A Newton-type Incremental Method with a Superlinear Convergence Rate*

Anton Rodomanov
Higher School of Economics
Moscow, Russia
anton.rodomanov@gmail.com

Dmitry Kropotov
Lomonosov Moscow State University
Moscow, Russia
dmitry.kropotov@gmail.com

Abstract

We consider the problem of optimizing the strongly convex sum of a finite number of convex functions. Standard algorithms for solving this problem in the class of incremental/stochastic methods have at most a linear convergence rate. We propose a new incremental method whose convergence rate is superlinear—the Newton-type incremental method (NIM). The idea of the method is to introduce an overall quadratic model of the objective with the same sum-of-functions structure and further update a single component per iteration. We prove that NIM has a superlinear local convergence rate and linear global convergence rate. Experiments show that the method is very effective for problems with a large number of functions and a small number of variables.

1 Introduction

In this paper we consider the following strongly convex unconstrained optimization problem:

$$\min_{x \in \mathbf{R}^d} \left[f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) + \frac{\mu}{2} \|x\|_2^2 \right], \tag{1}$$

where $f_i : \mathbf{R}^d \to \mathbf{R}$, $i = 1, \dots, n$, are twice continuously differentiable convex functions and $\mu > 0$. A typical example of such a problem is ℓ_2-regularized empirical risk minimization for training a machine learning algorithm. In this case the variables x are the parameters of the model and the value $f_i(x)$ measures the error of the model on the ith training sample. Since the objective f is strongly convex, it has the unique minimizer which we denote by x^*.

Context. We consider the case when the number of functions n may be very large. In this situation for minimizing f it is convenient to use incremental optimization methods [1] since the complexity of their iteration does not depend on n. Unlike classic optimization methods which operate on all the n functions at every iteration, incremental methods operate only on a single component f_i at every iteration. If each incremental iteration tends to make reasonable progress in some "average" sense, then an incremental method may significantly outperform its non-incremental counterpart [1].

Contributions. In this work we propose a new incremental optimization method that has a fast *superlinear* local convergence rate—the Newton-type incremental method (NIM). To the best of our knowledge, this is the first method in the class of incremental methods whose convergence rate is superlinear. We provide a theoretical analysis of the convergence of NIM (both local and global) and show in particular that NIM accelerates to a quadratic rate w.r.t. epochs.

*This research was financially supported by RFBR grant #15-31-20596mol-a-ved, Microsoft Research, research initiative: Computer vision collaborative research in Russia, Skoltech SDP Initiative, applications A1 and A2.

Related work. Our work is related to the literature on incremental optimization methods. Incremental optimization methods are an actively developing research area [5, 3, 15]. They all can be divided into two groups depending on their convergence rate.

The first group contains the stochastic gradient method (SGD) and other methods with iterations of the form, $x_{k+1} = x_k - \alpha_k B_k(\nabla f_{i_k}(x_k) + \mu x_k)$, where B_k is some scaling matrix. For instance, in SGD B_k is just the identity matrix; in the oBFGS and oLBFGS methods [15], the matrix B_k is set to a quasi-Newton estimate of the inverse Hessian which is calculated according to the BFGS and L-BFGS formulas; in the SGD-QN method [3] the matrix B_k is a diagonal matrix whose entries are estimated from the secant equation; the SQN method [5] is an advanced version of oLBFGS; instead of estimating the gradient difference $\nabla f(x_{k+1}) - \nabla f(x_k)$ by subtracting two stochastic gradients, SQN estimates it by multiplying a stochastic Hessian by the deterministic difference $x_{k+1} - x_k$. Compared to SGD, all these methods may have better practical performance. However, none of them qualitatively improves on the convergence rate—any method from this group has a slow sublinear rate, including the method with $B_k = \nabla^2 f(x_k)^{-1}$ [4].

The second group contains methods such as SAG [14], IAG [2], SVRG [11], FINITO [8], SAGA [7], MISO [13] etc. The common property of these methods is that they use such estimates of the objective gradient $\nabla f(x_k)$ whose error tends to zero as the iterate x_k approaches the optimum x^*. As a result, these methods may converge with constant step size, and their convergence rate is linear.

Recently Gurbuzbalaban et al. have proposed the incremental Newton (IN) method [10]. Despite a similar name, the IN method is quite different to NIM. The IN method belongs to the first group of incremental methods with B_k equal to a partial sum of $\nabla^2 f_i$, and its convergence rate is sublinear.

The most closely-related method to NIM is SFO [16]. This method also uses a second-order model for each function f_i, but, instead of the true Hessian $\nabla^2 f_i(v_i^k)$, it works with its approximation obtained with the help of an L-BFGS-like technique. Although no convergence analysis of SFO is given in [16], experiments show that its convergence rate is linear. This shows that the Hessian approximation of SFO is not accurate enough to ensure a superlinear rate.

2 Method NIM

We begin the derivation of NIM by constructing a quadratic model of f at the current iteration k. First, we form the following convex quadratic model of each f_i:

$$f_i(x) \approx m_k^i(x) := f_i(v_k^i) + \nabla f_i(v_k^i)^\top (x - v_k^i) + \frac{1}{2}(x - v_k^i)^\top \nabla^2 f_i(v_k^i)(x - v_k^i)$$

for some point $v_k^i \in \mathbf{R}^d$. Once we have a model of each f_i, we can naturally build a model of f:

$$f(x) \approx m_k(x) := \frac{1}{n}\sum_{i=1}^n m_k^i(x) + \frac{\mu}{2}\|x\|_2^2.$$

Note that the model m_k is a strongly convex function and hence has the unique minimizer $\bar{x}_k := \operatorname{argmin}_x m_k(x)$. By setting the gradient of m_k to zero, we obtain the following formula for $\bar{x}_k$:

$$\bar{x}_k = (H_k + \mu I)^{-1}(u_k - g_k), \tag{2}$$

where we use the notation

$$H_k := \frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(v_k^i), \qquad u_k := \frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(v_k^i) v_k^i, \qquad g_k := \frac{1}{n}\sum_{i=1}^n \nabla f_i(v_k^i). \tag{3}$$

To obtain the new iterate x_{k+1}, NIM makes a step in the direction of the minimum of the model m_k:

$$x_{k+1} := \alpha_k \bar{x}_k + (1 - \alpha_k) x_k, \tag{4}$$

where $\alpha_k \in (0, 1]$ is the step size. After the step is done, we update the model m or, equivalently, the centers v. To keep the iteration complexity low, we update only one component of the full model at every iteration:

$$v_{k+1}^i := \begin{cases} x_{k+1} & \text{if } i = i_k, \\ v_k^i & \text{otherwise,} \end{cases}$$

where $i_k \in \{1, \dots, n\}$ is the index of the component to update.

