$
  $x_i = x_{i-1} + (\phi_i/\rho_i) w_i$
  $w_{i+1} = q_{i+1} - (\theta_{i+1}/\rho_i) w_i$
end for

From the above description, the procedure of updating x_k is constituted of two parts: the Givens QR factorization and the computation of x_i and w_{i+1}. First, we investigate the choice of proper computing precision for the Givens QR factorization. Denote the roundoff unit used in this process by $\tilde{\mathbf{u}}$ and assume the computations of other parts are exact. In finite precision arithmetic, after the above process, we have computed the k-th iterative solution $\tilde{x}_k = Q_k \tilde{y}_k$ with $\tilde{R}_k \tilde{y}_k = \tilde{f}_k$, where $\begin{pmatrix} \tilde{R}_k & \tilde{f}_k \\ & \tilde{\phi}_{k+1} \end{pmatrix}$ is the computed R-factor of $\left(B_k \ \beta_1 e_1^{(k+1)}\right)$. Using the backward error analysis result about the Givens QR factorization as (4.1), there exist an orthogonal matrix $\tilde{Q}_k \in \mathbb{R}^{(k+1)\times(k+1)}$ such that

$$\tilde{Q}_k \left[\left(B_k \ \beta_1 e_1^{(k+1)} \right) + \left(\Delta_k^B \ \delta_k^\beta \right) \right] = \begin{pmatrix} \tilde{R}_k & \tilde{f}_k \\ & \tilde{\phi}_{k+1} \end{pmatrix} \quad (4.5)$$

where $\tilde{R}_k \in \mathbb{R}^{k\times k}$ is upper triangular and

$$\left\| \Delta_k^B \right\| / \|B_k\| \le c_1(k)\tilde{\mathbf{u}} + \mathcal{O}(\tilde{\mathbf{u}}^2), \quad \|\delta_k^\beta\|/\beta_1 \le c_1(k)\tilde{\mathbf{u}} + \mathcal{O}(\mathbf{u}^2)$$

with a moderate value $c_1(k)$ depending on k; see [24, Theorem 19.10]. The above relation means that $\tilde{Q}_k$ is the Q-factor of a perturbed $\left(B_k \ \beta_1 e_1^{(k+1)}\right)$, and $\tilde{R}_k$ is the R-factor of a perturbed B_k. Using the perturbation analysis result about QR factorizations, we have

$$\frac{\|\tilde{R}_k - R_k\|}{\|R_k\|} \le c_2(k)\kappa(R_k)\tilde{\mathbf{u}} + \mathcal{O}(\tilde{\mathbf{u}}^2), \tag{4.6}$$

$$\|\tilde{Q}_k - \hat{Q}_k\| \le c_3(k)\kappa(R_k)\tilde{\mathbf{u}} + \mathcal{O}(\tilde{\mathbf{u}}^2), \tag{4.7}$$

where $\kappa(R_k) = \|R_k\|\|R_k^{-1}\|$ is the condition number of R_k, and $c_2(k)$ and $c_3(k)$ are two moderate values depending on k; see [24, §19.9]. Note that the F-norm result appeared in [24, §19.9] also applies to the above 2-norm result besides a difference of multiplicative factor depending on k. Thus, we have

$$\begin{aligned}\|\tilde{f}_k - f_k\| &\le \left\| \tilde{Q}_k \left(\beta_1 e_1^{(k+1)} + \delta_k^\beta\right) - \hat{Q}_k \beta_1 e_1^{(k+1)} \right\| \\ &\le \beta_1 \|\tilde{Q}_k - \hat{Q}_k\| + \|\tilde{Q}_k \delta_k^\beta\| \\ &\le \beta_1 \left[c_3(k)\kappa(R_k) + c_1(k)\right]\tilde{\mathbf{u}} + \mathcal{O}(\tilde{\mathbf{u}}^2). \end{aligned} \tag{4.8}$$

Note that $R_k y_k = f_k$. Using the perturbation analysis result about this linear system [15, §2.6.4], if $\|R_k^{-1}(\tilde{R}_k - R_k)\| < 1$, we get

$$\begin{aligned}\frac{\|\tilde{y}_k - y_k\|}{\|y_k\|} &\le \frac{\kappa(R_k)}{1 - \kappa(R_k)\frac{\|\tilde{R}_k - R_k\|}{\|R_k\|}} \left(\frac{\|\tilde{R}_k - R_k\|}{\|R_k\|} + \frac{\|\tilde{f}_k - f_k\|}{\|f_k\|} \right) \\ &\le \frac{\kappa(R_k)}{1 - c_2(k)\kappa(R_k)^2\tilde{\mathbf{u}}} \left(c_2(k)\kappa(R_k) + \frac{\beta_1 c_3(k)\kappa(R_k) + c_1(k)}{(\beta_1^2 - \bar{\phi}_{k+1}^2)^{1/2}} \right) \tilde{\mathbf{u}} + \mathcal{O}(\tilde{\mathbf{u}}^2). \end{aligned}$$

Notice that $x_k = Q_k y_k$ and $\tilde{x}_k = Q_k \tilde{y}_k$, where Q_k is computed by LBFRO in finite precision with roundoff unit $\mathbf{u}$. We get $\|\tilde{x}_k - x_k\| \le \|Q_k\|\|\tilde{y}_k - y_k\| \le \|\tilde{y}_k - y_k\|\,(1 + \mathcal{O}(\mathbf{u}))$, where we have used $\|Q_k - \bar{Q}_k\| = \mathcal{O}(\mathbf{u})$ and $\|\bar{Q}_k\| = 1$ by (2.13). Using Theorem 3.1 and $\|y_k\| = \|\bar{Q}_k y_k\| = \|\bar{x}_k\|$, we obtain

$$\begin{aligned}\frac{\|\tilde{x}_k - x_k\|}{\|x_k\|} &= \frac{\|\tilde{x}_k - x_k\|}{\|\bar{x}_k\|}\frac{\|\bar{x}_k\|}{\|x_k\|} \le \frac{\|\tilde{y}_k - y_k\|\,(1 + \mathcal{O}(\mathbf{u}))}{\|y_k\|}\,(1 + \mathcal{O}(\mathbf{u})) \\ &\le \frac{\kappa(R_k)}{1 - c_2(k)\kappa(R_k)^2\tilde{\mathbf{u}}} \left(c_2(k)\kappa(R_k) + \frac{\beta_1 c_3(k)\kappa(R_k) + c_1(k)}{(\beta_1^2 - \bar{\phi}_{k+1}^2)^{1/2}} \right) \tilde{\mathbf{u}} \\ &\quad + \mathcal{O}(\mathbf{u}\tilde{\mathbf{u}} + \tilde{\mathbf{u}}^2). \end{aligned} \tag{4.9}$$

The upper bound (4.9) grows up at the speed of $\kappa(R_k)^2\tilde{\mathbf{u}} = \kappa(B_k)^2\tilde{\mathbf{u}}$. By Theorem 2.1, B_k is the projection of $A+E$ on span$\{\bar{P}_{k+1}\}$ and span$\{\bar{Q}_k\}$, and it gradually becomes ill-conditioned since $A + E$ is a slight perturbation of the ill-conditioned matrix A. This implies that a lower computing precision used by the Givens QR factorization will lead to a loss of accuracy of the computed solution. Therefore, in practical computation, we need to use double precision to perform it. Fortunately, the Givens QR factorization (Line 3–6 in Algorithm 2) can always be performed very quickly since only operations of scalars are involved. In contrast, the

updating of x_i and w_{i+1} involves vector operations, thereby it is better to be performed using lower precision. The proper choice of the computing precision for it will be analyzed in the following part.

In practical computation of the iterative regularization algorithm, the k-step LBFRO need not to be implemented in advance, while it should be done in tandem with the updating procedure. An early stopping criterion such as DP or L-curve criterion is used to estimate the semi-convergence point. The whole process can be summarized in Algorithm 3 as a mixed precision variant of LSQR for linear ill-posed problems.

