UDC 519.6 *http://dx.doi.org/10.30970/vam.2024.32.12744*

ON METHODS WITH ASYNCHRONOUS APPROXIMATION OF THE INVERSE OPERATOR FOR NONLINEAR LEAST SQUARES PROBLEMS

S. Shakhno, Y. Shunkin, N. Hudzeliak

Ivan Franko National University of Lviv,
1, Universytetska str., 79000, Lviv, Ukraine,
e-mails: stepan.shakhno@lnu.edu.ua, yuriy.shunkin@lnu.edu.ua,
nataliia.hudzeliak@lnu.edu.ua

This paper addresses the development and study of iterative methods for solving nonlinear least squares problems that avoid the direct computation of matrix inverses. Specifically, we investigate successive, synchronous, and asynchronous strategies for approximating the inverse operator within the Gauss-Newton method, the secant method, and a method with third-order convergence. We present the theoretical foundations of these approaches, including their convergence conditions, and provide details on how they can be implemented in parallel computing environments. Numerical experiments on a series of benchmark problems illustrate the comparative performance of each method variant. In particular, we show that methods using asynchronous approximation of the inverse operator often converge in fewer iterations and with reduced computational time compared to both their synchronous and successive counterparts.

Key words: parallel iterative methods; nonlinear least square problem; inverse operator approximation; asynchronous computing.

1. Introduction

A nonlinear least squares problem is a powerful tool for solving a wide range of applied problems. These problems arise, in particular, when solving overdetermined systems of equations, building nonlinear regression models, estimating parameters of physical processes from experimental data, testing hypotheses in mathematical statistics, and controlling objects or processes [3].

The Gauss-Newton method is one of the most widely used techniques for solving such problems. However, its application becomes significantly more complicated if computing derivatives is difficult or infeasible. In these scenarios, the secant method can serve as an effective alternative, as it only relies on function evaluations rather than derivatives.

Despite their popularity, many classic methods for solving least squares problems, including the Gauss-Newton and secant methods, require computing an inverse matrix (or solving an equivalent system), which can be both computationally expensive and numerically unstable. To overcome this drawback, methods that approximate the inverse operator for nonlinear least squares problems have been proposed. These methods allow for finding approximate solutions without explicitly computing an inverse matrix. Moreover, they naturally lend themselves to parallel or asynchronous implementations, making them particularly appealing for large-scale problems or problems with high performance requirements.

In this work, we investigate the efficiency of iterative methods that use both *synchronous* and *asynchronous* approximation of the inverse operator for nonlinear least squares problems. We build these methods upon three primary algorithmic foundations:

– The Gauss-Newton method;
– The secant method;
– A method with third-order convergence.

By adapting each of these approaches to approximate the inverse operator rather than compute it directly, we aim to mitigate numerical challenges and reduce overall computational costs. In addition, we show how parallel and asynchronous implementations can further enhance efficiency by effectively distributing the required computations.

A synchronous parallel approximation typically involves updating the approximate solution and the inverse-operator approximation *in lock-step* at each iteration; the next iteration begins only after both computations have finished. By contrast, an asynchronous implementation allows these updates to occur without strict synchronization, letting each update proceed whenever its data are ready. Consequently, asynchronous methods can reduce idle time in multi-core or multi-processor environments, although at the cost of more complex control schemes.

We carry out numerical experiments on test problems of varying complexity to assess how effectively these methods address the nonlinear least squares problem:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\, F(x)^T F(x), \tag{1}$$

where $F:\mathbb{R}^n\to\mathbb{R}^m$ $(m\geq n)$ is a residual function, assumed to be Frechet-differentiable in x. We seek an x^* such that

$$f(x^*) \;=\; \min_{x\in\mathbb{R}^n} f(x), \quad \text{where}\;\; f(x) \;=\; \tfrac{1}{2}\, F(x)^T F(x).$$

If $f(x^*)=0$, then the problem has a zero residual; otherwise, the residual is nonzero.

Among the most common methods for solving (1) are the Gauss-Newton and Levenberg-Marquardt methods [5], both of which require solving a system of linear equations at each iteration. As an alternative, methods with successive and parallel approximation of the inverse operator can be employed to avoid directly solving these systems [10]. In general, these methods consist of two key parts:

1. Finding the approximate solution to (1);
2. Approximating the inverse operator needed in that solution step.

Many scientific and applied tasks are reduced to solving a nonlinear least squares problem, further emphasizing the importance of robust and scalable methods. For instance, in *positioning based on distance measurements*, one may seek the location of an object $x\in\mathbb{R}^3$ from measurements of distances to known points [3]. Another example is *fitting a nonlinear model $y\approx f(x;\theta)$* by minimizing the sum of squared residuals for a given dataset. One can also consider fitting geometric shapes (e.g., circles, ellipses, spheres) to data in computational geometry or computer graphics. All these illustrate the breadth and practical significance of nonlinear least squares problems.