3 Convergence analysis

In this section we state the results about local and global convergence rates of NIM. The proof of these results can be found in the Appendix.

Theorem 1 (local convergence rate)**.** *Suppose the Hessians $\nabla^2 f_i$ are Lipschitz-continuous:*

$$\left\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\right\|_2 \leq M \left\|x - y\right\|_2, \qquad i = 1, \ldots, n,$$

for all $x, y \in \mathbf{R}^d$. Let $\{x_k\}$ be the sequence of iterates generated by NIM with the unit step size $\alpha_k \equiv 1$ and cyclic order of component selection. If all the model centers are initialized close enough to the solution x^* of (1)*, $\left\|v_0^i - x^*\right\|_2 \leq (2\mu)/(M\sqrt{n}),\ i = 1, \ldots, n$, then the sequence $\{x_k\}$ converges to x^* at an R-superlinear rate:*

$$\left\|x_k - x^*\right\|_2 \leq r_k \qquad \textit{and} \qquad \lim_{k\to\infty} \frac{r_{k+1}}{r_k} = 0.$$

More precisely, the convergence rate of $\{x_k\}$ is n-step R-quadratic:

$$r_{k+n} \leq \frac{M}{2\mu} r_k^2, \qquad k = 2n, 2n+1, \ldots.$$

Theorem 2 (global convergence rate)**.** *Suppose the gradients ∇f_i are Lipschitz-continuous:*

$$\left\|\nabla f_i(x) - \nabla f_i(y)\right\|_2 \leq L_f \left\|x - y\right\|_2, \qquad i = 1, \ldots, n,$$

for all $x, y \in \mathbf{R}^d$. Denote the condition number of f as $\kappa := (L_f + \mu)/\mu$. Let $\{x_k\}$ be the sequence of iterates generated by NIM with the cyclic order of component selection and a constant step size $\alpha_k \equiv \alpha$, where $\alpha < \bar{\alpha} := 2\kappa^{-3}(1 + 19\kappa(n-1))^{-1}$. Then, for any initialization of the model centers $v_0^i,\ i = 1, \ldots, n$, the sequence $\{x_k\}$ converges to the solution x^* of (1) *at an R-linear rate:*

$$\left\|x_k - x_*\right\|_2 \leq \sqrt{\kappa} \cdot c^{k/2} \left\|x_0 - x^*\right\|_2$$

where $c := h^{1/(1+2(n-1))}$ for $h := 1 - 2\kappa^{-1}\alpha + \kappa^2(1 + 19\kappa(n-1))\alpha^2$.

4 Implementation details

Model update. Since we update only one component of the full model at every iteration we need not compute the sums in (3) every time. Instead, we keep track of the quantities H_k, u_k, g_k and update them as follows (here i denotes the index of the selected component at iteration k):

$$\begin{aligned} H_{k+1} &= H_k + \frac{1}{n}\left[\nabla^2 f_i(v_{k+1}^i) - \nabla^2 f_i(v_k^i)\right], \\ u_{k+1} &= u_k + \frac{1}{n}\left[\nabla^2 f_i(v_{k+1}^i) v_{k+1}^i - \nabla^2 f_i(v_k^i) v_k^i\right], \\ g_{k+1} &= g_k + \frac{1}{n}\left[\nabla f_i(v_{k+1}^i) - \nabla f_i(v_k^i)\right]. \end{aligned} \tag{5}$$

In order to do this, we need to store all the centers v_k^i in memory. Taking into account the cost of storing H_k, u_k, g_k, the total storage cost of NIM is $O(nd + d^2)$. Note that we do not store the separate Hessians $\nabla^2 f_i(v_k^i)$ in memory because it would cost $O(nd^2)$. Therefore, to update the model, we need to compute the selected f_i twice—once at the new point $v_{k+1}^i = x_{k+1}$ and once at the previous point v_k^i.

Order of component selection. We have experimented with two standard strategies for choosing the component i_k to update: 1) *cyclic* when $i_k = (k \bmod n) + 1$ and 2) *randomized* when at every iteration $i_k \in \{1, \ldots, n\}$ is chosen uniformly at random. In all our experiments we observed that NIM always converges faster under the cyclic order. This is quite different to incremental gradient methods for which it is always better to use randomized order both in theory and practice[1, 6].

Linear models. In many machine learning problems the functions f_i have the following linearly-parameterized form: $f_i(x) := \phi_i(a_i^\top x)$, where $\phi_i : \mathbf{R} \to \mathbf{R}$ is a univariate function and $a_i \in \mathbf{R}^d$ is some known vector. In this case, by exploiting the structure of the functions f_i, we can substantially

reduce the iteration complexity and storage cost of NIM. Namely, denoting $\nu_k^i := a_i^\top v_k^i$, we can rewrite (5) as follows:

$$H_{k+1} = H_k + \frac{1}{n}\left[\phi_i''(\nu_{k+1}^i) - \phi_i''(\nu_k^i)\right] a_i a_i^\top,$$
$$u_{k+1} = u_k + \frac{1}{n}\left[\phi_i''(\nu_{k+1}^i)\nu_{k+1}^i - \phi_i''(\nu_k^i)\nu_k^i\right] a_i,$$
$$g_{k+1} = g_k + \frac{1}{n}\left[\phi_i'(\nu_{k+1}^i) - \phi_i'(\nu_k^i)\right] a_i.$$