Algorithm 3 Mixed precision variant of LSQR for (1.1)

Input: $A, b, x_0 = \mathbf{0}$
1: **for** $k = 1, 2, \ldots,$ **do**
2: Compute $p_k, q_k, \alpha_k, \beta_k$ by the LBFRO ▷ roundoff unit is $\mathbf{u}$
3: Compute $\rho_k, \theta_{k+1}, \bar{\rho}_{k+1}, \phi_k, \bar{\phi}_{k+1}$ by the updating procedure ▷ double precision
4: Compute x_k, w_{k+1} by the updating procedure ▷ roundoff unit is $\bar{\mathbf{u}}$
5: **if** Early stopping criterion is satisfied **then** ▷ DP or L-curve criterion
6: The semi-convergence point is estimated as k_1
7: Terminate the iteration
8: **end if**
9: **end for**
Output: Final regularized solution $\hat{x}_{k_1}$ ▷ Computed solution corresponding to x_{k_1}

To analyze the choice of $\bar{\mathbf{u}}$, we use the following model [15, §2.7.3] for the floating point arithmetic:

$$\mathrm{fl}(a \ \mathrm{op}\ b) = (a \ \mathrm{op}\ b)(1+\epsilon), \quad |\epsilon| \le \bar{\mathbf{u}}, \quad \mathrm{op} = +, -, *, /. \tag{4.10}$$

Under this model, we have the following rounding error results for matrix and vector computations [15, §2.7.8]:

$$\mathrm{fl}(u + \alpha v) = u + \alpha v + w, \quad |w| \le (|u| + 2|\alpha v|)\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2), \tag{4.11}$$

$$\mathrm{fl}(AB) = AB + X, \quad |X| \le n|A||B|\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2). \tag{4.12}$$

where u, v are vectors, α is a scalar, and A, B are two matrices of orders $m \times n$ and $n \times l$, respectively. In (4.11) and (4.12), the notation $|\cdot|$ is used to denote the absolute value of a matrix or vector and " $\le''$ means the relation " $\le''$ holds componentwise. We remark that (4.12) applies to both dot-product and outer-product based procedures for matrix multiplications; see [15, §1.1, §2.7] for these two types of computation of matrix multiplications and the corresponding rounding error analysis results.

Denote by $\hat{x}_k$ and $\hat{w}_k$ the computed quantities where the roundoff unit of the computing precision is $\bar{\mathbf{u}}$ and the Givens QR factorization is performed with double precision. To avoid cumbersome using of notations, in the following analysis, notations such as v_k, f_k, x_k, w_k denote the computed quantities for the process that the LBFRO is implemented in finite precision with roundoff unit $\mathbf{u}$ while the Givens QR factorization is implemented in double precision and other computations are exact. Using the above model, the computation of $\hat{w}_i$ in finite precision arithmetic can be formed as:

$$\begin{cases} \hat{w}_1 = v_1 + \delta_1^w \\ \hat{w}_2 = v_2 - (\theta_2/\rho_1)\hat{w}_1 + \delta_2^w \\ \quad \vdots \\ \hat{w}_k = v_k - (\theta_k/\rho_{k-1})\hat{w}_{k-1} + \delta_k^w \end{cases}$$

where

$$\|\delta_1^w\| \le \|v_1\|\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2) = \bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2)$$

and

$$\|\delta_i^w\| \le (\|v_i\| + 2\|(\theta_i/\rho_{i-1})\hat{w}_{i-1}\|)\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2) = (1 + 2\|(\theta_i/\rho_{i-1})\hat{w}_{i-1}\|)\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2).$$

Let $h_i = \hat{w}_i - w_i$. Then we have

$$h_{i+1} = v_{i+1} - (\theta_{i+1}/\rho_i)\hat{w}_i + \delta_i^w - (v_{i+1} - (\theta_{i+1}/\rho_i)w_i) = -(\theta_{i+1}/\rho_i)h_i + \delta_i^w,$$

which leads to

$$H_k\widehat{R}_k = \Delta_k^w \tag{4.13}$$

with $H_k = (h_1, \ldots, h_k)$ and $\Delta_k^w = (\delta_1^w, \ldots, \delta_k^w)$. Therefore, we have

$$\begin{aligned}\|\Delta_k^w\| &\le \sqrt{k}\max_{1\le i\le k}\|\delta_i^w\| \le \sqrt{k}[1 + 2\max_{1\le i\le k}(\theta_{i+1}/\rho_i)\|w_i + h_i\|]\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2)\\ &\le \sqrt{k}[1 + 2\|\widehat{R}_k\|(\|W_k\| + \|\Delta_k^w\|\|\widehat{R}_k^{-1}\|)]\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2),\end{aligned}$$

where we have used $|\theta_{i+1}/\rho_i| \le \|\widehat{R}_k\|$ and $\|h_i\| \le \|H_k\| \le \|\Delta_k^w\|\|\widehat{R}_k^{-1}\|$. This inequality leads to

$$(1 - 2\sqrt{k}\kappa(\widehat{R}_k)\bar{\mathbf{u}})\|\Delta_k^w\| \le \sqrt{k}(1 + 2\|\widehat{R}_k\|\|W_k\|)\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2),$$

where $\kappa(\widehat{R}_k) = \|\widehat{R}_k\|\|\widehat{R}_k^{-1}\|$ is the condition number of $\widehat{R}_k$. Notice that $\|W_k\| \le \|Q_k\|\|\widehat{R}_k^{-1}\| = \|\widehat{R}_k^{-1}\| + \mathcal{O}(\mathbf{u})$ where we have used $\|Q_k - \bar{Q}_k\| = \mathcal{O}(\mathbf{u})$ and $\|\bar{Q}_k\| = 1$ by (2.13). We obtain the upper bound on $\|\Delta_k^w\|$:

$$\|\Delta_k^w\| \le \sqrt{k}[1 + 2(1 + \sqrt{k})\kappa(\widehat{R}_k)]\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2 + \bar{\mathbf{u}}\mathbf{u}). \tag{4.14}$$

Now we can analyze the accuracy of $\hat{x}_k$.

Theorem 4.1 *In Algorithm 3, denote by x_k the computed solution where the Givens QR factorization is performed in double precision and other computations of the updating procedure are exact. Then at each iteration we have*

$$\frac{\|\hat{x}_k - x_k\|}{\|x_k\|} \le \sqrt{k}[1 + (2 + 2\sqrt{k} + k)\kappa(\widehat{R}_k)]\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2 + \bar{\mathbf{u}}\mathbf{u}). \tag{4.15}$$

Proof Notice from (4.3) and (4.4) that the formation of $\hat{x}_k$ is the matrix multiplication between $\widehat{W}_k = (\hat{w}_1, \ldots, \hat{w}_k)$ and $D_k^{-1}f_k$ by the outer-product based procedure. Using model (4.12) we have

$$\hat{x}_k - x_k = \widehat{W}_k D_k^{-1} f_k + \Delta_k^x - W_k D_k^{-1} f_k, \tag{4.16}$$

where $\hat{x}_k = \widehat{W}_k(D_k^{-1}f_k) + \Delta_k^x$ with $|\Delta_k^x| \le k|\widehat{W}_k||D_k^{-1}f_k|\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2)$. Therefore, we have

$$\begin{aligned}\|\Delta_k^x\| &\le \||\Delta_k^x|\| \le k\||\widehat{W}_k|\|_F\||D_k^{-1}f_k|\|\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2)\\ &= k\|\widehat{W}_k\|_F\|D_k^{-1}f_k\|\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2)\\ &\le k^{3/2}\|\widehat{W}_k\|\|D_k^{-1}f_k\|\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2),\end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix. By (4.13), we have $\widehat{W}_k - W_k = H_k = \Delta_k^w \widehat{R}_k^{-1}$, and thus

$$\|\widehat{W}_k\| \le \|W_k\| + \|H_k\| \le \|W_k\| + \|\Delta_k^w\|\|\widehat{R}_k^{-1}\|.$$

