Les Cahiers du GERAD ISSN: 0711–2440

RipQP: A multi-precision regularized predictor-corrector method for convex quadratic optimization

G. Leconte, D. Orban

G–2021–03

January 2021

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2021
– Bibliothèque et Archives Canada, 2021

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2021
– Library and Archives Canada, 2021

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

RipQP: A multi-precision regularized predictor-corrector method for convex quadratic optimization

Geoffroy Leconte [a, b]

Dominique Orban [a, c]

[a] *GERAD, Montréal (Québec), Canada*

[b] *École Centrale, Nantes, France*

[c] *Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Québec), Canada*

geoffroy.leconte@eleves.ec-nantes.fr

dominique.orban@gerad.ca

January 2021
Les Cahiers du GERAD
G–2021–03

Abstract : We describe a Julia implementation of Mehrotra's predictor-corrector method for convex quadratic optimization that is entirely open source and generic in that it is able to accommodate computations in multiple floating-point systems. As a result, our code, named RipQP [24], can be initialized in a low-precision system such as Float16 as a form of warm start, and gradually transition through higher-precision systems until it reaches the accuracy tolerances prescribed by the user. On platforms with hardware for various floating-point systems, our strategy results in savings in terms of time, number of normalized iterations, and energy expended during the computations. RipQP employs primal and dual regularization as described by Friedlander and Orban [8], solves symmetric and quasi-definite systems at each iteration, and is competitive with Gurobi, CPLEX and Xpress on standard collections of linear and quadratic optimization problems in double precision in terms of number of iterations and time.

Keywords: Interior-point methods, convex quadratic optimization, predictor-corrector, multi-precision, regularization

Acknowledgements: Research of G. Leconte is supported by an IVADO Undergraduate Research Scholarship. Research of D. Orban is partially supported by an NSERC Discovery Grant.

1 Introduction

We describe an efficient implementation of the Mehrotra [22] predictor-corrector method named RipQP for the convex quadratic problem

$$\underset{x}{\text{minimize}}\ \ c^T x + \tfrac{1}{2} x^T Q x \quad \text{subject to}\ \ Ax = b,\ \ \ell \le x \le u, \tag{1}$$

where $c \in \mathbb{R}^n$, $Q = Q^T \in \mathbb{R}^{n\times n}$ is positive semi-definite, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $\ell \in \mathbb{R}^n$, $u \in \mathbb{R}^n$, and inequalities are understood elementwise. Although there already exist several commercial and freely-available implementation of the predictor-corrector method, ours has a number of distinguishing features, including:

1. treatment of free variables, weak convexity and rank-deficient constraints by way of exact primal-dual regularization [8];
2. effective scaling strategy based on that of Ruiz [30];
3. multiple centrality corrections as suggested by Gondzio [10];
4. our algorithm is entirely implemented in the Julia language [5] with no compiled or operating-system-specific dependencies, and therefore is portable, is compiled on the fly, and does not require a separate compiler;
5. a generic implementation that can be executed in any floating-point arithmetic available on the host platform and supported by Julia;
6. a multi-precision heuristic attempting to save time, computations and energy on appropriate platforms;
7. double precision efficiency competitive with that of the commercial libraries CPLEX [15], Gurobi [12] and Xpress [7];
8. our software is open source and freely available from https://github.com/JuliaSmoothOptimizers/RipQP.jl.

For simplicity of exposition, we assume that all elements of ℓ and u are finite. Accommodating infinite bounds, or more precisely, absence of bounds, is easily achieved by deleting rows and/or columns from matrices involving those bounds. If linear inequality constraints $b_\ell \le Ax \le b_u$ are present, our implementation adds slack variables $Ax - t = 0$ and the bounds $b_\ell \le t \le b_u$ to recover the form (1).

The dual of (1) is

$$\begin{aligned} &\underset{x,y,s_\ell,s_u}{\text{maximize}}\ y^T b + s_\ell^T \ell - s_u^T u - \tfrac{1}{2} x^T Q x \\ &\text{subject to}\ -Qx + A^T y + s_\ell - s_u = c,\ \ s_\ell \ge 0,\ \ s_u \ge 0, \end{aligned} \tag{2}$$

where $y \in \mathbb{R}^m$, $s_\ell \in \mathbb{R}^n$ and $s_u \in \mathbb{R}^n$ are vectors of Lagrange multipliers associated with the equality constraints, lower bounds, and upper bounds of (1), respectively. A solution (x, y, s_ℓ, s_u) of (1)–(2), when one exists, satisfies the Karush-Kuhn-Tucker (KKT) conditions

$$F(x, y, s_\ell, s_u) := \begin{bmatrix} -Qx + A^T y + s_\ell - s_u - c \\ Ax - b \\ S_\ell(x - \ell) \\ S_u(u - x) \end{bmatrix} = 0, \tag{3a}$$

$$s_\ell \ge 0, \quad s_u \ge 0, \quad \ell \le x \le u, \tag{3b}$$

where $S_\ell := \operatorname{diag}(s_\ell)$ and $S_u := \operatorname{diag}(s_u)$. The conditions (3) can be generalized to problems with infinite bounds in ℓ and u by removing the associated components of x and columns of S_ℓ and/or S_u.

The Mehrotra [22] predictor-corrector algorithm is one of the most popular and successful computational procedures to solve (1). Its main computational cost consists in solving a linear system often cast as the symmetric indefinite

$$\begin{bmatrix} -(Q + D) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}, \tag{4}$$

where D is positive semi-definite and diagonal, sometimes known as the K_2 formulation [9, 11]. The formulation (4) may be numerically unstable in the sense that its condition number grows without bound as convergence occurs [11]. Like other formulations, (4) is singular if A is rank deficient and does not lend itself to a convenient treatment of free variables when $Q = 0$. For those reasons, we use the exact primal-dual regularization analyzed by Friedlander and Orban [8]

$$\begin{bmatrix} -(Q + D + \rho I) & A^T \\ A & \delta I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}, \tag{5}$$

where the iteration-dependent positive regularization parameters ρ and δ have small values (typically close to the square root of machine precision). We also employ a scaling procedure inspired from that of Ruiz [30] to improve numerical stability, and an initialization inspired from those of Mehrotra [22] and Friedlander and Orban [8].

System (1) is commonly solved in double floating-point arithmetic. However, on platforms with hardware able to work with coarser floating-point systems, it is convenient to perform part of the computations in lower precision as lower-precision computations are performed faster and expend less energy than higher-precision computations, as observed by Haidar et al. [13]. In certain applications, the data, although given in double precision, spans many orders of magnitude, and it is necessary to identify and precision to higher accuracy than double precision allows, which lead Ma et al. [20] to design a quadruple-precision version of the solver MINOS.

Our implementation can be employed in any fixed floating-point system supported by Julia, including half, single, double, quadruple, and so-called BigFloat systems, which permit arbitrary-precision arithmetic simulated at the software level. RipQP is also able to perform iterations in a lower precision system until a relaxed optimality tolerance is attained, and gradually transition to higher precision systems. The Julia programming language is ideal for such an implementation thanks to its dispatch features, which make it possible to implement generic type-stable functions, a specialized version of which will be compiled on demand to suit the type of its input arguments. In particular, we implemented a single function to compute the predictor-corrector iterates, which is used regardless of the precision used.

Improvements such as the *zoom* procedure of Ma et al. [20] or the iterative refinement of Weber et al. [34] are the subject of ongoing work. Nevertheless, we are able to obtain residuals significantly smaller than we would in double precision on the problems of [20].

Related research

Greif et al. [11] and Ghannad et al. [9] discuss the formulation and condition number of several systems, including (4), and each provide a formulation with bounded condition number.

Friedlander and Orban [8] introduce the exact primal-dual regularization process using a slightly different formulation of (1). A computational benefit is to be able to work with an augmented system such as (4), which is often sparser than the normal equations

$$A(Q + D)^{-1} A^T \Delta y = f_y + A(Q + D)^{-1} f_x,$$

but, thank to the modified diagonal, is symmetric and quasi-definite (SQD), and can be solved by a sparse indefinite Cholesky factorization [33]. Friedlander and Orban [8] also establish convergence of the regularized approach.

Altman and Gondzio [3] perform the regularization and the factorization simultaneously, as a form of dynamic regularization, which departs from the convergence theory but is effective is practice.

The Julia language is receiving increasing attention from the optimization community. Anjos et al. [4] describe the implementation of an interior-point solver named Tulip for linear optimization based on the homogeneous self-dual form, with special focus on structured problems and customized linear algebra. Tulip also employs exact primal-dual regularization.

Using finer floating-point systems than double precision is a way to solve problems where conventional double precision solvers fail or are otherwise insufficient. Ma et al. [20] describe the *zoom* method, which is able perform iterations in quadruple precision to solve difficult multiscale problems.

Notation

We use Householder notation throughout: capital Latin letters such as A, B, and Q represent matrices. Lowercase letters such as s, x, and y represent real vectors. e is a vector of ones. I is the identity matrix. Lowercase Greek letters such as α and μ are scalars, except when we use Δx, which denotes a search direction in the variables x. If x is a real vector, we use X to denote the diagonal matrix $\mathrm{diag}(x)$. For conciseness, we sometimes abuse notation and write (x, y, s) to denote the vector $\begin{bmatrix} x^T & y^T & s^T \end{bmatrix}^T$.

2 A regularized predictor-corrector algorithm

In this section, we describe the main structure of RipQP's algorithm in a fixed precision. The algorithm follows closely the infeasible Mehrotra [22] predictor-corrector algorithm adapted to convex quadratic problems with both lower and upper bounds, which differs slightly from Friedlander and Orban [8].

2.1 Overview of the predictor-corrector algorithm

In this section, we provide a condensed overview of the main stages of the well-known Mehrotra [22] predictor-corrector algorithm. We refer the interested reader to the original paper and to [35] and references therein for a thorough review of primal-dual interior-point methods.

Infeasible primal-dual interior-point methods generate iterates (x, y, s_ℓ, s_u) that are strictly feasible with respect to the bounds (3b) and approximately satisfy a perturbation of (3a) defined by

$$F(x, y, s_\ell, s_u) = (0, 0, \omega e, \omega e), \tag{6}$$

where $\omega > 0$ is a parameter. Points (x, y, s_ℓ, s_u) that are strictly feasible with respect to (3b) and exactly satisfy (6) define a parametrized curve named the *central path* and the hope is that as $\omega \to 0$, the central path leads to a solution of (1) and (2).

If (x, y, s_ℓ, s_u) is strictly feasible with respect to (3b), we define the primal residual, dual residual, and duality measure as

$$r_b := Ax - b \tag{7a}$$

$$r_c := -Qx + A^T y + s_\ell - s_u - c \tag{7b}$$

$$\mu := \frac{s_\ell^T (x - \ell) + s_u^T (u - x)}{2n}. \tag{7c}$$

Indeed if (x, y, s_ℓ, s_u) satisfied the equality constraints of (1) and (2), μ would be equal to the duality gap. Most interior-point methods generate iterates that only satisfy (6) approximately for parameter values $\omega := \sigma\mu$, where σ is an iteration-dependent *centering* parameter.

The Mehrotra predictor-corrector algorithm is an infeasible interior-point methods in which each iteration consists in computing two steps. The first step is the predictor, or affine-scaling, step, and is a Newton step applied to (6) with $\sigma = 0$, i.e.,

$$\begin{bmatrix} -Q & A^T & I & -I \\ A & 0 & 0 & 0 \\ S_\ell & 0 & X - L & 0 \\ S_u & 0 & 0 & X - U \end{bmatrix} \begin{bmatrix} \Delta x^{\mathrm{aff}} \\ \Delta y^{\mathrm{aff}} \\ \Delta {s_\ell}^{\mathrm{aff}} \\ \Delta {s_u}^{\mathrm{aff}} \end{bmatrix} = - \begin{bmatrix} r_c \\ r_b \\ S_\ell (x - \ell) \\ S_u (x - u) \end{bmatrix}, \tag{8}$$

where $X := \mathrm{diag}(x)$, $L := \mathrm{diag}(\ell)$ and $U := \mathrm{diag}(u)$. We then compute a steplength along the primal and dual search directions that preserves sufficient strict feasibility with respect to (3b):

$$\alpha_{\mathrm{pri}}^{\mathrm{aff}} = \arg\max\{\alpha \in (0, 1] \mid x + \alpha \Delta x^{\mathrm{aff}} \in [\ell + \tau(x - \ell), u - \tau(u - x)]\} \tag{9a}$$

$$\alpha_{\text{dual}}^{\text{aff}} = \arg\max\{\alpha \in (0,1] \mid s_\ell + \alpha \Delta s_\ell{}^{\text{aff}} \geq \tau s_\ell \text{ and } s_u + \alpha \Delta s_u{}^{\text{aff}} \geq \tau s_u\} \tag{9b}$$

for a small user-defined $\tau \in (0,1)$. The objective of the predictor step is to estimate a sensible value of σ to be used when computing the second step.

The second step aims to strike a balance between aiming for optimality ($\sigma \approx 0$) and bringing the next iterate towards the central path ($\sigma \approx 1$). That is achieved by computing the duality measure that would result from the predictor step, and setting the centering parameter according to the heuristic of Mehrotra

$$x^{\text{aff}} := x + \alpha_{\text{pri}}^{\text{aff}} \Delta x^{\text{aff}} \tag{10a}$$

$$(y^{\text{aff}}, s_\ell^{\text{aff}}, s_u^{\text{aff}}) := (y, s_\ell, s_u) + \alpha_{\text{dual}}^{\text{aff}}(\Delta y^{\text{aff}}, \Delta s_\ell^{\text{aff}}, \Delta s_u^{\text{aff}}) \tag{10b}$$

$$\sigma := (\mu^{\text{aff}}/\mu)^3. \tag{10c}$$

The centering-corrector step is then defined as

$$\begin{bmatrix} -Q & A^T & I & -I \\ A & 0 & 0 & 0 \\ S_\ell & 0 & X-L & 0 \\ S_u & 0 & 0 & X-U \end{bmatrix} \begin{bmatrix} \Delta x^{\text{cc}} \\ \Delta y^{\text{cc}} \\ \Delta s_\ell^{\text{cc}} \\ \Delta s_u^{\text{cc}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma\mu e - \Delta S_\ell^{\text{aff}^T} \Delta x^{\text{aff}} \\ -\sigma\mu e - \Delta S_u^{\text{aff}^T} \Delta x^{\text{aff}} \end{bmatrix}. \tag{11}$$

The overall search direction is the combined step

$$(\Delta x, \Delta y, \Delta s_\ell, \Delta s_u) := (\Delta x, \Delta y, \Delta s_\ell, \Delta s_u)^{\text{aff}} + (\Delta x, \Delta y, \Delta s_\ell, \Delta s_u)^{\text{cc}}, \tag{12}$$

where we use the shorthand $(\Delta x, \Delta y, \Delta s_\ell, \Delta s_u)^{\text{aff}}$ to denote the solution of (8) and similarly for (11) for conciseness. We again compute a steplength that preserves sufficient strict feasibility

$$\alpha_{\text{pri}} = \arg\max\{\alpha \in (0,1] \mid x + \alpha\Delta x \in [\ell + \tau(x-\ell), u - \tau(u-x)]\} \tag{13a}$$

$$\alpha_{\text{dual}} = \arg\max\{\alpha \in (0,1] \mid s_\ell + \alpha\Delta s_\ell \geq \tau s_\ell \text{ and } s_u + \alpha\Delta s_u \geq \tau s_u\} \tag{13b}$$

and update the current iterate

$$x^{k+1} := x^k + \alpha_{\text{pri}}^k \Delta x^k \tag{14a}$$

$$(y^{k+1}, s_\ell^{k+1}, s_u^{k+1}) := (y^k, s_\ell^k, s_u^k) + \alpha_{\text{dual}}^k(\Delta y^k, \Delta s_\ell^k, \Delta s_u^k). \tag{14b}$$

In order to benefit from a sparse symmetric factorization, it is common to symmetrize (8) and (11). To that end, we eliminate Δs_ℓ and Δs_u from (8) and (11), and obtain the smaller, but sparse and symmetric systems

$$\begin{bmatrix} -(Q+D) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x^{\text{aff}} \\ \Delta y^{\text{aff}} \end{bmatrix} = -\begin{bmatrix} r_c - s_\ell + s_u \\ r_b \end{bmatrix} \tag{15a}$$

$$\Delta s_\ell^{\text{aff}} = -s_\ell - (X-L)^{-1} S_\ell \Delta x^{\text{aff}} \tag{15b}$$

$$\Delta s_u^{\text{aff}} = -s_u + (U-X)^{-1} S_u \Delta x^{\text{aff}}, \tag{15c}$$

and

$$\begin{bmatrix} -(Q+D) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x^{\text{cc}} \\ \Delta y^{\text{cc}} \end{bmatrix} = \begin{bmatrix} (X-L)^{-1} r_\ell + (U-X)^{-1} r_u \\ 0 \end{bmatrix} \tag{16a}$$

$$\Delta s_\ell^{\text{cc}} = -(X-L)^{-1}(r_\ell + S_\ell \Delta x^{\text{cc}}) \tag{16b}$$

$$\Delta s_u^{\text{cc}} = \quad (U-X)^{-1}(r_u + S_u \Delta x^{\text{cc}}), \tag{16c}$$

where

$$\begin{aligned} D &= (X-L)^{-1} S_\ell + (U-X)^{-1} S_u \\ r_\ell &= -\sigma\mu e + \Delta X_\ell^{\text{aff}} \Delta s_\ell^{\text{aff}} \\ r_u &= \quad \sigma\mu e + \Delta X_u^{\text{aff}} \Delta s_u^{\text{aff}}. \end{aligned}$$

The fomulation (15)–(16) is sometimes referred to as the K_2 formulation [9, 11].

2.2 Overview of the exact primal-dual regularization

If $Q + D$ is nearly singular or if A is nearly rank deficient, the factorization of the matrix of (15) and (16) may fail or be unstable. In addition, its condition number grows unbounded as convergence occurs [11]. Exact primal-dual regularization can be used to bypass those difficulties and to make the matrix SQD, which allows us to use an indefinite Cholesky factorization without pivoting for stability. Regularization changes the matrix of (15) and (16) to

$$\begin{bmatrix} -(Q + D + \rho I) & A^T \\ A & \delta I \end{bmatrix}, \tag{17}$$

where $\rho > 0$ and $\delta > 0$ are iteration-dependent regularization parameters. Friedlander and Orban [8] give an interpretation of the perturbation (17) in terms of a proximal method of multipliers applied to (1)–(2) and expose strong duality connections between the perturbed problems. Our implementation follows [8] and starts with relatively large values of ρ and δ, and divides them by 10 at each iteration until they reach a precision-dependant threshold (typically the square root of machine precision). Theorem 1 provides convergence properties of the regularized approach.

Theorem 1 (8, Theorem 5.8)**.** Suppose Algorithm 2 generates the bounded sequence $\{(x^k, y^k, s_\ell^k, s_u^k)\}$. Suppose also that there exist $k_0 \in \mathbb{N}$ and $\alpha^* \in (0, 1]$ such that $\alpha_{\text{pri}}^k \geq \alpha^*$ and $\alpha_{\text{dual}}^k \geq \alpha^*$for all $k \geq k_0$. Then every limit point of $\{(x^k, y^k, s_\ell^k, s_u^k)\}$ determines a primal-dual solution of (1)–(2).

Friedlander and Orban [8] indicate that the assumptions on α^* and k_0 may be avoided using an analysis in Wright [35, Chapter 6].

3 Implementation

In this section, we describe the aspects of our implementation that differ from other common implementation of the predictor-corrector method.

3.1 Starting point

In order to generalize the procedure of Mehrotra to two-sided bounds, we first solve

$$\begin{bmatrix} -(Q + \rho_0 I) & A^T \\ A & \delta_0 I \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}, \tag{18}$$

where $\rho_0 \geq 0$ and $\delta_0 \geq 0$ are initial regularization parameters, and set

$$\tilde{s_\ell} = Q\tilde{x} - A^T\tilde{y} + c \tag{19a}$$

$$\tilde{s_u} = -(Q\tilde{x} - A^T\tilde{y} + c). \tag{19b}$$

As in Friedlander and Orban [8], (18) has the same sparsity pattern as (17), which is substituted into (15) and (16) during the iterations. As that sparsity pattern remains constant along the iterations, we perform the symbolic analysis of the LDL^T factorization of (17) only once during the initialization. We adapt the starting point precedure described by Mehrotra [22] to two-sided bounds by introducing $\delta_{x_\ell} = \max(-\frac{3}{2}\min(\tilde{x} - \ell), 0)$, $\delta_{x_u} = \max(-\frac{3}{2}\min(u - \tilde{x}), 0)$, $\delta_{s_\ell} = \max(-\frac{3}{2}\min(s_\ell), 0)$, and $\delta_{s_u} = \max(-\frac{3}{2}\min(s_u), 0)$, and setting

$$\tilde{\delta}_{x_\ell} = \delta_{x_\ell} + \tfrac{1}{2}\frac{\tilde{s}_\ell^T(\tilde{x} + \delta_{x_\ell} - \ell)}{\sum_i (s_{\ell i} + \delta_{s_\ell})} \tag{20a}$$

$$\tilde{\delta}_{x_u} = \delta_{x_u} + \tfrac{1}{2}\frac{\tilde{s}_u^T(u - \tilde{x} - \delta_{x_u})}{\sum_i (s_{u i} + \delta_{s_u})} \tag{20b}$$

$$\tilde{\delta}_{s_\ell} = \delta_{s_\ell} + \tfrac{1}{2}\frac{\tilde{s}_\ell^T(\tilde{x} + \delta_{x_\ell} - \ell)}{\sum_i (\tilde{x}_i + \delta_{x_\ell} - \ell_i)} \tag{20c}$$

$$\tilde{\delta}_{s_u} = \delta_{s_u} + \tfrac{1}{2}\frac{\tilde{s}_u^T(u - \tilde{x} - \delta_{x_u})}{\sum_i (u_i - \tilde{x}_i - \delta_{x_u})}. \tag{20d}$$

Finally, we set the initial values

$$x_0 = \tilde{x} + \tilde{\delta}_{x_\ell} - \tilde{\delta}_{x_u} \tag{21a}$$

$$s_{\ell,0} = \tilde{s}_\ell + \tilde{\delta}_{s_\ell} \tag{21b}$$

$$s_{u,0} = \tilde{s}_u + \tilde{\delta}_{s_u}. \tag{21c}$$

By construction, δ_{s_ℓ} and δ_{s_u} ensure that $s_{\ell,0} \geq 0$ and $s_{u,0} \geq 0$. The additional shifts in (20) ensure that $s_{\ell,0}$ and $s_{u,0}$ are sufficiently positive. However, there may exist $i \in \{1, \dots, n\}$ such that $x_i \notin [\ell_i, u_i]$. In this case, we set x_i close to its closest bound. For example, if $x_{0,i} \leq \ell_i$, we set $x_{0,i} = \ell_i + \delta_i$, where $\delta_i > 0$.