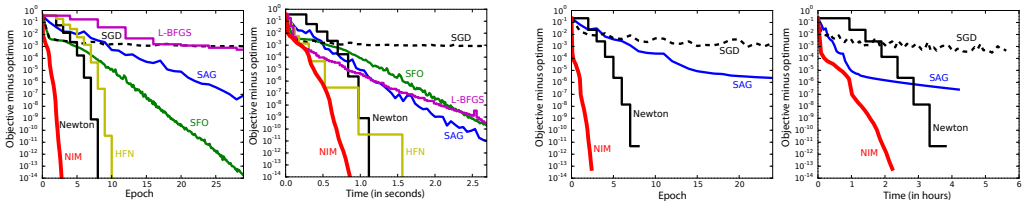
Note that the update of H_k is a matrix rank-1 update. Therefore, using the Sherman-Morrison formula, we can write the update for the inverse matrix $B_k := (H_k + \mu I)^{-1}$:

$$B_{k+1} = B_k - \frac{\delta_k B_k a_i a_i^\top B_k}{n + \delta_k a_i^\top B_k a_i}, \qquad \text{where } \delta_k := \phi_i''(\nu_{k+1}^i) - \phi_i''(\nu_k^i).$$

Once we have the matrix B_k, the cost of finding $\bar{x}_k$ by formula (2) reduces from $O(d^3)$ to $O(d^2)$, so, instead of working with matrices H_k, we can work directly with matrices B_k. Also, instead of storing n vectors v_k^i, now we need to store only n scalars ν_k^i. This reduces the memory requirements of NIM from $O(nd + d^2)$ to $O(n + d^2)$.

5 Experiments

We compare the empirical performance of NIM with that of several other methods for solving (1) on the ℓ_2-regularized logistic regression problem. We use the following methods in our comparison: 1) *NIM*: the proposed method NIM (the version for linear models with the unit step size and cyclic order of component selection; see Section 4), 2) *SGD*: the classic stochastic gradient method (with step size $\alpha_k = n/(10k)$) as a representative of sublinearly convergent incremental methods, 3) *SAG*: the stochastic average gradient [14] (the version for linear models with constant step size $\alpha = 1/L$, where L is the analytically calculated Lipschitz constant) as a representative of linearly convergent incremental methods, 4) *Newton*: the classic Newton method (with the unit step size) as a non-incremental variant of NIM, 5) *L-BFGS*: the limited-memory BFGS method [12] (with the history of size 10) and 6) *HFN*: the Hessian-free (inexact) Newton method [9] as the two most popular variants of the classic Newton method, and 7) *SFO*: the sum of functions optimizer [16] (with default parameters) as the most closely-related algorithm to NIM. All the methods except SFO are implemented in C++; for SFO we use its public implementation in Python. As training data, we use the following two data sets from the LIBSVM site that correspond to the cases of moderate and large n in problem (1): 1) *a9a* (~32k samples, 123 features), and 2) a binarized *mnist8m* data set[1] (~8m samples and 784 features). We set the regularization coefficient μ to $1/n$ and run all methods from $x_0 = 0$. Figure 1 shows the results of our experiments[2][3].



(a) The case of moderate n.

(b) The case of large n.

Figure 1: Comparison of NIM with other methods.

[1] To binarize the original *mnist8m* data set, we group digits with labels 0, 1, 2, 3, 4 into the first class, and digits with labels 5, 6, 7, 8, 9 into the second class.

[2] We use a stairstep plot for non-incremental methods (Newton, L-BFGS and HFN) to highlight that their iterations are very expensive (require a full pass over the training set) and cannot be interrupted in the middle.

[3] Since methods L-BFGS and HFN cannot be efficiently applied for large n, they are not shown in Figure 1b. We also do not show SFO in this experiment because its Python implementation is quite slow on large data sets.

References

[1] Dimitri P Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010:1–38, 2011.

[2] Doron Blatt, Alfred O Hero, and Hillel Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2007.

[3] Antoine Bordes, Léon Bottou, and Patrick Gallinari. SGD-QN: Careful quasi-newton stochastic gradient descent. *The Journal of Machine Learning Research*, 10:1737–1754, 2009.

[4] Olivier Bousquet and Léon Bottou. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pages 161–168, 2008.

[5] Richard H Byrd, SL Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *arXiv preprint arXiv:1401.7020*, 2014.

[6] Aaron Defazio. *New Optimization Methods for Machine Learning*. PhD thesis, Australian National University, 2014, 2014.

[7] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

[8] Aaron J Defazio, Tibério S Caetano, and Justin Domke. Finito: A faster, permutable incremental gradient method for big data problems. In *Proceedings of 31st International Conference on Machine Learning*, 2014.

[9] Ron S Dembo, Stanley C Eisenstat, and Trond Steihaug. Inexact Newton methods. *SIAM Journal on Numerical analysis*, 19(2):400–408, 1982.

[10] Mert Gürbüzbalaban, Asuman Ozdaglar, and Pablo Parrilo. A globally convergent incremental newton method. *Mathematical Programming*, 151(1):283–313, 2015.

[11] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[12] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[13] Julien Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

[14] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.

[15] Nicol N Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-newton method for online convex optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 436–443, 2007.

[16] Jascha Sohl-Dickstein, Ben Poole, and Surya Ganguli. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 604–612, 2014.

A Local convergence rate: proof

In this section we prove Theorem 1 about the local convergence rate of NIM.

A.1 Outline of the proof

The proof is organized as follows:

1. First, we show that the sequence $\{\|x_k - x^*\|_2\}$ of the residuals of NIM is bounded above by the following recurrent sequence (Lemma 1):
$$r_k := \frac{M}{2\mu n}(r_{k-1}^2 + r_{k-2}^2 + \cdots + r_{k-n}^2), \qquad k = n, n+1, \ldots,$$
$$r_k := \|x_k - x^*\|_2, \qquad k = 0, \ldots, n-1.$$
2. Then we prove that, starting from some moment, the sequence $\{r_k\}$ decreases monotonically (Lemma 5):
$$r_{k+1} \le r_k, \qquad k = 2n, 2n+1, \ldots.$$
3. From this it follows that the convergence rate of $\{r_k\}$ is n-step quadratic (Lemma 6):
$$r_k \le \frac{M}{2\mu} r_{k-n}^2, \qquad k = 3n, 3n+1, \ldots.$$
4. Then with the help of Lemma 7 and Lemma 8 we build a majorizing sequence $\{q_k\}$ for the ratio r_{k+1}/r_k:
$$\frac{r_{k+1}}{r_k} \le q_k, \qquad k = 3n, 3n+1, \ldots,$$
where $q_k \to 0$ as $k \to \infty$. This estimate proves the superlinear convergence rate:
$$\lim_{k\to\infty} \frac{r_{k+1}}{r_k} = 0.$$

Lemma 2 is an auxiliary lemma which is used inside the proofs of all the other lemmas. Lemma 3 is used to prove Lemma 4, which in turn is used to prove Lemma 5. The proof of Theorem 1 itself is placed at the end of this section.