Substituting it into the inequality about $\|\Delta_k^x\|$ and noticing $\widehat{R}_k^{-1} D_k^{-1} f_k = R_k^{-1} f_k = y_k$, we obtain

$$\begin{aligned}\|\Delta_k^x\| &\le k^{3/2}(\|W_k\| + \|\Delta_k^w\|\|\widehat{R}_k^{-1}\|)\|\widehat{R}_k(\widehat{R}_k^{-1} D_k^{-1} f_k)\|\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2)\\ &\le k^{3/2}\kappa(\widehat{R}_k)\|y_k\|\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2 + \bar{\mathbf{u}}\mathbf{u}),\end{aligned}$$

where we have used $\|W_k\| \le \|\widehat{R}_k^{-1}\| + \mathcal{O}(\mathbf{u})$. Using $\widehat{W}_k - W_k = \Delta_k^w \widehat{R}_k^{-1}$ again, we get

$$\|\widehat{W}_k D_k^{-1} f_k - W_k D_k^{-1} f_k\| = \|(\widehat{W}_k - W_k) D_k^{-1} f_k\| = \|\Delta_k^w (\widehat{R}_k^{-1} D_k^{-1} f_k)\| \le \|\Delta_k^w\|\|y_k\|.$$

By (4.16) and combining with (4.14), we obtain

$$\begin{aligned}\|\hat{x}_k - x_k\| &\le \|\Delta_k^x\| + \|\widehat{W}_k D_k^{-1} f_k - W_k D_k^{-1} f_k\|\\ &\le \sqrt{k}[1 + (2 + 2\sqrt{k} + k)\kappa(\widehat{R}_k)]\|y_k\|\bar{\mathbf{u}} + \mathcal{O}(\bar{\mathbf{u}}^2 + \bar{\mathbf{u}}\mathbf{u}).\end{aligned}$$

Since $\|Q_k - \bar{Q}_k\| = \mathcal{O}(\mathbf{u})$ and $\bar{Q}_k$ is orthonormal, we have $\|Q_k^{-1}\| \le 1/(1 - \mathcal{O}(\mathbf{u})) = 1 + \mathcal{O}(\mathbf{u})$. Using the relations $\|y_k\| = \|Q_k^{-1} x_k\| \le \|x_k\|\,(1 + \mathcal{O}(\mathbf{u}))$ and $\|\hat{x}_k - x_k\|/\|x_k\| = \|\hat{x}_k - x_k\|/\|y_k\| \cdot \|y_k\|/\|x_k\|$, we finally obtain (4.15). □

Note that $\widehat{R}_k$ is obtained by scaling R_k using the diagonal of it, and the diagonal scaling step can often dramatically reduces the condition number of R_k. In fact, we will show in the numerical experiments section that $\kappa(\widehat{R}_k)$ is a moderate value even for an iteration k bigger than the semi-convergence point. By Theorem 4.1, if $\mathbf{u}$ has been chosen such that the best solution among x_k can achieve the same accuracy as the best LSQR regularized solution to (1.1) obtained in exact arithmetic, in order to make the practical updated $\hat{x}_k$ can also achieve the same accuracy, $\bar{\mathbf{u}}$ should be chosen such that the upper bound in (4.15) is much smaller than $\|x_{opt} - x_{ex}\|/\|x_{ex}\|$. Thanks to (2.5), for a not very small noise level, the single precision roundoff unit $\bar{\mathbf{u}}$ is enough. This ensures that we can use lower precision for updating x_k, which is more efficient than using double precision.

Now we discuss methods for estimating the optimal early stopping iteration, i.e., the semi-convergence point. By 3.1 and (4.2) we have

$$\bar{\phi}_{k+1} = \|B_k y_k - \beta_1 e_1^{(k+1)}\| = \|(A + E)x_k - (b + \delta_b)\|.$$

For the proper choice of $\mathbf{u}$, the noise norm of (3.2) is $\|e - Ex_{ex} + \delta_b\| \approx \|e\|$ since $\| - Ex_{ex} + \delta_b\| \ll \|e\|$. Since $\hat{x}_k$, x_k and $\bar{x}_k$ have the same accuracy for the proper $\mathbf{u}$ and $\bar{\mathbf{u}}$, we only need to estimate the semi-convergence point of LSQR applied to (3.2). The discrepancy principle corresponding to (3.2) can be written as

$$\|(A + E)x_k - (b + \delta_b)\| \lesssim \tau\|e\|$$

with $\tau > 1$ slightly, and we should stop iteration at the first k satisfying

$$\bar{\phi}_{k+1} = \|B_k y_k - \beta_1 e_1^{(k+1)}\| \le \tau\|e\|, \tag{4.17}$$

and use this k as the estimate of semi-convergence point, where $\bar{\phi}_{k+1}$ can be efficiently computed by using 2. Numerical experiments will show that this estimate is almost the same as that obtained by the discrepancy principle for LSQR in double precision arithmetic. The

discrepancy principle method usually suffers from under-estimating and thus the solution is over-regularized.

Another approach is the L-curve criterion, which does not need $\|e\|$ in advance. The motivation is that one can plot $(\log \|Ax_k - b\|, \log \|x_k\|)$ in the shape of an L-curve, and the corner of the curve is a good estimate of the semi-convergence point. The L-curve for (3.2) is

$$(\log \|(A + E)x_k - (b + \delta_b)\|, \log \|x_k\|),$$

which is just

$$\left(\log \bar{\phi}_{k+1}, \log \|x_k\|\right), \tag{4.18}$$

where the norm of x_k should be computed at each iteration. A modification of (4.18) computes the norm of $\hat{x}_k$ instead of x_k, and this may make a little difference with the estimate by (4.18). Numerical experiments will show that these two estimates are almost the same as that obtained by the L-curve criterion for LSQR in double precision arithmetic.

Finally, we give a model for comparing computing efficiency between the double and mixed precision implementations of LSQR. We also perform full reorthogonalization of the Lanczos bidiagonalization for the double precision implementation, since without reorthogonalization the convergence behavior is irregular and the convergence rate is much slow. We count the computations involving matrix/vector operations in the two main parts of the algorithm:

- For the LBFRO process, at each step it takes $\mathcal{O}(mn)$ flops for matrix–vector products and $\mathcal{O}(m+n)$ flops for scalar-vector multiplications; besides, the reorthogonalization at the k-th step takes $\mathcal{O}((m+n)k^2)$ flops. Therefore, at each k-th iteration, LBFRO takes $\mathcal{O}(mn + (m+n)(k^2+1))$ flops.
- For the updating procedure, the most time-consuming part is the computation of x_i and w_{i+1}, and it takes $\mathcal{O}(n)$ flops.

From the above investigation, we find that the matrix–vector products in LBFRO are the most dominant computations in the entire algorithm. In the ideal case, the performance of 32-bit operations is at least twice as fast as that of 64-bit operations on modern computing architectures [1]. Therefore, the proposed mixed precision algorithm can save approximately half the time compared to the original double precision algorithm.

In pracital computations, to give a convincing comparison between the two implementations, the mixed precision algorithm need to be performed on a specific computing architecture supporting well for lower precision computations such as NVIDIA Tesla V100 GPU [36], and the codes should be optimized to take full advantage of the computing power. This will de considered in our future work.

5 Numerical Experiments

In this section, we present some numerical experiments to justify the theoretical results obtained. Two mixed precision variants of LSQR are implemented to be compared with the double precision LSQR for several test linear ill-posed problems. We use "d" to denote the algorithm implemented using double precision, and use "s+d" and "s+s" to denote the algorithms that use single precision for LBFRO while use double and single precisions for updating x_k, respectively. Note that for "d" the Lanczos bidiagonalization is also implemented using full reorthogonalization to avoid delay of convergence.

Table 2 The description of test problems

Problem	$m \times n$	Ill-posedness	Description
shaw	1000×1000	Severe	1-D image restoration model
deriv2	1000×1000	Moderate	Computation of second derivative
gravity	2000×2000	Severe	1-D gravity surveying problem
heat	2000×2000	Moderate	Inverse heat equation
PRblurspeckle	16384×16384	Mild	2-D image deblurring problem
PRblurdefocus	65536×65536	Mild	2-D image deblurring problem

For these different implementations, we compare accuracy of the regularized solutions by using the relative reconstruction error

$$\mathrm{RE}(k) = \frac{\|x_k - x_{ex}\|}{\|x_{ex}\|} \tag{5.1}$$

to plot semi-convergence curves, where x_k (for "s+s" it should be $\hat{x}_k$) denote the computed solutions produced by the three implementations. We emphasis that computing efficiency in terms of time-to-solution between "d", "s+d" and "s+s" is not compared here, since the purpose of this paper is to verify the feasibility of lower precision LSQR.