In this paper, we focus on methods with synchronous and asynchronous approximation of the inverse operator. After providing the theoretical underpinnings of these methods, we detail their numerical implementation and compare their effectiveness on a variety of test problems. Our results highlight the notable advantages of asynchronous schemes in particular, which often reduce idle CPU time and accelerate convergence without requiring explicit matrix inversions.

2. Analysis of Methods with Synchronous and Asynchronous Approximation of the Inverse Operator

2.1. Gauss-Newton Method

The Gauss-Newton method is one of the most popular methods for solving problem (1). This method exhibits quadratic convergence in the case of zero residual. It is a modification of Newton's method and has the advantage that it does not require the computation of second-order derivatives.

The iterative process of the classical Gauss-Newton method can be written as following [3]:

$$x_{k+1} = x_k - (F'(x_k)^T F'(x_k))^{-1} F'(x_k)^T F(x_k), \quad k = 0, 1, \dots \tag{2}$$

Let us consider a method with successive approximation for solving the normal equation of the nonlinear least squares problem

$$F'(x)^T F(x) = 0, \tag{3}$$

built on the basis of the Gauss-Newton method:

$$x_{k+1} = x_k - A_k\, J(x_k)^T F(x_k), \tag{4}$$

$$A_{k+1} = A_k \left[2E - J(x_{k+1})^T J(x_{k+1})\, A_k\right], \quad k = 0, 1, \dots \tag{5}$$

where E is the identity operator (identity matrix), $J(x) = F'(x)$, and x_0, A_0 are initial approximations to the exact solution x^* of equation (3) and to the inverse operator $A^* = [\, J(x^*)^T J(x^*)\,]^{-1}$, respectively. The process (4)–(5) consists of two branches, executed successively one after the other.

The following theorem [9] provides the convergence conditions for the iterative process (4) (5) in the case of zero residual $F(x^*) = 0$.

Theorem 1. *Suppose the following conditions hold:*

1. *The problem* (3) *has a solution* x^* *with* $F(x^*) = 0$, *and there exists* $A^* = [\, J(x^*)^T J(x^*)\,]^{-1}$ *such that* $\|A^*\| \le B$.
2. *In the neighborhood* $\Omega_* = \{\, x : \|x - x^*\| \le r_0\}$, *the following estimates hold:*

$$\|F''(x)\| \le L, \quad \|J(x)^T - J(x^*)^T\| \le L\, \|x - x^*\|.$$

3. $\max\{\|J(x^*)\|, \, \|J(x^*)^T\|\} \le C$.
4. $$h_0 = \max\{K, \; C^2 + (B + r_0)^2\, L\, K\, (2C + Lr_0)\} < \frac{1}{r_0},$$

where

$$r_0 = \max\{\|x_0 - x^*\|, \; \|A_0 - A^*\|\},$$

$$K = C^2 + (B + r_0)\big(2.5\, L\, C + 1.5\, L^2\, r_0\big).$$

Then the sequences $\{x_k\}$ *and* $\{A_k\}, k = 0, 1, \dots$, *converge to* x^* *and* A^*, *respectively, and satisfy*

$$r_k = \max\{\, \|x_k - x^*\|, \; \|A_k - A^*\|\} \;\le\; \big(h_0\, r_0\big)^{2^k - 1} r_0, \quad k = 0, 1, \dots \tag{6}$$

Next, consider the iterative method for solving equation (3) described in [5], intended for parallel computing processors that share memory. This process is defined as:

$$x_{k+1} = x_k - A_k\, J(x_k)^T\, F(x_k), \tag{7}$$

$$A_{k+1} = A_k\left[2E - J(x_k)^T\, J(x_k)\, A_k\right], \quad k = 0, 1, \ldots \tag{8}$$

where similarly x_0 and A_0 are initial approximations to the exact solution x^* of (3) and to the inverse operator $A^* = [\,J(x^*)^T\, J(x^*)\,]^{-1}$, respectively. Like the previous method (4)–(5), the method (7)–(8) also has two branches, but they can be executed in parallel. In addition, computations on each branch can be further parallelized by using linear algebra routines, such as those in the BLAS (Basic Linear Algebra Subprograms) and LAPACK (Linear Algebra PACKage) libraries.