The above can be adapted to infinite bounds. If ℓ_i or u_i is infinite, the above procedure guarantees that $x_i \in [\ell_i, u_i]$.

3.2 Scaling

In order to improve performance and robustness, we scale A using the method described by Ruiz [30], and then we scale Q. Let D_1, D_2 and D_3 be diagonal matrices used for the scaling procedure: D_1 and D_2 scale the rows and columns of A, respectively, and D_3 scales the rows and columns of Q. The process is described in Algorithm 1. As Q is symmetric, we apply Algorithm 1 with $A = Q$ and set D_3 to the common final value of $D_1^{(k)}$ and $D_2^{(k)}$ in the algorithm.

Algorithm 1 [30, Algorithm 2.1]: scale the $m \times n$ matrix A

Require: $\hat{A}^{(0)} = A$, $D_1^{(0)} = I_m$, and $D_2^{(0)} = I_n$.
1: **for** $k = 0, 1, 2, \dots$ until convergence **do**
2: Let $r_i^{(k)}$, $i \in \{1, ..., m\}$, and $c_j^{(k)}$, $j \in \{1, ..., n\}$, be the rows and columns of $\hat{A}^{(k)}$.
3: Set $D_R = \mathrm{diag}(\sqrt{\|r_i^{(k)}\|_\infty})_{i=1,\dots,m}$, and $D_C = \mathrm{diag}(\sqrt{\|c_i^{(k)}\|_\infty})_{j=1,\dots,m}$.
4: Set $\hat{A}^{(k+1)} = D_R^{-1}\hat{A}^{(k)}D_C^{-1}$.
5: Set $D_1^{(k+1)} = D_1^{(k)}D_R^{(-1)}$, and $D_2^{(k+1)} = D_2^{(k)}D_C^{-1}$.
6: **end for**

After scaling, (1) becomes

$$\underset{\hat{x} \in \mathbb{R}^n}{\text{minimize}} \;\; \hat{c}^T\hat{x} + \tfrac{1}{2}\hat{x}^T\hat{Q}\hat{x} \quad \text{subject to} \;\; \hat{A}\hat{x} = \hat{b}, \; \hat{\ell} \leq \hat{x} \leq \hat{u}, \tag{22}$$

where

$$\hat{Q} = D_3D_2QD_2D_3 \tag{23a}$$

$$\hat{A} = D_1AD_2D_3 \tag{23b}$$

$$\hat{c} = D_3D_2c \tag{23c}$$

$$\hat{b} = D_1b \tag{23d}$$

$$\hat{\ell} = (D_3D_2)^{-1}\ell \tag{23e}$$

$$\hat{u} = (D_3D_2)^{-1}u. \tag{23f}$$

At the end of the algorithm, we unscale the primal-dual variables using

$$x = D_2D_3\hat{x} \tag{24a}$$

$$y = D_1\hat{y} \tag{24b}$$

$$s_\ell = (D_2D_3)^{-1}\hat{s_\ell} \tag{24c}$$

$$s_u = (D_2D_3)^{-1}\hat{s_u}. \tag{24d}$$

3.3 Stopping criteria

Our algorithm stops when the following conditions hold:

$$\frac{\left|c^T x + x^T Q x - y^T b + s_\ell^T \ell - s_u^T u\right|}{1 + \left|c^T x + \frac{1}{2} x^T Q x + c_0\right|} \leq \epsilon_{pdd} \tag{25a}$$

$$\|r_c\|_\infty \leq (1 + \epsilon_{r_c}) \|r_{c,0}\|_\infty \tag{25b}$$

$$\|r_b\|_\infty \leq (1 + \epsilon_{r_b}) \|r_{b,0}\|_\infty, \tag{25c}$$

where c_0 is the constant term appearing in the objective of (1), or zero if there is none. The condition (25a) states that the scaled duality gap is sufficiently small, while (25b) and (25c) state that the dual and primal residuals are sufficiently small, respectively. In double precision, we use the default values: $\epsilon_{pdd} = 10^{-8}$, $\epsilon_{r_c} = 10^{-6}$, and $\epsilon_{r_b} = 10^{-6}$.

The overall algorithm may be summarized as Algorithm 2 where, for simplicity, we drop iteration indices.

Algorithm 2 RipQP

Require: Q, A, c, c_0, b, ℓ, u defined in (1).
1: Initialize (x, y, s_ℓ, s_u) as described in Section 3.1.
2: Initialize $\rho > 0$ and $\delta > 0$.
3: Set ε to the machine epsilon.
4: **for** $k = 1, 2, \dots$ **do**
5: If (25) is verified, declare convergence.
6: Solve (8) with matrix (17) to obtain $(\Delta x, \Delta y, \Delta s_\ell, \Delta s_u)^{\text{aff}}$.
7: Compute $(x, y, s_\ell, s_u)^{\text{aff}}$ and σ according to (9) and (10).
8: Solve (11) with matrix (17) to obtain $(\Delta x, \Delta y, \Delta s_\ell, \Delta s_u)^{\text{cc}}$.
9: Compute $(\Delta x, \Delta y, \Delta s_\ell, \Delta s_u)$ according to (12).
10: Update (x, y, s_ℓ, s_u) according to (13) and (14).
11: $\rho \leftarrow \max(\rho/10, \sqrt{\varepsilon})$, $\delta \leftarrow \max(\delta/10, \sqrt{\varepsilon})$.
12: **end for**

3.4 Multiple centrality corrections

RipQP has the possibility to use multiple centrality corrections, as described by Gondzio [10], whose objective is to reduce the number of iterations by computing higher-quality steps and better centered iterates. The number of corrections at each iteration has to be determined prior to the main computations. Too many corrections may slow progress, while too few corrections may result in badly-centered iterates. Gondzio [10] provides a heuristic that works well for the linear problems that we tested. However, it was significantly slower when applied to quadratic problems in spite of a reduced number of iterations. Currently, multiple centrality corrections are disabled by default in our implementation, but it is possible to activate then by modifying a parameter. We activate them in the numerical experiments of Section 6.

4 Numerical results

4.1 Implementation environment

RipQP implements Algorithm 2 in the Julia language [5], which is a high-level, high-performance programming language in which code is compiled just in time transparently to the user. One of its main advantages is that it is possible to write generic functions whose behavior depends on the types of their input arguments—a useful feature when working with several floating-point systems. We also make use of Julia packages from the JuliaSmoothOptimizers organization of Orban and Siqueira [25], including:

- QuadraticModels [27]: a package to model linear and quadratic problems using the QuadraticModel type, and pass them to solvers;
- NLPModels [26]: a modeling package for nonlinear problems with utilities such as the addition of slack variables to a problem;

- LDLFactorizations [23]: a package to factorize symmetric and quasi-definite matrices such as (17) and solve systems involving them using a sparse LDL^T factorization with diagonal D;
- SolverTools [29] and SolverBenchmark [28]: two packages to standardize the output of solvers so as to perform systematic benchmarks, produce performance profiles and tables of results.

In the remainder of this section, we summarize our finding in the form of Dolan and Moré [6] performance profiles. Complete numerical results can be found in Appendix A.

4.2 Comparisons with existing solvers in double precision

We tested RipQP on the Netlib linear optimization dataset [2], and the convex quadratic problems from the Maros and Mészáros [21] dataset. Since our implementation does not include a presolve procedure yet, we excluded problems with fixed variables.

We compare RipQP to the commercial solvers CPLEX [15], Gurobi [12], and Xpress [7]. We disabled presolve and the crossover, and we used the stopping criteria listed in Table 1. We note that as the solvers perform scaling methods differently, the comparisons may not be strictly accurate, however, this gives us a good order of magnitude of the performances of RipQP compared to other solvers. Moreover, RipQP's tolerances ϵ_{r_b} and ϵ_{r_c} are relative to their initial value computed with point obtained in Section 3.1.

Table 1: Stopping criteria of the tested solvers.

Solver	ϵ_{pdd}	ϵ_{r_b}	ϵ_{r_c}	scaling	crossover	presolve
Gurobi	1e-8	1e-6	1e-6	yes	no	no
CPLEX	1e-8	1e-6	1e-6	yes	no	no
Xpress	1e-8	1e-6	1e-6	yes	no	no
RipQP	1e-8	1e-6	1e-6	yes	no	no

Each solver is run with their barrier method. In order to streamline our numerical experiments, we developed thin interfaces to each solver that allow us to pass an instance of a QuadraticModel as input. Our interfaces are available as part of the JuliaSmoothOptimizers organization as QuadraticModelsCPLEX [17], QuadraticModelsGurobi [18], and QuadraticModelsXpress [19]. Each rests on the complete interfaces implemented in the packages CPLEX.jl,[1] Gurobi.jl[2] and Xpress.jl.[3]

On certain problems, the stopping conditions could not be attained even when ρ_k and δ_k were at their lowest allowed value equal to the square root of machine epsilon. That is likely due to the fact that D in (17) becomes arbitrarily ill conditioned as convergence occurs if there is at least one active bound at the solution. Allowing ρ and δ to decrease further allowed us to attain the stopping conditions but potentially makes the indefinite Cholesky factorization unstable. We found that, with the machine precision ε, $\rho_{\min} = 10^{-6}\sqrt{\varepsilon}$ and $\delta_{\min} = 10^{-1}\sqrt{\varepsilon}$ gave good results in double precision on the problems tested. If the factorization encounters numerical issues, we increase those values to $\rho_{\min} = 10\sqrt{\varepsilon}$, and $\delta_{\min} = 10\sqrt{\varepsilon}$ (and continue to increase them if the factorization encounters later numerical issues). In addition, provided at least one element of D exceeds a threshold value, we divide $\rho_{\min}$ by 10.

The performance profiles of Figure 1 report our findings. The measure used in both profiles is the CPU time.

The left profile of Figure 1 indicates that CPLEX is on average the fastest on the Netlib problems, but that RipQP's performance is intermediate between that of Gurobi an Xpress. On about 80% of the problems, RipQP's run time is within a factor of at most four of CPLEX. The right profile indicates that, on the Maros and Mészáros [21] dataset, RipQP's performance is almost on par with that of CPLEX, and exceeds that of both Gurobi and Xpress.

[1] https://github.com/jump-dev/CPLEX.jl
[2] https://github.com/jump-dev/Gurobi.jl
[3] https://github.com/jump-dev/Xpress.jl

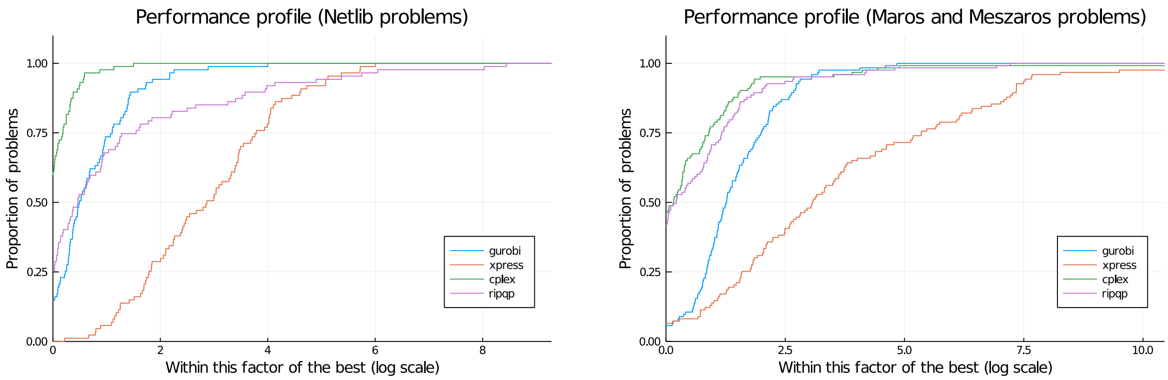

Figure 1: Time performance profiles between CPLEX, Gurobi, Xpress and RipQP in double precision on linear problems (left) and quadratic problems (right). The presolve and crossover are deactivated for CPLEX, Gurobi and Xpress.

5 Iterations in multiple floating-point systems

Certain multiscale problems require solutions to a degree of accuracy that is not attainable in double precision [20]. Thanks to Julia's multiple dispatch features, it is possible to run RipQP entirely in quadruple precision, or in arbitrary precision arithmetic without rewriting any code. On commodity hardware, high-precision operations are simulated at the software level via libraries such as GFortran's quadruple precision library [32], Intel's DecFP library for quadruple precision [16], the GNU Multi-Precision library [1], or a number of others. Such simulations are typically slow but the premise is of course that a processor with quadruple precision arithmetic implemented in hardware would execute similarly, only more efficiently.

Still, running the solver entirely in high-precision arithmetic can be wasteful and there may be better options on certain types of hardware. An increasing number of platforms nowadays have hardware offering access to several floating-point systems, including certain GPUs that feature half, single and double precision chips. Performing most of the computations in low-precision arithmetic may improve performance. For example, single precision operations execute faster than corresponding double precision operations. In addition, appropriate low-precision chips consume less energy and release less heat when performing operations than higher-precision chips [13, 31].

RipQP is able to perform initial computations in low-precision arithmetic as a form of warm-start, and gradually transition to higher-precision arithmetic. With such a strategy, we incur an increase in storage requirements as the problem data must be replicated and converted to each floating-point system employed during the solve.

5.1 Method

Algorithm 3 describes the simple strategy currently implemented in RipQP to enable multi-precision mode in the case of a solve in single and double precision. The process generalizes easily to more floating-point systems. The scaling is performed in the precision of the input data, which is, in our implementation, the highest precision used in the algorithm.

Algorithm 3 RipQP multi precision mode (single then double precision)

Require: Q, A, c, c_0, b, ℓ, u defined in (1) in double precision.
1: Duplicate Q, A, c, c_0, b, ℓ, u in single precision.
2: Apply Algorithm 2 in single precision until a relaxed stopping criteria is reached.
3: Convert (x, y, s_ℓ, s_u) and all the necessary storage to double precision.
4: Apply Algorithm 2 in double precision, bypassing the initialization procedure.

The main difficulty is to find adequate relaxed stopping criteria. If there is no numerical issues, the default parameters used for the relaxed stopping criterion in single precision are $\epsilon_{\tilde{pdd}} = 10^{-2}$ and

$\epsilon_{r_c}^{\sim} = \epsilon_{r_b}^{\sim} = 10^{-4}$. On some of the problems tested, letting the regularization parameters drop below the square root of machine epsilon is insufficient and still sometimes does not allow satisfaction of the stopping conditions. A simple strategy is to decrease the maximum number of iteration in order to avoid such situations. In addition, we transition to double precision if the single precision factorization fails.

5.2 Comparison with the mono-precision mode

We compare the number of iteration made by RipQP in mono-precision mode and in multi-precision mode (single precision then double precision). We choose to count four iterations in single precision for one iteration in double precision in order to simulate the amount of energy expended by operations in each floating-point system. Figure 2 summarizes our results.

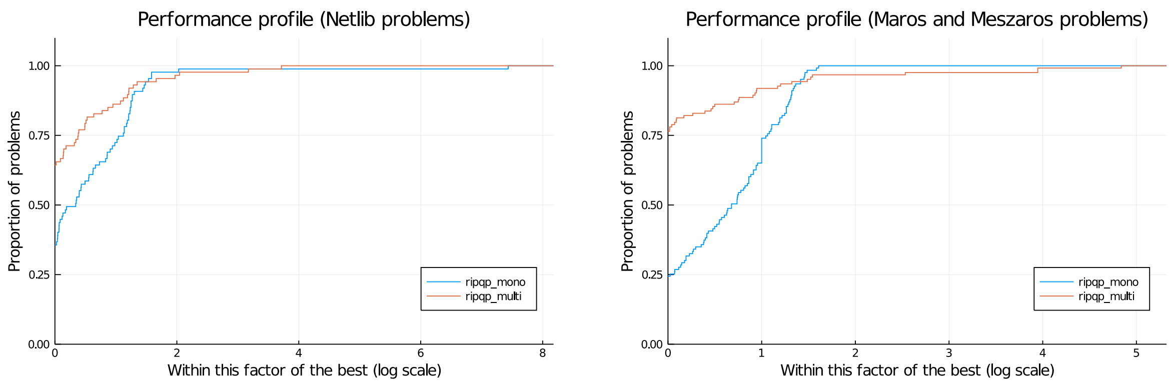

Figure 2: Iteration performance profile between RipQP in double precision (mono) mode, and in dual-precision (multi) mode on linear (left) and quadratic (right) optimization problems. Each iteration in double precision counts for four iterations.

While some problems require more iterations because single precision is not sufficient to make significant improvements towards a solution, for most of them, the number of iterations is smaller in multi-precision mode. Figure 2 shows that the difference can be as large as a factor of about four on linear and quadratic problems. We are well aware that the strategy implemented is rather simplistic, but it is illuminating to see that there are large gains to be made, and we feel that the preliminary results above justify further investigation into more sophisticated strategies.

6 Quadruple precision solution of multiscale problems

RipQP is able to perform iterations in high-precision systems such as quadruple precision. In this section, we illustrate results on the three difficult multiscale problems described by Ma et al. [20].

The multiscale nature of those problems makes it difficult to attain the accuracy required without presolve. If we first ask CPLEX [15] to presolve the problems, we obtain the results of Table 3. Since these problems were too hard to solve with RipQP's default transition values in multi-precision mode, we show the values we used in Table 2. 32 caracterises stopping criteria in single precision, and 64 in double precision. k is the maximum number of iterations. The mode column states if the algorithm uses several floating point systems (multi) transitioning from single to quadruple precision, or only one floating point system (mono) in quadruple precision. We set the final tolerances to 0, and we stopped the algorithm once it would not make progress anymore (when α_{pri} and α_{dual} in (13) would be stuck close to 0), so that we could see what would be the lowest attainable residuals.

Problem GlcAlift was not solved using the mono precision mode, because numerical issues with the factorization led quickly to transitioning to quadruple precision. The multiple centrality corrections method was only used with GlcAlift because we found that it leads to numerical issues in single precision for these difficult problems. The final residuals are not as small as those found by Ma et al. [20], but they are small enough to conclude on the convergence of the models, and the objective values are also very close (the eleven first significant digits that we found match all the digits shown in [20]).

Table 2: RipQP parameters on the problems described in [20].

Problem	mode	k_{32}	k_{64}	$\epsilon_{pdd,64}$	$\epsilon_{r_b,64}$	$\epsilon_{r_c,64}$	corrections
TMA_ME	multi	200	2000	1e4	1e-4	1e-4	no
GlcAerWT	multi	200	2000	1e4	1e-2	1e-2	no
GlcAlift	mono	/	/	/	/	/	yes

Table 3: RipQP results using quadruple precision on the problems described in [20].

Problem	objective	$\|r_b\|_\infty$	$\|r_c\|_\infty$
TMA_ME	8.7036315385e−07	3e−20	3e−17
GlcAerWT	−7.0382449681e+05	3e−19	7e−21
GlcAlift	−7.0434008750e+05	2e−26	7e−18

7 Dynamic regularization

As RipQP is used across several floating-point systems, it becomes inconvenient to tune ρ and δ, and to balance efficiency and robustness while ensuring that (17) is sufficiently quasi definite. An alternative is to solve (15) and (16) with dynamic regularization, computed on the fly during the factorization, as described by Altman and Gondzio [3]. Instead of using constant ρ and δ, we regularize pivots only if their absolute value is smaller than the infinite norm of (17) multiplied by the machine precision. The values of the corrections, if needed, are the same as in [3]: $-\varepsilon^{3/4}$ for the n first pivots, and $\varepsilon^{1/2}$ for the m last pivots.

The profiles in Figure 3 show that the number of iterations with dynamic regularization is close to that of the regularization strategy of Section 2.2.

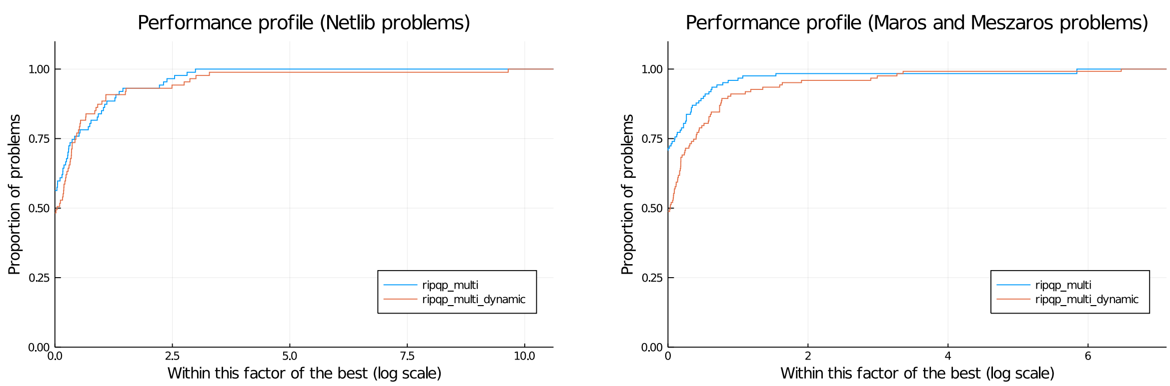

Figure 3: Iteration performance profile between RipQP in dual precision (multi) mode with and without dynamic regularization on linear (left) and quadratic (right) optimization problems. Each iteration in double precision counts for four iterations.

