Report: Multi-Precision Quadratic Regularization Algorithm

D. Monnet, F. Rahbarnia

March 18, 2022

1 Problem Statement and Background

We consider the problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

with f a continuously differentiable function. We suppose that the computations are run in finite precision. As a consequence, it is not possible to evaluate f nor the gradient ∇f exactly, but only their finite precision couterparts denoted $\hat{f}$ and g. We further suppose that models for the errors ω_f and ω_g are provided as functions of x,

$$||f(x) - \hat{f}(x)|| \leq \omega_f(x), \quad ||\nabla f(x) - g(x)|| \leq \omega_g(x)||g(x)||. \tag{1}$$

The following assumption is made.

AS. 1. *The gradient of f is Lipschitz continuous.*

$$\exists L > 0, \forall x, y \in \mathbb{R}^n, ||\nabla f(x) - \nabla f(y)|| \leq L||x - y||. \tag{2}$$

From assubmption AS.1, one derives that f is continuously differentiable

The regularized quadratic method, detailed in Algorithm 1, is based on a first order model of the objective function. Let T be the Taylor expansion of f truncated at order 1,

$$T(x, s) = f(x) + \nabla f(x)^T s. \tag{3}$$

In practice f and $\nabla f(x)$ might be costly to evaluate. Instead of using T directly, we use the approximated Taylor series defined with an inexact gradient g and the inexact evaluation of ,

$$\bar{T}(x, s) = f(x) + g(x)^T s. \tag{4}$$

The regularized approximated Taylor series is used as the local model in the algorithm, defined as,

$$m(x, s) = \bar{T}(x, s) + \frac{1}{2}\sigma||s||^2, \tag{5}$$

with σ the regularization parameter that controls the step size. The step is chosen as the minimizer of the local model given by,

$$s \in \underset{s \in \mathbb{R}^n}{\operatorname{argmin}}\ m(x,s) = \left\{ -\frac{g(x)}{\sigma} \right\}. \tag{6}$$

2 Finite Precision Computations

Finite precision computations induce not only inexact evaluations of the objective function and the gradient, but also other errors that must be taken into account. In this section, several sources of errors due to finite precision computation are detailed. We also introduce some tools and results necessary to proove the convergence of Algorithm 1. The following assumption is made in the rest of the paper.

AS. 2. *Finite precision computations comply with IEEE 754 norm, and underflow and overflow do not occur during algorithm execution.*

We focus on floating point numbers of radix 2. The *machine precision* is

$$u = 2^{1-p}/2 = 2^{-p}$$

with p the precision of the considered floating point representation (*e.g.*16, 32, 64, ... bits). The IEEE norm requires that the standard operations produce a result with a rounding error such that

$$fl\,(x \square y) = (x \square y)(1+\delta), \quad |\delta| \leq u, \tag{7}$$

with x and y two floating point numbers, $\square$ one of the operators $+$, $-$, $*$, $/$, and $fl\,(.)$ denotes the finite precision computation. In the following, δ represents a rounding error bounded as in (7).

The notations used to study the rounding errors propagation are the one introduced in [2],

$$\prod_{i}^{n}(1+\delta_i) = (1+\theta_n), \quad |\theta_n| \leq \gamma_n. \tag{8}$$

Several formulas exist for γ_n, recaped in Table 2. Note that these bounds can be very pessimistic because they account for the worst-case propagation of errors. TODO @Farhad: table update

Table 1: Formula for γ_n

ref			
γ_n	$\frac{n}{1-nu}$	$\frac{n}{1-nu/2}$	nu

In the following, we simplify the notations as follows:

- δ is any rounding error, an index is added if there is a need to keep track of a particular rounding error,
- θ_n models any rounding error propagation. The notation ϑ is used is there is a need to keep track of a particular θ,
- given a vector $x \in \mathbb{R}^n$, we denote abusively,

$$x(1+\delta) = (x_1(1+\delta_1), \ldots, x_n(1+\delta_n)), \quad x\delta = (x_1\delta_1, \ldots, x_n\delta_n)$$

  if there is no need to keep track of the rounding errors δ_i. The same goes for θ_n.

2.1 Inexact Step and Pseudo-Taylor Series

When computing the step s_k at iteration k with finite precision arithmetic, it happens that the descent direction is not $g_k = g(x_k)$ but $\tilde{g}_k$ that includes rounding error. Denoting $\hat{s}_k$ the finite precision step, one has,

$$\hat{s}_k = fl\left(-\frac{g_k}{\sigma_k}\right) = -\frac{\tilde{g}_k}{\sigma_k} \tag{9}$$

with $\tilde{g}_k = g_k \odot ((1+\delta_1), \ldots, (1+\delta_n))^T$, where $\odot$ denotes the element-wise multiplication operator.

Lemma 1. *The norm of the difference between g_k and $\tilde{g}_k$ is bounded as,*

$$||g_k - \tilde{g}_k|| \leq u||g_k||, \tag{10}$$

and $||\tilde{g}_k|| \geq ||g_k||(1-u)$

Proof. Denote $g_k = (g_k^1, \ldots, g_k^n)^T$. One has,

$$\begin{aligned} ||g_k - \tilde{g}_k||^2 &= \sum_{i=1}^{n} (g_k^i - \tilde{g}_k^i)^2 \\ &= \sum_{i=1}^{n} (g_k^i - g_k^i(1+\delta_i))^2 \\ &= \sum_{i=1}^{n} \delta_i^2 (g_k^i)^2 \\ &\leq \sum_{i=1}^{n} u^2 (g_k^i)^2 \\ &= u^2 ||g||^2. \end{aligned} \tag{11}$$

This proves the first inequality. Following a similar reasoning,

$$\begin{aligned} ||\tilde{g}||^2 &= \sum_{i=1}^{n} (g_k^i(1+\delta_i))^2 \\ &\geq (1-u)^2 ||g||^2, \end{aligned} \tag{12}$$

which proves the second inequality. □

We define the *pseudo-Taylor series* as,

$$\tilde{T}(x_k, s) = \hat{f}(x_k) + \tilde{g}^T s. \tag{13}$$

The pseudo-Taylor series can be viewed as the approximated Taylor series $\bar{T}$ but slightly modified, such that the rounding error is taken into account. Indeed, $\tilde{T}(x_k, s)$ is expressed with $\tilde{g}_k$ which is the descent direction of the inexact step $\hat{s}_k$. As such, considering the pseudo-Taylor series $\tilde{T}$ instead of the $\bar{T}$ makes it easier to prove the convergence of the algorithm when considering rounding errors due to finite precision computations (see Section 4).