The sufficient convergence conditions for the method (7)–(8) are established in Theorem 2.

Theorem 2. *Suppose the following conditions hold:*

1. *The problem* (3) *has a solution* x^* *with* $F(x^*) = 0$, *and there exists* $A^* = [\,J(x^*)^T\, J(x^*)\,]^{-1}$ *such that* $\|A^*\| \le B$.
2. *In the neighborhood* $\Omega_* = \{x : \|x - x^*\| \le r_0\}$, *the following estimates hold:*

$$\|F''(x)\| \le L, \quad \|J(x)^T - J(x^*)^T\| \le L\,\|x - x^*\|.$$

3. $\max\{\|J(x^*)\|,\ \|J(x^*)^T\|\} \le C$.
4. $$h_0 = \max\{K\, r_0,\ G\} < 1,$$

where

$$r_0 = \max\{\|x_0 - x^*\|,\ \|A_0 - A^*\|\},$$

$$K = C^2 + (B + r_0)\big(2.5\,L\,C + 1.5\,L^2\, r_0\big), \quad G = C^2\, r_0 + L\,(2C + L\, r_0)\,(B + r_0)^2.$$

Then the sequences $\{x_k\}$ *and* $\{A_k\}, k = 0, 1, \ldots$ *converge to* x^* *and* A^*, *respectively, and there exist constants* γ_1 *and* γ_2 *such that the following estimates hold:*

$$\|x_k - x^*\| \;\le\; h^{c_k}\, r_0, \quad \|A_k - A^*\| \;\le\; h^{g_k}\, r_0, \tag{9}$$

where

$$c_k = \gamma_1\, t_1^k + \gamma_2\, t_2^k - 2, \quad g_k = c_{k-1} + 1, \quad c_{-1} = -1,$$

$$t_1 = \frac{1 + \sqrt{5}}{2} \approx 1.618, \quad t_2 = \frac{1 - \sqrt{5}}{2} \approx -0.618.$$

The proof of this Theorem is carried out similar to the proof of theorem for nonlinear equations given by A. Rooze [11].

Despite certain advantages of the method (7)–(8), it also has drawbacks. First, the convergence rate decreases to approximately 1.618... $<$ 2, which is lower than that of the method (4)–(5). Second, the uneven distribution of computational load due to the different numbers of computations in formulas (7) and (8) can lead to idle time on one of the processors or threads. Nonetheless, despite not fully utilizing the available computational resources during parallelization, the overall problem-solving time is still expected to decrease.

To overcome these shortcomings, one can implement the computational process without synchronizing computations across different processors. In this case, an asynchronous version of the method (7)–(8) can be written as [5]:

$$x_{k+1}^{m+1} = x_{k+1}^{m} - A_k\, J\big(x_k^{m_k-1}\big)^T\, F\big(x_{k+1}^{m}\big), \quad m = 0, 1, \ldots, m_{k+1} - 1, \tag{10}$$

$$A_{k+1} = A_k \left[2E - J\big(x_k^{m_k-1}\big)^T\, J\big(x_k^{m_k-1}\big)\, A_k\right], \quad k = 0, 1, \ldots, \tag{11}$$

where $x_k^{m_k}$ is the latest approximation to the exact solution x^* of (3), for which the inverse operator approximation A_{k-1} is applied, and $x_k^{m_k} = x_k = x_{k+1}^0$. Moreover, x_0, A_0 are the initial approximations for x^* and A^*, respectively. This method implements an asynchronous approximation of the inverse operator.

The main advantage of (10)–(11) over the parallel approximation of the inverse operator (7)–(8) is that, on architectures with shared memory, processors do not remain idle. In addition, the order of convergence of (10)–(11) equals 2, which exceeds the convergence order of about 1.618 for the parallel approximation of the inverse operator (7)–(8).

Sufficient convergence conditions for the iterative process (10)–(11) are given in the next theorem.

Theorem 3. *Suppose the following conditions hold:*

1. *The problem* (3) *has a solution* x^*, $F(x^*) = 0$, *and there exists*

$$A^* = \left[\, J(x^*)^T\, J(x^*)\, \right]^{-1}$$

such that $\|A^*\| \le B$.