Figure 4 shows that the time in mono-precision mode with dynamic regularization is also close to that with the strategy of Section 2.2.

The above results suggest that dynamic regularization may be a suitable option in multi-precision mode, where managing ρ and δ explicitly is delicate.

8 Discussion

Despite being a very young implementation, RipQP is efficient and competitive with commercial solvers on our test set. The ability to perform iterations in several floating-point systems as a form of

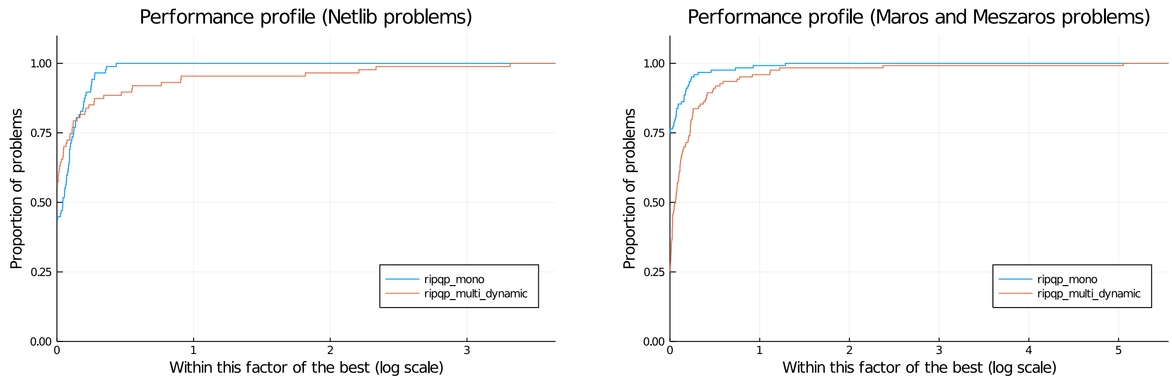

Figure 4: Time performance profiles between RipQP in mono precision mode with and without dynamic regularization on linear (left) and quadratic (right) optimization problems.

warm start is also promising and opens the door to sophisticated strategies that perform most of the computations in low-precision arithmetic, yet are able to attain high-precision tolerances. However, the transition thresholds between floating-point systems needs to be improved. One drawback of the method is that it is mandatory to duplicate the problem data to lower precision systems, which increases the required memory.

We have not yet added the possibility to perform iterations in half precision because numerous numerical difficulties occur. We are currently investigating scaling strategies such as those suggested by Higham et al. [14].

It is also encouraging that RipQP is able to satisfactory accuracy on the multiscale problems of Ma et al. [20].

Dynamic regularization, despite being slightly slower than our default implementation, is convenient because it removes the difficulty of finding good values for the regularization parameters ρ and δ.

RipQP is in active development. Among other improvements, we are working to add other formulations than (15) and (16), such as the $K_{2.5}$ formulation of Ghannad et al. [9], which has bounded condition number.

Appendix

A RipQP detailed results

To compare single and double precision, we decided that one iteration in double precision accounts for four iterations.

Table 4: RipQP results in mono precision mode (double floating-point precision) on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
25FV47	1571	821	acceptable	5.50e+03	1.81e−01	104	2.14e−10	1.03e−10
ADLITTLE	97	56	acceptable	2.25e+05	2.19e−03	56	8.52e−10	2.13e−10
AFIRO	32	27	acceptable	−4.65e+02	7.49e−04	36	5.68e−14	1.08e−11
AGG	163	488	acceptable	−3.60e+07	1.84e−02	84	2.23e−07	1.96e−08
AGG2	302	516	acceptable	−2.02e+07	7.20e−02	92	2.31e−05	6.20e−06
AGG3	302	516	acceptable	1.03e+07	6.68e−02	88	6.73e−08	8.13e−08
BANDM	472	305	acceptable	−1.59e+02	5.68e−01	68	1.40e−11	1.64e−11
BEACONFD	262	173	acceptable	3.36e+04	9.25e−03	48	2.66e−10	1.47e−08
BLEND	83	74	acceptable	−3.08e+01	1.87e−03	44	8.11e−14	7.11e−11
BNL1	1175	643	acceptable	1.98e+03	5.80e−02	120	2.52e−09	4.85e−11

Continued on next page

Table 4: RipQP results in mono precision mode (double floating-point precision) on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
BNL2	3489	2324	acceptable	1.81e+03	7.44e−01	156	1.16e−10	6.96e−10
BOEING1	384	351	acceptable	−3.35e+02	3.23e−02	112	4.08e−05	1.17e−10
BOEING2	143	166	acceptable	−3.15e+02	8.41e−03	88	2.06e−11	3.99e−14
BRANDY	249	220	acceptable	1.52e+03	1.09e−02	68	6.14e−12	2.91e−11
CRE-A	4067	3516	acceptable	2.36e+07	1.98e−01	120	1.09e−08	1.98e−08
CRE-B	72447	9648	acceptable	2.31e+07	3.13e+01	164	4.14e−07	1.40e−09
CRE-C	3678	3068	acceptable	2.53e+07	1.73e−01	124	1.94e−09	2.10e−08
CRE-D	69980	8926	acceptable	2.45e+07	2.18e+01	156	4.91e−10	1.03e−08
CYCLE	2857	1903	acceptable	−5.23e+00	3.62e−01	104	1.09e−11	5.01e−13
D2Q06C	5167	2171	acceptable	1.23e+05	1.55e+00	136	2.01e−08	4.28e−06
D6CUBE	6184	415	acceptable	3.15e+02	5.09e−01	92	2.87e−10	3.55e−15
DEGEN2	534	444	acceptable	−1.44e+03	3.80e−02	56	2.08e−11	1.48e−11
DEGEN3	1818	1503	acceptable	−9.87e+02	5.72e−01	72	1.45e−11	3.26e−10
DFL001	12230	6071	acceptable	1.13e+07	1.18e+02	172	1.88e−05	1.67e−08
E226	282	223	acceptable	−1.16e+01	1.59e−02	80	4.49e−10	4.88e−12
FFFFF800	854	524	acceptable	5.56e+05	1.06e−01	140	7.06e−10	7.74e−07
FIT1D	1026	24	acceptable	−9.15e+03	3.63e−02	80	1.21e−10	3.41e−13
FIT1P	1677	627	acceptable	9.15e+03	3.03e−02	52	6.96e−11	3.56e−11
FIT2D	10500	25	acceptable	−6.85e+04	4.81e−01	92	3.86e−10	3.41e−13
FIT2P	13525	3000	acceptable	6.85e+04	3.15e−01	72	7.26e−10	1.98e−10
GANGES	1681	1309	acceptable	−1.10e+05	6.39e−02	104	1.60e−07	1.75e−10
GFRD-PNC	1092	616	acceptable	6.90e+06	1.66e−02	88	6.45e−10	1.86e−10
GROW15	645	300	acceptable	−1.07e+08	2.01e−02	56	7.87e−07	1.48e−13
GROW22	946	440	acceptable	−1.61e+08	2.95e−02	60	7.87e−07	5.83e−14
GROW7	301	140	acceptable	−4.78e+07	9.23e−03	56	7.79e−09	4.97e−14
ISRAEL	142	174	acceptable	−8.97e+05	1.08e−02	80	3.49e−10	5.77e−12
KB2	41	43	acceptable	−1.75e+03	1.75e−03	72	4.39e−09	1.13e−08
KEN-07	3602	2426	acceptable	−6.80e+08	5.54e−02	64	9.00e−10	5.73e−09
KEN-11	21349	14694	acceptable	−6.97e+09	8.34e−01	96	9.16e−10	5.87e−08
KEN-13	42659	28632	acceptable	−1.03e+10	2.92e+00	120	3.42e−08	1.27e−10
KEN-18	154699	105127	acceptable	−5.22e+10	4.69e+01	168	4.42e−07	1.81e−09
LOTFI	308	153	acceptable	−2.53e+01	7.45e−03	88	1.86e−09	1.55e−15
MAROS-R7	9408	3136	acceptable	1.50e+06	1.35e+01	68	1.71e−10	2.54e−12
MODSZK1	1620	687	acceptable	3.21e+02	2.96e−02	104	5.82e−11	4.55e−13
OSA-07	23949	1118	acceptable	5.36e+05	1.00e+00	80	7.02e−09	6.37e−12
OSA-14	52460	2337	acceptable	1.11e+06	3.91e+00	104	4.84e−09	3.64e−11
OSA-30	100024	4350	acceptable	2.14e+06	1.10e+01	164	5.93e−09	6.96e−11
OSA-60	232966	10280	acceptable	4.04e+06	2.48e+01	120	1.55e−08	4.96e−11
PDS-02	7535	2953	acceptable	2.89e+10	3.00e−01	140	7.97e−09	1.12e−06
PDS-06	28655	9881	acceptable	2.78e+10	1.45e+01	164	7.67e−09	1.88e−06
PDS-10	48763	16558	acceptable	2.67e+10	9.43e+01	204	7.18e−09	4.20e−08
PDS-20	105728	33874	acceptable	2.38e+10	9.47e+02	240	6.92e−07	4.64e−08
QAP12	8856	3192	acceptable	5.23e+02	6.83e+01	72	3.73e−11	3.23e−10
QAP15	22275	6330	acceptable	1.04e+03	6.22e+02	88	6.55e−12	1.17e−10
QAP8	1632	912	acceptable	2.04e+02	8.48e−01	36	7.90e−13	4.67e−14
SC105	103	105	acceptable	−5.22e+01	2.05e−03	44	8.31e−13	7.51e−11
SC205	203	205	acceptable	−5.22e+01	4.21e−03	60	4.68e−12	5.34e−11
SC50A	48	50	acceptable	−6.46e+01	8.98e−04	36	5.97e−12	5.34e−12
SC50B	48	50	acceptable	−7.00e+01	9.23e−04	36	3.08e−13	9.63e−11
SCAGR25	500	471	acceptable	−1.48e+07	8.94e−03	64	4.70e−09	1.32e−09
SCAGR7	140	129	acceptable	−2.33e+06	2.43e−03	56	1.10e−10	2.44e−12
SCFXM1	457	330	acceptable	1.84e+04	1.60e−02	76	4.67e−09	7.25e−11
SCFXM2	914	660	acceptable	3.67e+04	3.63e−02	84	4.12e−10	1.09e−10
SCFXM3	1371	990	acceptable	5.49e+04	5.41e−02	84	6.90e−09	4.67e−10
SCORPION	358	388	acceptable	1.88e+03	6.25e−03	52	2.06e−10	1.39e−11
SCRS8	1169	490	acceptable	9.04e+02	1.89e−02	68	1.11e−10	3.46e−11
SCSD1	760	77	acceptable	8.67e+00	5.83e−03	32	1.59e−13	2.36e−13
SCSD6	1350	147	acceptable	5.05e+01	1.17e−02	40	1.80e−13	1.38e−14
SCSD8	2750	397	acceptable	9.05e+02	2.34e−02	40	3.24e−13	1.36e−12
SCTAP1	480	300	acceptable	1.41e+03	9.22e−03	60	1.10e−12	8.53e−14
SCTAP2	1880	1090	acceptable	1.72e+03	3.81e−02	56	2.20e−13	4.26e−14
SCTAP3	2480	1480	acceptable	1.42e+03	5.37e−02	60	1.13e−11	9.70e−13
SEBA	1028	515	acceptable	1.57e+04	2.38e−02	92	5.67e−12	1.85e−12
SHARE1B	225	117	acceptable	−7.66e+04	5.94e−03	84	6.99e−07	8.83e−07
SHARE2B	79	96	acceptable	−4.16e+02	2.84e−03	52	2.99e−12	7.43e−10

Continued on next page

Table 4: RipQP results in mono precision mode (double floating-point precision) on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
SHIP04L	2118	402	acceptable	1.79e+06	2.45e−02	56	2.67e−10	1.96e−11
SHIP04S	1458	402	acceptable	1.80e+06	1.78e−02	60	6.49e−10	6.64e−12
SHIP08L	4283	778	acceptable	1.91e+06	6.55e−02	84	1.05e−09	4.55e−12
SHIP08S	2387	778	acceptable	1.92e+06	3.31e−02	72	1.80e−09	2.01e−11
SHIP12L	5427	1151	acceptable	1.47e+06	7.04e−02	68	3.66e−11	7.93e−12
SHIP12S	2763	1151	acceptable	1.49e+06	3.30e−02	60	1.57e−09	1.58e−09
STOCFOR1	111	117	acceptable	−4.11e+04	2.27e−03	44	2.37e−10	3.05e−11
STOCFOR2	2031	2157	acceptable	−3.90e+04	6.44e−02	76	1.29e−09	3.23e−10
STOCFOR3	15695	16675	acceptable	−4.00e+04	9.14e−01	132	1.73e−10	2.56e−10
TRUSS	8806	1000	acceptable	4.59e+05	1.97e−01	68	4.86e−11	5.84e−10
WOOD1P	2594	244	acceptable	1.44e+00	2.81e−01	60	1.01e−09	1.71e−13
WOODW	8405	1098	acceptable	1.30e+00	3.56e−01	112	2.88e−10	5.46e−12

Table 5: RipQP results in mono precision mode (double floating-point precision) on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
AUG2D	20200	10000	acceptable	1.69e+06	2.17e−01	20	5.31e−12	2.13e−13
AUG2DC	20200	10000	acceptable	1.82e+06	2.06e−01	20	6.84e−12	2.68e−13
AUG2DCQP	20200	10000	acceptable	6.50e+06	4.52e−01	48	4.85e−12	1.55e−05
AUG2DQP	20200	10000	acceptable	6.24e+06	5.01e−01	48	7.14e−13	1.94e−05
AUG3D	3873	1000	acceptable	5.54e+02	2.20e−02	12	5.79e−11	3.85e−11
AUG3DC	3873	1000	acceptable	7.71e+02	2.23e−02	12	1.46e−10	6.68e−11
AUG3DCQP	3873	1000	acceptable	9.93e+02	6.89e−02	48	2.66e−14	4.39e−08
AUG3DQP	3873	1000	acceptable	6.75e+02	6.89e−02	48	4.00e−15	1.33e−15
BOYD1	93261	18	acceptable	−6.17e+07	2.35e+00	64	3.47e−02	4.54e−03
BOYD2	93263	186531	acceptable	2.13e+01	2.06e+01	368	1.21e−02	5.31e−10
CONT-050	2597	2401	acceptable	−4.56e+00	1.26e−01	28	1.04e−13	2.27e−12
CONT-100	10197	9801	acceptable	−4.64e+00	1.67e+00	44	9.55e−15	2.29e−13
CONT-101	10197	10098	acceptable	1.96e−01	6.47e+00	188	7.11e−15	1.46e−13
CONT-200	40397	39601	acceptable	−4.68e+00	1.96e+01	52	1.33e−13	1.29e−12
CONT-201	40397	40198	acceptable	1.92e−01	5.21e+01	144	8.88e−15	2.83e−12
CONT-300	90597	90298	acceptable	1.92e−01	2.01e+02	128	6.86e−14	1.59e−11
CVXQP1_L	10000	5000	acceptable	1.09e+08	7.04e+02	356	3.55e−15	3.05e−08
CVXQP1_M	1000	500	acceptable	1.09e+06	2.83e−01	60	3.75e−09	1.39e−05
CVXQP1_S	100	50	acceptable	1.16e+04	1.60e−03	24	1.52e−08	1.64e−05
CVXQP2_L	10000	2500	acceptable	8.18e+07	6.06e+01	40	4.48e−08	9.49e−04
CVXQP2_M	1000	250	acceptable	8.20e+05	9.98e−02	36	1.15e−09	5.72e−04
CVXQP2_S	100	25	acceptable	8.12e+03	1.65e−03	28	2.39e−11	1.42e−07
CVXQP3_L	10000	7500	acceptable	1.16e+08	7.98e+02	372	2.44e−10	2.23e−04
CVXQP3_M	1000	750	acceptable	1.36e+06	5.07e−01	96	5.31e−09	1.20e−09
CVXQP3_S	100	75	acceptable	1.19e+04	2.13e−03	32	5.46e−11	6.59e−07
DPKLO1	133	77	acceptable	3.70e−01	2.55e−03	12	1.08e−10	1.56e−10
DUAL1	85	1	acceptable	3.50e−02	4.57e−03	36	8.88e−16	4.95e−07
DUAL2	96	1	acceptable	3.37e−02	5.14e−03	28	2.55e−15	7.64e−07
DUAL3	111	1	acceptable	1.36e−01	8.95e−03	36	1.44e−15	1.78e−06
DUAL4	75	1	acceptable	7.46e−01	3.29e−03	32	1.09e−14	4.96e−13
DUALC1	9	215	acceptable	6.16e+03	5.28e−03	48	2.80e−11	5.33e−06
DUALC2	7	229	acceptable	3.55e+03	4.08e−03	40	1.39e−12	1.79e−07
DUALC5	8	278	acceptable	4.27e+02	3.78e−03	24	1.25e−12	3.92e−08
DUALC8	8	503	acceptable	1.83e+04	8.75e−03	36	6.82e−13	4.37e−08
EXDATA	3000	3001	acceptable	−1.42e+02	2.45e+01	48	1.78e−13	1.59e−07
GENHS28	10	8	acceptable	9.27e−01	2.25e−04	12	7.09e−12	1.99e−11
GOULDQP2	699	349	acceptable	1.84e−04	3.97e−02	400	4.27e−15	1.08e−06
GOULDQP3	699	349	acceptable	2.06e+00	8.16e−03	56	4.39e−15	3.78e−11
HS118	15	17	acceptable	6.65e+02	6.11e−04	60	7.82e−13	9.05e−09
HS21	2	1	acceptable	−1.00e+02	1.78e−04	20	2.82e−12	1.26e−08
HS268	5	5	acceptable	3.64e−10	3.08e−04	52	1.78e−14	2.18e−11
HS35	3	1	acceptable	1.11e−01	1.89e−04	28	1.15e−14	7.81e−11
HS51	5	3	acceptable	0.00e+00	1.23e−04	12	5.71e−10	2.74e−10
HS52	5	3	acceptable	5.33e+00	1.17e−04	12	1.52e−09	1.19e−09
HS53	5	3	acceptable	4.09e+00	1.32e−04	16	3.68e−13	6.22e−10
HS76	4	3	acceptable	−4.68e+00	2.09e−04	24	3.53e−13	5.62e−10
HUES-MOD	10000	2	acceptable	3.48e+07	4.94e−02	36	2.27e−12	9.14e−08