The main difficulty is that $\tilde{g}$ is unknown, and as a consequence $\tilde{T}$ is also unknown. This is an issue for relying the pseudo-Taylor series since the algorithm requires the decrease to be computed. However, computing the decrease of the approximated Taylor series $\bar{T}$ with finite precision arithmetic provide the decrease of the pseudo-Taylor series with a relative error. This is what is stated by Lemma 2. Note that the notation ϑ_{n+2} is used because it is necessary to keep track of this particular propagation error model in the following.

Lemma 2. *Let $\widehat{\Delta T}_k$ be the finite precision computation of $\bar{T}(x_k, 0) - \bar{T}(x_k, \hat{s}_k)$. One has,*

$$\widehat{\Delta T}_k = \left(\tilde{T}(x, 0) - \tilde{T}(x, \hat{s}_k)\right)(1 + \vartheta_{n+2}) = -\tilde{g}^T \hat{s}_k (1 + \vartheta_{n+2}). \tag{14}$$

Proof.

$$\begin{aligned}
\widehat{\Delta T}_k &= fl\left(-g_k^T \hat{s}_k\right) \\
&= -fl\left(g_k^T \frac{\tilde{g}}{\sigma_k}\right) \\
&= -fl\left(\sum_{i=1}^{n} \frac{(\tilde{g}_k^i)^2(1+\delta)}{\sigma_k(1+\delta_i)}\right) \\
&= -\sum_{i=1}^{n} \frac{(\tilde{g}_k^i)^2(1+\delta)}{\sigma_k(1+\delta_i)}(1+\theta_n) \\
&= -\frac{(\tilde{g}_k^i)^2}{\sigma_k}(1+\vartheta_{n+2}) \\
&= -\tilde{g}^T \hat{s}_k (1+\vartheta_{n+2})
\end{aligned} \tag{15}$$

□

2.2 Actual Candidate and Descent Direction

At each iteration, the candidate c_k is computed as $c_k = x_k + \hat{s}_k$. With finite precision computation, one has

$$\hat{c}_k = fl\left(x_k + \hat{s}_k\right) = (x_k + \hat{s}_k)(1 + \delta). \tag{16}$$

As a consequence, the computed step $\hat{s}_k$ is not the actual step. The actual step is $x_k\delta + \hat{s}_k(1+\delta)$, and therefore the actual descent direction is not $\tilde{g}$ but $\sigma_k(\hat{c}_k - x_k) = \tilde{g}(1+\delta) + \sigma_k x_k \delta$. Since we have no relationship between g_k and

x_k, it is not possible to find a relationship between the actual descent direction and g_k as we did in the previous subsection between g_k and $\tilde{g}_k$.

In order to deal with this inexact cadidate computation, we introduce ϕ_k an upper bound on the ratio $||x_k||/||s_k||$. If ϕ_k is great, then $x\delta$ is not neglectable compare to $\hat{s}_k$ and the difference between g_k and the actual descent direction can be huge. If ϕ_k is too big, then the convergence of the finite precision quadratic regularization algorithm cannot be ensured. This is included in the termination condition in Step 2 in Algorithm 1 since μ_k increases with ϕ_k.

2.3 Finite Precision Norm Computation

In Algorithm 1, it is necessary to compute $||g_k||$, $||s_k||$ and $||x_k||$. Due to rounding errors, norm computations are inexact. Lemma 3 states an upper bound on the error for norm computation.

Lemma 3. *Let $x \in \mathbb{R}^n$ be an n-dimentional floating point vector. The error on norm computation of x in finite precision can be modeled*

$$fl\left(||x||\right) = ||x||(1+\theta_{\lceil \frac{n}{2} \rceil + r}), \tag{17}$$

with $fl\left(\sqrt{y}\right) = \sqrt{y}(1+\theta_r)$, y being a floating point number, and $\lceil . \rceil$ the ceiling function.

Proof. The error on dot product can be modeled as [2],

$$fl\left(|x|.|x|\right) = |x|.|x|(1+\theta_n) = ||x||^2(1+\theta_n). \tag{18}$$

As a consequence,

$$\begin{aligned} fl\left(||x||\right) &= fl\left(\sqrt{||x||^2}\right) \\ &= ||x||\sqrt{(1+\theta_n)}(1+\theta_r) \\ &= ||x||(1+\theta_{\lceil \frac{n}{2} \rceil + r}). \end{aligned} \tag{19}$$

□

3 Finite Precision Algorithm

The algorithm detailed in this section implements strategies to deal with inexact evaluation of f, its gradient and the approximated Taylor serie. In order to make the equations easier to read, we define

$$\alpha_n = \frac{1}{1-\gamma_{n+2}}. \tag{20}$$

AS. 3. *The machine precision is such that $1 - \gamma_{n+2} > 0$.*

Algorithm 1 is the adaptation of the classical quadratic regularization algorithm that takes the objective function and gradient evaluation errors into account. It takes as input the usual parameters η_1, η_2, that decides wether or not an iteration is successful and γ_1, γ_2, γ_3 that controls how the regularization parameter σ_k is updated. It also takes as input the parameter η_0 which controls the objective function evaluation precision to ensure the convergence of the algorithm (Step 3). Condition (21) initialized the parameters such that the convergence is ensured. Algorithm 1 stops either when one of the three stopping criteria (22) is reached:

- Step 1: the termination condition ensures that ensures $||\nabla f(x_k)|| \leq \epsilon$ by taking into account the gradient evaluation error $\omega_g(x_k)$ and the error of norm computation in finite precision modeled in Lemma 3.
- Step 2: the value μ_k has to be lower than κ_μ given in input. μ_k can be viewed as the worst-case relative error on the exact gradient $\nabla f(x_k)$ that takes into account $\omega_g(x_k)$ and the errrors due to inexact step and candidate computations. Note that if we suppose that s_k and c_k are computed exactly, *i.e.* $u = 0$, one has $\mu_k = \omega(x_k)$.
- Step 3: the termination condition ensures that the objective function evaluation error remains small enough to ensure the convergence.

Termination conditions at Steps 2 and 3 are necessary to guaratee the convergence (see Section 4 for the proof of convergence). At Step 2, ϕ_k is a guaranteed upper bound on the ratio $||x_k||/||s_k||$ by taking into account the norm computaion errors as given in Lemma 3. In the following, we suppose that μ_k, ϕ_k and ρ_k are computed exactly, that is why the word "define" instead of "compute" is used in Algorithm 1.

AS. 4. *α_n, γ_n, ϕ_k, μ_k and ρ_k are computed exactly.*

These quantities involve few computations and make this assumption reasonable.

TODO: provide a better analysis for ρ_k. What happen if $f_k \approx f_k^+ >> 1$?