2. *In the neighborhood* $\Omega_* = \{\, x : \|x - x^*\| \le r_0\}$, *the following estimates hold:*

$$\|F''(x)\| \le L, \quad \|J(x)^T - J(x^*)^T\| \le L\, \|x - x^*\|.$$

3. $\max\{\|J(x^*)\|, \|J(x^*)^T\|\} \le C$.

4.

$$h_0 = \max\{\, K\, r_0,\ G\} < 1,$$

where

$$r_0 = \max\{\|x_0 - x^*\|,\ \|A_0 - A^*\|\},$$

$$K = C^2 + (B + r_0)\big(2.5\, L\, C + 1.5\, L^2\, r_0\big), \quad G = C^2\, r_0 + L\,(2C + L r_0)\,(B + r_0)^2.$$

5. $m_k > N_1,\ N_1 \ge 2;\ k = 1, 2, \ldots$

Then the sequences $\{x_k\}$ *and* $\{A_k\}$, $k = 0, 1, \ldots$, *converge to* x^* *and* A^*, *respectively, and satisfy*

$$\begin{aligned} \|x_k - x^*\| \ &\le\ h^{\,(N_1+1)\,(2^k-1)}\, r_0, \\ \|A_k - A^*\| \ &\le\ h^{2^k-1}\, r_0, \quad k = 0, 1, \ldots \end{aligned} \tag{12}$$

The estimates (12) indicate that the iterative process (10)–(11) has quadratic convergence order [9].

2.2. Secant Method

In certain cases, due to the properties of nonlinear functions (for example, F may be non-differentiable), using the Gauss-Newton method to solve a nonlinear least squares problem can be impossible. Then it is prudent to use alternative approaches, such as the secant method, which does not require computing a derivative of the function.

In the general case, the secant method for solving a nonlinear least squares problem can be presented as [6]:

$$x_{k+1} = x_k - \left(A_k^T A_k\right)^{-1} A_k^T F(x_k), \quad k = 0, 1, \dots, \tag{13}$$

where $A_k = F\bigl(x_k,\ x_k + \alpha_k (x_{k-1} - x_k)\bigr)$, $[x, y; F]$ is the first-order divided difference of $F(x)$ at points x and y, $\alpha_k \in [0, 1]$, α_k is a nonincreasing sequence, and x_{-1}, x_0 are given initial values.

If $\alpha_k = 1$, then to find an approximate solution to problem (1), a secant method was proposed as a modification of the Gauss-Newton method [7]:

$$x_{k+1} = x_k - \left([x_k, x_{k-1}; F]^T [x_k, x_{k-1}; F]\right)^{-1} [x_k, x_{k-1}; F]^T F(x_k), \quad k = 0, 1, \dots, \tag{14}$$

where $x_{-1}, x_0 \in D$ are given initial approximations, and $[x_k, x_{k-1}; F]$ is the first-order divided difference of $F(x)$ at points x_k and x_{k-1}.

Like the Gauss-Newton method (2), the secant method (14) requires either computing the inverse matrix or solving a system of linear equations to find the solution of the nonlinear least squares problem. Thus, iterative methods with approximation of the inverse operator can also be applied to this method.

One can approximate the inverse operator A^{-1} of a linear operator A using Newton's method:

$$A_{k+1} = A_k \left(2E - A\, A_k\right), \quad k = 0, 1, \dots, \tag{15}$$

where E is the identity operator, and A_0 is an initial approximation to A^{-1} [12]. Ulm also described a Newton method with successive approximation of the inverse operator for solving the nonlinear equation

$$\begin{aligned} x_{k+1} &= x_k - A_k F(x_k), \\ A_{k+1} &= A_k\bigl(2E - F'(x_{k+1})\, A_k\bigr), \quad k = 0, 1, \dots, \end{aligned} \tag{16}$$

where A_0 is the initial approximation of $\left(F'(x^*)\right)^{-1}$, and x_0 is the initial approximation to the exact solution of $F(x) = 0$.

The secant method with successive approximation of the inverse operator for solving (1) can be written as [10]:

$$\begin{aligned} x_{k+1} &= x_k - A_k B_k^T F(x_k), \\ A_{k+1} &= A_k\bigl(2E - B_{k+1}^T B_{k+1} A_k\bigr), \quad k = 0, 1, \dots, \end{aligned} \tag{17}$$

where $B_k = [x_k, x_{k-1}; F]$; x_{-1}, x_0 are given initial approximations to x_*; A_0 is the initial approximation to $\left(F'(x_*)^T F'(x_*)\right)^{-1}$ (e.g., $A_0 = (B_0^T B_0)^{-1}$); and E is the identity matrix. If $B_k = F'(x_k)$, then one obtains a Gauss-Newton type method, which was discussed in Section 2.1.