Continued on next page

Table 5: RipQP results in mono precision mode (double floating-point precision) on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
HUESTIS	10000	2	acceptable	3.48e+11	3.99e−02	28	9.29e−10	7.31e−03
KSIP	20	1001	acceptable	5.76e−01	1.15e−01	92	3.61e−13	8.51e−09
LASER	1002	1000	acceptable	2.41e+06	1.16e−01	516	5.21e−10	4.54e−07
LISWET1	10002	10000	acceptable	3.61e+01	4.28e−01	180	4.44e−16	2.45e−10
LISWET10	10002	10000	acceptable	4.95e+01	9.51e−01	432	6.68e−16	1.16e−07
LISWET11	10002	10000	acceptable	4.95e+01	8.79e−01	396	6.66e−16	3.01e−08
LISWET12	10002	10000	acceptable	1.74e+03	1.18e+00	548	5.10e−16	2.85e−09
LISWET2	10002	10000	acceptable	2.50e+01	4.23e−01	184	2.50e−14	4.34e−10
LISWET3	10002	10000	acceptable	2.50e+01	2.39e−01	96	8.03e−12	4.48e−07
LISWET4	10002	10000	acceptable	2.50e+01	3.62e−01	152	1.59e−11	5.57e−08
LISWET5	10002	10000	acceptable	2.50e+01	2.69e−01	108	1.36e−11	8.68e−08
LISWET6	10002	10000	acceptable	2.50e+01	3.32e−01	140	2.13e−11	5.94e−07
LISWET7	10002	10000	acceptable	4.99e+02	3.69e−01	156	3.33e−16	1.65e−09
LISWET8	10002	10000	acceptable	7.14e+02	8.29e−01	380	6.53e−16	2.53e−07
LISWET9	10002	10000	acceptable	1.96e+03	1.19e+00	552	4.43e−16	1.16e−09
LOTSCHD	12	7	acceptable	2.40e+03	3.16e−04	32	2.42e−13	2.06e−11
MOSARQP1	2500	700	acceptable	−9.53e+02	2.64e−02	36	7.39e−12	2.52e−06
MOSARQP2	900	600	acceptable	−1.60e+03	2.50e−02	40	3.31e−12	5.28e−06
POWELL20	10000	10000	acceptable	5.21e+10	3.19e−01	124	2.52e−10	1.19e−07
PRIMAL1	325	85	acceptable	−3.50e−02	1.34e−02	36	2.18e−13	8.81e−09
PRIMAL2	649	96	acceptable	−3.37e−02	1.80e−02	32	1.57e−13	1.25e−09
PRIMAL3	745	111	acceptable	−1.36e−01	5.80e−02	40	3.34e−12	1.47e−08
PRIMAL4	1489	75	acceptable	−7.46e−01	4.14e−02	40	1.07e−11	3.16e−15
PRIMALC1	230	9	acceptable	−6.16e+03	2.71e−03	40	4.44e−09	5.39e−11
PRIMALC2	231	7	acceptable	−3.55e+03	2.47e−03	48	1.46e−11	1.88e−09
PRIMALC5	287	8	acceptable	−4.27e+02	3.22e−03	44	3.41e−12	1.18e−09
PRIMALC8	520	8	acceptable	−1.83e+04	4.73e−03	32	2.26e−09	3.36e−07
Q25FV47	1571	820	acceptable	1.37e+07	7.95e−01	104	6.89e−09	2.25e−06
QADLITTL	97	56	acceptable	4.80e+05	1.74e−03	44	1.19e−09	3.56e−06
QAFIRO	32	27	acceptable	−1.59e+00	5.44e−04	36	2.13e−14	1.87e−09
QBANDM	472	305	acceptable	1.64e+04	1.28e−02	72	7.72e−09	1.86e−04
QBEACONF	262	173	acceptable	1.65e+05	1.12e−02	64	7.82e−11	9.24e−09
QBRANDY	249	220	acceptable	2.84e+04	9.81e−03	60	4.32e−10	4.76e−08
QE226	282	223	acceptable	2.13e+02	1.56e−02	68	2.13e−12	9.22e−08
QFFFFF80	854	524	acceptable	8.73e+05	1.22e−01	124	5.41e−09	1.39e−06
QGFRDXPN	1092	616	acceptable	1.01e+11	1.76e−02	92	4.58e−06	4.72e−06
QGROW15	645	300	acceptable	−1.02e+08	2.40e−02	68	3.73e−08	4.96e−01
QGROW22	946	440	acceptable	−1.50e+08	4.00e−02	80	1.66e−08	2.29e−01
QGROW7	301	140	acceptable	−4.28e+07	1.15e−02	68	1.13e−08	1.25e−01
QISRAEL	142	174	acceptable	2.53e+07	1.06e−02	72	1.57e−06	6.02e−03
QPCBLEND	83	74	acceptable	−7.84e−03	2.07e−03	52	3.16e−11	2.29e−07
QPCBOEI1	384	351	acceptable	1.15e+07	3.28e−02	120	1.44e−08	1.06e−04
QPCBOEI2	143	166	acceptable	8.17e+06	1.29e−02	152	2.55e−09	2.74e−04
QPTEST	2	2	acceptable	4.37e+00	2.68e−04	56	1.33e−15	2.25e−10
QSC205	203	205	acceptable	−5.81e−03	2.92e−03	44	2.99e−13	7.21e−07
QSCAGR25	500	471	acceptable	2.02e+08	8.51e−03	68	6.27e−10	2.52e−06
QSCAGR7	140	129	acceptable	2.69e+07	2.48e−03	64	7.38e−10	2.09e−06
QSCFXM1	457	330	acceptable	1.69e+07	1.88e−02	92	3.25e−08	3.02e−03
QSCFXM2	914	660	acceptable	2.78e+07	4.33e−02	108	6.69e−08	3.03e−04
QSCFXM3	1371	990	acceptable	3.08e+07	6.83e−02	116	6.75e−09	2.23e−05
QSCORPIO	358	388	acceptable	1.88e+03	5.79e−03	52	8.51e−11	1.02e−08
QSCRS8	1169	490	acceptable	9.05e+02	1.74e−02	68	2.90e−11	9.85e−07
QSCSD1	760	77	acceptable	8.67e+00	6.07e−03	32	1.56e−13	8.62e−11
QSCSD6	1350	147	acceptable	5.08e+01	1.48e−02	48	2.83e−13	5.40e−08
QSCSD8	2750	397	acceptable	9.41e+02	2.91e−02	44	3.11e−11	9.88e−07
QSCTAP1	480	300	acceptable	1.42e+03	9.32e−03	64	7.59e−12	9.43e−10
QSCTAP2	1880	1090	acceptable	1.74e+03	4.03e−02	52	8.17e−12	1.22e−07
QSCTAP3	2480	1480	acceptable	1.44e+03	6.18e−02	64	1.06e−12	8.99e−11
QSEBA	1028	515	acceptable	8.15e+07	3.77e−02	164	5.04e−10	1.86e−08
QSHARE1B	225	117	acceptable	7.20e+05	6.00e−03	92	4.51e−08	4.24e−07
QSHARE2B	79	96	acceptable	1.17e+04	2.83e−03	56	5.53e−11	1.26e−07
QSHIP04L	2118	402	acceptable	2.42e+06	2.16e−02	52	4.45e−10	2.16e−06
QSHIP04S	1458	402	acceptable	2.42e+06	1.57e−02	56	1.72e−10	5.53e−09
QSHIP08L	4283	778	acceptable	2.38e+06	5.61e−01	56	2.12e−09	7.15e−04
QSHIP08S	2387	778	acceptable	2.39e+06	8.75e−02	56	5.32e−10	7.61e−04

Continued on next page

Table 5: RipQP results in mono precision mode (double floating-point precision) on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
QSHIP12L	5427	1151	acceptable	3.02e+06	9.69e−01	64	6.67e−10	1.34e−04
QSHIP12S	2763	1151	acceptable	3.06e+06	9.88e−02	64	1.43e−09	7.41e−06
S268	5	5	acceptable	3.64e−10	3.95e−04	52	1.78e−14	2.18e−11
STADAT1	2001	3999	acceptable	−2.85e+07	4.04e−01	600	3.29e−08	7.21e−07
STADAT2	2001	3999	acceptable	−3.26e+01	1.70e−01	244	1.73e−11	7.19e−12
STADAT3	4001	7999	acceptable	−3.58e+01	4.34e−01	268	2.02e−10	6.38e−11
STCQP1	4097	2052	acceptable	1.55e+05	1.78e−01	24	5.08e−10	9.20e−06
STCQP2	4097	2052	acceptable	2.23e+04	4.17e−01	28	1.94e−11	2.79e−06
TAME	2	1	acceptable	0.00e+00	1.78e−04	16	7.27e−14	7.87e−13
VALUES	202	1	acceptable	−1.40e+00	6.22e−03	48	1.23e−16	1.36e−06
ZECEVIC2	2	2	acceptable	−4.12e+00	2.43e−04	28	2.44e−14	2.58e−10

Table 6: RipQP results in multi precision mode (single than double floating-point precision) on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
25FV47	1571	821	acceptable	5.50e+03	1.38e+01	57	5.69e−09	2.86e−10
ADLITTLE	97	56	acceptable	2.25e+05	2.49e−03	23	3.02e−07	2.31e−07
AFIRO	32	27	acceptable	−4.65e+02	8.04e−04	14	1.61e−08	2.64e−10
AGG	163	488	acceptable	−3.60e+07	1.22e−01	193	1.26e−08	1.29e−07
AGG2	302	516	acceptable	−2.02e+07	1.43e−01	98	5.98e−07	2.91e−07
AGG3	302	516	max_iter	NaN	6.71e−01	802	NaN	NaN
BANDM	472	305	acceptable	−1.59e+02	5.13e−01	62	1.63e−09	5.79e−10
BEACONFD	262	173	acceptable	3.36e+04	1.39e−02	53	8.28e−07	7.00e−09
BLEND	83	74	acceptable	−3.08e+01	2.55e−03	23	3.68e−11	5.57e−11
BNL1	1175	643	acceptable	1.98e+03	1.78e−01	206	2.25e−10	7.65e−10
BNL2	3489	2324	acceptable	1.81e+03	6.04e−01	118	4.58e−10	1.82e−10
BOEING1	384	351	acceptable	−3.35e+02	6.71e−02	162	7.29e−09	4.70e−11
BOEING2	143	166	acceptable	−3.15e+02	1.78e−02	110	1.82e−11	4.13e−12
BRANDY	249	220	acceptable	1.52e+03	1.23e−02	48	4.39e−09	2.25e−10
CRE-A	4067	3516	acceptable	2.36e+07	2.66e−01	89	1.01e−07	1.11e−08
CRE-B	72447	9648	acceptable	2.31e+07	3.07e+01	90	2.12e−08	1.22e−07
CRE-C	3678	3068	acceptable	2.53e+07	2.62e−01	110	5.19e−08	3.46e−07
CRE-D	69980	8926	acceptable	2.45e+07	2.60e+01	106	6.12e−08	1.11e−07
CYCLE	2857	1903	acceptable	−5.23e+00	5.65e−01	81	1.71e−09	5.97e−11
D2Q06C	5167	2171	acceptable	1.23e+05	1.51e+00	103	2.69e−07	1.24e−08
D6CUBE	6184	415	acceptable	3.15e+02	4.70e−01	30	2.01e−07	3.98e−09
DEGEN2	534	444	acceptable	−1.44e+03	4.22e−02	23	1.90e−11	3.84e−10
DEGEN3	1818	1503	acceptable	−9.87e+02	5.71e−01	30	6.18e−10	6.49e−10
DFL001	12230	6071	exception	∞	∞	0	∞	∞
E226	282	223	acceptable	−1.16e+01	1.95e−02	39	1.39e−10	6.91e−11
FFFFF800	854	524	acceptable	5.56e+05	2.13e−01	197	4.67e−07	7.44e−06
FIT1D	1026	24	acceptable	−9.15e+03	4.62e−02	40	1.90e−09	1.33e−10
FIT1P	1677	627	acceptable	9.15e+03	3.54e−02	21	5.55e−07	2.94e−05
FIT2D	10500	25	acceptable	−6.85e+04	6.08e−01	89	2.90e−09	2.74e−11
FIT2P	13525	3000	acceptable	6.85e+04	2.66e−01	37	3.60e−10	8.42e−09
GANGES	1681	1309	acceptable	−1.10e+05	2.06e−01	242	6.69e−09	1.24e−10
GFRD-PNC	1092	616	acceptable	6.90e+06	4.18e−02	81	1.82e−08	1.06e−09
GROW15	645	300	acceptable	−1.07e+08	1.07e−01	234	3.11e−07	3.48e−10
GROW22	946	440	acceptable	−1.61e+08	1.55e−01	238	4.16e−07	8.67e−12
GROW7	301	140	acceptable	−4.78e+07	3.81e−02	138	3.90e−07	2.04e−10
ISRAEL	142	174	acceptable	−8.97e+05	4.45e−02	70	5.64e−08	3.17e−09
KB2	41	43	acceptable	−1.75e+03	4.26e−03	70	5.54e−08	2.26e−07
KEN-07	3602	2426	acceptable	−6.80e+08	6.77e−02	61	9.02e−10	7.15e−09
KEN-11	21349	14694	acceptable	−6.97e+09	8.68e−01	93	9.11e−10	5.15e−08
KEN-13	42659	28632	acceptable	−1.03e+10	2.93e+00	113	9.35e−08	2.35e−10
KEN-18	154699	105127	acceptable	−5.22e+10	4.58e+01	165	2.19e−08	1.51e−10
LOTFI	308	153	acceptable	−2.53e+01	1.31e−02	84	3.73e−09	2.18e−14
MAROS-R7	9408	3136	acceptable	1.50e+06	2.48e+01	89	1.34e−10	1.87e−11
MODSZK1	1620	687	acceptable	3.21e+02	3.96e−02	99	5.83e−11	5.46e−13
OSA-07	23949	1118	acceptable	5.36e+05	1.02e+00	34	2.54e−07	3.70e−09
OSA-14	52460	2337	acceptable	1.11e+06	3.60e+00	34	1.06e−06	9.41e−08
OSA-30	100024	4350	acceptable	2.14e+06	8.91e+00	58	9.15e−08	3.18e−12
OSA-60	232966	10280	acceptable	4.04e+06	2.86e+01	54	8.96e−07	1.24e−09

Continued on next page

Table 6: RipQP results in multi precision mode (single than double floating-point precision) on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
PDS-02	7535	2953	acceptable	2.89e+10	5.66e−01	182	8.72e−08	1.61e−07
PDS-06	28655	9881	acceptable	2.78e+10	2.88e+01	231	2.98e−08	1.41e−07
PDS-10	48763	16558	acceptable	2.67e+10	1.78e+02	259	9.77e−09	1.36e−07
PDS-20	105728	33874	max_time	2.36e+10	1.21e+03	58	5.86e+00	1.60e−01
QAP12	8856	3192	acceptable	5.23e+02	6.96e+01	43	2.77e−12	1.29e−11
QAP15	22275	6330	acceptable	1.04e+03	5.64e+02	49	1.59e−11	1.11e−10
QAP8	1632	912	acceptable	2.03e+02	8.05e−01	15	7.54e−11	6.16e−09
SC105	103	105	acceptable	−5.22e+01	6.55e−03	86	1.79e−12	2.52e−12
SC205	203	205	max_iter	−2.70e+01	4.80e−02	800	2.09e−05	1.60e−02
SC50A	48	50	acceptable	−6.46e+01	1.52e−03	17	1.20e−08	1.66e−09
SC50B	48	50	acceptable	−7.00e+01	1.59e−03	41	7.46e−14	2.80e−12
SCAGR25	500	471	acceptable	−1.48e+07	1.08e−02	40	2.71e−09	2.11e−10
SCAGR7	140	129	acceptable	−2.33e+06	3.41e−03	44	1.43e−09	5.93e−12
SCFXM1	457	330	acceptable	1.84e+04	1.77e−02	49	1.60e−08	5.74e−10
SCFXM2	914	660	acceptable	3.67e+04	4.10e−02	62	1.08e−09	1.05e−10
SCFXM3	1371	990	acceptable	5.49e+04	5.78e−02	54	3.36e−09	1.33e−10
SCORPION	358	388	acceptable	1.88e+03	1.80e−02	134	2.32e−09	7.05e−08
SCRS8	1169	490	acceptable	9.04e+02	5.18e−02	125	9.43e−09	1.79e−10
SCSD1	760	77	acceptable	8.67e+00	6.00e−03	14	4.73e−11	8.01e−10
SCSD6	1350	147	acceptable	5.05e+01	1.20e−02	17	3.92e−10	1.06e−08
SCSD8	2750	397	acceptable	9.05e+02	2.21e−02	16	4.02e−09	4.19e−09
SCTAP1	480	300	acceptable	1.41e+03	1.04e−02	27	8.80e−11	9.86e−11
SCTAP2	1880	1090	acceptable	1.72e+03	3.70e−02	20	1.14e−08	9.53e−09
SCTAP3	2480	1480	acceptable	1.42e+03	5.86e−02	26	1.30e−10	9.26e−11
SEBA	1028	515	acceptable	1.57e+04	2.30e−02	72	8.86e−10	6.88e−10
SHARE1B	225	117	acceptable	−7.66e+04	1.18e−02	120	1.15e−04	1.10e−08
SHARE2B	79	96	acceptable	−4.16e+02	4.04e−03	35	1.11e−08	1.72e−08
SHIP04L	2118	402	acceptable	1.79e+06	3.92e−02	62	1.58e−08	9.43e−09
SHIP04S	1458	402	acceptable	1.80e+06	2.95e−02	66	3.07e−06	1.77e−08
SHIP08L	4283	778	acceptable	1.91e+06	1.76e−01	178	2.16e−10	4.05e−10
SHIP08S	2387	778	acceptable	1.92e+06	8.72e−02	158	2.73e−11	8.01e−10
SHIP12L	5427	1151	acceptable	1.47e+06	1.69e−01	159	2.77e−11	4.58e−10
SHIP12S	2763	1151	acceptable	1.49e+06	1.10e−01	192	1.71e−09	1.30e−09
STOCFOR1	111	117	acceptable	−4.11e+04	4.92e−03	69	9.83e−11	3.83e−11
STOCFOR2	2031	2157	acceptable	−3.90e+04	1.19e−01	100	2.50e−10	1.19e−11
STOCFOR3	15695	16675	acceptable	−4.00e+04	1.20e+00	134	2.19e−09	1.33e−10
TRUSS	8806	1000	acceptable	4.59e+05	1.97e−01	24	2.38e−07	2.05e−07
WOOD1P	2594	244	acceptable	1.44e+00	3.30e−01	25	1.20e−06	2.34e−08
WOODW	8405	1098	acceptable	1.30e+00	2.94e−01	38	1.82e−09	2.06e−10

Table 7: RipQP results in multi precision mode (single than double floating-point precision) on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|rb\|_\infty$	$\|rc\|_\infty$
AUG2D	20200	10000	acceptable	1.69e+06	2.43e−01	12	5.11e−10	1.99e−11
AUG2DC	20200	10000	acceptable	1.82e+06	2.42e−01	12	6.53e−10	2.28e−11
AUG2DCQP	20200	10000	acceptable	6.50e+06	5.41e−01	24	1.21e−09	4.07e−05
AUG2DQP	20200	10000	acceptable	6.24e+06	7.91e−01	20	2.52e−09	1.60e−08
AUG3D	3873	1000	acceptable	5.54e+02	2.58e−02	6	4.77e−07	1.19e−07
AUG3DC	3873	1000	acceptable	7.71e+02	2.51e−02	6	5.96e−07	1.12e−07
AUG3DCQP	3873	1000	acceptable	9.93e+02	7.49e−02	30	2.18e−14	3.84e−08
AUG3DQP	3873	1000	acceptable	6.75e+02	7.01e−02	26	2.22e−15	2.83e−13
BOYD1	93261	18	acceptable	−6.17e+07	2.66e+00	49	5.42e−02	1.62e−03
BOYD2	93263	186531	acceptable	2.13e+01	2.91e+01	442	1.28e−03	2.80e−09
CONT-050	2597	2401	acceptable	−4.56e+00	5.46e−01	54	7.24e−13	3.58e−11
CONT-100	10197	9801	acceptable	−4.64e+00	5.57e+00	74	3.89e−14	5.68e−13
CONT-101	10197	10098	acceptable	1.96e−01	6.10e+00	90	1.07e−14	3.39e−13
CONT-200	40397	39601	acceptable	−4.68e+00	1.23e+02	301	1.89e−12	3.48e−10
CONT-201	40397	40198	acceptable	1.92e−01	7.81e+01	126	8.88e−15	8.51e−13
CONT-300	90597	90298	acceptable	1.92e−01	3.34e+02	126	7.28e−13	4.16e−11
CVXQP1_L	10000	5000	acceptable	1.09e+08	1.14e+03	468	2.45e−10	9.72e−04
CVXQP1_M	1000	500	acceptable	1.09e+06	8.56e−01	138	1.23e−09	1.35e−03
CVXQP1_S	100	50	acceptable	1.16e+04	3.85e−03	33	1.83e−10	2.49e−08