Algorithm 1 Finite precision quadratic regularization algorithm

Step 0: Initialization: Initial point x_0, initial value σ_0, minimal value σ_{min} for σ, final gradient accuracy ϵ, formula for γ_n. Constant values η_0, η_1, η_2, γ_1, γ_2, γ_3, κ_μ such that,

$$0 < \eta_1 \leq \eta_2 < 1 \quad 0 < \gamma_1 < 1 < \gamma_2 \leq \gamma_3, \quad \eta_0 < \frac{1}{2}\eta_1 \quad \eta_0 + \frac{\kappa_\mu}{2} \leq \frac{1}{2}(1-\eta_2) \tag{21}$$

Set $k = 0$, compute $f_0 = \hat{f}(x_0)$.

Step 1: Check for termination: If $k = 0$ or $x_k \neq x_{k-1}$, compute $g_k = g(x_k)$. Compute $\widehat{||g_k||} = fl(||g_k||)$. Terminate if

$$\widehat{||g_k||} \leq \frac{1}{1+\gamma_{\lceil \frac{n}{2} \rceil + r}} \frac{\epsilon}{(1+\omega_g(x_k))} \tag{22}$$

Step 2: Step calculation: Compute $\hat{s}_k = fl\,(g_k/\sigma_k)$.
Compute $\hat{\phi}_k = fl\left(\dfrac{||x_k||}{||s_k||}\right)$. Define $\phi_k = \hat{\phi}_k \dfrac{1+\gamma_{\lceil \frac{n}{2} \rceil + r}}{1-\gamma_{\lceil \frac{n}{2} \rceil + r}}$
Define $\mu_k = \dfrac{\alpha_n \omega_g(x_k)(u+\phi_k u+1) + u(\phi_k \alpha_n + \alpha_n + 1) + \frac{\gamma_{n+2}}{1+\gamma_{n+2}}}{1-u}$.
Terminate if $\mu_k > \kappa_\mu$
Compute $\hat{c}_k = fl\,(x_k + \hat{s}_k)$.
Compute approximated taylor series decrease $\widehat{\Delta T}_k = fl\left(g_k^T \hat{s}_k\right)$.

Step 3: Evaluate the objective function: Compute $f_k^+ = \hat{f}(\hat{c}_k)$. Terminate if $\omega_f(x_k) > \eta_0 \widehat{\Delta T}_k$ or $\omega_f(\hat{c}_k) > \eta_0 \widehat{\Delta T}_k$

Step 4: Acceptance of the trial point: Define the ratio

$$\rho_k = \frac{f_k - f_k^+}{\widehat{\Delta T}_k} \tag{23}$$

If $\rho_k \geq \eta_1$, then $x_{k+1} = x_k + \hat{s}_k$, $f_{k+1} = f_k^+$. Otherwise set $x_{k+1} = x_k$, $f_{k+1} = f_k$.

Step 5: Regularization parameter update:

$$\sigma_{k+1} \in \begin{cases} [\max(\sigma_{min}, \gamma_1 \sigma_k), \sigma_k] & \text{if } \rho_k \geq \eta_2 \\ [\sigma_k, \gamma_2 \sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2) \\ [\gamma_2 \sigma_k, \gamma_3 \sigma_k] & \text{if } \rho_k \in < \eta_1 \end{cases} \tag{24}$$

$k = k + 1$, go to Step 1.

4 Proof of convergence

AS. 5. *There exsits κ_{low} such that,*

$$\forall x \in \mathbb{R}^n,\, f(x) \geq \kappa_{low}. \tag{AS.1}$$

Lemma 4. *For all $k > 0$,*

$$\begin{aligned} |f_k - f(\hat{c}_k) - \widehat{\Delta T}_k| \leq \quad & |f(x_k) - f_k| + \tfrac{1}{2}L||x_k\delta + \hat{s}_k(1+\delta)||^2 \\ & +\delta||\nabla_f(x_k)||\,(||x_k|| + ||\hat{s}_k||) + |\tilde{g}_k^T\hat{s}_k(1+\vartheta_{n+2}) - \nabla_f(x_k)^T\hat{s}_k|. \end{aligned} \tag{25}$$

Proof. With the triangular inequality, one has,

$$|f_k - f(\hat{c}_k) - \widehat{\Delta T}_k| \quad \leq |f_k - f(x_k)| + |f(x_k) - f(\hat{c}_k) - \widehat{\Delta T}_k|. \tag{26}$$

By Lemma 2,

$$\begin{aligned} & |f(x_k) - f(\hat{c}_k) - \widehat{\Delta T}_k| \\ \leq \quad & |f(x_k) - f(\hat{c}_k) + \tilde{g}_k^T\hat{s}_k(1+\vartheta_{n+2})| \\ \leq \quad & |f(x_k) - f(\hat{c}_k) + \nabla f(x_k)^T\hat{s}_k| + |\tilde{g}_k^T\hat{s}_k(1+\vartheta_{n+2}) - \nabla f(x_k)^T\hat{s}_k|. \end{aligned} \tag{27}$$

By triangular inequality,

$$\begin{aligned} & |f(x_k) - f(\hat{c}_k) + \nabla f(x_k)^T\hat{s}_k| \\ \leq \quad & |f(x_k) - f(\hat{c}_k) + \nabla f(x_k)^T(x_k\delta + \hat{s}_k(1+\delta)| + |\nabla f(x_k)^T\hat{s}_k - \nabla f(x_k)^T(x_k\delta + \hat{s}_k(1+\delta)| \end{aligned} \tag{28}$$

Recalling that ([1])

$$|f(x_k + s) - T(x_k, s)| \leq \frac{1}{2}L||s||^2, \tag{29}$$

one has,

$$\begin{aligned} |f(x_k) - f(\hat{c}_k) + \nabla f(x_k)^T(x_k\delta + \hat{s}_k(1+\delta))| \quad & = |T(x_k, x_k\delta + \hat{s}_k(1+\delta)) - f(\hat{c}_k)| \\ & \leq \frac{1}{2}L||x_k\delta + \hat{s}_k(1+\delta)||^2. \end{aligned} \tag{30}$$

By triangular inequality,

$$\begin{aligned} |\nabla f(x_k)^T\hat{s}_k - \nabla f(x_k)^T(x_k\delta + \hat{s}_k(1+\delta))| \quad & = \delta|\nabla f(x_k)^T(x_k + \hat{s}_k)| \\ & \leq \delta||\nabla f(x_k)^T||\,||(x_k + \hat{s}_k)|| \\ & \leq \delta||\nabla f(x_k)^T||\,(||x_k|| + ||\hat{s}_k||). \end{aligned} \tag{31}$$