We now examine the local convergence of (17) in the zero-residual case under classical Lipschitz conditions for first-order divided differences.

For the divided-difference operator $[x, y; F]$, a Lipschitz condition with constant L holds in domain D:

$$\|[x, y; F] - [u, v; F]\| \;\le\; L\big(\|x-u\| + \|y-v\|\big), \quad \forall x, y, u, v \in D. \tag{18}$$

Let $U(x_0, r) = \{x : \|x - x_0\| < r\}$ be a sphere of radius r centered at x_0. Then the classical Lipschitz conditions can be expressed as follows:

1. The condition $\|[x, y; F] - F'(x_0)\| \le L\big(\|x - x_0\| + \|y - x_0\|\big)$ $\forall x, y \in U(x_0, r)$ is called the *central Lipschitz condition* on $U(x_0, r)$ with Lipschitz constant L.
2. The condition $\|F'(x_0)^{-1}\,[x, x_0; F] - I\| \le L\,\|x - x_0\|$ $\forall x \in U(x_0, r)$ is called the *radial Lipschitz condition* on $U(x_0, r)$ with Lipschitz constant L.

Theorem 4. [10] Let F be a nonlinear operator defined on an open convex set $D \subset R^p$ with values in a Banach space R^m. Suppose that

1. The problem (1) has a solution $x^* \in D$, $F(x^*) = 0$, and there exists $A^* = [\,F'(x^*)^T\,F'(x^*)\,]^{-1}$ such that
$$\|A^*\| \;\le\; B. \tag{19}$$
2. In the closed sphere $\overline{U(x^*, r)} = \{\,x : \|x - x^*\| \le r_0\}$, where
$$r_0 = \max\{\|x_0 - x^*\|,\ \|x_{-1} - x^*\|,\ \|A_0 - A^*\|\}, \tag{20}$$
the following estimates hold:
$$\max\{\|F'(x^*)\|,\ \|F'(x^*)^T\|\} \;\le\; C, \tag{21}$$
$$\|F'(x^*) - [x, y; F]\| \;\le\; L\big(\|x - x^*\| + \|y - x^*\|\big). \tag{22}$$
3. The initial approximations x_{-1}, x_0, and A_0 are such that
$$q < 1, \tag{23}$$
where $q = \max\{a_1\,r_0,\ a_2\}$, $a_1 = C^2 + (B + r_0)\big(3CL + 2L^2\,r_0\big)$, and $a_2 = C^2\,r_0 + (B + r_0)^2\,[\,4CL + 4L^2\,r_0\,]$.

Then the sequences $\{x_k\}$ and $\{A_k\}, k = 0, 1, \ldots,$ converge to x^* and A^*, respectively, and satisfy

$$\|x_k - x^*\| \;\le\; q^{c_k}\,r_0, \quad \|A_k - A^*\| \;\le\; q^{g_k}\,r_0, \tag{24}$$

where $c_{-1} = -1$, $c_0 = 0$, $c_k = c_{k-2} + c_{k-1}$, $k = 1, 2, \ldots$, $g_k = c_{k-1} + 1$, $k = 0, 1, \ldots$.

Similar to the Gauss-Newton type method, one can consider an iterative process with a parallel approximation of the inverse operator for the secant method as well. This process has the form [10]:

$$\begin{aligned} x_{k+1} &= x_k - A_k\,B_k^T\,F(x_k), \\ A_{k+1} &= A_k\big(2E - B_k^T\,B_k\,A_k\big), \quad k = 0, 1, \ldots, \end{aligned} \tag{25}$$

where $B_k = [x_k, x_{k-1}; F]$; x_{-1}, x_0 are given initial approximations to the solution x_*; A_0 is an initial approximation to $\big(F'(x_*)^T\,F'(x_*)\big)^{-1}$; and E is the identity matrix.

Thus, the iterative process of the form

$$x_{k+1} = x_k - [\,A_k^T\,A_k\,]^{-1}\,A_k^T\,F(x_k), \quad k = 0,1,\ldots, \tag{26}$$

where $A_k = [x_k, x_{k-1}; F]$ is the first-order divided difference operator of $F(x)$, has a convergence order of $\frac{1+\sqrt{5}}{2} = 1.618\ldots$ in the zero-residual case [7]. In the case of the secant method, applying asynchronous approximation of the inverse operator is also highly practical because it avoids synchronization between computations, reducing the likelihood of processor idling in a parallel environment and improving overall computational efficiency.