A.2 Main estimate

The next lemma provides a recurrent estimate for the sequence of residuals $\tilde{r}_k := \|x_k - x^*\|_2$. The proof of this lemma is almost identical to the proof of the quadratic convergence rate of the classic Newton method.

Lemma 1 (main estimate)**.** *Assume the conditions of theorem 1 hold. Then the sequence of residuals satisfies the following recurrent inequality:*

$$\tilde{r}_k \le \frac{M}{2\mu n}(\tilde{r}_{k-1}^2 + \tilde{r}_{k-2}^2 + \cdots + \tilde{r}_{k-n}^2), \qquad k = n, n+1, \ldots. \tag{6}$$

Proof. Let $k \ge n-1$. To simplify formulas, let us introduce the following notation:

$$A_k := \frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v_k^i) + \mu I.$$

Since the step size is unit, $\alpha_k \equiv 1$, then, according to (4), the next iterate x_{k+1} is exactly the minimizer of the model m_k: $x_{k+1} \equiv \bar{x}_k$. Using formulas (2) and (3), we obtain the following expression for the next iterate:

$$x_{k+1} = A_k^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} \nabla^2 f_i(v_k^i) v_k^i - \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(v_k^i)\right).$$

By the first-order optimality condition, $0 = \nabla f(x^*) = (1/n)\sum_{i=1}^n \nabla f_i(x^*) + \mu x^*$, so we can write

$$\begin{aligned} x_{k+1} - x^* &= A_k^{-1}\left(\frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(v_k^i)v_k^i - \frac{1}{n}\sum_{i=1}^n \nabla f_i(v_k^i) - \frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(v_k^i)x^* - \mu x^*\right) \\ &= A_k^{-1}\left(\frac{1}{n}\sum_{i=1}^n \nabla^2 f_i(v_k^i)(v_k^i - x^*) - \frac{1}{n}\sum_{i=1}^n \left[\nabla f_i(v_k^i) - \nabla f_i(x^*)\right]\right). \end{aligned}$$

For the gradient difference we use the Taylor formula:

$$\nabla f_i(v_k^i) - \nabla f_i(x^*) = \int_0^1 \nabla^2 f_i(tv_k^i + (1-t)x^*)(v_k^i - x^*)dt.$$

Then

$$x_{k+1} - x^* = \frac{A_k^{-1}}{n}\sum_{i=1}^n \int_0^1 \left[\nabla^2 f_i(v_k^i) - \nabla^2 f_i(tv_k^i + (1-t)x^*)\right](v_k^i - x^*)dt.$$

Taking the norms and using the Lipschitz condition, we get

$$\tilde{r}_{k+1} \le \left\|A_k^{-1}\right\|_2 \frac{M}{2n}\sum_{i=1}^n \left\|v_k^i - x^*\right\|_2^2. \tag{7}$$

Since all the functions f_i are convex, their Hessians $\nabla^2 f_i(v_k^i)$ are positive-semidefinite. Therefore,

$$\|A_k\|_2 \ge \mu \qquad \text{and} \qquad \left\|A_k^{-1}\right\| \le \frac{1}{\mu}. \tag{8}$$

Also, because we use the cyclic order of component selection, we have

$$\sum_{i=1}^n \left\|v_k^i - x^*\right\|_2^2 = \tilde{r}_k^2 + \tilde{r}_{k-1}^2 + \cdots + \tilde{r}_{k-n+1}^2. \tag{9}$$

Plugging (8) and (9) into (7), we obtain the estimate

$$\tilde{r}_{k+1} \le \frac{M}{2\mu n}(\tilde{r}_k^2 + \tilde{r}_{k-1}^2 + \cdots + \tilde{r}_{k-n+1}^2),$$

which is the same as (6) to within a shift of indices. □

A.3 Auxiliary lemmas

In the rest of the proof we analyze the following recurrent sequence:

$$r_k := \frac{C}{n}(r_{k-1}^2 + r_{k-2}^2 \cdots + r_{k-n}^2), \qquad k = n, n+1, \ldots, \tag{10}$$

where $C := M/(2\mu) > 0$ and $r_k := \|x_k - x^*\|_2$, $k = 0, \ldots, n-1$. According to Lemma 1, this sequence is an upper bound for the sequence of residuals: $\|x_k - x^*\|_2 \le r_k$. Each of the following lemmas proves one little fact about the sequence $\{r_k\}$. Almost in every lemma we assume that the initial elements $r_0, r_1, \ldots, r_{n-1}$ of $\{r_k\}$ are small enough. This assumption corresponds to the assumption about locality (i.e. that we initialize the centers in NIM sufficiently close to the optimum).

Lemma 2 (basic estimate)**.** *The sequence $\{r_k\}$ satisfies the following two recurrent inequalities:*

$$\frac{C}{n}\max\{r_{k-1}, r_{k-2}, \ldots, r_{k-n}\}^2 \le r_k \le C\max\{r_{k-1}, r_{k-2}, \ldots, r_{k-n}\}^2.$$

Proof. Follows from the definition (10) and the fact that

$$\max\{r_{k-1}^2, r_{k-2}^2, \ldots, r_{k-n}^2\} = \max\{r_{k-1}, r_{k-2}, \ldots, r_{k-n}\}^2.$$ □

Lemma 3 (boundedness)**.** *If the initial elements of the sequence* $\{r_k\}$ *are bounded,*

$$\max\{r_0, r_1, \ldots, r_{n-1}\} \le \frac{1}{C\sqrt{n}},$$

then all the elements of this sequence are bounded:

$$r_k \le \frac{1}{C\sqrt{n}}, \qquad k = 0, 1, 2, \ldots. \tag{11}$$

Proof. By induction. Assume bound (11) is true for all the indices from 0 to $k-1$ inclusive. Let us prove that this bound is also true for index k. Using Lemma 2 about the basic estimate and the induction hypothesis, we get

$$r_k \le C \max\{r_{k-1}, r_{k-2}, \ldots, r_{k-n}\}^2 \le C\frac{1}{C^2 n} = \frac{1}{Cn} \le \frac{1}{C\sqrt{n}}. \qquad \square$$