In this paper, we implement the MATLAB codes with MATLAB R2019b to perform numerical experiments, where the roundoff units for double and single precision are $2^{-53} \approx 1.11 \times 10^{-16}$ and $2^{-24} \approx 5.96 \times 10^{-8}$, respectively. The codes are available at https://github.com/Machealb/Lower_precision_solver. We choose some one dimensional (1-D) problems from the regularization toolbox [21], and two dimensional (2-D) image deblurring problems from [13]. The description of all test examples is listed in Table 2.

5.1 One Dimensional Case

For one dimensional problems shaw, deriv2, gravity and heat, we use the codes from [21] to generate A, x_{ex} and $b_{ex} = Ax_{ex}$, and then add a white Gaussian noise e with a prescribed noise level $\varepsilon = \|e\|/\|b_{ex}\|$ to b_{ex} and form the noisy $b = b_{ex} + e$.

First, we compare the relative errors $\mathrm{RE}(k)$ for the three different implementations of LSQR when $\varepsilon = 10^{-3}$. From Fig. 1 we can find the convergence behaviors of the three implementations "d", "s+d" and "s+s" are of highly consistence. The semi-convergence points k_0 are the same and the relative error curves coincide until many steps after semi-convergence, and thus the optimal regularized solutions computed by "d", "s+d" and "s+s" have the same accuracy. In order to give a more clear comparison about accuracy of solutions, we also plot the relative error curves of $x_k/\hat{x}_k$ computed by "s+d"/"s+s" with respect to that by "d". Figure 2 shows that these two relative errors are much smaller than $\mathrm{RE}(k)$ of "d" until semi-convergence occurs and this is also true for many iterations afterwards. These results confirm that both the LBFRO and updating procedure can be implemented using single precision without sacrificing any accuracy of final regularized solutions for $\varepsilon = 10^{-3}$.

To further compare the accuracy of solutions computed by "s+d" and "s+s", we plot in Fig. 3 the relative error curves of these two solutions and their upper bounds in (4.15). Here we set the upper bounds as $\kappa(\widehat{R}_k)\bar{\mathbf{u}}$ with $\bar{\mathbf{u}}$ the roundoff unit of single precision. From Fig. 3 we can find that $\kappa(\widehat{R}_k)$ for the four test problems grow very slightly, which lead to the upper

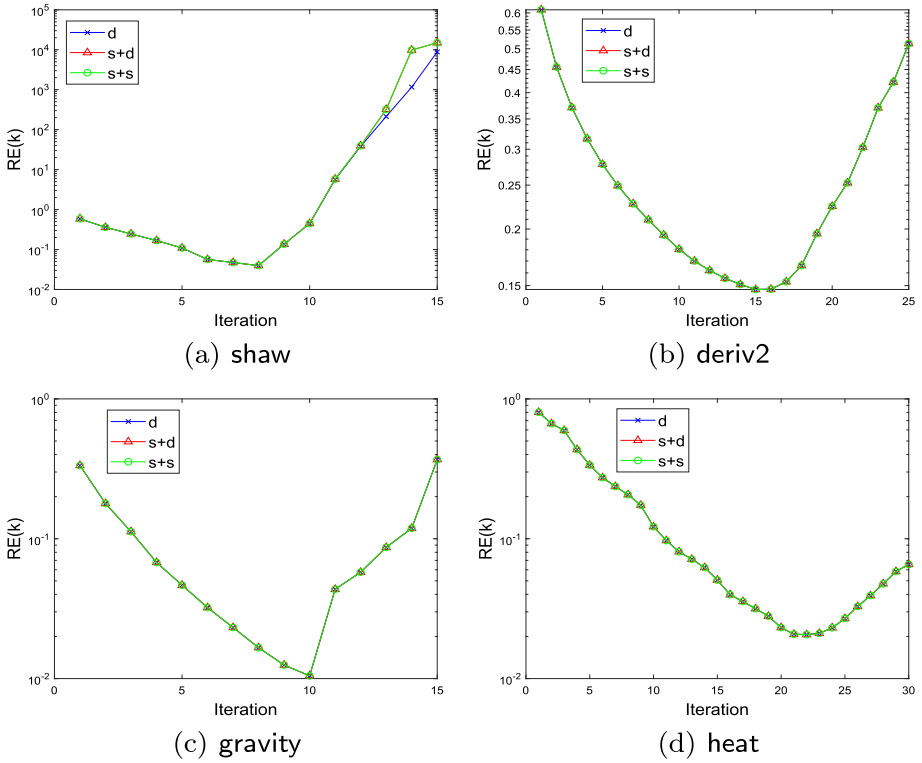


Fig. 1 Semi-convergence curves for LSQR implemented using different computing precisions, $\varepsilon = 10^{-3}$

bounds much smaller than RE(k). Therefore, the regularized solutions computed by "s+d" and "s+s" have the same accuracy, which has already been clearly observed from Fig. 1.

Table 3 shows the relative errors of the regularized solutions at the semi-convergence point k_0 and the estimates of k_0 by L-curve criterion and discrepancy principle, where the corresponding iteration number is in brackets. We find that the three optimal iterations and corresponding RE(k) for "d", "s+d" and "s+s" are the same, which has also been observed from Fig. 1. This is also true for the L-curve criterion, and the method gets an over-estimate of k_0 for heat. For the discrepancy principle, by (4.17) we know that the estimates of k_0 for "s+d" and "s+s" are always the same since they compute the same $\bar{\phi}_{k+1}$, and we find these estimates are the same at that for "d". The discrepancy principle gets under-estimates of k_0 for the four test problems and thus the solutions are over-regularized.

From the above experimental results, we can make sure that the two mixed precision variants "s+d" and "s+s" can compute regularized solutions with the same accuracy as the double precision LSQR when $\varepsilon = 10^{-3}$. We have also made numerical experiments for $\varepsilon = 10^{-4}, 10^{-5}$ and get similar results. Here we only show the semi-convergence curves for $\varepsilon = 10^{-5}$ in Fig. 4. For this noise level, single precision computing of the LBFRO and updating x_k is also enough for LSQR for solving the four test problems. Note from subfigure (a) that the relative errors for "s+d" and "s+s" quickly become bigger than that for "d" after semi-convergence. This reminds us that for shaw, if ε is smaller than 10^{-5}, simply implementing the LBFRO with single precision will lead to a loss of accuracy of regularized solutions. For extremely small noise level $\varepsilon = 10^{-7}$, we plot the semi-convergence curves for gravity and

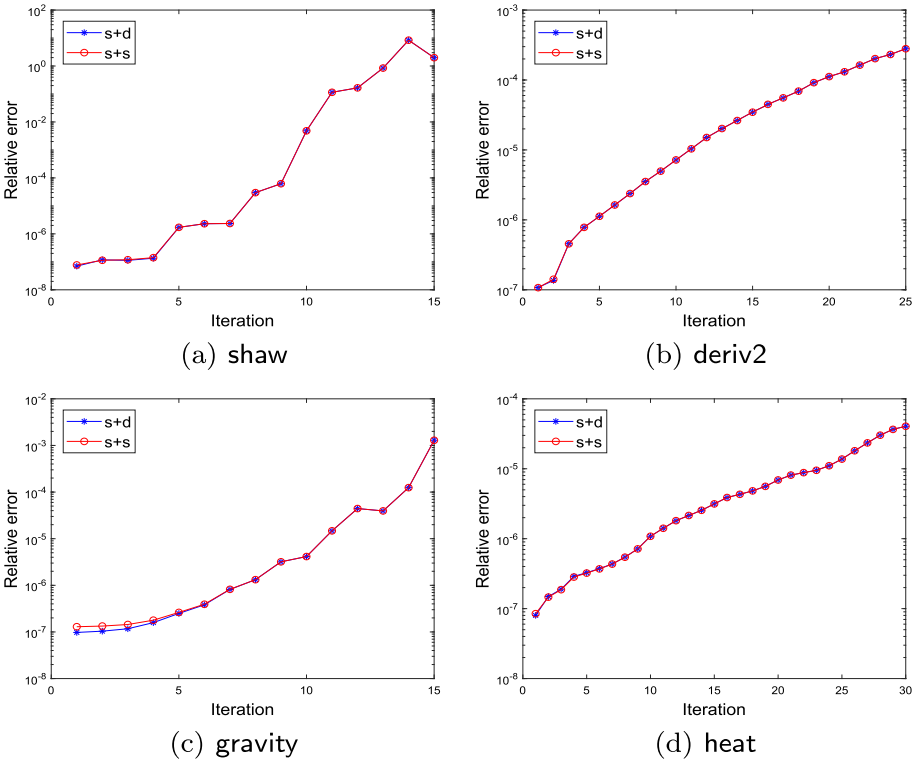


Fig. 2 Relative errors of the regularized solutions computed by "s+d"/"s+s" with respect to that by "d", $\varepsilon = 10^{-3}$