Injecting Inequalities (31) and (30) into (28),

$$|f(x_k) - f(\hat{c}_k) - \nabla f(x_k)^T\hat{s}_k| \leq \frac{1}{2}L||x_k\delta + \hat{s}_k(1+\delta)||^2 + \delta||\nabla f(x_k)^T||\,(||x_k|| + ||\hat{s}_k||) \tag{32}$$

Putting together Inequalities (26), (27) and (32) provides the inequality stated by Lemma 4 □

Lemma 5. *For all $k > 0$,*

$$\left| \frac{\tilde{g}_k^T \hat{s}_k(1+\vartheta_{n+2}) - \nabla_f(x_k)^T \hat{s}_k}{\widehat{\Delta T}_k} \right| \leq \frac{u + \frac{\gamma_{n+2}}{1+\gamma_{n+2}} + \alpha_n \omega_g(x_k)}{1-u}. \tag{33}$$

Proof. With Lemma 2 and (9),

$$\begin{aligned} \widehat{\Delta T}_k &= -\tilde{g}_k^T \hat{s}_k(1+\vartheta_{n+2}) \\ &= \frac{||\tilde{g}_k||^2}{\sigma_k}(1+\vartheta_{n+2}), \end{aligned}$$

As a consequence,

$$\left| \frac{\tilde{g}_k^T \hat{s}_k(1+\vartheta_{n+2}) - \nabla_f(x_k)^T \hat{s}_k}{\widehat{\Delta T}_k} \right| \leq \frac{||\tilde{g}_k(1+\vartheta_{n+2}) - \nabla f(x_k)||\,||\hat{s}_k||}{||\tilde{g}_k||^2/\sigma_k(1+\vartheta_{n+2})} \tag{34}$$

Since $||\hat{s}_k|| = ||\tilde{g}_k||/\sigma_k$,

$$\begin{aligned} \left| \frac{\tilde{g}_k^T \hat{s}_k(1+\vartheta_{n+2}) - \nabla_f(x_k)^T \hat{s}_k}{\widehat{\Delta T}_k} \right| &\leq \frac{||\tilde{g}_k(1+\vartheta_{n+2}) - \nabla f(x_k)||}{||\tilde{g}_k||(1+\vartheta_{n+2})} \\ &\leq \frac{||\tilde{g}_k(1+\vartheta_{n+2}) - g_k||}{||\tilde{g}_k||(1+\vartheta_{n+2})} + \frac{||g_k - \nabla f(x_k)||}{||\tilde{g}_k||(1+\vartheta_{n+2})} \\ &\leq \frac{||\tilde{g}_k(1+\vartheta_{n+2}) - g_k||}{||\tilde{g}_k||(1+\vartheta_{n+2})} + \alpha_n \frac{||g_k - \nabla f(x_k)||}{||\tilde{g}_k||}. \end{aligned} \tag{35}$$

By Lemma 1,

$$\begin{aligned} \alpha_n \frac{||g_k - \nabla f(x_k)||}{||\tilde{g}_k||} &\leq \alpha_n \frac{\omega_g(x_k)||g_k||}{||\tilde{g}_k||} \\ &\leq \alpha_n \frac{\omega_g(x_k)}{1-u}. \end{aligned} \tag{36}$$

Furthermore, because ϑ_{n+2} is a particular rounding error propagation model,

$$\begin{aligned} \frac{||\tilde{g}_k(1+\vartheta_{n+2}) - g_k||}{||\tilde{g}_k||(1+\vartheta_{n+2})} &\leq \frac{||\tilde{g}_k - g_k||}{||\tilde{g}_k||} + \frac{||g_k(1+\vartheta_{n+2}) - g_k||}{||\tilde{g}_k||(1+\vartheta_{n+2})} \\ &= \frac{||\tilde{g}_k - g_k||}{||\tilde{g}_k||} + \frac{||g_k||\vartheta_{n+2}}{||\tilde{g}_k||(1+\vartheta_{n+2})} \\ &\leq \frac{||\tilde{g}_k - g_k||}{||\tilde{g}_k||} + \frac{\gamma_{n+2}}{1+\gamma_{n+2}} \frac{||g_k||}{||\tilde{g}_k||}. \end{aligned} \tag{37}$$

By Lemma 1, it follows that,

$$\frac{||\tilde{g}_k(1+\vartheta_{n+2}) - g_k||}{||\tilde{g}_k||(1+\vartheta_{n+2})} \leq \frac{u + \frac{\gamma_{n+2}}{1+\gamma_{n+2}}}{1-u}. \tag{38}$$

Putting Inequalities (36) and (38) in Inequality (35), one obtains the inequality stated by Lemma 5 □

Lemma 6. *For all $k > 0$,*

$$\frac{1}{\sigma_k} \leq \left[1 - \eta_2 - \eta_0 - \frac{\kappa_\mu}{2}\right] \frac{1}{\alpha_n L(1+\lambda_k)} \implies \rho_k \geq \eta_2, \tag{39}$$

with $\lambda_k = u^2\phi_k^2 + u^2\phi_k + u\phi_k + 2u + u^2$.

Proof. By triangular inequality,

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{f_k - f_k^+ - \widehat{\Delta T}_k}{\widehat{\Delta T}_k} \right| \\ &\leq \left| \frac{f((x_k + \hat{s}_k)(1+\delta)) - f_k^+}{\widehat{\Delta T}_k} \right| + \left| \frac{f_k - f((x_k + \hat{s}_k)(1+\delta)) - \widehat{\Delta T}_k}{\widehat{\Delta T}_k} \right| \end{aligned} \tag{40}$$

The termination condition at Step 3 ensures,

$$\left| \frac{f((x_k + \hat{s}_k)(1+\delta)) - f_k^+}{\widehat{\Delta T}_k} \right| \leq \eta_0. \tag{41}$$

By Lemma 4 and Lemma 5,

$$\begin{aligned} &\left| \frac{f_k - f((x_k + \hat{s}_k)(1+\delta)) - \widehat{\Delta T}_k}{\widehat{\Delta T}_k} \right| \\ \leq\ & \eta_0 + \frac{\frac{1}{2}L||x_k\delta + \hat{s}_k(1+\delta)||^2}{\widehat{\Delta T}_k} + \frac{\delta||\nabla_f(x_k)||\,(||x_k|| + ||\hat{s}_k||)}{\widehat{\Delta T}_k} + \frac{u + \frac{\gamma_{n+2}}{1+\gamma_{n+2}} + \alpha_n\omega_g(x_k)}{1-u}. \end{aligned} \tag{42}$$