Lemma 4 (block quadratic convergence)**.** *Let the initial elements of the sequence* $\{r_k\}$ *be bounded:*

$$\max\{r_0, r_1, \ldots, r_{n-1}\} \le \frac{1}{C\sqrt{n}}.$$

Then the sequence of the maximums over successive n *elements converges quadratically:*

$$\max\{r_{k-n+1}, r_{k-n+2}, \ldots, r_k\} \le C \max\{r_{k-1}, r_{k-2}, \ldots, r_{k-n}\}^2, \qquad k = n, n+1, \ldots. \tag{12}$$

Proof. Let us fix a number $k \ge n$. From Lemma 3 about boundedness it follows that

$$\begin{aligned} \max\{Cr_{k-1}^2, Cr_{k-2}^2, \ldots, Cr_{k-n}^2\} &\le C\frac{1}{C\sqrt{n}} \max\{r_{k-1}, r_{k-2}, \ldots, r_{k-n}\} \\ &\le \max\{r_{k-1}, r_{k-2}, \ldots, r_{k-n}\}. \end{aligned} \tag{13}$$

According to Lemma 2 about the basic estimate,

$$r_k \le C \max\{r_{k-1}, r_{k-2}, \ldots, r_{k-n}\}^2. \tag{14}$$

Using inequalities (14) and (13), we obtain

$$\begin{aligned} r_{k+1} &\le C \max\left\{r_k, r_{k-1}, \ldots, r_{k-n+1}\right\}^2 \\ &\le C \max\left\{\max\left\{Cr_{k-1}^2, Cr_{k-2}^2, \ldots, Cr_{k-n}^2\right\}, r_{k-1}, \ldots, r_{k-n+1}\right\}^2 \\ &\le C \max\left\{r_{k-1}, r_{k-2}, \ldots, r_{k-n}\right\}^2. \end{aligned} \tag{15}$$

Now, combining inequalities (15), (14) and (13), we have

$$\begin{aligned} r_{k+2} &\le C \max\left\{r_{k+1}, r_k, r_{k-1} \ldots, r_{k-n+2}\right\}^2 \\ &\le C \max\left\{\max\left\{Cr_{k-1}^2, \ldots, Cr_{k-n}^2\right\}, \max\left\{Cr_{k-1}^2, \ldots, Cr_{k-n}^2\right\}, r_{k-1}, \ldots, r_{k-n+2}\right\}^2 \\ &\le C \max\left\{r_{k-1}, r_{k-2}, \ldots, r_{k-n}\right\}^2. \end{aligned}$$

Applying the same procedure successively for $r_{k+3}, \ldots, r_{k+n-1}$, we get (12). $\square$

Lemma 5 (monotonicity)**.** *Let the initial elements of the sequence* $\{r_k\}$ *be bounded:*

$$\max\{r_0, r_1, \ldots, r_{n-1}\} \le \frac{1}{C\sqrt{n}}.$$

Then, starting from $k = 2n$*, the sequence* $\{r_k\}$ *decreases monotonically:*

$$r_{k+1} \le r_k, \qquad k = 2n, 2n+1, \ldots.$$

Proof. Let us fix a number $k \ge 2n$. Note that by the definition (10) of the sequence $\{r_k\}$ the inequality $r_{k+1} \le r_k$ is equivalent to the inequality $r_k \le r_{k-n}$. Therefore we will prove that

$r_k \le r_{k-n}$. Applying Lemma 2 about the basic estimate and Lemma 4 about block quadratic convergence, we have

$$\begin{aligned} r_k &\le C \max\{r_{k-1}, r_{k-2}, \dots, r_{k-n}\}^2 \\ &\le C \left(C \max\{r_{k-n-1}, r_{k-n-2}, \dots, r_{k-2n}\}^2\right)^2 \\ &= C^3 \max\{r_{k-n-1}, r_{k-n-2}, \dots, r_{k-2n}\}^4. \end{aligned}$$

Applying Lemma 2 about the basic estimate, we can write

$$r_{k-n} \ge \frac{C}{n} \max\{r_{k-n-1}, r_{k-n-2}, \dots, r_{k-2n}\}^2.$$

Comparing the right-hand sides of the last two inequalities and using Lemma 3 about boundedness, we obtain the inequality $r_k \le r_{k-n}$. □

Lemma 6 (n-step quadratic convergence)**.** *Let the initial elements of the sequence $\{r_k\}$ be bounded:*

$$\max\{r_0, r_1, \dots, r_{n-1}\} \le \frac{1}{C\sqrt{n}}.$$

Then the convergence rate of this sequence is n-step quadratic:

$$r_k \le C r_{k-n}^2, \qquad k = 3n, 3n+1, \dots. \tag{16}$$

Proof. Follows from Lemma 2 about the basic estimate and Lemma 5 about monotonicity. □

Lemma 7 (linear convergence rate)**.** *Let the initial elements of the sequence $\{r_k\}$ be bounded:*

$$\max\{r_0, r_1, \dots, r_{n-1}\} \le \frac{1}{C\sqrt{n}}.$$

Then the convergence rate of this sequence is at least linear[4]:

$$r_{k+1} \le \left(1 - \frac{n-1}{n^2}\right) r_k, \qquad 3n, 3n+1, \dots.$$

Proof. Let us fix a number $k \ge 3n$. From Lemma 6 about n-step quadratic convergence and Lemma 3 about boundedness we have

$$r_k \le (C r_{k-n}) r_{k-n} \le \frac{r_{k-n}}{\sqrt{n}}.$$

Then

$$\begin{aligned} r_{k+1} &= \frac{C}{n}(r_k^2 + r_{k-1}^2 + \dots + r_{k-n+1}^2) \\ &\le \frac{C}{n}\left(\frac{r_{k-n}^2}{n} + r_{k-1}^2 + \dots + r_{k-n+1}^2\right) \\ &= \frac{C}{n}\left(\frac{r_{k-n}^2}{n} + r_{k-1}^2 + \dots + r_{k-n+1}^2 + r_{k-n}^2 - r_{k-n}^2\right) \\ &= r_k - \frac{n-1}{n^2} C r_{k-n}^2. \end{aligned}$$

Combining this estimate with inequality (16), we get the claimed statement:

$$r_{k+1} \le r_k - \frac{n-1}{n^2} r_k = \left(1 - \frac{n-1}{n^2}\right) r_k.$$ □

Lemma 8 (improving the constant)**.** *Assume that, starting from number k_0, the sequence $\{r_k\}$ decays linearly to zero with constant $c_0 \in (0, 1)$:*

$$r_{k+1} \le c_0 r_k, \qquad k = k_0, k_0+1, \dots. \tag{17}$$

Then, starting from number $k_0 + n$, the constant c_0 can be replaced with its square:

$$r_{k+1} \le c_0^2 r_k, \qquad k = k_0 + n, k_0 + n + 1, \dots.$$

[4]Here we assume that $n \ge 2$. When $n = 1$, the statement follows from the definition (10).