Continued on next page

Table 7: RipQP results in multi precision mode (single than double floating-point precision) on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|rb\|_\infty$	$\|rc\|_\infty$
CVXQP2_L	10000	2500	acceptable	8.18e+07	5.43e+02	90	1.13e−08	2.74e−03
CVXQP2_M	1000	250	acceptable	8.20e+05	2.62e−01	37	3.13e−10	7.16e−04
CVXQP2_S	100	25	acceptable	8.12e+03	2.06e−03	14	2.91e−07	3.32e−05
CVXQP3_L	10000	7500	acceptable	1.16e+08	5.02e+02	135	2.26e−08	1.60e−04
CVXQP3_M	1000	750	acceptable	1.36e+06	1.71e+00	269	8.77e−09	1.36e−05
CVXQP3_S	100	75	acceptable	1.19e+04	4.12e−03	34	8.14e−11	1.65e−05
DPKLO1	133	77	acceptable	3.70e−01	2.80e−03	9	7.19e−12	3.53e−12
DUAL1	85	1	acceptable	3.50e−02	4.70e−03	17	5.25e−13	6.75e−07
DUAL2	96	1	acceptable	3.37e−02	4.61e−03	11	6.06e−11	9.73e−07
DUAL3	111	1	acceptable	1.36e−01	8.74e−03	20	3.80e−13	5.50e−08
DUAL4	75	1	acceptable	7.46e−01	3.27e−03	16	1.77e−12	4.60e−10
DUALC1	9	215	acceptable	6.16e+03	5.72e−03	18	2.28e−07	1.13e−04
DUALC2	7	229	acceptable	3.55e+03	4.38e−03	16	2.51e−07	2.61e−06
DUALC5	8	278	acceptable	4.27e+02	4.16e−03	12	7.60e−08	5.66e−07
DUALC8	8	503	acceptable	1.83e+04	7.38e−03	12	2.43e−07	1.61e−04
EXDATA	3000	3001	acceptable	−1.42e+02	2.49e+01	25	6.80e−13	1.68e−08
GENHS28	10	8	acceptable	9.27e−01	6.91e−04	5	9.63e−08	1.19e−07
GOULDQP2	699	349	acceptable	1.84e−04	2.23e−02	194	3.60e−13	7.20e−08
GOULDQP3	699	349	acceptable	2.06e+00	1.34e−02	56	3.83e−15	4.24e−12
HS118	15	17	acceptable	6.65e+02	1.06e−03	28	1.15e−12	6.82e−11
HS21	2	1	acceptable	−1.00e+02	3.53e−04	8	1.03e−06	1.19e−08
HS268	5	5	max_iter	NaN	2.99e−03	802	NaN	NaN
HS35	3	1	acceptable	1.11e−01	3.74e−04	13	1.97e−10	3.04e−10
HS51	5	3	acceptable	0.00e+00	3.04e−04	10	1.19e−13	1.67e−13
HS52	5	3	acceptable	5.33e+00	2.80e−04	6	4.38e−09	2.56e−07
HS53	5	3	acceptable	4.09e+00	3.00e−04	7	7.11e−08	4.37e−07
HS76	4	3	acceptable	−4.68e+00	3.82e−04	12	1.80e−10	5.40e−10
HUES-MOD	10000	2	acceptable	3.48e+07	4.78e−02	18	1.23e−11	1.64e−07
HUESTIS	10000	2	acceptable	3.48e+11	4.97e−02	16	8.67e−07	4.57e−04
KSIP	20	1001	acceptable	5.76e−01	9.85e−02	37	2.63e−12	2.37e−08
LASER	1002	1000	acceptable	2.41e+06	1.25e−01	522	1.21e−09	4.41e−07
LISWET1	10002	10000	acceptable	3.61e+01	3.71e−01	123	4.44e−16	3.49e−10
LISWET10	10002	10000	acceptable	4.95e+01	7.65e−01	310	5.49e−16	5.56e−10
LISWET11	10002	10000	acceptable	4.95e+01	8.41e−01	338	2.03e−14	5.46e−10
LISWET12	10002	10000	acceptable	1.74e+03	1.35e+00	555	9.03e−14	3.50e−07
LISWET2	10002	10000	acceptable	2.50e+01	3.85e−01	126	1.95e−13	1.24e−06
LISWET3	10002	10000	acceptable	2.50e+01	2.39e−01	62	5.56e−10	5.13e−07
LISWET4	10002	10000	acceptable	2.50e+01	3.40e−01	106	2.86e−13	1.83e−10
LISWET5	10002	10000	acceptable	2.50e+01	2.61e−01	70	7.99e−11	6.94e−08
LISWET6	10002	10000	acceptable	2.50e+01	3.03e−01	90	8.79e−11	2.02e−06
LISWET7	10002	10000	acceptable	4.99e+02	3.72e−01	122	3.33e−16	1.54e−09
LISWET8	10002	10000	acceptable	7.14e+02	9.18e−01	362	6.44e−15	1.63e−06
LISWET9	10002	10000	acceptable	1.96e+03	1.23e+00	499	3.12e−14	1.19e−07
LOTSCHD	12	7	acceptable	2.40e+03	7.30e−04	13	2.51e−08	3.23e−06
MOSARQP1	2500	700	acceptable	−9.53e+02	2.80e−02	15	9.66e−09	2.50e−06
MOSARQP2	900	600	acceptable	−1.60e+03	3.03e−02	25	8.98e−12	5.26e−06
POWELL20	10000	10000	acceptable	5.21e+10	3.71e−01	50	1.67e−10	5.85e−08
PRIMAL1	325	85	acceptable	−3.50e−02	1.44e−02	21	5.95e−13	2.57e−08
PRIMAL2	649	96	acceptable	−3.37e−02	1.91e−02	17	2.10e−12	1.31e−09
PRIMAL3	745	111	acceptable	−1.36e−01	6.07e−02	22	4.04e−13	1.45e−08
PRIMAL4	1489	75	acceptable	−7.46e−01	4.30e−02	22	3.40e−13	3.20e−11
PRIMALC1	230	9	acceptable	−6.16e+03	4.79e−03	42	4.74e−09	5.47e−09
PRIMALC2	231	7	acceptable	−3.55e+03	3.46e−03	34	1.58e−09	2.69e−07
PRIMALC5	287	8	acceptable	−4.27e+02	4.45e−03	33	2.10e−09	7.02e−09
PRIMALC8	520	8	acceptable	−1.83e+04	5.99e−03	19	1.45e−07	1.77e−09
Q25FV47	1571	820	acceptable	1.37e+07	9.21e−01	70	2.46e−07	1.89e−08
QADLITTL	97	56	acceptable	4.80e+05	2.41e−03	18	9.76e−07	1.92e−06
QAFIRO	32	27	acceptable	−1.59e+00	7.26e−04	14	5.46e−09	1.83e−10
QBANDM	472	305	acceptable	1.64e+04	1.43e−02	43	3.42e−10	3.18e−07
QBEACONF	262	173	acceptable	1.65e+05	9.99e−03	33	3.33e−06	3.19e−05
QBRANDY	249	220	acceptable	2.84e+04	1.19e−02	33	6.64e−10	4.95e−07
QE226	282	223	acceptable	2.13e+02	1.72e−02	30	2.18e−08	1.17e−07
QFFFFF80	854	524	acceptable	8.73e+05	1.45e−01	101	1.39e−07	3.00e−06
QGFRDXPN	1092	616	acceptable	1.01e+11	1.87e−02	85	1.05e−05	1.65e−03

Continued on next page

Table 7: RipQP results in multi precision mode (single than double floating-point precision) on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|rb\|_\infty$	$\|rc\|_\infty$
QGROW15	645	300	acceptable	−1.02e+08	9.10e−02	198	5.91e−06	5.76e−01
QGROW22	946	440	acceptable	−1.50e+08	1.49e−01	230	8.14e−06	2.38e−01
QGROW7	301	140	acceptable	−4.28e+07	3.97e−02	170	1.56e−05	6.75e−02
QISRAEL	142	174	acceptable	2.53e+07	1.15e−02	36	1.79e−06	7.32e−03
QPCBLEND	83	74	acceptable	−7.84e−03	4.85e−03	88	1.27e−11	1.64e−06
QPCBOEI1	384	351	acceptable	1.15e+07	4.04e−02	91	3.64e−07	8.17e−03
QPCBOEI2	143	166	acceptable	8.17e+06	1.44e−02	91	1.26e−09	2.96e−04
QPTEST	2	2	acceptable	4.37e+00	5.80e−04	20	1.64e−10	3.76e−08
QSC205	203	205	acceptable	−5.81e−03	3.43e−03	29	2.77e−13	1.12e−11
QSCAGR25	500	471	acceptable	2.02e+08	8.51e−03	28	1.36e−07	4.60e−05
QSCAGR7	140	129	acceptable	2.69e+07	2.85e−03	34	3.31e−07	8.89e−08
QSCFXM1	457	330	acceptable	1.69e+07	3.20e−02	98	9.31e−08	2.47e−03
QSCFXM2	914	660	acceptable	2.78e+07	6.05e−02	83	3.49e−06	9.95e−05
QSCFXM3	1371	990	acceptable	3.08e+07	1.01e−01	86	4.63e−07	1.11e−03
QSCORPIO	358	388	acceptable	1.88e+03	1.40e−02	99	3.83e−10	1.78e−08
QSCRS8	1169	490	acceptable	9.05e+02	4.58e−02	111	5.19e−09	6.92e−07
QSCSD1	760	77	acceptable	8.67e+00	7.08e−03	14	9.28e−11	6.14e−10
QSCSD6	1350	147	acceptable	5.08e+01	1.70e−02	19	4.24e−10	6.57e−08
QSCSD8	2750	397	acceptable	9.41e+02	2.85e−02	16	5.83e−09	8.66e−06
QSCTAP1	480	300	acceptable	1.42e+03	1.10e−02	27	8.76e−11	2.52e−09
QSCTAP2	1880	1090	acceptable	1.74e+03	4.44e−02	19	1.05e−08	4.15e−08
QSCTAP3	2480	1480	acceptable	1.44e+03	6.31e−02	21	2.39e−08	3.02e−08
QSEBA	1028	515	acceptable	8.15e+07	5.01e−02	156	3.67e−09	9.96e−08
QSHARE1B	225	117	acceptable	7.20e+05	1.38e−02	130	1.10e−06	1.20e−07
QSHARE2B	79	96	acceptable	1.17e+04	4.59e−03	49	7.23e−08	3.04e−07
QSHIP04L	2118	402	acceptable	2.42e+06	3.20e−02	46	2.31e−09	4.77e−06
QSHIP04S	1458	402	acceptable	2.42e+06	3.28e−02	78	1.27e−10	7.17e−07
QSHIP08L	4283	778	acceptable	2.38e+06	9.00e−01	51	5.19e−09	5.83e−05
QSHIP08S	2387	778	acceptable	2.39e+06	1.55e−01	63	4.20e−10	3.11e−04
QSHIP12L	5427	1151	acceptable	3.02e+06	2.70e+00	123	1.81e−09	8.46e−05
QSHIP12S	2763	1151	acceptable	3.06e+06	2.57e−01	120	2.24e−09	7.89e−05
S268	5	5	max_iter	NaN	3.43e−03	802	NaN	NaN
STADAT1	2001	3999	acceptable	−2.85e+07	2.44e−01	258	3.85e−09	8.51e−08
STADAT2	2001	3999	acceptable	−3.26e+01	1.66e−01	138	1.27e−08	1.33e−10
STADAT3	4001	7999	acceptable	−3.58e+01	5.09e−01	222	1.22e−10	2.56e−11
STCQP1	4097	2052	acceptable	1.55e+05	1.71e−01	12	1.66e−08	3.03e−05
STCQP2	4097	2052	max_iter	6.70e+04	1.08e+01	801	2.00e+00	4.40e+09
TAME	2	1	acceptable	0.00e+00	5.78e−04	6	3.82e−08	7.72e−10
VALUES	202	1	acceptable	−1.40e+00	7.39e−03	25	4.81e−13	7.49e−09
ZECEVIC2	2	2	acceptable	−4.12e+00	4.79e−04	13	5.43e−11	3.12e−10

B Gurobi detailed results

Table 8: Gurobi results without presolve and crossover on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
25FV47	1571	821	acceptable	5.50e+03	2.43e−01	22	3.00e−10	1.47e−13
ADLITTLE	97	56	acceptable	2.25e+05	2.35e−03	13	3.18e−12	2.33e−12
AFIRO	32	27	acceptable	−4.65e+02	6.61e−04	7	3.58e−11	3.22e−15
AGG	163	488	acceptable	−3.60e+07	3.47e−02	17	3.20e−09	5.00e−12
AGG2	302	516	acceptable	−2.02e+07	5.11e−02	17	2.40e−10	1.82e−12
AGG3	302	516	acceptable	1.03e+07	4.80e−02	16	5.80e−10	2.73e−12
BANDM	472	305	acceptable	−1.59e+02	1.61e−02	14	1.93e−09	5.68e−14
BEACONFD	262	173	acceptable	3.36e+04	1.01e−02	10	9.38e−11	1.01e−11
BLEND	83	74	acceptable	−3.08e+01	6.93e−03	9	5.62e−11	1.02e−14
BNL1	1175	643	acceptable	1.98e+03	1.37e−01	60	8.47e−09	1.28e−12
BNL2	3489	2324	acceptable	1.81e+03	3.85e−01	24	3.28e−09	1.35e−13
BOEING1	384	351	acceptable	−3.35e+02	2.98e−02	21	1.59e−09	3.13e−13
BOEING2	143	166	acceptable	−3.15e+02	7.93e−03	14	1.26e−10	2.84e−14
BRANDY	249	220	acceptable	1.52e+03	1.51e−02	19	2.49e−07	1.70e−12
CRE-A	4067	3516	acceptable	2.36e+07	2.27e−01	25	1.13e−08	9.31e−10

Continued on next page

Table 8: Gurobi results without presolve and crossover on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
CRE-B	72447	9648	acceptable	2.31e+07	2.97e+00	31	1.79e−08	2.05e−09
CRE-C	3678	3068	acceptable	2.53e+07	2.25e−01	27	2.33e−09	7.42e−10
CRE-D	69980	8926	acceptable	2.45e+07	2.98e+00	36	7.64e−10	3.20e−09
CYCLE	2857	1903	acceptable	−5.23e+00	2.75e−01	19	2.55e−11	1.98e−13
D2Q06C	5167	2171	acceptable	1.23e+05	3.66e−01	30	4.80e−08	9.09e−13
D6CUBE	6184	415	acceptable	3.15e+02	1.76e−01	19	1.70e−08	1.49e−14
DEGEN2	534	444	acceptable	−1.44e+03	3.61e−02	11	6.44e−12	2.27e−13
DEGEN3	1818	1503	acceptable	−9.87e+02	2.34e−01	19	3.87e−08	9.95e−14
DFL001	12230	6071	acceptable	1.13e+07	2.09e+00	38	3.15e−06	1.49e−08
E226	282	223	acceptable	−1.16e+01	1.37e−02	16	1.29e−10	4.55e−13
FFFFF800	854	524	acceptable	5.56e+05	2.31e−01	79	4.76e−09	6.27e−11
FIT1D	1026	24	acceptable	−9.15e+03	2.40e−02	18	1.78e−06	2.91e−12
FIT1P	1677	627	acceptable	9.15e+03	4.43e−02	15	4.21e−11	1.02e−08
FIT2D	10500	25	acceptable	−6.85e+04	3.40e−01	30	2.07e−09	6.91e−12
FIT2P	13525	3000	acceptable	6.85e+04	2.82e−01	18	4.41e−13	1.24e−09
GANGES	1681	1309	acceptable	−1.10e+05	9.82e−02	18	1.25e−08	2.56e−13
GFRD-PNC	1092	616	acceptable	6.90e+06	2.06e−02	16	8.26e−08	2.63e−11
GROW15	645	300	acceptable	−1.07e+08	1.89e−02	12	6.76e−08	3.55e−14
GROW22	946	440	acceptable	−1.61e+08	3.17e−02	14	3.52e−08	2.27e−13
GROW7	301	140	acceptable	−4.78e+07	9.19e−03	13	3.57e−10	9.95e−14
ISRAEL	142	174	acceptable	−8.97e+05	2.10e−02	14	5.12e−09	4.92e−13
KB2	41	43	acceptable	−1.75e+03	1.60e−03	14	9.15e−08	1.60e−13
KEN-07	3602	2426	acceptable	−6.80e+08	9.62e−02	14	7.50e−12	9.46e−11
KEN-11	21349	14694	acceptable	−6.97e+09	8.96e−01	19	2.13e−11	1.36e−10
KEN-13	42659	28632	acceptable	−1.03e+10	1.79e+00	22	5.75e−09	9.46e−11
KEN-18	154699	105127	acceptable	−5.22e+10	1.25e+01	30	1.03e−06	5.82e−11
LOTFI	308	153	acceptable	−2.53e+01	8.16e−03	15	2.23e−08	3.53e−12
MAROS-R7	9408	3136	acceptable	1.50e+06	1.23e+00	13	1.00e−07	3.55e−15
MODSZK1	1620	687	acceptable	3.21e+02	6.48e−02	23	9.17e−10	4.17e−12
OSA-07	23949	1118	acceptable	5.36e+05	3.51e−01	17	1.01e−09	2.73e−12
OSA-14	52460	2337	acceptable	1.11e+06	1.32e+00	23	2.42e−09	2.73e−12
OSA-30	100024	4350	acceptable	2.14e+06	2.11e+00	21	3.99e−09	2.73e−12
OSA-60	232966	10280	acceptable	4.04e+06	5.49e+00	20	1.58e−08	5.00e−12
PDS-02	7535	2953	acceptable	2.89e+10	5.20e−01	45	1.46e−10	1.11e−09
PDS-06	28655	9881	acceptable	2.78e+10	2.31e+00	55	9.90e−10	7.03e−09
PDS-10	48763	16558	acceptable	2.67e+10	6.17e+00	66	1.89e−10	5.50e−09
PDS-20	105728	33874	acceptable	2.38e+10	3.16e+01	118	1.23e−08	1.27e−09
QAP12	8856	3192	acceptable	5.23e+02	2.00e+00	16	4.30e−10	6.04e−13
QAP15	22275	6330	acceptable	1.04e+03	8.04e+00	18	1.30e−09	1.01e−12
QAP8	1632	912	acceptable	2.04e+02	2.06e−01	9	7.40e−13	2.07e−14
SC105	103	105	acceptable	−5.22e+01	2.63e−03	9	2.35e−10	1.11e−15
SC205	203	205	acceptable	−5.22e+01	4.38e−03	10	4.72e−11	3.39e−14
SC50A	48	50	acceptable	−6.46e+01	1.39e−03	8	7.88e−11	2.44e−15
SC50B	48	50	acceptable	−7.00e+01	1.12e−03	7	2.61e−10	3.08e−15
SCAGR25	500	471	acceptable	−1.48e+07	1.61e−02	17	3.54e−07	2.27e−12
SCAGR7	140	129	acceptable	−2.33e+06	4.71e−03	16	9.80e−08	2.27e−12
SCFXM1	457	330	acceptable	1.84e+04	1.91e−02	17	2.73e−08	7.94e−14
SCFXM2	914	660	acceptable	3.67e+04	4.00e−02	18	1.88e−08	1.99e−13
SCFXM3	1371	990	acceptable	5.49e+04	5.98e−02	18	1.64e−09	1.42e−13
SCORPION	358	388	acceptable	1.88e+03	1.18e−02	15	1.98e−10	2.84e−13
SCRS8	1169	490	acceptable	9.04e+02	3.09e−02	16	3.02e−11	4.55e−12
SCSD1	760	77	acceptable	8.67e+00	4.22e−03	7	5.48e−14	4.44e−15
SCSD6	1350	147	acceptable	5.05e+01	8.48e−03	9	4.77e−12	5.33e−15
SCSD8	2750	397	acceptable	9.05e+02	2.05e−02	11	9.38e−14	1.42e−14
SCTAP1	480	300	acceptable	1.41e+03	1.35e−02	18	4.53e−09	6.82e−14
SCTAP2	1880	1090	acceptable	1.72e+03	3.65e−02	10	2.79e−11	3.55e−14
SCTAP3	2480	1480	acceptable	1.42e+03	5.25e−02	11	7.48e−10	3.91e−14
SEBA	1028	515	acceptable	1.57e+04	2.33e−02	13	6.38e−12	2.18e−12
SHARE1B	225	117	acceptable	−7.66e+04	6.55e−03	20	1.51e−08	7.77e−13
SHARE2B	79	96	acceptable	−4.16e+02	2.96e−03	14	7.38e−09	5.68e−14
SHIP04L	2118	402	acceptable	1.79e+06	2.50e−02	16	2.72e−10	3.64e−12
SHIP04S	1458	402	acceptable	1.80e+06	1.98e−02	16	1.06e−11	4.55e−12
SHIP08L	4283	778	acceptable	1.91e+06	7.58e−02	24	5.62e−09	1.18e−11
SHIP08S	2387	778	acceptable	1.92e+06	3.75e−02	17	5.11e−10	2.64e−11
SHIP12L	5427	1151	acceptable	1.47e+06	8.66e−02	22	1.87e−10	3.64e−12

Continued on next page

Table 8: Gurobi results without presolve and crossover on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
SHIP12S	2763	1151	acceptable	1.49e+06	5.73e−02	20	2.00e−11	7.28e−12
STOCFOR1	111	117	acceptable	−4.11e+04	3.30e−03	10	9.17e−11	5.12e−13
STOCFOR2	2031	2157	acceptable	−3.90e+04	1.14e−01	20	1.12e−08	7.39e−13
STOCFOR3	15695	16675	acceptable	−4.00e+04	1.68e+00	33	4.11e−08	2.27e−12
TRUSS	8806	1000	acceptable	4.59e+05	3.22e−01	27	9.07e−11	3.73e−11
WOOD1P	2594	244	acceptable	1.44e+00	1.28e−01	13	7.43e−07	2.40e−13
WOODW	8405	1098	acceptable	1.30e+00	2.94e−01	26	6.37e−09	3.55e−15

Table 9: Gurobi results without presolve and crossover on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
AUG2D	20200	10000	acceptable	1.69e+06	3.69e−01	5	7.49e−12	3.58e−12
AUG2DC	20200	10000	acceptable	1.82e+06	3.00e−01	5	7.62e−12	4.32e−12
AUG2DCQP	20200	10000	acceptable	6.50e+06	7.72e−01	17	1.47e−14	2.18e−11
AUG2DQP	20200	10000	acceptable	6.24e+06	1.16e+00	20	2.86e−14	5.68e−10
AUG3D	3873	1000	acceptable	5.54e+02	8.24e−02	5	1.55e−12	9.45e−13
AUG3DC	3873	1000	acceptable	7.71e+02	5.83e−02	5	1.56e−12	1.09e−12
AUG3DCQP	3873	1000	acceptable	9.93e+02	1.27e−01	19	1.55e−15	8.89e−11
AUG3DQP	3873	1000	acceptable	6.75e+02	1.36e−01	22	2.66e−15	2.59e−09
BOYD1	93261	18	acceptable	−6.17e+07	1.81e+00	30	5.62e−02	3.78e−10
BOYD2	93263	186531	acceptable	2.13e+01	8.26e+01	75	1.48e−02	4.66e−10
CONT-050	2597	2401	acceptable	−4.56e+00	2.25e−01	15	2.27e−15	1.15e−13
CONT-100	10197	9801	acceptable	−4.64e+00	6.58e−01	16	2.65e−15	6.63e−14
CONT-101	10197	10098	acceptable	1.96e−01	6.58e−01	15	1.32e−11	1.15e−10
CONT-200	40397	39601	acceptable	−4.68e+00	2.68e+00	16	5.59e−15	3.60e−14
CONT-201	40397	40198	acceptable	1.92e−01	2.69e+00	16	3.83e−11	1.14e−11
CONT-300	90597	90298	acceptable	1.92e−01	6.74e+00	18	5.17e−10	2.20e−11
CVXQP1_L	10000	5000	exception	1.09e+08	9.39e+01	105	1.96e−09	6.65e+06
CVXQP1_M	1000	500	acceptable	1.09e+06	5.27e−01	25	1.24e−08	1.14e−05
CVXQP1_S	100	50	acceptable	1.16e+04	1.06e−02	11	1.39e−11	9.28e−10
CVXQP2_L	10000	2500	acceptable	8.18e+07	1.18e+01	21	7.88e−07	5.90e−03
CVXQP2_M	1000	250	acceptable	8.20e+05	3.29e−01	13	1.24e−09	2.03e−07
CVXQP2_S	100	25	acceptable	8.12e+03	7.87e−03	11	1.09e−10	3.85e−09
CVXQP3_L	10000	7500	exception	1.16e+08	1.53e+02	122	1.54e−07	1.67e+07
CVXQP3_M	1000	750	acceptable	1.36e+06	1.43e+00	51	2.33e−08	1.21e−05
CVXQP3_S	100	75	acceptable	1.19e+04	1.69e−02	12	4.28e−10	6.85e−08
DPKLO1	133	77	acceptable	3.70e−01	3.35e−03	5	1.68e−11	6.99e−13
DUAL1	85	1	acceptable	3.50e−02	1.75e−02	17	5.36e−12	3.21e−08
DUAL2	96	1	acceptable	3.37e−02	2.23e−02	16	9.55e−15	1.26e−10
DUAL3	111	1	acceptable	1.36e−01	3.06e−02	17	5.27e−12	2.71e−09
DUAL4	75	1	acceptable	7.46e−01	1.25e−02	16	2.79e−13	7.40e−09
DUALC1	9	215	acceptable	6.16e+03	6.01e−03	11	6.82e−13	5.75e−07
DUALC2	7	229	acceptable	3.55e+03	5.01e−03	11	6.82e−13	2.67e−07
DUALC5	8	278	acceptable	4.27e+02	2.54e−02	33	3.41e−13	5.94e−02
DUALC8	8	503	acceptable	1.83e+04	1.23e−02	14	2.96e−12	1.46e−09
EXDATA	3000	3001	acceptable	−1.42e+02	1.06e+01	13	2.19e−08	1.98e−08
GENHS28	10	8	acceptable	9.27e−01	1.59e−03	5	4.53e−12	1.40e−13
GOULDQP2	699	349	acceptable	1.84e−04	2.21e−02	15	1.61e−14	2.86e−10
GOULDQP3	699	349	acceptable	2.06e+00	3.01e−02	12	4.56e−14	3.50e−12
HS118	15	17	acceptable	6.65e+02	8.24e−04	13	6.22e−15	6.29e−16
HS21	2	1	acceptable	−1.00e+02	2.79e−04	9	0.00e+00	1.13e−14
HS268	5	5	acceptable	6.71e−09	9.69e−04	14	3.01e−13	2.84e−10
HS35	3	1	acceptable	1.11e−01	7.03e−04	10	1.08e−15	1.62e−15
HS51	5	3	acceptable	8.88e−16	7.10e−04	5	4.00e−12	2.28e−13
HS52	5	3	acceptable	5.33e+00	6.80e−04	5	6.06e−12	4.56e−13
HS53	5	3	acceptable	4.09e+00	7.28e−04	9	1.22e−14	1.75e−13
HS76	4	3	acceptable	−4.68e+00	6.99e−04	9	7.67e−16	1.49e−15
HUES-MOD	10000	2	acceptable	3.48e+07	5.72e−02	22	1.05e−11	8.15e−10
HUESTIS	10000	2	acceptable	3.48e+11	1.12e−01	34	1.36e−12	6.43e+08
KSIP	20	1001	acceptable	5.76e−01	6.34e−02	18	7.05e−16	9.85e−11
LASER	1002	1000	acceptable	2.41e+06	7.14e−02	15	2.21e−11	4.38e−10
LISWET1	10002	10000	acceptable	3.61e+01	4.01e−01	23	4.47e−14	2.23e−10
LISWET10	10002	10000	acceptable	4.95e+01	7.02e−01	42	1.58e−15	2.12e−10