Table 3 Comparison of relative errors RE(k) and estimates of the optimal iteration k_0 by L-curve criterion and DP ($\tau = 1.001$), $\varepsilon = 10^{-3}$

Work precision	Shaw	Deriv2	Gravity	Heat
Optimal				
d	0.0396 (8)	0.1471 (15)	0.0105 (10)	0.0206 (22)
s+d	0.0396 (8)	0.1471 (15)	0.0105 (10)	0.0206 (22)
s+s	0.0396 (8)	0.1471 (15)	0.0105 (10)	0.0206 (22)
L-curve				
d	0.0396 (8)	0.1529 (17)	0.0105 (10)	0.0392 (27)
s+d	0.0396 (8)	0.1529 (17)	0.0105 (10)	0.0392 (27)
s+s	0.0396 (8)	0.1529 (17)	0.0105 (10)	0.0392 (27)
Discrepancy principle				
d	0.0473 (7)	0.1557 (13)	0.0166 (8)	0.0280 (19)
s+d/s	0.0473 (7)	0.1557 (13)	0.0166 (8)	0.0280 (19)

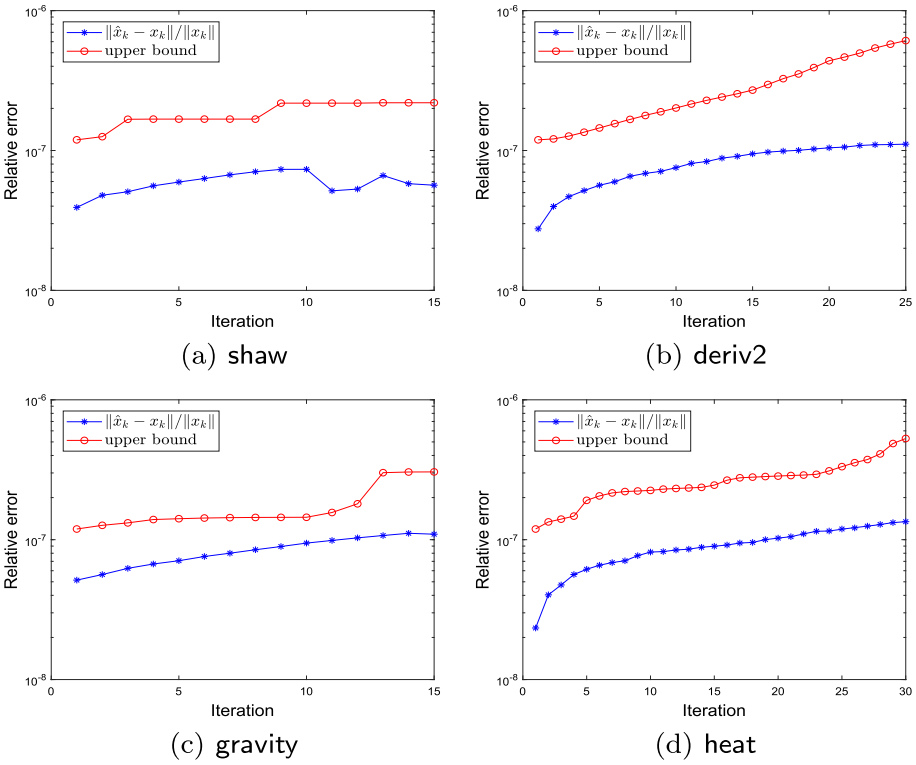


Fig. 3 Relative errors between regularized solutions computed by "s+d" and "s+s", $\varepsilon = 10^{-3}$

heat in Fig. 5. We can clearly find that neither "s+d" nor "s+s" can compute a regularized solution with accuracy as good as the best one achieved by "d". In real applications, the noise levels are seldom so small, and it is almost always possible to implement LSQR using lower precisions to compute a regularized solution without sacrificing accuracy.

5.2 Two Dimensional Case

We generate two image deblurring problems using codes from [13] for testing two dimensional linear ill-posed problems, with the goal to restore an image from a blurred and noisy one $b = Ax_{ex} + e$, where x_{ex} denotes the true image and A denotes the blurring operator. For the background of image deblurring, we refer the readers to [23]. For PRblurspeckle, which simulates spatially invariant blurring caused by atmospheric turbulence, we use the true image "Hubble Space Telescope" with image size of $N = 128$ (i.e., the true and blurred images have 128×128 pixels), and the blur level is set to be medium. For PRblurdefocus, which simulates a spatially invariant, out-of-focus blur, we use the true image "Cameraman" with image size of $N = 256$, and the blur level is set to be severe. Zero boundary condition is used for both the two blurs to construct A. The two true images are shown in Fig. 6.

For PRblurspeckle with image "Hubble Space Telescope", the noise levels are set as $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$. Figure 7 depicts the semi-convergence curves for "d", "s+d" and "s+s" as well as relative error curves of x_k computed by "s+d" and "s+s" with respect to

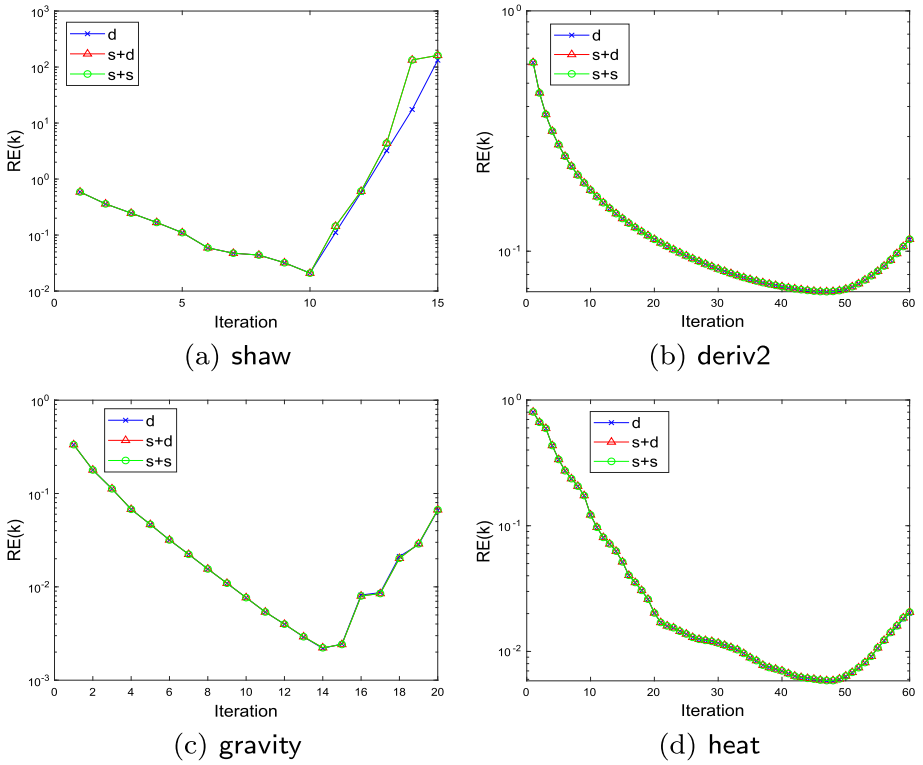


Fig. 4 Semi-convergence curves for LSQR implemented using different computing precisions, $\varepsilon = 10^{-5}$