Proof. Let us fix a number $k \geq k_0 + n$. Using the definition (10) of $\{r_k\}$ and inequality (17), we have

$$\begin{aligned} r_{k+1} &= \frac{C}{n}(r_k^2 + r_{k-1}^2 + \cdots + r_{k-n+1}^2) \\ &\leq \frac{C}{n}(c_0^2 r_{k-1}^2 + c_0^2 r_{k-2}^2 + \cdots + c_0^2 r_{k-n}^2) \\ &= c_0^2 \frac{C}{n}(r_{k-1}^2 + r_{k-2}^2 + \cdots + r_{k-n}^2) \\ &= c_0^2 r_k. \quad \square \end{aligned}$$

A.4 Proof of the theorem

Proof. According to Lemma 1 about the main estimate, the sequence of residuals $\|x_k - x^*\|_2$ is bounded above by the sequence $\{r_k\}$ defined in (10) with $C := M/(2\mu)$ and $r_k := \|x_k - x^*\|_2$, $k = 0, \ldots, n-1$. The statement about the n-step quadratic convergence rate is proved in Lemma 6. Let us prove the statement about the superlinear convergence rate[5]. Using Lemma 7 about linear convergence and Lemma 8 about improving the constant, we can write the following sequence of estimates:

$$\begin{aligned} &\frac{r_{k+1}}{r_k} \leq q, && k = 3n, 3n+1, \ldots, \\ &\frac{r_{k+1}}{r_k} \leq q^2, && k = 4n, 4n+1, \ldots, \\ &\frac{r_{k+1}}{r_k} \leq q^4, && k = 5n, 5n+1, \ldots, \\ &\frac{r_{k+1}}{r_k} \leq q^8, && k = 6n, 6n+1, \ldots, \\ &\quad \ldots \end{aligned}$$

where $q := 1 - (n-1)/n^2$. Combining all these estimates together, we get

$$\frac{r_{k+1}}{r_k} \leq q^{2^{\lfloor k/n \rfloor - 3}}, \qquad k = 3n, 3n+1, \ldots.$$

Since the right-hand side in this inequality converges to zero as $k \to \infty$, then r_{k+1}/r_k also converges to zero as $k \to \infty$. $\square$

B Global convergence rate: proof

In this section we prove Theorem 2 about the global convergence rate of NIM.

B.1 Outline of the proof and notation

Note that according to (4), (2) and (3) we can write the step of NIM as follows:

$$x_{k+1} = x_k + \alpha p_k,$$

where $p_k := \bar{x}_k - x_k$ is the search direction given by:

$$p_k = A_k^{-1} \left(\frac{1}{n} \sum_{i=1}^n \nabla^2 f_i(v_k^i)(v_k^i - x_k) - \frac{1}{n} \sum_{i=1}^n \nabla f_i(v_k^i) - \mu x_k \right), \tag{18}$$

where, for convenience, we define $A_k := (1/n) \sum_{i=1}^n \nabla^2 f_i(v_k^i) + \mu I$.

To analyze NIM, we view it as a perturbed scaled gradient method:

$$x_{k+1} = x_k + \alpha p_k^{\mathrm{SG}} + \alpha e_k, \tag{19}$$

where we use the same matrix A_k to scale the gradient, $p_k^{\mathrm{SG}} := -A_k^{-1} \nabla f(x_k)$, and $e_k := p_k - p_k^{\mathrm{SG}}$ is the error in the approximation of p_k^{SG} by p_k.

The most important step is to prove that the norm of the error e_k can be bounded by a term proportional to the step size α and the norms of gradients at previous points. This is done in Section B.3.

[5] Here we assume that $n \geq 2$. When $n = 1$, the statement follows from n-step quadratic convergence.

B.2 Auxiliary facts

In this section we state several useful inequalities that we will use throughout the rest of the proof.

Since A_k is a symmetric positive definite matrix and $\lambda_{\min}(A_k) \geq \mu$, we can write

$$\left\|A_k^{-1}\right\|_2 \leq \frac{1}{\mu}. \tag{20}$$

and

$$\left\|p_k^{\mathrm{SG}}\right\|_2 \leq \left\|A_k^{-1}\right\|_2 \left\|\nabla f(x_k)\right\|_2 \leq \frac{1}{\mu}\left\|\nabla f(x_k)\right\|_2. \tag{21}$$

Lemma 9. *For a twice continuously differentiable strongly convex function $f : \mathbf{R}^d \to \mathbf{R}$ with constant $\mu > 0$ we have the following inequality for any $x \in \mathbf{R}^d$:*

$$\left\|x - x^*\right\|_2 \leq \frac{1}{\mu}\left\|\nabla f(x)\right\|_2, \tag{22}$$

where x^* is the optimum of f.

Proof. Take any $x, y \in \mathbf{R}^d$. Using the Taylor formula, we obtain

$$\nabla f(x) - \nabla f(y) = \nabla^2 f(z)(x - y),$$

where $z \in [x, y]$. By the strong convexity of f, for any $z \in \mathbf{R}^d$ the matrix $\nabla^2 f(z)$ is symmetric positive definite with $\lambda_{\min}(\nabla^2 f(z)) \geq \mu$. Therefore,

$$\left\|\nabla f(x) - \nabla f(y)\right\|_2 \geq \mu\left\|x - y\right\|_2.$$

Now it remains to set $y = x^*$ and divide both sides by μ. □

Lemma 10. *Let $f : \mathbf{R}^d \to \mathbf{R}$ be a continuously differentiable strongly convex function f with constant $\mu > 0$ and Lipschitz-continuous gradient with constant $L > 0$. Then for any $x \in \mathbf{R}^d$ we have:*

$$2\mu[f(x) - f(x^*)] \leq \left\|\nabla f(x)\right\|_2^2 \leq 2L[f(x) - f(x^*)], \tag{23}$$

where x^* is the optimum of f.