Continued on next page

Table 9: Gurobi results without presolve and crossover on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
LISWET11	10002	10000	acceptable	4.95e+01	5.05e−01	29	7.99e−16	5.21e−11
LISWET12	10002	10000	exception	2.72e+01	2.31e+00	89	1.16e−06	1.40e−10
LISWET2	10002	10000	acceptable	2.50e+01	4.26e−01	15	1.09e−12	3.42e−13
LISWET3	10002	10000	acceptable	2.50e+01	4.33e−01	17	4.40e−16	3.11e−15
LISWET4	10002	10000	acceptable	2.50e+01	3.16e−01	18	1.47e−15	6.21e−15
LISWET5	10002	10000	acceptable	2.50e+01	2.74e−01	15	2.68e−15	2.44e−15
LISWET6	10002	10000	acceptable	2.50e+01	2.68e−01	15	6.50e−15	6.11e−15
LISWET7	10002	10000	exception	2.51e+01	1.26e+00	49	8.84e−08	2.66e−11
LISWET8	10002	10000	exception	2.57e+01	1.66e+00	67	3.24e−07	5.29e−11
LISWET9	10002	10000	exception	2.84e+01	1.66e+00	85	1.18e−06	1.41e−10
LOTSCHD	12	7	acceptable	2.40e+03	7.55e−04	12	1.59e−10	1.03e−12
MOSARQP1	2500	700	acceptable	−9.53e+02	5.29e−02	15	1.33e−15	1.67e−14
MOSARQP2	900	600	acceptable	−1.60e+03	4.20e−02	15	1.33e−15	5.33e−15
POWELL20	10000	10000	acceptable	5.21e+10	1.15e+00	65	2.73e−11	1.78e−09
PRIMAL1	325	85	acceptable	−3.50e−02	6.09e−02	19	3.80e−13	9.37e−17
PRIMAL2	649	96	acceptable	−3.37e−02	5.02e−02	18	4.15e−13	1.04e−16
PRIMAL3	745	111	acceptable	−1.36e−01	1.46e−01	19	5.94e−12	3.65e−16
PRIMAL4	1489	75	acceptable	−7.46e−01	8.46e−02	17	3.09e−11	2.58e−16
PRIMALC1	230	9	acceptable	−6.16e+03	3.80e−03	12	1.86e−09	6.82e−13
PRIMALC2	231	7	acceptable	−3.55e+03	3.41e−03	14	1.54e−11	1.36e−12
PRIMALC5	287	8	acceptable	−4.27e+02	4.12e−03	12	1.72e−12	3.41e−13
PRIMALC8	520	8	acceptable	−1.83e+04	6.63e−03	12	5.27e−11	6.82e−13
Q25FV47	1571	820	exception	1.37e+07	8.92e−01	36	6.44e−08	1.43e+06
QADLITTL	97	56	acceptable	4.80e+05	3.62e−03	12	2.21e−11	2.11e−11
QAFIRO	32	27	acceptable	−1.59e+00	1.55e−03	16	3.02e−14	4.04e−11
QBANDM	472	305	acceptable	1.64e+04	2.74e−02	23	1.63e−09	6.01e−10
QBEACONF	262	173	acceptable	1.65e+05	1.69e−02	17	2.19e−10	8.73e−11
QBRANDY	249	220	acceptable	2.84e+04	3.00e−02	17	9.92e−09	2.33e−10
QE226	282	223	acceptable	2.13e+02	2.88e−02	20	3.64e−12	1.86e−09
QFFFFF80	854	524	acceptable	8.73e+05	3.51e−01	51	9.46e−11	1.17e+07
QGFRDXPN	1092	616	acceptable	1.01e+11	6.27e−02	31	3.39e−09	7.92e−06
QGROW15	645	300	acceptable	−1.02e+08	4.10e−02	19	2.91e−10	3.89e−11
QGROW22	946	440	acceptable	−1.50e+08	5.82e−02	21	5.53e−10	4.84e−10
QGROW7	301	140	acceptable	−4.28e+07	1.64e−02	18	2.62e−10	4.78e−12
QISRAEL	142	174	acceptable	2.53e+07	4.00e−02	19	8.73e−10	2.71e−11
QPCBLEND	83	74	acceptable	−7.84e−03	4.98e−03	25	1.63e−17	2.49e−09
QPCBOEI1	384	351	acceptable	1.15e+07	2.91e−02	21	2.91e−11	5.51e−10
QPCBOEI2	143	166	acceptable	8.17e+06	1.28e−02	25	7.28e−12	1.49e−08
QPTEST	2	2	acceptable	4.37e+00	7.61e−04	11	1.36e−14	8.37e−14
QSC205	203	205	acceptable	−5.81e−03	8.45e−03	21	4.55e−13	8.00e−11
QSCAGR25	500	471	acceptable	2.02e+08	2.20e−02	20	3.61e−10	5.82e−09
QSCAGR7	140	129	acceptable	2.69e+07	6.19e−03	19	1.46e−11	9.46e−11
QSCFXM1	457	330	acceptable	1.69e+07	3.67e−02	23	3.58e−09	1.01e−09
QSCFXM2	914	660	acceptable	2.78e+07	8.93e−02	29	9.99e−09	8.43e−10
QSCFXM3	1371	990	acceptable	3.08e+07	1.37e−01	30	4.03e−10	8.96e−10
QSCORPIO	358	388	acceptable	1.88e+03	1.43e−02	16	1.21e−13	2.91e−11
QSCRS8	1169	490	acceptable	9.05e+02	4.24e−02	23	2.91e−11	1.86e−09
QSCSD1	760	77	acceptable	8.67e+00	1.38e−02	16	2.05e−15	1.57e−09
QSCSD6	1350	147	acceptable	5.08e+01	3.19e−02	19	7.55e−15	4.95e−09
QSCSD8	2750	397	acceptable	9.41e+02	6.99e−02	16	2.24e−11	2.26e−09
QSCTAP1	480	300	acceptable	1.42e+03	1.79e−02	20	1.43e−10	2.79e−10
QSCTAP2	1880	1090	acceptable	1.74e+03	7.32e−02	18	6.93e−13	3.03e−10
QSCTAP3	2480	1480	acceptable	1.44e+03	1.20e−01	21	6.46e−12	1.41e−09
QSEBA	1028	515	acceptable	8.15e+07	1.61e−01	41	1.06e−12	6.14e−09
QSHARE1B	225	117	acceptable	7.20e+05	9.01e−03	22	1.15e−09	3.17e−09
QSHARE2B	79	96	acceptable	1.17e+04	4.97e−03	19	1.16e−09	2.63e−09
QSHIP04L	2118	402	acceptable	2.42e+06	3.10e−02	17	1.03e−12	4.38e−11
QSHIP04S	1458	402	acceptable	2.42e+06	2.54e−02	18	7.93e−13	4.64e−11
QSHIP08L	4283	778	acceptable	2.38e+06	3.88e−01	18	9.30e−07	3.91e−05
QSHIP08S	2387	778	acceptable	2.39e+06	2.13e−01	18	1.60e−09	1.42e−07
QSHIP12L	5427	1151	acceptable	3.02e+06	5.97e−01	25	5.01e−09	2.28e−07
QSHIP12S	2763	1151	acceptable	3.06e+06	2.36e−01	22	8.72e−10	3.35e−08
S268	5	5	acceptable	6.71e−09	1.06e−03	14	3.01e−13	2.84e−10
STADAT1	2001	3999	exception	−2.87e+07	1.62e+00	278	8.28e−02	6.96e−06
STADAT2	2001	3999	acceptable	−3.26e+01	1.35e−01	17	2.73e−11	6.54e−14

Continued on next page

Table 9: Gurobi results without presolve and crossover on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
STADAT3	4001	7999	acceptable	−3.58e+01	2.05e−01	16	1.97e−09	5.36e−14
STCQP1	4097	2052	acceptable	1.55e+05	6.84e−01	13	6.11e−09	8.88e−07
STCQP2	4097	2052	acceptable	2.23e+04	7.95e−01	13	9.64e−10	4.89e−09
TAME	2	1	acceptable	0.00e+00	1.19e−03	7	7.99e−15	9.97e−13
VALUES	202	1	exception	∞	∞	0	∞	∞
ZECEVIC2	2	2	acceptable	−4.12e+00	5.23e−04	9	1.11e−15	2.37e−17

C CPLEX detailed results

Table 10: CPLEX results without presolve and crossover on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
25FV47	1571	821	acceptable	5.50e+03	1.14e−01	24	4.25e−13	5.60e−07
ADLITTLE	97	56	acceptable	2.25e+05	2.50e−03	12	1.68e−13	8.74e−07
AFIRO	32	27	acceptable	−4.65e+02	8.89e−04	8	2.16e−14	2.95e−10
AGG	163	488	acceptable	−3.60e+07	1.06e−02	20	3.50e−10	1.24e−08
AGG2	302	516	acceptable	−2.02e+07	2.10e−02	21	1.71e−10	1.47e−10
AGG3	302	516	acceptable	1.03e+07	2.20e−02	22	1.67e−10	7.48e−12
BANDM	472	305	acceptable	−1.59e+02	1.37e−02	17	1.94e−13	4.33e−12
BEACONFD	262	173	acceptable	3.36e+04	7.62e−03	12	7.13e−11	8.74e−10
BLEND	83	74	acceptable	−3.08e+01	2.65e−03	10	1.75e−14	4.09e−11
BNL1	1175	643	acceptable	1.98e+03	7.22e−02	41	2.01e−13	1.82e−09
BNL2	3489	2324	acceptable	1.81e+03	3.08e−01	35	1.37e−13	3.28e−10
BOEING1	384	351	acceptable	−3.35e+02	2.34e−02	20	7.95e−12	9.24e−14
BOEING2	143	166	acceptable	−3.15e+02	5.62e−03	13	6.74e−12	1.96e−13
BRANDY	249	220	acceptable	1.52e+03	9.91e−03	16	2.51e−13	1.40e−10
CRE-A	4067	3516	acceptable	2.36e+07	1.74e−01	30	1.29e−12	3.29e−08
CRE-B	72447	9648	acceptable	2.31e+07	1.79e+00	38	2.89e−12	2.03e−10
CRE-C	3678	3068	acceptable	2.53e+07	1.61e−01	34	2.98e−12	1.43e−10
CRE-D	69980	8926	acceptable	2.45e+07	1.38e+00	36	4.94e−12	4.33e−08
CYCLE	2857	1903	acceptable	−5.23e+00	1.96e−01	27	3.22e−12	5.83e−14
D2Q06C	5167	2171	acceptable	1.23e+05	3.41e−01	25	5.35e−11	3.35e−08
D6CUBE	6184	415	acceptable	3.15e+02	1.42e−01	13	1.19e−11	1.09e−09
DEGEN2	534	444	acceptable	−1.44e+03	2.96e−02	13	4.01e−15	2.73e−11
DEGEN3	1818	1503	acceptable	−9.87e+02	2.63e−01	15	2.03e−14	8.32e−13
DFL001	12230	6071	acceptable	1.13e+07	6.23e+00	24	3.29e−14	1.60e−04
E226	282	223	acceptable	−1.16e+01	1.23e−02	18	5.16e−14	2.03e−09
FFFFF800	854	524	acceptable	5.56e+05	5.50e−02	26	2.91e−11	7.88e−09
FIT1D	1026	24	acceptable	−9.15e+03	2.34e−02	13	2.36e−11	1.59e−13
FIT1P	1677	627	acceptable	9.15e+03	4.56e−02	15	2.74e−14	5.26e−07
FIT2D	10500	25	acceptable	−6.85e+04	2.98e−01	16	7.62e−10	4.47e−12
FIT2P	13525	3000	acceptable	6.85e+04	3.12e−01	24	7.22e−14	1.02e−10
GANGES	1681	1309	acceptable	−1.10e+05	6.26e−02	16	7.28e−12	6.67e−11
GFRD-PNC	1092	616	acceptable	6.90e+06	2.15e−02	20	3.92e−12	4.26e−10
GROW15	645	300	acceptable	−1.07e+08	1.92e−02	11	2.47e−10	2.20e−09
GROW22	946	440	acceptable	−1.61e+08	2.96e−02	12	4.01e−10	2.00e−10
GROW7	301	140	acceptable	−4.78e+07	8.35e−03	10	3.05e−10	4.85e−10
ISRAEL	142	174	acceptable	−8.97e+05	1.09e−02	15	9.14e−11	1.63e−06
KB2	41	43	acceptable	−1.75e+03	1.72e−03	13	1.40e−12	8.81e−12
KEN-07	3602	2426	acceptable	−6.80e+08	4.55e−02	14	1.60e−12	6.29e−08
KEN-11	21349	14694	acceptable	−6.97e+09	5.53e−01	20	3.61e−12	2.60e−10
KEN-13	42659	28632	acceptable	−1.03e+10	1.55e+00	22	8.91e−12	2.74e−10
KEN-18	154699	105127	acceptable	−5.22e+10	1.16e+01	33	2.63e−11	5.99e−09
LOTFI	308	153	acceptable	−2.53e+01	7.26e−03	16	3.98e−09	1.04e−09
MAROS-R7	9408	3136	acceptable	1.50e+06	2.80e+00	13	1.94e−11	6.52e−12
MODSZK1	1620	687	acceptable	3.21e+02	5.40e−02	27	3.93e−11	2.84e−12
OSA-07	23949	1118	acceptable	5.36e+05	1.77e−01	19	6.84e−12	1.65e−12
OSA-14	52460	2337	acceptable	1.11e+06	4.10e−01	23	8.83e−11	1.69e−12
OSA-30	100024	4350	acceptable	2.14e+06	9.24e−01	34	3.35e−11	2.02e−12
OSA-60	232966	10280	acceptable	4.04e+06	2.15e+00	38	1.49e−10	2.03e−12
PDS-02	7535	2953	acceptable	2.89e+10	1.97e−01	26	7.09e−11	4.89e−07
PDS-06	28655	9881	acceptable	2.78e+10	2.34e+00	36	1.22e−11	4.87e−07

Continued on next page

Table 10: CPLEX results without presolve and crossover on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
PDS-10	48763	16558	acceptable	2.67e+10	9.10e+00	41	8.54e−12	1.59e−06
PDS-20	105728	33874	acceptable	2.38e+10	4.00e+01	48	5.96e−11	7.82e−06
QAP12	8856	3192	acceptable	5.23e+02	2.61e−01	13	1.67e−16	5.99e−13
QAP15	22275	6330	acceptable	1.04e+03	1.79e+00	14	2.36e−16	8.84e−14
QAP8	1632	912	acceptable	2.04e+02	1.28e−02	8	2.22e−16	8.99e−15
SC105	103	105	acceptable	−5.22e+01	2.55e−03	11	4.26e−14	1.81e−06
SC205	203	205	acceptable	−5.22e+01	4.48e−03	12	2.27e−13	7.38e−08
SC50A	48	50	acceptable	−6.46e+01	1.33e−03	10	2.84e−14	4.64e−10
SC50B	48	50	acceptable	−7.00e+01	1.10e−03	7	1.42e−14	8.89e−10
SCAGR25	500	471	acceptable	−1.48e+07	1.09e−02	16	2.59e−12	1.38e−09
SCAGR7	140	129	acceptable	−2.33e+06	3.32e−03	14	1.88e−12	1.06e−10
SCFXM1	457	330	acceptable	1.84e+04	1.57e−02	15	4.60e−12	3.95e−12
SCFXM2	914	660	acceptable	3.67e+04	3.47e−02	17	6.48e−12	4.31e−11
SCFXM3	1371	990	acceptable	5.49e+04	5.16e−02	17	6.53e−12	6.36e−11
SCORPION	358	388	acceptable	1.88e+03	6.78e−03	9	3.51e−16	7.37e−05
SCRS8	1169	490	acceptable	9.04e+02	2.65e−02	20	6.37e−14	3.07e−10
SCSD1	760	77	acceptable	8.67e+00	5.32e−03	10	1.62e−16	2.44e−15
SCSD6	1350	147	acceptable	5.05e+01	1.01e−02	11	6.13e−16	3.73e−15
SCSD8	2750	397	acceptable	9.05e+02	1.89e−02	9	7.21e−15	3.64e−10
SCTAP1	480	300	acceptable	1.41e+03	8.69e−03	14	1.68e−14	1.98e−09
SCTAP2	1880	1090	acceptable	1.72e+03	3.04e−02	11	1.83e−14	4.75e−12
SCTAP3	2480	1480	acceptable	1.42e+03	4.17e−02	11	1.62e−14	1.03e−08
SEBA	1028	515	acceptable	1.57e+04	7.71e−03	10	8.38e−13	1.85e−13
SHARE1B	225	117	acceptable	−7.66e+04	6.22e−03	18	3.01e−10	3.33e−10
SHARE2B	79	96	acceptable	−4.16e+02	3.68e−03	18	3.72e−13	6.46e−09
SHIP04L	2118	402	acceptable	1.79e+06	1.32e−02	12	6.74e−14	1.62e−12
SHIP04S	1458	402	acceptable	1.80e+06	9.36e−03	12	4.39e−14	1.82e−12
SHIP08L	4283	778	acceptable	1.91e+06	2.76e−02	16	2.43e−14	2.29e−12
SHIP08S	2387	778	acceptable	1.92e+06	1.40e−02	11	2.34e−14	1.43e−07
SHIP12L	5427	1151	acceptable	1.47e+06	3.75e−02	17	2.45e−14	2.47e−12
SHIP12S	2763	1151	acceptable	1.49e+06	1.61e−02	14	2.79e−14	1.56e−12
STOCFOR1	111	117	acceptable	−4.11e+04	2.53e−03	9	8.09e−13	2.15e−09
STOCFOR2	2031	2157	acceptable	−3.90e+04	7.42e−02	16	7.70e−13	3.63e−11
STOCFOR3	15695	16675	acceptable	−4.00e+04	9.76e−01	25	9.70e−13	5.61e−12
TRUSS	8806	1000	acceptable	4.59e+05	1.41e−01	17	1.55e−13	5.51e−11
WOOD1P	2594	244	acceptable	1.44e+00	1.22e−01	13	1.27e−13	3.18e−11
WOODW	8405	1098	acceptable	1.30e+00	1.22e−01	18	1.42e−13	1.47e−10