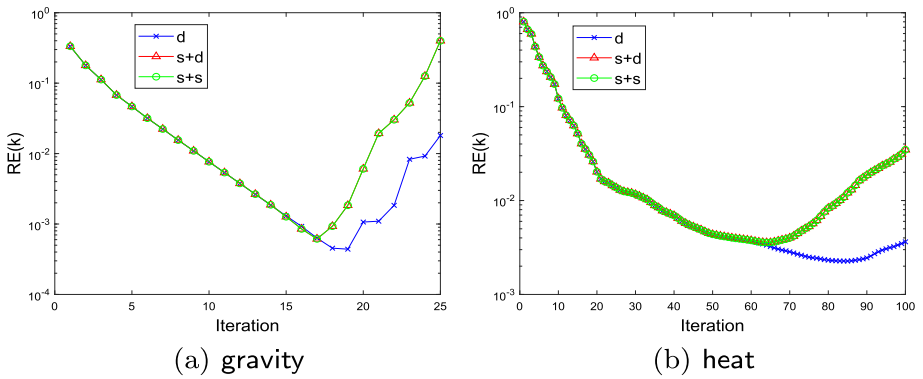
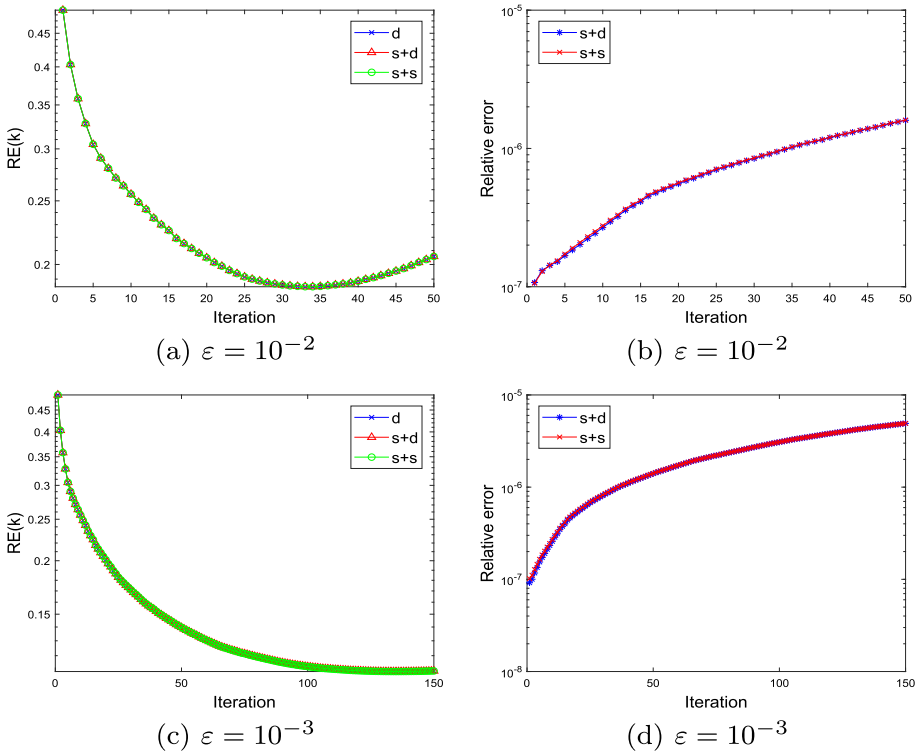


Fig. 5 Semi-convergence curves for LSQR implemented using different computing precisions, $\varepsilon = 10^{-7}$

that by "d". For both the two noise levels, we find that the curves of RE(k) coincide until many steps after semi-convergence and the three semi-convergence points are the same. The relative errors of $x_k/\hat{x}_k$ computed by "s+d"/"s+s" with respect to that by "d" are shown in subfigures (c) and (d), and they are much smaller than RE(k) of "d" until many steps after semi-convergence. The numerical results confirm that for PRblurspeckle, the LSQR can be implemented using lower precisions without sacrificing accuracy of regularized solutions for

(a) Hubble Space Telescope (b) Cameraman

Fig. 6 True images for testing PRblurspeckle and PRblurdefocus



(a) $\varepsilon = 10^{-2}$ (b) $\varepsilon = 10^{-2}$

(c) $\varepsilon = 10^{-3}$ (d) $\varepsilon = 10^{-3}$

Fig. 7 Semi-convergence curves for LSQR implemented using different computing precisions, and relative errors of regularized solutions computed by "s+d"/"s+s" with respect to that by "d", PRblurspeckle

$\varepsilon = 10^{-2}$ or $\varepsilon = 10^{-3}$. Figure 8 shows the blurred images and corresponding restored ones for the two noise levels, where the restored images are obtained from the best regularized solution at the semi-convergence point (k_0 and $\mathrm{RE}(k_0)$ are the same for "d", "s+d" and "s+s", and we choose $\hat{x}_{k_0}$ computed by "s+s"). The result shows a good deblurring effect

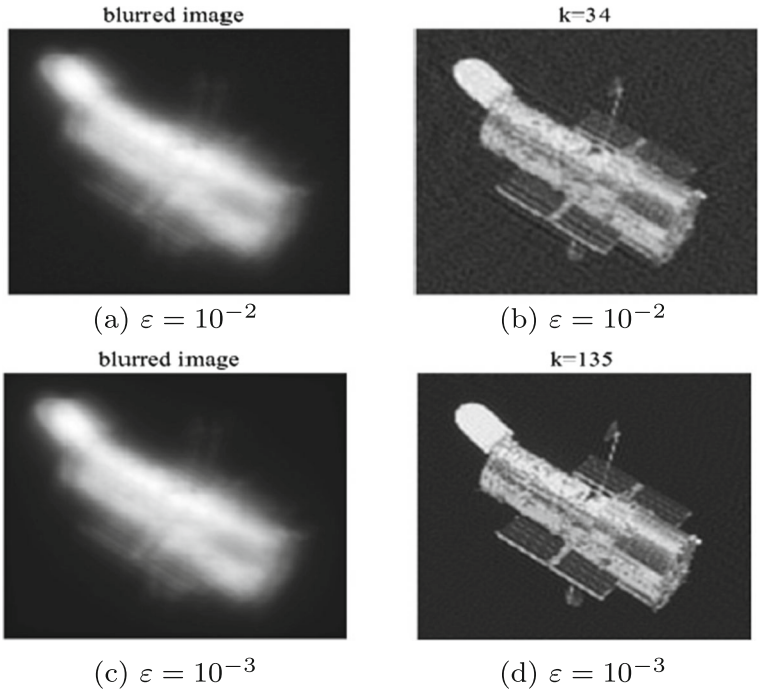


Fig. 8 Images "Hubble Space Telescope" blurred by PRblurspeckle and restored by the mixed precision LSQR "s+s" at the optimal iteration: **a**, **b** $\varepsilon = 10^{-2}$; **c**, **d** $\varepsilon = 10^{-3}$

of LSQR implemented using single precision. For noise levels smaller than 10^{-4}, we find that the semi-convergence behavior does not appear, which means that noise amplification is tolerable even without regularization, and thus we need not test for smaller noise cases.

For PRblurdefocus with image "Cameraman", the noise levels are set as $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-4}$. For noise levels smaller than 10^{-5}, the semi-convergence behavior will not appear and we need not test for those cases. Figures 9 and 10 show the relative error curves and blurred and restored images. Subfigures (b) and (d) of Fig. 9 depict the relative errors between regularized solutions computed by "s+d" and "s+s" with upper bounds $\kappa(\widehat{R}_k)\bar{\mathbf{u}}$. We can find that $\kappa(\widehat{R}_k)$ for the two noise levels grow very slightly, which lead to the upper bounds much smaller than $\mathrm{RE}(k)$. Therefore, the regularized solutions computed by "s+d" and "s+s" have the same accuracy, which can clearly observed from subfigures (a) and (c). The other experimental results are similar to those of PRblurdefocus and we do not illustrate them in detail any longer.