The iterative process of the secant method with asynchronous approximation of the inverse operator has the following form:

$$\begin{aligned} x_{k+1}^{m+1} &= x_{k+1}^{m} - A_k{}^T[x_k^{m_k-1}, x_k^{m_k-2}; F]^T\,F\big(x_{k+1}^{m}\big), \quad m = 0,1,\ldots,m_{k+1}-1,\\ A_{k+1} &= A_k\big[\,2E - [x_k^{m_k-1}, x_k^{m_k-2}; F]^T[x_k^{m_k}, x_k^{m_k-1}; F]\,A_k\big], \quad k = 0,1,\ldots, \end{aligned} \tag{27}$$

where $x_k^{m_k}$ is the latest approximation to the exact solution x^*, in which the approximation of the inverse operator A_{k-1} is used; $x_k^{m_k} = x_k = x_{k+1}^0$. Moreover, x_0 and A_0 are the initial approximations for x^* and A^*, respectively.

Compared to the method with parallel approximation of the inverse operator (25), this asynchronous version of the secant method also avoids processor idle time on shared-memory architectures and provides a higher order of convergence.

2.3. Two-Step Iterative Methods

To find a solution to the nonlinear least squares problem (1), one may use a difference modification of the two-step Gauss-Newton method of the form [8]:

$$\begin{aligned} x_{k+1} &= x_k - (A_k^T A_k)^{-1} A_k^T F(x_k),\\ y_{k+1} &= x_{k+1} - (A_k^T A_k)^{-1} A_k^T F(x_{k+1}), \quad k = 0,1,\ldots, \end{aligned} \tag{28}$$

where $A_k = [x_k, y_k; F]$ is the first-order divided difference of the function $F(x)$ at points x_k and y_k; x_0, y_0 are given initial approximations.

The residual is zero, the order of convergence of the iterative process (28) equals $1+\sqrt{2}$.

Similarly to previous iterative processes, the inverse operator $(A_k^T A_k)^{-1}$ in the difference modification of the two-step Gauss-Newton method can also be approximated.

Thus, in the case of an approximate inverse operator, the two-step iterative process (28) can be written as [8]:

$$\begin{aligned} x_{k+1} &= x_k - A_k B_k^T F(x_k),\\ y_{k+1} &= x_{k+1} - A_k B_k^T F(x_{k+1}),\\ A_{k+1} &= A_k\big(2E - B_{k+1}^T B_{k+1} A_k\big), \quad k = 0,1,\ldots, \end{aligned} \tag{29}$$

where $B_k = [x_k, y_k; F]$; x_0, y_0 are given initial approximations; A_0 is the initial approximation of the operator, and E is the identity matrix.

Hence, at each iteration (29), a single matrix A_k is used to compute x_{k+1} and y_{k+1}.

In 1983, methods of the third, fifth, and sixth order that use approximation of the inverse operator were proposed for solving operator equations. The third-order method for operator equations has the form [2]:

$$\begin{aligned} &y_k = x_k - A_k F(x_k),\\ &x_{k+1} = y_k - A_k F(y_k),\\ &B_k = A_k\big(2E - [x_{k+1}, u_{k+1}; F]A_k\big),\\ &A_{k+1} = B_k\big(2E - [x_{k+1}, u_{k+1}; F]B_k\big), \quad k = 0, 1, \dots, \end{aligned} \tag{30}$$

where $u_{k+1} = x_{k+1} - \epsilon_{k+1}F(x_{k+1})$, and ϵ_{k+1} is a real parameter.

To increase the order of convergence, the efficiency, and the potential applications of the Ulm-type method given by (30), the authors of [1] proposed replacing divided differences with Frechet derivatives, leading to an improved method that can be written as:

$$\begin{aligned} &y_k = x_k - A_k F(x_k),\\ &x_{k+1} = y_k - A_k F(y_k),\\ &B_k = A_k(2E - F'(x_{k+1})A_k),\\ &A_{k+1} = B_k(2E - F'(x_{k+1})B_k), \quad k = 0, 1, \dots, \end{aligned} \tag{31}$$

Based on method (31), which was defined for operator equations, one can derive a similar method for solving nonlinear least squares problems. Thus, the third-order convergence method with successive approximation of the inverse operator for nonlinear least squares problems will have the form:

$$\begin{aligned} &y_k = x_k - A_k F'(x_k)^T F(x_k),\\ &x_{k+1} = y_k - A_k F'(y_k)^T F(y_k),\\ &B_k = A_k\big(2E - F'(x_{k+1})^T F'(x_{k+1})A_k\big),\\ &A_{k+1} = B_k\big(2E - F'(x_{k+1})^T F'(x_{k+1})B_k\big), \quad k = 0, 1, \dots, \end{aligned} \tag{32}$$

where x_0, y_0 are given initial approximations to the solution x^*, A_0 is the initial approximation to $\big(F'(x^*)^T F'(x^*)\big)^{-1}$, and E is the identity matrix.