Table 11: CPLEX results without presolve and crossover on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
AUG2D	20200	10000	acceptable	1.69e+06	9.76e−02	1	3.55e−15	5.68e−14
AUG2DC	20200	10000	acceptable	1.82e+06	9.99e−02	1	3.55e−15	5.68e−14
AUG2DCQP	20200	10000	acceptable	6.50e+06	4.38e−01	15	9.33e−15	4.11e−13
AUG2DQP	20200	10000	acceptable	6.24e+06	5.23e−01	15	9.55e−15	3.65e−13
AUG3D	3873	1000	acceptable	5.54e+02	9.09e−03	1	6.66e−16	4.44e−16
AUG3DC	3873	1000	acceptable	7.71e+02	2.19e−02	1	8.88e−16	4.44e−16
AUG3DCQP	3873	1000	acceptable	9.93e+02	5.01e−02	10	1.33e−15	9.99e−16
AUG3DQP	3873	1000	acceptable	6.75e+02	3.13e−02	7	8.88e−16	8.88e−16
BOYD1	93261	18	acceptable	−6.17e+07	1.36e+00	20	3.01e−02	4.81e−10
BOYD2	93263	186531	acceptable	2.13e+01	9.57e+00	51	1.24e−03	6.17e−11
CONT-050	2597	2401	acceptable	−4.56e+00	1.23e−01	10	8.88e−16	4.40e−17
CONT-100	10197	9801	acceptable	−4.64e+00	8.50e−01	12	1.11e−15	6.70e−17
CONT-101	10197	10098	acceptable	1.96e−01	7.84e−01	10	3.55e−15	1.55e−16
CONT-200	40397	39601	acceptable	−4.68e+00	5.29e+00	11	8.88e−16	7.39e−17
CONT-201	40397	40198	acceptable	1.92e−01	4.81e+00	10	3.55e−15	1.95e−16
CONT-300	90597	90298	acceptable	1.92e−01	1.36e+01	10	3.55e−15	2.32e−16
CVXQP1_L	10000	5000	acceptable	1.09e+08	8.63e+01	7	1.33e−15	1.49e−08
CVXQP1_M	1000	500	acceptable	1.09e+06	6.20e−01	7	1.11e−15	4.82e−11
CVXQP1_S	100	50	acceptable	1.16e+04	5.55e−03	7	7.49e−16	3.41e−13
CVXQP2_L	10000	2500	acceptable	8.18e+07	3.86e+01	8	1.22e−15	5.82e−11
CVXQP2_M	1000	250	acceptable	8.20e+05	3.28e−01	7	1.05e−15	1.82e−12
CVXQP2_S	100	25	acceptable	8.12e+03	4.42e−03	8	8.74e−16	1.14e−13

Continued on next page

Table 11: CPLEX results without presolve and crossover on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
CVXQP3_L	10000	7500	acceptable	1.16e+08	1.59e+02	9	1.11e−15	1.34e−07
CVXQP3_M	1000	750	acceptable	1.36e+06	1.18e+00	8	1.11e−15	3.48e−10
CVXQP3_S	100	75	acceptable	1.19e+04	7.65e−03	8	8.19e−16	2.27e−13
DPKLO1	133	77	acceptable	3.70e−01	1.92e−03	3	6.17e−15	2.80e−14
DUAL1	85	1	acceptable	3.50e−02	1.01e−02	14	9.06e−17	1.17e−15
DUAL2	96	1	acceptable	3.37e−02	1.18e−02	12	2.52e−16	6.75e−16
DUAL3	111	1	acceptable	1.36e−01	1.69e−02	13	7.14e−17	1.36e−15
DUAL4	75	1	acceptable	7.46e−01	6.91e−03	12	8.28e−17	1.30e−15
DUALC1	9	215	acceptable	6.16e+03	1.22e−03	11	5.16e−13	1.89e−10
DUALC2	7	229	acceptable	3.55e+03	1.01e−03	10	4.67e−13	7.28e−12
DUALC5	8	278	acceptable	4.27e+02	1.06e−03	10	1.03e−13	7.11e−13
DUALC8	8	503	acceptable	1.83e+04	1.56e−03	8	2.65e−13	5.82e−11
EXDATA	3000	3001	acceptable	−1.42e+02	8.33e+01	13	1.73e−14	7.57e−14
GENHS28	10	8	acceptable	9.27e−01	7.67e−04	2	1.39e−16	0.00e+00
GOULDQP2	699	349	acceptable	1.84e−04	1.41e−02	14	0.00e+00	2.20e−08
GOULDQP3	699	349	acceptable	2.06e+00	1.32e−02	8	0.00e+00	7.11e−15
HS118	15	17	acceptable	6.65e+02	7.76e−04	11	6.66e−15	3.17e−16
HS21	2	1	acceptable	−1.00e+02	4.41e−04	15	2.35e−16	0.00e+00
HS268	5	5	acceptable	1.17e−05	5.71e−04	11	8.88e−16	1.82e−12
HS35	3	1	acceptable	1.11e−01	4.93e−04	15	0.00e+00	8.88e−16
HS51	5	3	acceptable	0.00e+00	3.61e−04	2	0.00e+00	4.44e−16
HS52	5	3	acceptable	5.33e+00	3.33e−04	2	0.00e+00	4.44e−16
HS53	5	3	acceptable	4.09e+00	5.18e−04	13	2.78e−17	4.44e−16
HS76	4	3	acceptable	−4.68e+00	5.00e−04	12	1.34e−16	4.44e−16
HUES-MOD	10000	2	acceptable	3.48e+07	3.44e−02	10	1.47e−11	1.64e−15
HUESTIS	10000	2	infeasible	3.63e+19	3.38e−02	10	1.20e−11	2.91e−11
KSIP	20	1001	acceptable	5.76e−01	1.08e+00	13	4.24e−16	2.50e−16
LASER	1002	1000	acceptable	2.41e+06	4.43e−02	11	3.40e−14	1.71e−13
LISWET1	10002	10000	acceptable	3.61e+01	2.72e−01	26	1.11e−16	2.91e−11
LISWET10	10002	10000	acceptable	4.95e+01	4.25e−01	43	1.11e−16	2.91e−11
LISWET11	10002	10000	acceptable	4.95e+01	2.71e−01	26	1.11e−16	7.27e−12
LISWET12	10002	10000	acceptable	1.74e+03	6.27e−01	66	1.11e−16	2.56e−10
LISWET2	10002	10000	acceptable	2.50e+01	1.45e−01	12	1.11e−16	2.84e−14
LISWET3	10002	10000	acceptable	2.50e+01	1.81e−01	16	5.55e−17	6.41e−16
LISWET4	10002	10000	acceptable	2.50e+01	1.98e−01	18	5.55e−17	8.91e−16
LISWET5	10002	10000	acceptable	2.50e+01	1.64e−01	14	2.22e−16	8.88e−16
LISWET6	10002	10000	acceptable	2.50e+01	1.71e−01	15	0.00e+00	8.88e−16
LISWET7	10002	10000	acceptable	4.99e+02	1.92e−01	17	0.00e+00	2.32e−10
LISWET8	10002	10000	acceptable	7.14e+02	6.29e−01	66	5.55e−17	1.17e−10
LISWET9	10002	10000	acceptable	1.96e+03	6.00e−01	63	1.39e−17	1.19e−10
LOTSCHD	12	7	acceptable	2.40e+03	6.28e−04	9	1.08e−14	1.42e−14
MOSARQP1	2500	700	acceptable	−9.53e+02	6.46e−02	14	2.69e−16	4.44e−16
MOSARQP2	900	600	acceptable	−1.60e+03	3.65e−02	12	2.50e−16	1.55e−15
POWELL20	10000	10000	acceptable	5.21e+10	4.17e−01	39	9.09e−13	2.27e−13
PRIMAL1	325	85	acceptable	−3.50e−02	1.28e−02	16	4.85e−17	1.70e−16
PRIMAL2	649	96	acceptable	−3.37e−02	1.56e−02	15	4.74e−17	4.31e−17
PRIMAL3	745	111	acceptable	−1.36e−01	3.95e−02	14	3.04e−16	2.60e−16
PRIMAL4	1489	75	acceptable	−7.46e−01	2.17e−02	10	1.29e−15	2.23e−16
PRIMALC1	230	9	acceptable	−6.16e+03	1.20e−03	12	1.99e−10	3.83e−13
PRIMALC2	231	7	acceptable	−3.55e+03	9.26e−04	11	1.50e−12	3.41e−13
PRIMALC5	287	8	acceptable	−4.27e+02	1.10e−03	12	1.11e−13	8.34e−14
PRIMALC8	520	8	acceptable	−1.83e+04	1.52e−03	9	1.10e−11	3.10e−13
Q25FV47	1571	820	acceptable	1.37e+07	9.19e−01	20	2.01e−12	3.30e+00
QADLITTL	97	56	acceptable	4.80e+05	2.27e−03	10	2.07e−13	7.64e−11
QAFIRO	32	27	acceptable	−1.59e+00	1.11e−03	15	1.47e−14	1.97e−15
QBANDM	472	305	acceptable	1.64e+04	1.24e−02	20	1.28e−13	3.00e+00
QBEACONF	262	173	acceptable	1.65e+05	4.13e−03	13	5.91e−11	1.78e+01
QBRANDY	249	220	acceptable	2.84e+04	7.78e−03	16	7.15e−13	1.88e−11
QE226	282	223	acceptable	2.13e+02	1.74e−02	20	4.30e−14	1.96e−10
QFFFFF80	854	524	acceptable	8.73e+05	6.65e−02	25	2.91e−11	9.84e+02
QGFRDXPN	1092	616	acceptable	1.01e+11	2.52e−02	28	7.28e−12	2.10e−08
QGROW15	645	300	acceptable	−1.02e+08	2.06e−02	16	4.62e−10	1.97e−12
QGROW22	946	440	acceptable	−1.50e+08	3.81e−02	20	3.39e−10	3.52e−14
QGROW7	301	140	acceptable	−4.28e+07	1.07e−02	17	2.07e−10	2.85e−14
QISRAEL	142	174	acceptable	2.53e+07	1.73e−02	19	4.61e−11	3.42e−09

Continued on next page

Table 11: CPLEX results without presolve and crossover on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
QPCBLEND	83	74	acceptable	−7.84e−03	2.60e−03	15	1.66e−15	1.25e−15
QPCBOEI1	384	351	acceptable	1.15e+07	2.44e−02	27	8.21e−12	2.52e−11
QPCBOEI2	143	166	acceptable	8.17e+06	1.16e−02	42	5.32e−12	3.09e−10
QPTEST	2	2	acceptable	4.37e+00	5.11e−04	15	1.67e−16	0.00e+00
QSC205	203	205	acceptable	−5.81e−03	5.16e−03	18	4.09e−14	1.39e−12
QSCAGR25	500	471	acceptable	2.02e+08	1.07e−02	18	4.39e−12	5.00e+02
QSCAGR7	140	129	acceptable	2.69e+07	3.05e−03	15	1.17e−12	5.00e+02
QSCFXM1	457	330	acceptable	1.69e+07	2.40e−02	24	2.70e−12	6.30e+00
QSCFXM2	914	660	acceptable	2.78e+07	5.09e−02	28	7.92e−12	6.30e+00
QSCFXM3	1371	990	acceptable	3.08e+07	7.66e−02	30	5.13e−12	6.30e+00
QSCORPIO	358	388	acceptable	1.88e+03	7.55e−03	14	3.31e−16	4.99e+01
QSCRS8	1169	490	acceptable	9.05e+02	2.30e−02	23	5.42e−14	1.97e−11
QSCSD1	760	77	acceptable	8.67e+00	8.09e−03	13	1.35e−16	1.85e−15
QSCSD6	1350	147	acceptable	5.08e+01	1.72e−02	16	3.52e−16	3.14e−15
QSCSD8	2750	397	acceptable	9.41e+02	3.71e−02	14	6.36e−15	3.15e−09
QSCTAP1	480	300	acceptable	1.42e+03	1.03e−02	17	1.67e−14	2.12e−11
QSCTAP2	1880	1090	acceptable	1.74e+03	4.55e−02	16	1.64e−14	5.77e−11
QSCTAP3	2480	1480	acceptable	1.44e+03	6.43e−02	17	1.06e−14	1.04e−10
QSEBA	1028	515	acceptable	8.15e+07	8.98e−03	19	6.75e−13	1.06e−09
QSHARE1B	225	117	acceptable	7.20e+05	6.16e−03	21	1.12e−10	1.14e−13
QSHARE2B	79	96	acceptable	1.17e+04	3.38e−03	16	3.00e−13	2.13e−09
QSHIP04L	2118	402	acceptable	2.42e+06	1.26e−02	12	1.93e−14	1.16e+02
QSHIP04S	1458	402	acceptable	2.42e+06	9.04e−03	12	4.68e−14	1.16e+02
QSHIP08L	4283	778	acceptable	2.38e+06	1.96e−01	12	2.70e−14	1.02e+02
QSHIP08S	2387	778	acceptable	2.39e+06	5.75e−02	12	1.22e−14	1.02e+02
QSHIP12L	5427	1151	acceptable	3.02e+06	3.30e−01	16	3.12e−14	9.02e+01
QSHIP12S	2763	1151	acceptable	3.06e+06	8.40e−02	15	4.04e−14	9.02e+01
S268	5	5	acceptable	1.17e−05	6.58e−04	11	8.88e−16	1.82e−12
STADAT1	2001	3999	acceptable	−2.85e+07	2.25e−01	52	1.75e−09	1.72e−05
STADAT2	2001	3999	acceptable	−3.26e+01	1.01e−01	21	3.30e−12	2.02e−09
STADAT3	4001	7999	acceptable	−3.58e+01	1.77e−01	17	7.13e−12	1.62e−09
STCQP1	4097	2052	acceptable	1.55e+05	7.65e−02	10	0.00e+00	1.25e−12
STCQP2	4097	2052	acceptable	2.23e+04	1.50e+00	9	0.00e+00	8.53e−14
TAME	2	1	acceptable	0.00e+00	5.10e−04	2	0.00e+00	0.00e+00
VALUES	202	1	exception	∞	∞	0	∞	∞
ZECEVIC2	2	2	acceptable	−4.12e+00	5.83e−04	12	2.22e−16	5.89e−17

D Xpress detailed results

Table 12: Xpress results without presolve and crossover on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
25FV47	1571	821	acceptable	5.50e+03	3.32e−01	15	3.23e−09	1.42e−14
ADLITTLE	97	56	acceptable	2.25e+05	7.13e−02	11	1.61e−08	2.33e−10
AFIRO	32	27	acceptable	−4.65e+02	4.79e−02	7	8.72e−11	1.78e−15
AGG	163	488	acceptable	−3.60e+07	1.07e−01	15	1.00e−06	2.10e−08
AGG2	302	516	acceptable	−2.02e+07	2.30e−01	14	1.28e−06	3.29e−07
AGG3	302	516	acceptable	1.03e+07	2.10e−01	14	1.77e−05	2.23e−06
BANDM	472	305	acceptable	−1.59e+02	9.85e−02	11	6.26e−10	2.09e−13
BEACONFD	262	173	acceptable	3.36e+04	1.06e−01	11	8.82e−09	4.55e−12
BLEND	83	74	acceptable	−3.08e+01	7.37e−02	12	2.65e−11	2.68e−15
BNL1	1175	643	acceptable	1.98e+03	2.18e−01	21	9.92e−10	9.09e−13
BNL2	3489	2324	acceptable	1.81e+03	9.66e−01	23	1.33e−06	1.86e−09
BOEING1	384	351	acceptable	−3.35e+02	1.77e−01	18	1.76e−10	1.32e−11
BOEING2	143	166	acceptable	−3.15e+02	9.12e−02	12	5.10e−09	3.76e−14
BRANDY	249	220	acceptable	1.52e+03	1.72e−01	14	1.61e−07	1.14e−13
CRE-A	4067	3516	acceptable	2.36e+07	5.35e−01	20	4.24e−09	6.98e−10
CRE-B	72447	9648	acceptable	2.31e+07	6.17e+00	21	7.71e−07	1.05e−10
CRE-C	3678	3068	acceptable	2.53e+07	6.00e−01	22	5.05e−09	9.31e−10
CRE-D	69980	8926	acceptable	2.45e+07	7.13e+00	28	3.89e−06	7.28e−11
CYCLE	2857	1903	acceptable	−5.23e+00	6.34e−01	19	6.98e−10	2.26e−10
D2Q06C	5167	2171	acceptable	1.23e+05	6.93e−01	16	5.80e−07	9.09e−12

Continued on next page

Table 12: Xpress results without presolve and crossover on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
D6CUBE	6184	415	acceptable	3.15e+02	6.23e−01	17	7.90e−11	5.33e−15
DEGEN2	534	444	acceptable	−1.44e+03	1.65e−01	9	3.43e−10	3.93e−14
DEGEN3	1818	1503	acceptable	−9.87e+02	5.56e−01	12	3.05e−09	5.71e−13
DFL001	12230	6071	acceptable	1.13e+07	7.86e+00	22	1.40e−07	2.38e−07
E226	282	223	acceptable	−1.16e+01	1.49e−01	16	3.49e−10	2.27e−13
FFFFF800	854	524	acceptable	5.56e+05	2.91e−01	21	6.91e−09	1.05e−09
FIT1D	1026	24	acceptable	−9.15e+03	1.19e−01	10	7.77e−10	5.70e−13
FIT1P	1677	627	acceptable	9.15e+03	2.48e−01	19	1.02e−08	2.99e−10
FIT2D	10500	25	acceptable	−6.85e+04	3.50e−01	13	8.79e−12	1.76e−12
FIT2P	13525	3000	acceptable	6.85e+04	5.31e−01	16	7.76e−10	1.53e−10
GANGES	1681	1309	acceptable	−1.10e+05	2.54e−01	12	9.07e−07	3.64e−12
GFRD-PNC	1092	616	acceptable	6.90e+06	1.22e−01	13	1.63e−08	2.33e−10
GROW15	645	300	acceptable	−1.07e+08	9.29e−02	9	2.76e−06	3.55e−14
GROW22	946	440	acceptable	−1.61e+08	1.21e−01	10	4.42e−06	3.55e−14
GROW7	301	140	acceptable	−4.78e+07	8.00e−02	9	1.05e−07	1.42e−14
ISRAEL	142	174	acceptable	−8.97e+05	1.14e−01	9	8.37e−11	9.09e−13
KB2	41	43	acceptable	−1.75e+03	5.84e−02	9	9.64e−10	7.11e−15
KEN-07	3602	2426	acceptable	−6.80e+08	2.66e−01	13	2.31e−08	1.09e−11
KEN-11	21349	14694	acceptable	−6.97e+09	1.11e+00	16	1.46e−06	1.46e−11
KEN-13	42659	28632	acceptable	−1.03e+10	3.31e+00	21	2.34e−07	1.66e−11
KEN-18	154699	105127	acceptable	−5.22e+10	2.27e+01	24	1.81e−05	2.18e−11
LOTFI	308	153	acceptable	−2.53e+01	6.63e−02	14	1.46e−10	4.44e−16
MAROS-R7	9408	3136	acceptable	1.50e+06	2.10e+00	9	3.23e−07	4.44e−15
MODSZK1	1620	687	acceptable	3.21e+02	3.52e−01	20	2.83e−05	4.55e−13
OSA-07	23949	1118	acceptable	5.36e+05	7.26e−01	14	4.41e−07	2.73e−12
OSA-14	52460	2337	acceptable	1.11e+06	1.83e+00	18	1.82e−07	2.73e−12
OSA-30	100024	4350	acceptable	2.14e+06	3.84e+00	24	2.13e−05	2.73e−12
OSA-60	232966	10280	acceptable	4.04e+06	9.40e+00	26	2.69e−05	1.92e−09
PDS-02	7535	2953	acceptable	2.89e+10	6.81e−01	22	1.13e−05	1.29e−10
PDS-06	28655	9881	acceptable	2.78e+10	5.22e+00	28	4.95e−06	2.20e−10
PDS-10	48763	16558	acceptable	2.67e+10	1.62e+01	39	8.62e−06	3.61e−10
PDS-20	105728	33874	acceptable	2.38e+10	3.49e+01	34	1.48e−06	3.17e−09
QAP12	8856	3192	acceptable	5.23e+02	4.37e+00	12	1.88e−10	4.09e−14
QAP15	22275	6330	acceptable	1.04e+03	1.86e+01	15	4.07e−10	1.67e−13
QAP8	1632	912	acceptable	2.04e+02	3.05e−01	7	4.88e−09	3.91e−14
SC105	103	105	acceptable	−5.22e+01	6.25e−02	9	1.15e−10	1.82e−12
SC205	203	205	acceptable	−5.22e+01	9.92e−02	11	5.21e−11	9.09e−13
SC50A	48	50	acceptable	−6.46e+01	5.07e−02	8	1.44e−11	6.57e−17
SC50B	48	50	acceptable	−7.00e+01	4.83e−02	7	9.36e−12	2.17e−11
SCAGR25	500	471	acceptable	−1.48e+07	1.05e−01	13	8.90e−09	1.82e−12
SCAGR7	140	129	acceptable	−2.33e+06	7.76e−02	12	2.10e−09	9.09e−13
SCFXM1	457	330	exception	1.84e+04	3.39e−01	23	6.38e−05	9.09e−13
SCFXM2	914	660	acceptable	3.67e+04	5.16e−01	37	6.25e−03	1.16e−10
SCFXM3	1371	990	acceptable	5.49e+04	5.66e−01	27	1.21e−05	3.73e−09
SCORPION	358	388	acceptable	1.88e+03	8.25e−02	10	1.09e−09	7.28e−12
SCRS8	1169	490	acceptable	9.04e+02	1.26e−01	15	1.04e−11	3.64e−11
SCSD1	760	77	acceptable	8.67e+00	6.87e−02	9	3.18e−13	2.22e−15
SCSD6	1350	147	acceptable	5.05e+01	1.08e−01	9	1.57e−10	4.00e−15
SCSD8	2750	397	acceptable	9.05e+02	1.07e−01	8	1.16e−11	9.08e−15
SCTAP1	480	300	acceptable	1.41e+03	9.60e−02	11	2.64e−09	5.68e−14
SCTAP2	1880	1090	acceptable	1.72e+03	1.47e−01	11	1.37e−11	2.84e−14
SCTAP3	2480	1480	acceptable	1.42e+03	2.29e−01	12	9.07e−12	3.46e−14
SEBA	1028	515	acceptable	1.57e+04	1.20e−01	10	2.64e−08	3.61e−12
SHARE1B	225	117	acceptable	−7.66e+04	8.99e−02	14	9.08e−07	3.55e−14
SHARE2B	79	96	acceptable	−4.16e+02	7.33e−02	12	8.51e−11	7.21e−14
SHIP04L	2118	402	acceptable	1.79e+06	1.38e−01	10	1.06e−09	1.42e−12
SHIP04S	1458	402	acceptable	1.80e+06	1.12e−01	10	1.02e−07	1.53e−12
SHIP08L	4283	778	acceptable	1.91e+06	2.15e−01	12	9.90e−09	2.05e−12
SHIP08S	2387	778	acceptable	1.92e+06	1.42e−01	11	3.41e−07	2.73e−12
SHIP12L	5427	1151	acceptable	1.47e+06	2.29e−01	12	2.73e−08	2.38e−12
SHIP12S	2763	1151	acceptable	1.49e+06	1.59e−01	11	2.78e−08	1.48e−12
STOCFOR1	111	117	acceptable	−4.11e+04	6.49e−02	9	5.90e−11	3.64e−12
STOCFOR2	2031	2157	acceptable	−3.90e+04	2.46e−01	13	5.96e−10	1.82e−12
STOCFOR3	15695	16675	acceptable	−4.00e+04	1.83e+00	20	7.93e−10	9.09e−13
TRUSS	8806	1000	acceptable	4.59e+05	2.65e−01	13	5.03e−08	1.70e−11