Table 4 shows the relative errors of the regularized solutions at the semi-convergence point k_0 and the estimates by L-curve criterion and discrepancy principle. For PRblurspeckle, the three optimal iterations k_0 and corresponding $\mathrm{RE}(k_0)$ for "d", "s+d" and "s+s" are the same, while for PRblurdefocus the k_0 for "d" and "s+d/s" are differed only by one and the corresponding $\mathrm{RE}(k_0)$ are the same. For the discrepancy principle, the estimates of k_0 are the same for "d" and "s+d/s", and the corresponding $\mathrm{RE}(k_0)$ is only slightly different for PRblurdefocus with $\varepsilon = 10^{-3}$. For the L-curve criterion, the estimates of k_0 and the corresponding $\mathrm{RE}(k_0)$ for "d", "s+d" and "s+s" are not the same for some cases, but the differences are very slight.

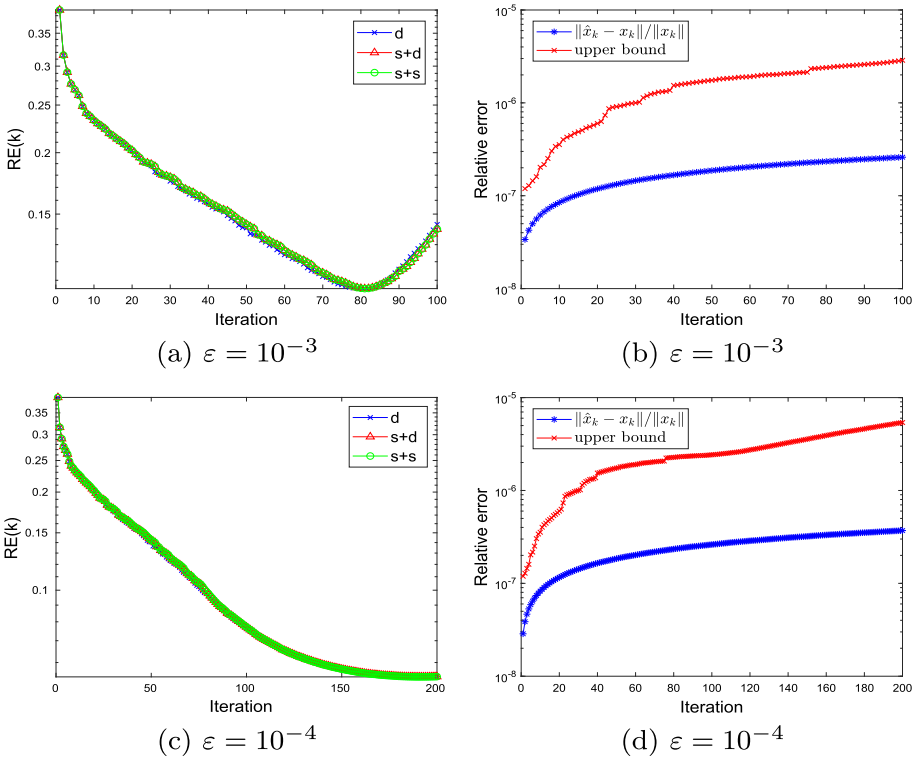


Fig. 9 Semi-convergence curves for LSQR implemented using different computing precisions, and relative errors between regularized solutions computed by "s+d" and "s+s", PRblurdefocus

From the above experimental results, we can make sure that single precision computing of LBFRO and updating x_k is enough for LSQR for solving the 2-D image deblurring ill-posed problems if the noise level is not extremely small. For those large scale problems, the mixed precision variant of LSQR has a great potential of defeating the double precision implementation in computation efficiency. We will consider implementing and optimizing mixed precision codes of LSQR for large scale problems on a high performance computing architecture in the future work.

6 Conclusion

For the most commonly used iterative regularization algorithm LSQR for solving linear discrete ill-posed problems, we have investigated how to get a mixed precision implementation by analyzing the choice of proper computing precision for the two main parts of the algorithm, including the construction of Krylov subspace and updating procedure of iterative solutions. Based on the commonly used regularization model for linear inverse problems, we have shown that, for not extremely small noise levels, single precision is enough for computing Lanczos vectors with full reorthogonalization without loss of any accuracy of the final regularized solution. For the updating procedure, we have shown that the update of x_k and w_k, which is the most time consuming part, can be performed using single precision without

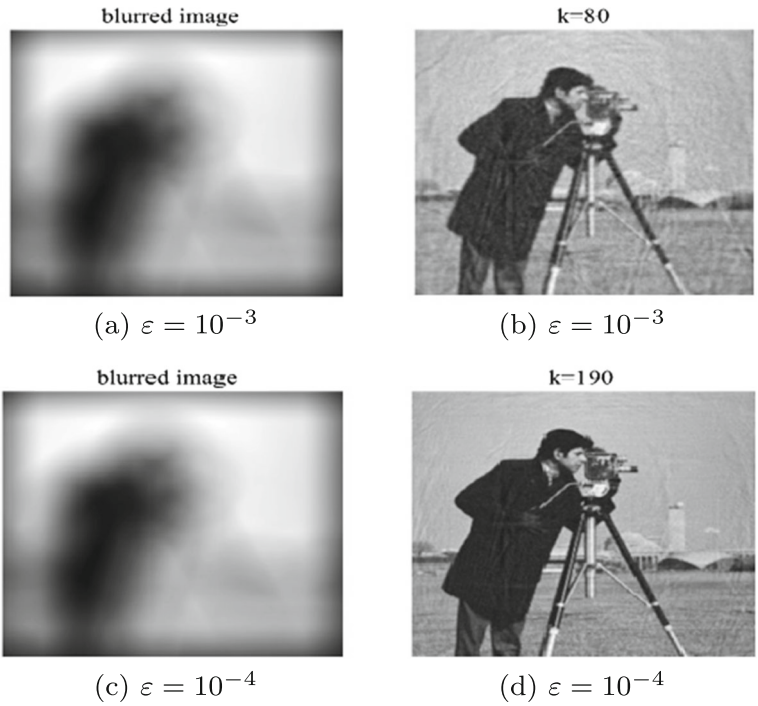


Fig. 10 Images "Cameraman" blurred by PRblurdefocus and restored by the mixed precision LSQR "s+s" at the optimal iteration: **a**, **b** $\varepsilon = 10^{-3}$; **c**, **d** $\varepsilon = 10^{-4}$

Table 4 Comparison of relative errors RE(k) and estimates of the optimal iterations k_0 by L-curve criterion and DP ($\tau = 1.001$)

Work precision	PRblurspeckle $\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	PRblurdefocus $\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$
Optimal				
d	0.1850 (34)	0.1102 (135)	0.1058 (80)	0.0542 (190)
s+d	0.1850 (34)	0.1102 (135)	0.1058 (81)	0.0542 (191)
s+s	0.1850 (34)	0.1102 (135)	0.1058 (81)	0.0542 (191)
L-curve				
d	0.1992 (47)	0.1105 (149)	0.1293 (96)	0.0764 (100)
s+d	0.1992 (47)	0.1105 (149)	0.1157 (91)	0.0771 (100)
s+s	0.1992 (47)	0.1103 (145)	0.1157 (91)	0.0771 (100)
Discrepancy principle				
d	0.1937 (24)	0.1200 (78)	0.1081 (74)	0.0599 (137)
s+d/s	0.1937 (24)	0.1200 (78)	0.1099 (74)	0.0599 (137)

sacrificing any accuracy as long as $\kappa(\widehat{R}_k)$ is not very big that is almost always satisfied. Several numerical experiments are made to test two mixed precision variants of LSQR and confirm the theoretical results.

Our results indicate that several highly time consuming parts of the algorithm can be implemented using lower precisions, and provide a theoretical guideline for implementing a robust and efficient mixed precision variant of LSQR for solving discrete linear ill-posed problems. Future work includes developing practical C codes on high performance computing architectures for specific real applications.

Acknowledgements The author thank Dr. Long Wang and Professor Weile Jia for many useful discussions about lower and mixed precision computations on high performance computing architectures. The author is grateful to the anonymous referees for their detailed reading of the manuscript and providing insightful comments that helped to improve the paper.

Funding This work was supported in part by the National Natural Science Foundation of China under Grant No. 3192270206.