One may also propose a method for nonlinear least squares problems that synchronously and in parallel computes both the approximate solution and the inverse operator:

$$\begin{aligned} &y_k = x_k - A_k F'(x_k)^T F(x_k),\\ &x_{k+1} = y_k - A_k F'(y_k)^T F(y_k),\\ &B_k = A_k(2E - F'(x_k)^T F'(x_k)A_k),\\ &A_{k+1} = B_k(2E - F'(x_k)^T F'(x_k)B_k), \quad k = 0, 1, \dots, \end{aligned} \tag{33}$$

Hence, each branch carries out operations, which are performed in parallel. A drawback of this method may be that computing the next approximation of the inverse operator is more time-consuming than computing the next approximation of the solution.

Consequently, one of the processors may remain idle for some time, which can be significant if the operations on the two branches differ greatly in their computational costs.

3. Numerical Results

We compare the methods with successive and parallel approximation of the inverse operator, applied to nonlinear least squares problems, assuming one processor is used in the first case and two identical processors in the second case. Let one iteration of the successive approximation of the inverse operator require time $\tau_1 + \tau_2$, where τ_1 is the time to compute the next approximation of the inverse operator, and τ_2 is the time to compute the next approximate solution. Then, when using two identical processors, one that runs the method with a parallel approximation of the inverse operator, and one that runs the successive method, one iteration of the parallel approximation method takes $\max\{\tau_1, \tau_2\}$, since both computations are performed simultaneously on different processors. Therefore, using two parallel branches for the method with a synchronous parallel approximation of the inverse operator makes sense only if $T_s - T_p > 0$, where T_s is the total runtime of the method with a successive approximation of the inverse operator, and T_p is the total runtime of the parallel method [11]. We can simulate the described approach by defining processes that engage one or two processor cores.

We examine the efficiency of the Gauss-Newton method, the secant method, and the third-order convergence method with synchronous and asynchronous approximation of the inverse operator, comparing them with methods that use successive approximation of the inverse operator. The methods were tested on the following functions [4, 7]:

Example 1. Brown's function. $n = 4$, $m = 4$,

$$F_i(x) = x_i + \sum_{j=1}^{n} x_j - n - 1, \quad i = 1, 2, \ldots, n-1,$$
$$F_n(x) = \prod_{j=1}^{n} x_j - 1;$$

$x_0 = (0.5, 0.5, 0.5, 0.5)$; $x^* = (\alpha, \alpha, \alpha, \alpha^{1-n})$, where α satisfies $n\alpha^n - (n+1)\alpha^{n-1} + 1 = 0$; $f(x^*) = 0$.

Remark. For Brown's function, the Gauss-Newton methods with an approximation of the inverse operator yielded the solution $(1, 1, 1, 1)$, whereas the classical Gauss-Newton method yielded the solution (0.868877, 0.868877, 0.868877, 1.524492). All the other methods found the solution $x^* = (1, 1, 1, 1)$.

Example 2. Freudenstein-Roth function. $n = 2$, $m = 2$,

$$F_1(x) = -13 + x_1 + ((5 - x_2)x_2 - 2)x_2,$$
$$F_2(x) = -29 + x_1 + ((x_2 + 1)x_2 - 14)x_2,$$

$x_0 = (7, 6)$; $\quad x^* = (5, 4)$; $\quad f(x^*) = 0$.

Example 3. Rosenbrock's function. $n = 8$, $m = 8$,

$$F_{2i-1} = 10(x_{2i} - x_{2i-1}^2),$$
$$F_{2i} = 1 - x_{2i-1}, \quad i = 1, 2, \ldots, n/2,$$

$x_0 = (1, 10, 1, 10, 1, 10, 1, 10)$; $\quad x^* = (1, 1, 1, 1, 1, 1, 1, 1)$; $\quad f(x^*) = 0$. We also consider options when $n = 16, m = 16$ and $n = 64, m = 64$.