By Lemma 2, and recalling that $||\hat{s}_k|| = ||\tilde{g}_k||/\sigma_k$ and $\widehat{\Delta T}_k = ||\tilde{g}_k||^2/\sigma_k$,

$$\begin{aligned} \frac{\frac{1}{2}L||x_k\delta + \hat{s}_k(1+\delta)||^2}{\widehat{\Delta T}_k} &= \frac{\frac{1}{2}L||x_k\delta + \hat{s}_k(1+\delta)||^2}{||\tilde{g}_k||^2/\sigma_k(1+\vartheta_{n+2})} \\ &\leq \alpha_n \frac{\frac{1}{2}L||x_k\delta + \hat{s}_k(1+\delta)||^2}{||\tilde{g}_k||^2/\sigma_k} \\ &= \alpha_n \frac{\frac{1}{2}L(||x_k\delta|| + ||\hat{s}_k(1+\delta)||)^2}{||\tilde{g}_k||^2/\sigma_k}, \end{aligned} \tag{43}$$

and since the mechanism at Step 2 ensures $||x_k|| \leq \phi_k||\hat{s}_k||$, one further has,

$$\begin{aligned} \frac{\frac{1}{2}L||x_k\delta + \hat{s}_k(1+\delta)||^2}{\widehat{\Delta T}_k} &\leq \alpha_n \frac{\frac{1}{2}L\left(\delta\phi_k||\hat{s}_k|| + (1+\delta)||\hat{s}_k||\right)^2}{||\tilde{g}_k||^2/\sigma_k} \\ &\leq \alpha_n \frac{\frac{1}{2}L||\hat{s}_k||^2(\delta^2\phi_k^2 + \delta\phi_k + \delta^2\phi_k + 1 + \delta + \delta^2)}{||\tilde{g}_k||^2/\sigma_k} \\ &\leq \alpha_n \frac{\frac{1}{2}L(1+\lambda_k)}{\sigma_k}, \end{aligned} \tag{44}$$

with λ_k as given in the Statement of the Lemma.

Furthermore, having $||x_k|| \leq \phi_k ||\hat{s}_k||$ ensures,

$$\begin{aligned} \frac{\delta||\nabla_f(x_k)||\,(||x_k|| + ||\hat{s}_k||)}{\widehat{\Delta T}_k} &\leq \alpha_n \frac{\delta||\nabla_f(x_k)||\,(\phi_k||\hat{s}_k|| + ||\hat{s}_k||)}{||\tilde{g}_k||^2/\sigma_k} \\ &= \alpha_n \frac{\delta||\nabla_f(x_k)||\,(\phi_k + 1)}{||\tilde{g}_k||}, \end{aligned} \tag{45}$$

and by Lemma 1 and triangular inequality,

$$\begin{aligned} \frac{\delta||\nabla_f(x_k)||\,(||x_k|| + ||\hat{s}_k||)}{\widehat{\Delta T}_k} &\leq \alpha_n \delta(\phi_k + 1)\left(\frac{||\nabla_f(x_k) - g_k||}{||\tilde{g}_k||} + \frac{||g_k||}{||\tilde{g}_k||}\right) \\ &\leq \alpha_n u(\phi_k + 1)\left(\frac{\omega_g(x_k)}{1-u} + \frac{1}{1-u}\right). \end{aligned} \tag{46}$$

Putting Inequalities (42), (44) and (46) together,

$$\begin{aligned} &\frac{\frac{1}{2}L||x_k\delta + \hat{s}_k(1+\delta)||^2}{\widehat{\Delta T}_k} \\ \leq\ & \eta_0 + \alpha_n \frac{\frac{1}{2}L(1+\lambda_k)}{\sigma_k} + \alpha_n u(\phi_k + 1)\left(\frac{\omega_g(x_k) + 1}{1-u}\right) + \frac{u + \frac{\gamma_{n+2}}{1+\gamma_{n+2}} + \alpha_n \omega_g(x_k)}{1-u}. \end{aligned} \tag{47}$$

Putting Inequalities (40), (41) and (47) together,

$$|\rho_k - 1| \leq 2\eta_0 + \alpha_n \frac{\frac{1}{2}L(1+\lambda_k)}{\sigma_k} + \frac{\alpha_n \omega_g(x_k)(u + \phi_k u + 1) + u(\phi_k \alpha_n + \alpha_n + 1) + \frac{\gamma_{n+2}}{1+\gamma_{n+2}}}{1-u} \tag{48}$$

Step 2 of Algorithm 1 ensures that

$$\mu_k = \frac{\alpha_n \omega_g(x_k)(u + \phi_k u + 1) + u(\phi_k \alpha_n + \alpha_n + 1) + \frac{\gamma_{n+2}}{1+\gamma_{n+2}}}{1-u} \leq \kappa_\mu,$$

and it follows that,

$$|\rho_k - 1| \leq 2\eta_0 + \kappa_\mu + \alpha_n \frac{\frac{1}{2}L(1+\lambda_k)}{\sigma_k}. \tag{49}$$

Therefore, with $1/\sigma_k \leq [1 - \eta_2 - 2\eta_0 - \kappa_\mu]\,\dfrac{1}{\alpha_n L(1+\lambda_k)}$, the inequality rewrites

$$|\rho - 1| \leq \tfrac{1}{2}(1 - \eta_2) + \eta_0 + \frac{\kappa_\mu}{2} \tag{50}$$

and since the parameters are chosen such that $\eta_0 + \dfrac{\kappa_\mu}{2} \leq \tfrac{1}{2}(1-\eta_2)$, it follows that $|\rho - 1| \leq 1 - \eta_2$. □

Lemma 7. *For all $k > 0$,*

$$\sigma_k \leq \sigma_{max} = \frac{\gamma_3 L(1+\lambda_{max})\alpha_n}{1 - \eta_2 - \eta_0 - \frac{\kappa_\mu}{2}}, \tag{51}$$

with

$$\lambda_{max} = \left((1-u)\frac{\kappa_\mu}{u}\right)^2 + (1-u)\frac{\kappa_\mu}{u} + u.$$

Proof. The termination condition at Step 2 ensures that

$$u(\phi_k \alpha_n + \alpha_n + 1) \leq \kappa_\mu (1 - u). \tag{52}$$

It follows that

$$u\phi_k + u \leq \frac{\kappa_\mu}{\alpha_n}(1 - u). \tag{53}$$

Rewritting

$$\lambda_k = (u\phi_k + u)^2 + u\phi_k + u + u, \tag{54}$$

one has, for all $k > 0$, $\lambda_k \leq \lambda_{max}$. The inequality stated by the lemma is straightforward from Lemma 6 and (24). □

Let $S_k = \{0 \leq j \leq k \,|\, \rho_j \geq \eta_1\}$ be the set of successful iterations and $U_k = \{0 \leq j \leq k \,|\, \rho_j < \eta_1\}$ be the set of unsuccessful iterations.