Data Availability The source code and data used in this work is available at https://github.com/Machealb/Lower_precision_solver.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this work.

References

1. Abdelfattah, A., Anzt, H., Boman, E.G., Carson, E., Cojean, T., Dongarra, J., Fox, A., Gates, M., Higham, N.J., Li, X.S., et al.: A survey of numerical linear algebra methods utilizing mixed-precision arithmetic. Int. J. High Perform Comput Appl **35**(4), 344–369 (2021). https://doi.org/10.1177/10943420211003313
2. Ahmad, K., Sundar, H., Hall, M.W.: Data-driven mixed precision sparse matrix vector multiplication for GPUs. ACM Trans. Archit. Code Optim. **16**, 51:1-51:24 (2019). https://doi.org/10.1145/3371275
3. Amestoy, P., Boiteau, O., Buttari, A., Gerest, M., Jézéquel, F., L'Excellent, J.Y., Mary, T.: Mixed precision low rank approximations and their application to block low rank LU factorization. Int. J. High Perform. Comput. Appl. (2021). https://hal.archives-ouvertes.fr/hal-03251738v1
4. Björck, Å.: A bidiagonalization algorithm for solving large and sparse ill-posed systems of linear equations. BIT Numer. Math. **28**(3), 659–670 (1988)
5. Björck, Å.: Numerical Methods for Least Squares Problems. SIAM, Philadelphia (1996)
6. Blanchard, P., Higham, N.J., Lopez, F., Mary, T., Pranesh, S.: Mixed precision block fused multiply-add: error analysis and application to GPU tensor cores. SIAM J. Sci. Comput. **42**(3), C124–C141 (2020). https://doi.org/10.1137/19M1289546
7. Borges, L.S., Bazán, F.S.V., Cunha, M.C.C.: Automatic stopping rule for iterative methods in discrete ill-posed problems. Comp. Appl. Math. **34**, 1175–1197 (2015)
8. Carson, E., Higham, N.J., Pranesh, S.: Three-precision GMRES-based iterative refinement for least squares problems. SIAM J. Sci. Comput. **42**(6), A4063–A4083 (2020). https://doi.org/10.1137/20M1316822
9. Chung, J., Gazzola, S.: Computational methods for large-scale inverse problems: a survey on hybrid projection methods. arXiv:2105.07221v2 (2021).
10. Chung, J., Nagy, J.G., O'Leary, D.P.: A weighted-GCV method for Lanczos-hybrid regularization. Electr. Trans. Numer. Anal. **28**(29), 149–167 (2008)
11. Durand, Y., Guthmuller, E., Fuguet, C., Fereyre, J., Bocco, A., Alidori, R.: Accelerating variants of the conjugate gradient with the variable precision processor. In: 2022 IEEE 29th Symposium on Computer Arithmetic (ARITH), pp. 51–57. IEEE (2022)
12. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of Inverse Problems. Kluwer Academic Publishers (2000)

13. Gazzola, S., Hansen, P.C., Nagy, J.G.: IR tools: a MATLAB package of iterative regularization methods and large-scale test problems. Numer. Algor. **81**(3), 773–811 (2019)
14. Gazzola, S., Novati, P., Russo, M.R.: On Krylov projection methods and Tikhonov regularization. Electr. Trans. Numer. Anal. **44**, 83–123 (2015)
15. Golub, G.H., Van Loan, C.F.: Matrix Computations, 4th edn. The Johns Hopkins University Press, Baltimore (2013)
16. Golub, G.H., Wahba, H.G.: Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics **21**(2), 215–223 (1979)
17. Gratton, S., Simon, E., Titley-Peloquin, D., Toint, P.: Exploiting variable precision in GMRES. arXiv:1907.10550v2 (2019).
18. Gratton, S., Simon, E., Titley-Peloquin, D., Toint, P.L.: Minimizing convex quadratics with variable precision conjugate gradients. Numer. Linear Algebra Appl. **28**(1), e2337 (2021)
19. Hansen, P.C.: Analysis of discrete ill-posed problems by means of the l-curve. SIAM Rev. **34**(4), 561–580 (1992)
20. Hansen, P.C.: Rank-deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion. SIAM, Philadelphia (1998)
21. Hansen, P.C.: Regularization Tools version 4.0 for Matlab 7.3. Numer. Algor. **46**(2), 189–194 (2007)
22. Hansen, P.C.: Discrete Inverse Problems: Insight and Algorithms. SIAM, Philadelphia (2010)
23. Hansen, P.C., Nagy, J.G., O'Leary, D.P.: Deblurring Images: Matrices. Spectra and Filtering. SIAM, Philadelphia (2006)
24. Higham, N.J.: Accuracy and Stability of Numerical Algorithms. SIAM, Philadelphia (2002)
25. Higham, N.J., Pranesh, S., Zounon, M.: Squeezing a matrix into half precision, with an application to solving linear systems. SIAM J. Sci. Comput. **41**(4), A2536–A2551 (2019). https://doi.org/10.1137/18M1229511
26. Hnětynková, I., Plešinger, M., Strakoš, Z.: The regularizing effect of the Golub-Kahan iterative bidiagonalization and revealing the noise level in the data. BIT Numer. Math. **49**(4), 669–696 (2009). https://doi.org/10.1007/s10543-009-0239-7
27. Jia, Z.: Regularization properties of LSQR for linear discrete ill-posed problems in the multiple singular value case and best, near best and general low rank approximations. Inverse Probl. **36**(8), 085009 (2020)
28. Kaipio, J., Somersalo, E.: Statistical and Computational Inverse Problems. Springer (2006)
29. Kilmer, M.E., O'Leary, D.P.: Choosing regularization parameters in iterative methods for ill-posed problems. SIAM J. Matrix Anal. Appl. **22**(4), 1204–1221 (2001)
30. Larsen, R.M.: Lanczos bidiagonalization with partial reorthogonalization. DAIMI Report Series (1998)
31. Li, H., Tan, G., Zhao, T.: Backward error analysis of the Lanczos bidiagonalization with reorthogonalization. arXiv:2210.10297v1 (2022).
32. Lopez, F., Mary, T.: Mixed precision LU factorization on GPU tensor cores: Reducing data movement and memory footprint. MIMS EPrint 2020, Manchester Institute for Mathematical Sciences, The University of Manchester, UK (2020). http://eprints.maths.manchester.ac.uk/2782/
33. Meurant, G., Strakoš, Z.: The Lanczos and conjugate gradient algorithms in finite precision arithmetic. Acta Numer. **15**, 471–542 (2006)
34. Morozov, V.A.: Regularization of incorrectly posed problems and the choice of regularization parameter. USSR Comput. Math. Math. Phys. **6**(1), 242–251 (1966)
35. Natterer, F.: The Mathematics of Computerized Tomography. SIAM, Philadelphia (2001)
36. NVIDIA: NVIDIA Tesla V100. https://www.nvidia.com/en-gb/data-center/tesla-v100/
37. Paige, C.C., Saunders, M.A.: LSQR: an algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Softw. **8**, 43–71 (1982)
38. Reichel, L., Sadok, H., Zhang, W.H.: Simple stopping criteria for the LSQR method applied to discrete ill-posed problems. Numer. Algor. **84**(4), 1381–1395 (2020)
39. Renaut, R.A., Vatankhah, S., Ardesta, V.E.: Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems. SIAM J. Sci. Statist. Comput. **39**(2), B221–B243 (2017)
40. Richter, M.: Inverse problems: basics. Theory Appl. Geophys. (2016). https://doi.org/10.1007/978-3-319-48384-9_3
41. Tikhonov, A.N., Arsenin, V.Y.: Solutions of Ill-Posed Problems. Washington, DC (1977)
42. Vogel, C.R.: Computational Methods for Inverse Problems. SIAM, Philadelphia (2002)
43. Wedin, P.Å.: Perturbation bounds in connection with singular value decomposition. BIT Numer. Math. **12**(1), 99–111 (1972)
44. Yang, L.M., Fox, A., Sanders, G.: Rounding error analysis of mixed precision block householder QR algorithms. SIAM J. Sci. Comput. **43**(3), A1723–A1753 (2021). https://doi.org/10.1137/19M1296367