Continued on next page

Table 12: Xpress results without presolve and crossover on Netlib problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
WOOD1P	2594	244	acceptable	1.44e+00	3.25e−01	15	5.21e−13	3.64e−12
WOODW	8405	1098	acceptable	1.30e+00	4.63e−01	19	2.01e−11	9.09e−13

Table 13: Xpress results without presolve and crossover on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
AUG2D	20200	10000	acceptable	1.69e+06	1.95e−01	1	1.60e−12	1.00e−04
AUG2DC	20200	10000	acceptable	1.82e+06	1.69e−01	1	1.29e−12	9.99e−05
AUG2DCQP	20200	10000	acceptable	6.50e+06	2.33e+00	25	3.44e−07	4.87e−07
AUG2DQP	20200	10000	acceptable	6.24e+06	2.13e+00	25	2.72e−07	3.90e−07
AUG3D	3873	1000	acceptable	5.54e+02	7.34e−02	2	3.11e−15	3.90e−12
AUG3DC	3873	1000	acceptable	7.71e+02	6.66e−02	2	2.22e−15	6.86e−11
AUG3DCQP	3873	1000	acceptable	9.93e+02	2.27e−01	9	2.66e−15	1.78e−15
AUG3DQP	3873	1000	acceptable	6.75e+02	2.04e−01	10	3.55e−15	1.44e−15
BOYD1	93261	18	acceptable	−6.17e+07	2.19e+00	16	1.15e−10	5.24e−10
BOYD2	93263	186531	acceptable	2.13e+01	4.51e+01	74	3.44e−07	2.62e−10
CONT-050	2597	2401	acceptable	−4.56e+00	3.02e−01	9	5.55e−16	6.16e−17
CONT-100	10197	9801	acceptable	−4.64e+00	1.22e+00	9	6.11e−16	1.26e−16
CONT-101	10197	10098	acceptable	1.96e−01	1.21e+00	9	8.91e−10	2.45e−16
CONT-200	40397	39601	acceptable	−4.68e+00	6.49e+00	9	1.78e−15	1.67e−16
CONT-201	40397	40198	acceptable	1.92e−01	7.17e+00	10	5.34e−09	2.32e−16
CONT-300	90597	90298	acceptable	1.92e−01	2.11e+01	11	1.95e−07	2.52e−16
CVXQP1_L	10000	5000	acceptable	1.09e+08	5.57e+00	9	2.90e−05	6.68e−07
CVXQP1_M	1000	500	acceptable	1.09e+06	5.44e−01	8	5.55e−10	1.37e−10
CVXQP1_S	100	50	acceptable	1.16e+04	7.13e−02	7	4.46e−12	2.34e−13
CVXQP2_L	10000	2500	acceptable	8.18e+07	4.06e+00	8	5.01e−07	9.04e−09
CVXQP2_M	1000	250	acceptable	8.20e+05	3.06e−01	8	3.30e−11	2.86e−12
CVXQP2_S	100	25	acceptable	8.12e+03	6.82e−02	8	6.47e−12	1.46e−13
CVXQP3_L	10000	7500	acceptable	1.16e+08	6.35e+00	9	1.16e−08	7.83e−08
CVXQP3_M	1000	750	acceptable	1.36e+06	1.83e+00	30	6.59e−08	2.81e−08
CVXQP3_S	100	75	acceptable	1.19e+04	8.56e−02	9	1.89e−10	3.74e−13
DPKLO1	133	77	acceptable	3.70e−01	3.11e−02	2	6.77e−15	1.80e−13
DUAL1	85	1	acceptable	3.50e−02	1.17e−01	11	3.89e−16	3.78e−15
DUAL2	96	1	acceptable	3.37e−02	9.71e−02	10	1.91e−17	2.99e−15
DUAL3	111	1	acceptable	1.36e−01	1.17e−01	10	3.73e−16	5.62e−15
DUAL4	75	1	acceptable	7.46e−01	8.42e−02	10	1.03e−16	3.74e−15
DUALC1	9	215	acceptable	6.16e+03	1.63e−01	8	3.11e−15	3.85e−10
DUALC2	7	229	acceptable	3.55e+03	1.17e−01	7	3.55e−15	1.06e−11
DUALC5	8	278	acceptable	4.27e+02	2.14e−01	9	1.02e−14	5.49e−12
DUALC8	8	503	exception	∞	∞	0	∞	∞
EXDATA	3000	3001	acceptable	−1.42e+02	4.16e+00	13	1.09e−13	2.09e−13
GENHS28	10	8	acceptable	9.27e−01	1.26e−02	1	2.30e−10	3.75e−08
GOULDQP2	699	349	acceptable	1.84e−04	1.45e+00	18	3.36e−13	1.61e−16
GOULDQP3	699	349	acceptable	2.06e+00	4.29e−01	8	4.83e−15	1.00e−14
HS118	15	17	acceptable	6.65e+02	5.73e−02	8	1.45e−14	5.13e−16
HS21	2	1	acceptable	−1.00e+02	3.98e−02	7	2.22e−16	2.90e−18
HS268	5	5	acceptable	3.60e−09	7.22e−02	12	1.11e−15	3.64e−12
HS35	3	1	acceptable	1.11e−01	4.86e−02	9	2.05e−17	1.67e−15
HS51	5	3	acceptable	0.00e+00	2.27e−02	2	0.00e+00	2.22e−16
HS52	5	3	acceptable	5.33e+00	2.18e−02	2	8.29e−12	1.29e−05
HS53	5	3	acceptable	4.09e+00	1.21e−01	25	2.45e−11	2.16e−10
HS76	4	3	acceptable	−4.68e+00	4.69e−02	7	2.31e−16	4.49e−16
HUES-MOD	10000	2	acceptable	3.48e+07	2.85e−01	17	3.77e−08	1.71e−13
HUESTIS	10000	2	acceptable	3.48e+11	2.54e−01	17	7.75e−08	9.31e−10
KSIP	20	1001	acceptable	5.76e−01	2.61e−01	20	6.32e−16	1.05e−10
LASER	1002	1000	acceptable	2.41e+06	1.78e−01	10	1.06e−09	8.42e−07
LISWET1	10002	10000	acceptable	2.51e+01	5.11e−01	8	1.31e−07	1.18e−05
LISWET10	10002	10000	acceptable	2.51e+01	3.74e−01	7	4.94e−08	4.19e−06
LISWET11	10002	10000	acceptable	2.51e+01	4.09e−01	6	2.00e−07	1.89e−05
LISWET12	10002	10000	exception	∞	∞	0	∞	∞
LISWET2	10002	10000	acceptable	2.50e+01	5.99e−01	10	1.01e−13	3.26e−09
LISWET3	10002	10000	acceptable	2.50e+01	6.68e−01	14	1.21e−16	9.57e−10
LISWET4	10002	10000	acceptable	2.50e+01	5.45e−01	12	1.50e−16	1.18e−09

Continued on next page

Table 13: Xpress results without presolve and crossover on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
LISWET5	10002	10000	acceptable	2.50e+01	5.95e−01	12	4.47e−16	1.09e−11
LISWET6	10002	10000	acceptable	2.50e+01	5.81e−01	13	2.22e−16	2.24e−11
LISWET7	10002	10000	acceptable	2.51e+01	3.29e−01	7	4.94e−08	9.91e−06
LISWET8	10002	10000	acceptable	2.51e+01	3.42e−01	6	1.96e−07	3.62e−05
LISWET9	10002	10000	acceptable	2.53e+01	3.34e−01	6	7.84e−07	6.66e−05
LOTSCHD	12	7	acceptable	2.40e+03	5.08e−02	8	4.00e−11	1.42e−14
MOSARQP1	2500	700	acceptable	−9.53e+02	1.72e−01	9	2.78e−16	2.55e−14
MOSARQP2	900	600	acceptable	−1.60e+03	1.26e−01	8	3.89e−16	1.42e−14
POWELL20	10000	10000	acceptable	5.21e+10	3.05e+00	66	1.25e−07	1.64e−06
PRIMAL1	325	85	acceptable	−3.50e−02	1.18e−01	13	6.37e−13	5.69e−13
PRIMAL2	649	96	acceptable	−3.37e−02	1.21e−01	12	5.38e−16	2.38e−11
PRIMAL3	745	111	acceptable	−1.36e−01	1.39e−01	10	8.84e−12	1.06e−13
PRIMAL4	1489	75	acceptable	−7.46e−01	1.19e−01	9	5.31e−11	1.54e−14
PRIMALC1	230	9	acceptable	−6.16e+03	1.13e−01	16	2.60e−11	4.00e−08
PRIMALC2	231	7	acceptable	−3.55e+03	9.42e−02	13	1.16e−12	6.21e−11
PRIMALC5	287	8	acceptable	−4.27e+02	8.94e−02	13	1.71e−11	4.58e−10
PRIMALC8	520	8	acceptable	−1.83e+04	9.85e−02	13	1.37e−08	5.86e−09
Q25FV47	1571	820	acceptable	1.37e+07	1.04e+00	19	2.30e−08	6.25e−10
QADLITTL	97	56	acceptable	4.80e+05	8.89e−02	11	4.37e−12	7.45e−09
QAFIRO	32	27	acceptable	−1.59e+00	6.91e−02	11	5.68e−14	1.78e−15
QBANDM	472	305	acceptable	1.64e+04	1.33e−01	13	2.27e−08	2.31e−08
QBEACONF	262	173	acceptable	1.65e+05	1.10e−01	12	4.88e−10	1.59e−10
QBRANDY	249	220	acceptable	2.84e+04	1.62e−01	17	1.86e−07	5.42e−09
QE226	282	223	acceptable	2.13e+02	1.53e−01	13	9.59e−12	2.45e−11
QFFFFF80	854	524	acceptable	8.73e+05	5.47e−01	17	3.02e−08	4.96e−05
QGFRDXPN	1092	616	acceptable	1.01e+11	2.03e−01	22	2.35e−06	8.76e−03
QGROW15	645	300	acceptable	−1.02e+08	1.83e−01	17	3.17e−08	4.26e−14
QGROW22	946	440	acceptable	−1.50e+08	3.17e−01	19	5.74e−08	7.11e−14
QGROW7	301	140	acceptable	−4.28e+07	1.42e−01	17	6.42e−09	2.84e−14
QISRAEL	142	174	acceptable	2.53e+07	2.44e−01	15	1.58e−09	8.58e−12
QPCBLEND	83	74	acceptable	−7.84e−03	1.06e−01	15	1.07e−14	1.99e−15
QPCBOEI1	384	351	acceptable	1.15e+07	2.67e−01	23	7.15e−08	3.12e−06
QPCBOEI2	143	166	acceptable	8.17e+06	1.88e−01	20	4.55e−12	2.38e−07
QPTEST	2	2	acceptable	4.37e+00	4.51e−02	7	8.88e−16	6.22e−17
QSC205	203	205	acceptable	−5.81e−03	8.08e−02	11	9.66e−13	1.14e−13
QSCAGR25	500	471	acceptable	2.02e+08	1.31e−01	13	4.10e−09	4.73e−07
QSCAGR7	140	129	acceptable	2.69e+07	1.18e−01	16	5.74e−11	2.57e−10
QSCFXM1	457	330	acceptable	1.69e+07	6.52e−01	42	6.29e−05	1.06e−06
QSCFXM2	914	660	acceptable	2.78e+07	5.56e−01	28	4.84e−06	5.32e−07
QSCFXM3	1371	990	acceptable	3.08e+07	6.53e−01	25	1.37e−06	8.29e−07
QSCORPIO	358	388	acceptable	1.88e+03	1.09e−01	10	1.02e−11	1.30e−10
QSCRS8	1169	490	acceptable	9.05e+02	1.63e−01	14	1.83e−12	4.04e−12
QSCSD1	760	77	acceptable	8.67e+00	8.42e−02	9	5.52e−15	2.66e−15
QSCSD6	1350	147	acceptable	5.08e+01	1.13e−01	11	1.56e−12	4.44e−15
QSCSD8	2750	397	acceptable	9.41e+02	1.39e−01	8	2.52e−11	2.37e−09
QSCTAP1	480	300	acceptable	1.42e+03	1.09e−01	11	7.43e−13	2.80e−10
QSCTAP2	1880	1090	acceptable	1.74e+03	1.89e−01	11	3.55e−11	1.03e−12
QSCTAP3	2480	1480	acceptable	1.44e+03	2.51e−01	12	5.25e−13	1.22e−10
QSEBA	1028	515	acceptable	8.15e+07	3.09e−01	27	1.59e−06	4.84e−05
QSHARE1B	225	117	acceptable	7.20e+05	1.37e−01	15	2.70e−09	5.13e−13
QSHARE2B	79	96	acceptable	1.17e+04	1.54e−01	22	5.52e−11	3.18e−12
QSHIP04L	2118	402	acceptable	2.42e+06	1.44e−01	9	1.02e−06	4.10e−07
QSHIP04S	1458	402	acceptable	2.42e+06	1.05e−01	9	6.81e−07	3.62e−07
QSHIP08L	4283	778	acceptable	2.38e+06	6.14e−01	9	4.70e−08	1.06e−07
QSHIP08S	2387	778	acceptable	2.39e+06	2.82e−01	10	1.10e−09	7.98e−07
QSHIP12L	5427	1151	acceptable	3.02e+06	8.34e−01	11	8.41e−09	5.90e−07
QSHIP12S	2763	1151	acceptable	3.06e+06	4.09e−01	12	1.96e−08	7.30e−07
S268	5	5	acceptable	3.60e−09	6.94e−02	12	1.11e−15	3.64e−12
STADAT1	2001	3999	acceptable	−2.85e+07	2.21e+00	56	7.29e−08	2.24e−08
STADAT2	2001	3999	acceptable	−3.26e+01	4.54e−01	15	1.81e−11	3.56e−09
STADAT3	4001	7999	acceptable	−3.58e+01	7.00e−01	14	5.18e−11	1.55e−11
STCQP1	4097	2052	acceptable	1.55e+05	3.74e−01	7	7.39e−11	1.02e−12
STCQP2	4097	2052	acceptable	2.23e+04	4.97e−01	8	2.65e−09	6.19e−12
TAME	2	1	acceptable	0.00e+00	3.60e−02	5	0.00e+00	0.00e+00
VALUES	202	1	exception	∞	∞	0	∞	∞

Continued on next page

Table 13: Xpress results without presolve and crossover on Maros and Meszaros problems.

name	n	m	status	objective	time	iter	$\|r_b\|_\infty$	$\|r_c\|_\infty$
ZECEVIC2	2	2	acceptable	−4.12e+00	4.81e−02	6	2.12e−16	2.91e−16

References

[1] The GNU Multiple Precision Arithmetic Library. https://gmplib.org/.

[2] Netlib. http://netlib.org/lp/.

[3] A. Altman and J. Gondzio. Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization. Optim. Method Softw., 11(1–4):275–302, 1999. DOI: 10.1080/10556789908805754. URL https://doi.org/10.1080/10556789908805754.

[4] M. F. Anjos, A. Lodi, and M. Tanneau. Tulip: An open-source interior-point linear optimization solver with abstract linear algebra. Cahier du GERAD G–2019–36, GERAD, Montréal, QC, Canada, 2020. URL https://www.gerad.ca/en/papers/G-2019-36.

[5] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. SIAM Rev., 59(1):65–98, 2017. URL https://doi.org/10.1137/141000671.

[6] E. Dolan and J. Moré. Benchmarking optimization software with performance profiles. Math. Program., Series B, 91:201–213, 2002. DOI: 10.1007/s101070100263.

[7] FICO Xpress Optimization. Xpress solver. https://www.fico.com/en/products/fico-xpress-solver.

[8] M. P. Friedlander and D. Orban. A primal-dual regularized interior-point method for convex quadratic programs. Math. Program. Comp., 4(1):71–107, 2012. DOI: 10.1007/s12532-012-0035-2.

[9] A. Ghannad, D. Orban, and M. A. Saunders. A symmetric formulation of the linear system arising in interior methods for convex optimization with bounded condition number. Cahier du GERAD G–2020–37, GERAD, Montréal, QC, Canada, 2020.

[10] J. Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. Comput. Optim. Appl., 6:137–156, 1995. DOI: 10.1007/BF00249643.

[11] C. Greif, E. Moulding, and D. Orban. Bounds on eigenvalues of matrices arising from interior-point methods. SIAM J. Optim., 24:49–83, 2014. DOI: 10.1137/120890600.

[12] Gurobi Optimization. Gurobi Optimizer. https://www.gurobi.com/products/gurobi-optimizer.

[13] A. Haidar, H. Bayraktar, S. Tomov, J. Dongarra, and N. J. Higham. Mixed-precision iterative refinement using tensor cores on gpus to accelerate solution of linear systems. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 476(2243):20200110, 2020. DOI: 10.1098/rspa.2020.0110. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2020.0110.

[14] N. J. Higham, S. Pranesh, and M. Zounon. Squeezing a matrix into half precision, with an application to solving linear systems. SIAM J. Sci. Comput., 41(4):A2536–A2551, 2019. DOI: 10.1137/18M1229511.

[15] IBM. IBM ILOG CPLEX Optimization Studio. https://www.ibm.com/products/ilog-cplex-optimization-studio.

[16] Intel. The Intel Decimal Floating-Point Math Library. https://software.intel.com/content/www/us/en/develop/articles/intel-decimal-floating-point-math-library.html.

[17] G. Leconte and contributors. QuadraticModelsCPLEX.jl: Optimization of QuadraticModels using CPLEX. https://github.com/JuliaSmoothOptimizers/QuadraticModelsCPLEX.jl, December 2020.

[18] G. Leconte and contributors. QuadraticModelsGurobi.jl: Optimization of QuadraticModels using Gurobi. https://github.com/JuliaSmoothOptimizers/QuadraticModelsGurobi.jl, December 2020.

[19] G. Leconte and contributors. QuadraticModelsXpress.jl: Optimization of QuadraticModels using Xpress. https://github.com/JuliaSmoothOptimizers/QuadraticModelsXpress.jl, December 2020.

[20] D. Ma, L. Yang, R. Fleming, I. Thiele, B. Palsson, and M. Saunders. Reliable and efficient solution of genome-scale models of metabolism and macromolecular expression. Scientific Reports, 7(40863), 2017. DOI: 10.1038/srep40863.

[21] I. Maros and C. Mészáros. A repository of convex quadratic programming problems. Optim. Method Softw., 11:671–681, 1999. DOI: 10.1080/10556789908805768.

[22] S. Mehrotra. On the implementation of a primal-dual interior point method. SIAM J. Optim., 2:575–601, 1992. DOI: 10.1137/0802028.

[23] D. Orban and contributors. LDLFactorizations.jl: Factorization of symmetric matrices. https://github.com/JuliaSmoothOptimizers/LDLFactorizations.jl, June 2020.

[24] D. Orban and G. Leconte. RipQP.jl: Regularized interior point solver for quadratic problems. https://github.com/JuliaSmoothOptimizers/RipQP.jl, November 2020.

[25] D. Orban and A. S. Siqueira. JuliaSmoothOptimizers: Infrastructure and solvers for continuous optimization in Julia, 2020. URL https://juliasmoothoptimizers.github.io.

[26] D. Orban, A. S. Siqueira, and contributors. NLPModels.jl: Data structures for optimization models. https://github.com/JuliaSmoothOptimizers/NLPModels.jl, July 2020.

[27] D. Orban, A. S. Siqueira, and contributors. QuadraticModels.jl: Data structures for linear and quadratic optimization problems based on NLPModels.jl. https://github.com/JuliaSmoothOptimizers/QuadraticModels.jl, December 2020.

[28] D. Orban, A. S. Siqueira, and contributors. SolverBenchmark.jl: Benchmark tools for solvers. https://github.com/JuliaSmoothOptimizers/SolverBenchmark.jl, December 2020.

[29] D. Orban, A. S. Siqueira, and contributors. SolverTools.jl: Tools for developing nonlinear optimization solvers. https://github.com/JuliaSmoothOptimizers/SolverTools.jl, December 2020.

[30] D. Ruiz. A scaling algorithm to equilibrate both rows and columns norms in matrices. Technical Report RAL-TR-2001-034, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 2011.

[31] Software Outlook Training. Mixed precision, Unit 3: Single vs double precision: floating-point operations: speed and energy consumption. https://www.softwareoutlook.ac.uk/sites/www.softwareoutlook.ac.uk/files/MixedPrecisionUnit3.pdf, 2020.

[32] The GNU Projet. GNU Fortran. https://gcc.gnu.org/fortran/.

[33] R. J. Vanderbei. Symmetric quasi-definite matrices. SIAM J. Optim., 5:100–113, 1995. DOI: https://doi.org/10.1137/0805005.

[34] T. Weber, S. Sager, and A. Gleixner. Solving quadratic programs to high precision using scaled iterative refinement. Math. Program. Comp., (11):421–455, 2019. DOI: 10.1007/s12532-019-00154-6.

[35] S. J. Wright. Primal-dual interior-point methods. SIAM, 1997.