Lemma 8. *For all $k > 0$,*

$$k \leq |S_k| \left(1 + \frac{|\log(\gamma_1)|}{\log(\gamma_2)}\right) + \frac{1}{\log(\gamma_2)} \log\left(\frac{\sigma_{max}}{\sigma_0}\right). \tag{55}$$

Proof. From the update formula of σ_k (24), one has for each $k > 0$,

$$\forall j \in S_k,\ \gamma_1 \sigma_j \leq \max\left[\gamma_1 \sigma_j, \sigma_{min}\right] \text{ and } \forall i \in U_k,\ \gamma_2 \sigma_i \leq \gamma_{i+1}.$$

It follows that,

$$\sigma_0 \gamma_1^{|S_k|} \gamma_2^{|U_k|} \leq \sigma_k. \tag{56}$$

With Lemma 7 one obtains,

$$|S_k| \log \gamma_1 + |U_k| \log \gamma_2 \leq \log\left(\frac{\sigma_{max}}{\sigma_0}\right). \tag{57}$$

Since $\gamma_2 > 1$, it follows that,

$$|U_k| \leq -|S_k| \frac{\log(\gamma_1)}{\log(\gamma_2)} + \frac{1}{\log(\gamma_2)} \log\left(\frac{\sigma_{max}}{\sigma_0}\right). \tag{58}$$

Since $k = |S_k| + |U_k|$, the statement of Lemma 8 holds true. □

Theorem 1. *If the stopping conditions at Steps 2 and 3 are not met, Algorithm 1 needs at most*

$$\epsilon^{-2} \kappa_s (f(x_0) - \kappa_{low}),$$

successful iterations, with

$$\kappa_s = \alpha_n \sigma_{max} (1 + \kappa_\mu)^2 \frac{1}{\eta_1 - 2\eta_0} \left(\frac{1 + \gamma_n}{1 - \gamma_{n+1}}\right)^2, \tag{59}$$

and at most

$$\epsilon^{-2} \kappa_s (f(x_0) - \kappa_{low}) \left(1 + \frac{|\log(\gamma_1)|}{\log(\gamma_2)}\right) + \frac{1}{\log(\gamma_2)} \log\left(\frac{\sigma_{max}}{\sigma_0}\right)$$

iterations to provide an iterate x_k such that $\nabla f(x_k) \leq \epsilon$.

Proof. One has,

$$\begin{aligned} f(x_0) - \kappa_{low} &\geq (\eta_1 - 2\eta_0) \sum_{j \in S_k} \widehat{\Delta T}_j \\ &\geq (\eta_1 - 2\eta_0)(1 - \gamma_{n+2}) \sum_{j \in S_k} \tilde{T}(x_j, 0) - \tilde{T}(x_j, \hat{s}_j) \\ &\geq (\eta_1 - 2\eta_0)(1 - \gamma_{n+2}) \sum_{j \in S_k} ||\tilde{g}_j||^2 / \sigma_j \end{aligned} \tag{60}$$

From the termination condition at Step 1, before termination $||g_k||$ can be lower bounded as,

$$\widehat{||g_k||} = ||g_k||(1 + \theta_n) > (1 - \gamma_n)\frac{\epsilon}{1 + \omega_g(x_k)} \implies ||g_k|| > \frac{1 - \gamma_n}{1 + \gamma_n}\frac{\epsilon}{1 + \omega_g(x_k)}, \tag{61}$$

and by Lemma 1 one has $||\tilde{g}_k|| \geq ||g_k||(1 - u)$. Furthermore,

$$\omega_g(x_k) \leq \kappa_\mu. \tag{62}$$

It follows that,

$$f(x_0) - \kappa_{low} \geq (\eta_1 - 2\eta_0)(1 - \gamma_{n+2})|S_k| \left(\frac{(1 - \gamma_n)(1 - u)}{1 + \gamma_n}\right)^2 \frac{\epsilon^2}{(1 + \kappa_\mu)^2 \sigma_{max}}. \tag{63}$$

This proves the first statement of Theorem 1. Using the above inequality with Lemma 8 proves the second statement. □

Lemma 9. *For all $k > 0$, the bound on objective function evaluation error required at Step 3 to ensure convergence is lower bounded as,*

$$\eta_0 \widehat{\Delta T}_k > \eta_0 \left(\frac{1 - \gamma_{n+1}}{1 + \gamma_n}\frac{\epsilon}{1 + \kappa_\mu}\right)^2 \frac{1}{\alpha_n \sigma_{max}}. \tag{64}$$

Proof. The stopping criterion at Step 1 ensure $||g_k|| > \dfrac{1 - \gamma_n}{1 + \gamma_n}\dfrac{\epsilon}{1 + \kappa_\mu}$. As a consequence, for all $k > 0$, one has,

$$\begin{aligned} \widehat{\Delta T}_k &= (1 + \vartheta_{n+2})\frac{||\tilde{g}_k||^2}{\sigma_k} \\ &\geq \frac{||g_k||^2(1 - u)^2}{\alpha_n \sigma_k} \\ &\geq \frac{||g_k||^2(1 - u)^2}{\alpha_n \sigma_{max}} \\ &> \left(\frac{\epsilon}{1 + \kappa_\mu}\frac{1 - \gamma_{n+1}}{1 + \gamma_n}\right)^2 \frac{1}{\alpha_n \sigma_{max}} \end{aligned}$$

The mechanism at Step 3 enforces $\omega_f(x_k) \leq \eta_0 \widehat{\Delta T}_k$. With the above equation, one has,

$$\eta_0 \widehat{\Delta T}_k > \eta_0 \left(\frac{\epsilon}{1 + \kappa_\mu}\frac{1 - \gamma_{n+1}}{1 + \gamma_n}\right)^2 \frac{1}{\alpha_n \sigma_{max}}.$$

□

5 Multi-Precision Algorithm

In the previous sections, it was supposed that every computation was performed with the same finite precision level. Suppose that the computations can be done with several precision levels (*e.g.* half, single, double precision). In order to save time and energy, one would want to perform the computations with the lowest precision level that still ensures convergence. The idea of multi-precision is to adapt at every iteration the precision level of the computations to do so.