Proof. From strong convexity and Lipschitz-continuity of the gradient, for any $x, y \in \mathbf{R}^d$:

$$f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\left\|y - x\right\|_2^2 \leq f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\left\|y - x\right\|_2^2.$$

Taking the minimum over y, we get:

$$f(x) - \frac{1}{2\mu}\left\|\nabla f(x)\right\|_2^2 \leq f(x^*) \leq f(x) - \frac{1}{2L}\left\|\nabla f(x)\right\|_2^2,$$

which coincides with (23) after rearranging. □

Note that in the conditions of Theorem 2 the function f has Lipschitz-continuous gradient with constant $L := L_f + \mu$.

B.3 Bounding the error norm

Our derivation of the bound on the error norm is based on the recent work of Gurbuzbalaban et al. [1]. The main result of this section is in Lemma 12. To prove it, we first state an auxiliary lemma.

Lemma 11. *The error norm satisfies the following two bounds, regardless of the way the sequence $\{x_k\}$ is constructed:*

$$\left\|e_k\right\|_2 \leq \frac{2L_f}{\mu} \max_{j=k-n+1,\ldots,k-1} \left\|x_j - x_k\right\|_2 \tag{24}$$

and

$$\left\|e_k\right\|_2 \leq \frac{4L_f}{\mu^2} \max_{j=k-n+1,\ldots,k} \left\|\nabla f(x_j)\right\|_2 \tag{25}$$

Proof. Plugging (18) into the definition of e_k, we have

$$e_k = A_k^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2 f_i(v_k^i)(v_k^i - x_k) - \frac{1}{n}\sum_{i=1}^{n}\left[\nabla f_i(v_k^i) - \nabla f_i(x_k)\right]\right).$$

Then,

$$\|e_k\|_2 \le \left\|A_k^{-1}\right\|_2\left(\frac{1}{n}\sum_{i=1}^{n}\left\|\nabla^2 f_i(v_k^i)\right\|_2\left\|v_k^i - x_k\right\|_2 + \frac{1}{n}\sum_{i=1}^{n}\left\|\nabla f_i(v_k^i) - \nabla f_i(x_k)\right\|_2\right).$$

As a bound on $\left\|A_k^{-1}\right\|_2$ we use (20). Next, by the Lipschitz-continuity of ∇f_i, we have that $\left\|\nabla^2 f_i(v_k^i)\right\|_2 \le L_f$ and $\left\|\nabla f_i(v_k^i) - \nabla f_i(x_k)\right\|_2 \le L_f\left\|v_k^i - x_k\right\|_2$. Therefore,

$$\|e_k\|_2 \le \frac{2L_f}{\mu n}\sum_{i=1}^{n}\left\|v_k^i - x_k\right\|_2.$$

Since the order of component selection is cyclic,

$$\sum_{i=1}^{n}\left\|v_k^i - x_k\right\|_2 = \sum_{j=k-n+1}^{k}\|x_j - x_k\|_2,$$

and

$$\|e_k\|_2 \le \frac{2L_f}{\mu n}\sum_{j=k-n+1}^{k}\|x_j - x_k\|_2 \le \frac{2L_f}{\mu}\max_{j=k-n+1,\ldots,k-1}\|x_j - x_k\|_2.$$

Thus, inequality (24) is proved.

To prove inequality (25) we use the triangle inequality inside the maximum in (24), getting

$$\|e_k\|_2 \le \frac{4L_f}{\mu}\max_{j=k-n+1,\ldots,k}\|x_j - x^*\|_2$$

and then apply (22). □

Note that in Lemma 11 we did not use the way (19) the sequence $\{x_k\}$ is constructed. As we show in a moment, by using this extra information in Lemma 11, we can obtain a bound similar to (25), but proportional to the step length α.

Lemma 12. *The error norm can be bounded as follows:*

$$\|e_k\|_2 \le \frac{8L^2(n-1)\alpha}{\mu^3}\max_{j=k-2n+2,\ldots,k-1}\|\nabla f(x_j)\|_2. \tag{26}$$

Proof. We start with (24):

$$\|e_k\|_2 \le \frac{2L_f}{\mu}\max_{j=k-n+1,\ldots,k-1}\|x_j - x_k\|_2.$$

By the triangle inequality, for any $j = k-n+1,\ldots,k-1$, we get

$$\|x_j - x_k\|_2 = \left\|\sum_{t=j}^{k-1}[x_t - x_{t+1}]\right\|_2 \le \sum_{t=j}^{k-1}\|x_t - x_{t+1}\|_2.$$

Therefore,

$$\|e_k\|_2 \le \frac{2L_f}{\mu}\sum_{t=k-n+1}^{k-1}\|x_t - x_{t+1}\|_2 \le \frac{2L_f(n-1)}{\mu}\max_{t=k-n+1,\ldots,k-1}\|x_t - x_{t+1}\|_2. \tag{27}$$

Using (19), we have

$$\|x_t - x_{t+1}\|_2 \le \alpha\left(\left\|p_t^{\mathrm{SG}}\right\|_2 + \|e_t\|_2\right).$$

To bound the first term, we use (21). For the second term we use (25):

$$\|e_t\|_2 \leq \frac{4L_f}{\mu^2} \max_{j=t-n+1,\ldots,t-1} \|\nabla f(x_j)\|_2 .$$

Therefore,

$$\|x_t - x_{t+1}\|_2 \leq \frac{(4L_f+\mu)\alpha}{\mu^2} \max_{j=t-n+1,\ldots,t} \|\nabla f(x_j)\|_2 \leq \frac{4L\alpha}{\mu^2} \max_{j=t-n+1,\ldots,t} \|\nabla f(x_j)\|_2 .$$