In the following, a precision level is denoted π. We suppose that there are several precision levels available $\{\pi_1, ..., \pi_m\}$, with π_1 the lowest precision level, *e.g.*, π_1 is 16 bits, π_2 is 32 bits and π_3 is 64 bits. The inexact functions $\hat{f}$ and g and the error functions ω are extended as,

$$||f(x) - \hat{f}(x,\pi)|| \leq \omega_f(x,\pi), \quad ||\nabla f(x) - g(x,\pi)|| \leq \omega_g(x,\pi)||g(x,\pi)||. \quad (65)$$

The precision level is added as a second argument to indicate the precision level with which is performed the computations. We also denote $fl\,(x,\pi)$ the operation of casting x from its initial precision level into the precision level π. We suppose that the operation $fl\,(x,\pi)$ is implicitly performed before evaluating $\hat{f}(x,\pi)$ or $g(x,\pi)$ if the precision level of x is different than π.

In order to limit rounding errors to occur, we suppose that γ_1 and γ_3 are powers of 2, positive and negative, respectively, and σ_{min} and σ_0 can be represented in the lowest precision available. This ensure that σ_k is representable in any precision and avoids rounding errors due to casting in a different precision. As a consequence, we can safely ignore the precision level of σ_k in the following.

One can choose to perform each computations in Algorithm 1 with a different precision level. However, doing so would lead to complex convergence analysis. As a consequence, we limit our study to five different levels of precision:

- π_k^x is the precision level of x_k at iteration k,
- π_k^g is the precision level for gradient evaluation at iteration k,
- π_k^f is the precision level for objective function evaluation at $\hat{c}_k$ at iteration k,
- $\pi_k^{f^-}$ is the precision level for objective function evaluation at x_k at iteration k,
- π_k^c is the precision level of the candidate $\hat{c}_k$ at iteration k.

An important remark is that evaluating the objective function or the gradient can be done only with a precision level greater or equal to the one of their arguments. For example, if x_k is represented in 32 bits and $g(x_k)$ is evaluated in 16 bits, it implies that x_k is casted into 16 bits, a rounding error occurs and therefore g is not evaluated at x_k but at close though different point, which is not acceptable. If the evaluation of g is done in a higher precision than x_k, it is harmless since no rounding error occurs when casting x_k to a higher precision

(*e.g.* 64 bits). Following this remark, Algorithm 2, the multi-precision extension of Algorithm 1, works as follows.

At Step 1 and 2, we enforce for simplicity that the computations are performed with the precision π_k^g. As a consequence, at iteration k, γ_n and α_n depend on the machine precision related to π_k^g, denoted u_k^g. In the multi-precision context, γ_n^k and α_n^k are denoted with the superscript k to indicate this dependency to the iterate k. Note also that the precision level of $\hat{s}_k$ computed at Step 2 is π_k^g. Step 1 remains unchanged compare to Algorithm 1. Step 2 however must be adapted to take into account the different precision levels. First, the precision level update mechanism in Step 4 (detailed below) ensures that π_k^g is greater than the precision level of x_k. Secondly, Algorithm 2 allows $\pi_k^g \neq \pi_k^c$, and it might be necessary to cast $\hat{c}_k$ into π_k^c. It follows that, after the casting operation,

$$fl\left(\hat{c}_k, \pi_k^c\right) = \hat{c}_k(1+\delta_k^c) = c_k(1+\delta_k^g)(1+\delta_k^c) = c_k(1+\tilde{\delta}_k),$$

with δ_k^g the error due to addition with the precision π_k^g, δ_k^c the rounding error due to casting from precision π_k^g into π_k^c, and $\tilde{\delta}_k = \delta_k^g + \delta_k^c + \delta_k^g\delta_k^c$. Note that $\tilde{\delta}_k = \delta_k^g$ if $\pi_k^c > \pi_k^g$ since no casting error occurs. Due to the rounding error $\tilde{\delta}_k$, that mixes errors related to π_k^g and π_k^c, μ_k is redefined as its counterpart $\tilde{\mu}_k$ defined with $\tilde{u}_k = u_k^g + u_k^g + u_k^g u_k^c$, with u_k^c the machine precision related to π_k^c. Step 2 enforces that $\tilde{\mu}_k \leq \kappa_\mu$ by either increasing π_k^c or increasing π_k^g to reduce $\omega_g(x_k)$.

At Step 3, Algorithm 2 chooses π_k^f such that the objective function evaluation error at $\hat{c}_k$ is lower than $\eta_0\widehat{\Delta T}_k$ to ensure convergence. In addition, it might be necessary to recompute the objective function evaluation at x_k (done at the previous iteration) with a precision level $\pi_k^{f^-} > \pi_{k-1}^f$ to ensure that the evaluation error is small enough to ensure convergence. Note that π_k^f and $\pi_k^{f^-}$ are chosen greater than π_k^c which avoids rounding errors due to implicit casting. As such, π_k^c is the lowest possible precision level for objective function evaluation.

Step 4 remains similar than in Algorithm 1, except that the precision levels are updated. π_{k+1}^g is chosen greater or equal than π_k^x to avoid rounding error due to casting of x_k into a smaller precision (see remark above). π_{k+1}^c can be chosen freely, and enables to lower the evaluation precisions. Indeed, if one simply compute $\hat{c}_k = fl\left(x_k + s_k\right)$, the precision level of $\hat{c}_k$ is $\pi_k^g \geq \pi_k^x$. As a consequence, $\pi_k^{x+1} \geq \pi_k^x$ and the precision levels π_k^g, π_k^f and $\pi_k^{f^-}$ can only increase over the iterations. Allowing to choose π_k^c freely, and in particular lower than π_x^k, allows to decrease the percision levels for evaluating f and g. Note that $\pi_k^f \leq \pi_k^c$ and if iteration k is successful, then π_{k+1}^g, $\pi_{k+1}^{f^-} \leq \pi_k^c$.

Algorithm 2 MR2: Multi-precision quadratic regularization algorithm

Step 0: Initialization: Initial point x_0, initial value σ_0, minimal value σ_{min} for σ, final gradient accuracy ϵ, formula for γ_n. Constant values η_0, η_1, η_2, γ_1, γ_2, γ_3, κ_μ such that,

$$0 < \eta_1 \leq \eta_2 < 1 \quad 0 < \gamma_1 < 1 < \gamma_2 \leq \gamma_3, \quad \eta_0 < \frac{1}{2}\eta_1 \quad \eta_0 + \frac{\kappa_\mu}{2} \leq \frac{1}{2}(1-\eta_2) \tag{66}$$

Set $k = 0$, compute $f_0 = \hat{f}(x_0)$. Set π_0^c and π_0^g as the precision of x_0.

Step 1: Check for termination: If $k = 0$ or $x_k \neq x_{k-1}$, compute $g_k = g(x_k, \pi_k^g)$. Compute $\widehat{||g_k||} = fl(||g_k||)$. Terminate if

$$\widehat{||g_k||} \leq \frac{(1-\gamma_{\lceil \frac{n}{2} \rceil+1}^k)\epsilon}{(1+\omega_g(x_k, \pi_k^g))} \tag{67}$$

Step 2: Step calculation: Compute $\hat{s}_k = fl\,(g_k/\sigma_k)$.

Compute $\hat{\phi}_k = fl\left(\frac{||x_k||}{||s_k||}\right)$. Define $\phi_k = \hat{\phi}_k \frac{1+\gamma_{\lceil \frac{n}{2} \rceil+1}^k}{1-\gamma_{\lceil \frac{n}{2} \rceil+1}^k}$. Define $\tilde{u}_k = u_k^g + u_k^c + u_k^g u_k^c$.

Define $\tilde{\mu}_k = \dfrac{\alpha_n^k \omega_g(x_k)(\tilde{u}_k + \phi_k \tilde{u}_k + 1) + \tilde{u}_k(\phi_k \alpha_n^k + \alpha_n^k + 1) + \frac{\gamma_{n+2}^k}{1+\gamma_{n+2}^k}}{1-\tilde{u}_k}$.

If $\tilde{\mu}_k > \kappa_\mu$: increase π_k^g and go to Step 1 or increase π_k^c. Terminates if no available precision levels (π_k^c, π_k^g) ensure the inequality.

Compute $\hat{c}_k = fl\,(x_k + \hat{c}_k)$.

Cast $\hat{c}_k$ into π_k^c.

Compute approximated taylor series decrease $\widehat{\Delta T}_k = fl\left(g_k^T \hat{s}_k\right)$.

Step 3: Evaluate the objective function: Choose $\pi_k^f \geq \pi_k^c$ such that $\omega_f(\hat{c}_k) \leq \eta_0 \widehat{\Delta T}_k$. Compute $f_k^+ = \hat{f}(\hat{c}_k, \pi_k^f)$. Terminates if such precision is not available.

If $\omega_f(x_k, \pi_{k-1}^f) > \eta_0 \widehat{\Delta T}_k$, choose $\pi_k^{f^-} > \pi_{k-1}^f$ such that $\omega_f(x_k, \pi_k^{f^-}) \leq \eta_0 \widehat{\Delta T}_k$. Re-compute $f_k = \hat{f}(x_k, \pi_k^{f^-})$. Terminates if such precsion is not available.

Step 4: Acceptance of the trial point: Define the ratio

$$\rho_k = \frac{f_k - f_k^+}{\widehat{\Delta T}_k} \tag{68}$$

If $\rho_k \geq \eta_1$, then $x_{k+1} = \hat{c}_k$, $f_{k+1} = f_k^+$. Otherwise set $x_{k+1} = x_k$, $f_{k+1} = f_k$. Select $\pi_{k+1}^g \geq \pi_k^x$, select π_{k+1}^c.

Step 5: Regularization parameter update:

$$\sigma_{k+1} \in \begin{cases} [\max(\sigma_{min}, \gamma_1 \sigma_k), \sigma_k] & \text{if } \rho_k \geq \eta_2 \\ [\sigma_k, \gamma_2 \sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2) \\ [\gamma_2 \sigma_k, \gamma_3 \sigma_k] & \text{if } \rho_k \in < \eta_1 \end{cases} \tag{69}$$

$k = k + 1$, go to Step 1.

6 Proof of convergence of MR2

Lemma 10. *For all $k > 0$,*

$$\frac{1}{\sigma_k} \leq \left[1 - \eta_2 - \eta_0 - \frac{\kappa_\mu}{2}\right] \frac{1}{\alpha_n^k L(1+\tilde{\lambda}_k)} \implies \rho_k \geq \eta_2, \tag{70}$$

with $\tilde{\lambda}_k = \tilde{u}_k^2\phi_k^2 + \tilde{u}_k^2\phi_k + \tilde{u}_k\phi_k + 2\tilde{u}_k + \tilde{u}_k^2$.

Proof. Same proof as for Lemma 6 by replacing δ by $\tilde{\delta}_k$. □

Lemma 11. *For all $k > 0$,*

$$\sigma_k \leq \tilde{\sigma}_{max} = \frac{\gamma_3 \alpha_n^{max} L(1+\tilde{\lambda}_{max})}{1 - \eta_2 - \eta_0 - \frac{\kappa_\mu}{2}}, \tag{71}$$

with α_n^{max} being α_n defined with π_1 the lowest precision level,

$$\tilde{\lambda}_{max} = \left((1-\tilde{u}_{max})\frac{\kappa_\mu}{\tilde{u}_{max}}\right)^2 + (1-\tilde{u}_{max})\frac{\kappa_\mu}{\tilde{u}_{max}} + \tilde{u}_{max}.$$

$$\tilde{u}_{max} = 2u_{max} + u_{max}^2,$$

and u_{max} the machine precision related to the lowest precision level.

Proof. One has, for all k, $\alpha_n^k \leq \alpha_n^{max}$ and $\lambda_{max} \leq \lambda_k$ (see proof of Lemma 7 for details). The inequality stated by the Lemma is straightforward from Lemma 10 and (69). □

Theorem 2. *If the stopping conditions at Steps 2 and 3 are not met, Algorithm 2 needs at most*

$$\epsilon^{-2}\kappa_s(f(x_0) - \kappa_{low}),$$

successful iterations, with

$$\kappa_s = \alpha_n^{max}\tilde{\sigma}_{max}(1+\kappa_\mu)^2 \frac{1}{\eta_1 - 2\eta_0}\left(\frac{1+\gamma_n^{max}}{1-\gamma_{n+1}^{max}}\right)^2, \tag{72}$$

γ_n^{max} being γ_n defined with the machine precision of π_1, and at most

$$\epsilon^{-2}\kappa_s(f(x_0) - \kappa_{low})\left(1 + \frac{|\log(\gamma_1)|}{\log(\gamma_2)}\right) + \frac{1}{\log(\gamma_2)}\log\left(\frac{\tilde{\sigma}_{max}}{\sigma_0}\right)$$

iterations to provide an iterate x_k such that $\nabla f(x_k) \leq \epsilon$.

7 Numerical Results

TODO

References

[1] Ernesto G Birgin, JL Gardenghi, José Mario Martínez, Sandra Augusta Santos, and Ph L Toint. Worst-case evaluation complexity for unconstrained nonlinear optimization using high-order regularized models. *Mathematical Programming*, 163(1):359–368, 2017.

[2] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.