Plugging this bound into (27), we get

$$\begin{aligned}\|e_k\|_2 &\leq \frac{8L_f L(n-1)\alpha}{\mu^3} \max_{j=k-2n+2,\ldots,k-1} \|\nabla f(x_j)\|_2 \\ &\leq \frac{8L^2(n-1)\alpha}{\mu^3} \max_{j=k-2n+2,\ldots,k-1} \|\nabla f(x_j)\|_2 .\end{aligned}$$

□

B.4 Proof of the theorem

Proof. Using the Lipschitz-continuity of the gradient and iteration (19), we get

$$\begin{aligned}f(x_{k+1}) - f(x_k) &\leq \nabla f(x_k)^\top (x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|_2^2 \\ &= \alpha \nabla f(x_k)^\top p_k^{\mathrm{SG}} + \alpha \nabla f(x_k)^\top e_k \\ &\quad + \frac{L\alpha^2}{2}\left\|p_k^{\mathrm{SG}}\right\|_2^2 + L\alpha^2 (p_k^{\mathrm{SG}})^\top e_k + \frac{L\alpha^2}{2}\|e_k\|_2^2 .\end{aligned}$$

Now we bound each term above in terms of the norms of previous gradients. The first term is a quadratic form whose matrix A_k is symmetric positive definite with $\lambda_{\max}(A_k) \leq L$. Therefore we can write the following bound:

$$\alpha \nabla f(x_k)^\top p_k^{\mathrm{SG}} = -\alpha \nabla f(x_k)^\top A_k^{-1} \nabla f(x_k) \leq -\frac{\alpha}{L}\|\nabla f(x_k)\|_2^2 .$$

For the second term we use the Cauchy-Schwarz inequality and bound (26):

$$\alpha \nabla f(x_k)^\top e_k \leq \alpha \|\nabla f(x_k)\|_2 \|e_k\|_2 \leq \frac{8L^2(n-1)\alpha^2}{\mu^3} \max_{j=k-2n+2,\ldots,k} \|\nabla f(x_j)\|_2^2 .$$

For the third term we use (21):

$$\frac{L\alpha^2}{2}\left\|p_k^{\mathrm{SG}}\right\|_2^2 \leq \frac{L\alpha^2}{2\mu^2}\|\nabla f(x_k)\|_2^2 .$$

For the fourth term we use the Cauchy-Schwarz inequality and bounds (21) and (26):

$$L\alpha^2 (p_k^{\mathrm{SG}})^\top e_k \leq L\alpha^2 \left\|p_k^{\mathrm{SG}}\right\|_2 \|e_k\|_2 \leq \frac{8L^3(n-1)\alpha^3}{\mu^4} \max_{j=k-2n+2,\ldots,k} \|\nabla f(x_j)\|_2^2 .$$

For the fifth term we use (26):

$$\frac{L\alpha^2}{2}\|e_k\|_2^2 \leq \frac{32L^5(n-1)^2\alpha^4}{\mu^6} \max_{j=k-2n+2,\ldots,k-1} \|\nabla f(x_j)\|_2^2 .$$

Combining these five bounds together, we get:

$$\begin{aligned}f(x_{k+1}) - f(x_k) \leq &-\frac{\alpha}{L}\|\nabla f(x_k)\|_2^2 + \frac{L\alpha^2}{2\mu^2}\|\nabla f(x_k)\|_2^2 \\ &+ \frac{8L^2(n-1)\alpha^2}{\mu^3}\left(1 + \frac{L\alpha}{\mu} + \frac{4L^3(n-1)\alpha^2}{\mu^3}\right) \max_{j=k-2n+2,\ldots,k} \|\nabla f(x_j)\|_2^2 .\end{aligned}$$

Now we introduce $V_k := f(x_k) - f(x^*)$ and replace all the gradients in the above expression with the corresponding V_j using (23). This leads us to the following recurrent inequality:

$$V_{k+1} \leq p(\alpha) V_k + q(\alpha) \max_{j=k-2n+2,\ldots,k} V_t$$

where

$$p(\alpha) := 1 - 2\kappa^{-1}\alpha + \kappa^2\alpha^2,$$
$$q(\alpha) := 16\kappa^3(n-1)\alpha^2\left[1 + \kappa\alpha + 4\kappa^3(n-1)\alpha^2\right].$$

and $\kappa := L/\mu \geq 1$ is the condition number.

According to Lemma 3.2 of [1], to finish the proof, we need to find α such that $p(\alpha) + q(\alpha) < 1$. This will guarantee the linear convergence of V_k with constant $c = (p+q)^{1/(1+2(n-1))}$.

First, assume $\alpha \leq \alpha_0$ for some α_0 that we will choose later. Then

$$p(\alpha) + q(\alpha) \leq 1 - 2\kappa^{-1}\alpha + \kappa^2\alpha^2 + 16\kappa^3(n-1)\alpha^2(1+\delta) =: h_\delta(\alpha),$$

where $\delta := 1 + \kappa\alpha_0 + 4\kappa^3(n-1)\alpha_0^2$. The condition $h_\delta(\alpha) < 1$ is equivalent to

$$\alpha < 2\kappa^{-3}(1 + 16\kappa(n-1)(1+\delta))^{-1}.$$

Let us choose $\alpha_0 := \kappa^{-4}(n-1)^{-1}/8$. Then

$$1 + \delta = 1 + \frac{\kappa^{-3}(n-1)^{-1}}{8} + \frac{\kappa^{-5}(n-1)^{-1}}{16} \leq 1 + \frac{1}{8} + \frac{1}{16} = \frac{19}{16}$$

and

$$2\kappa^{-3}(1 + 16\kappa(n-1)(1+\delta))^{-1} \geq 2\kappa^{-3}(1 + 19\kappa(n-1))^{-1} =: \bar{\alpha}.$$

Note that $\bar{\alpha} \leq \alpha_0$. Therefore, for all $\alpha < \bar{\alpha}$ we will have $p(\alpha) + q(\alpha) \leq h_\delta(\alpha) \leq h(\alpha) < 1$ with

$$h(\alpha) := 1 - 2\kappa^{-1}\alpha + \kappa^2(1 + 19\kappa(n-1))\alpha^2. \qquad \square$$

References

[1] M. Gurbuzbalaban, A. Ozdaglar, and P. Parrilo. On the Convergence Rate of Incremental Aggregated Gradient Algorithms. *ArXiv e-prints*, June 2015.