For Rosenbrock's function, we consider three variations in dimension, namely $n = m = 8$, $n = m = 16$, and $n = m = 64$. We can conclude that changing the dimension of the original problem does not affect the overall distribution of method efficiency.

Since this example requires more total iterations to find the approximate solution than the other test problems, the following tables show that the asynchronous methods demonstrate significant efficiency.

For this problem, the methods with successive and synchronous parallel approximation of the inverse operator based on the third-order convergence method are quite efficient. Meanwhile, the asynchronous secant method is outperformed by the asynchronous Gauss-Newton method.

For this test problem, where $m > n$, the most efficient methods are the asynchronous Gauss-Newton and secant methods, as well as the synchronous inverse-operator approximation method based on the third-order convergence method.

Example 4. Kowalik and Osborne function [7]. $n = 4$; $m = 11$,

$$F_i(x) = y_i - \frac{x_1(u_i^2 + u_i x_2)}{u_i^2 + u_i x_3 + x_4},$$

where

i	y_i	u_i	i	y_i	u_i	i	y_i	u_i	i	y_i	u_i
1	0.1957	4.0000	4	0.1600	0.5000	7	0.0456	0.1250	10	0.0235	0.0714
2	0.1947	2.0000	5	0.0844	0.2500	8	0.0342	0.1000	11	0.0246	0.0625
3	0.1735	1.0000	6	0.0627	0.1670	9	0.0323	0.0833			

$$\begin{aligned} x_0 &= (0.25;\ 0.39;\ 0.415;\ 0.39), \\ x_* &= (0.1928\ldots;\ 0.1912\ldots;\ 0.1230\ldots;\ 0.1360\ldots), \\ f(x_*) &= 3.07505\ldots \times 10^{-4}. \end{aligned}$$

Example 5. [7] $n = 4$; $m = 7$,

$$F_i(x) = x_1 e^{t_i x_3} + x_2 e^{t_i x_4} - y_i, \quad t_i = \frac{(u_i - 425)}{195},$$

where

u	230	295	360	425	490	555	620
y	64.0	66.0	69.5	74.0	80.8	91.0	103.5

$x_0 = (25;\ 45;\ 1;\ 0)$, $x_* = (30.716958;\ 43.423609;\ 0.759299;\ -0.134355)$,

$$f(x_*) = \frac{1}{2} F(x_*)^T F(x_*) = 0.1423405.$$

Example 6. Gnedenko-Veibull distribution [7]. $n = 2$; $m = 8$,

$$F_i(x) = 1 - e^{-\left(\frac{t_i}{x_1}\right)^{x_2}} - y_i,$$

i	1	2	3	4	5	6	7	8
t_i	0.1000	0.5000	0.7000	1.0000	1.2000	1.7000	2.2000	4.5000
y_i	0.0050	0.1175	0.2173	0.3939	0.5132	0.7643	0.9111	0.99961

$$x_0 = (1, 1), \quad x_* = (1.4140, 2.0000), \quad f(x_*) = 1.3833 \cdot 10^{-7}.$$

Example 7. Wood function [7]. $n = 4$; $m = 6$,

$$F_1(x) = 10(x_2 - x_1^2), \quad F_2(x) = 1 - x_1,$$

$$F_3(x) = \sqrt{90}(x_4 - x_3^2), \quad F_4(x) = 1 - x_3,$$

$$F_5(x) = \sqrt{10}(x_4 + x_2 - 2), \quad F_6(x) = (10)^{-1/2}(x_2 - x_4),$$

$$x_0 = (-3, -1, -3, -1); \quad x_* = (1, 1, 1, 1); \quad f(x_*) = 0.$$

All calculations were carried out with an accuracy of $\epsilon = 10^{-6}$. For the secant method, the additional approximation (x_{-1}) is constructed as $x_0 + 10^{-5}$. To stop the iterative process, we check whether $\|x_{k+1} - x_k\| \leq \epsilon$.

We summarize the program execution results for the test examples in comparative tables for each type of method, where K is the number of iterations (for asynchronous methods, the number of iterations to get an approximation of the inverse operator); t - the total execution time of the program in seconds; τ - the average percentage of wait time in the main branch; Q - the number of main-branch iterations. Program execution time was determined as the average time of 50 separate runs to get a more generalized result.

Table 1

Gauss-Newton method

Example	Successive (4)-(5)		Synchronous (7)-(8)			Asynchronous (10)-(11)		
	K	t	K	t	τ	K	t	Q
1	5	0.4657	6	0.3040	15%	2	0.1860	45
2	8	1.4535	10	1.2149	9%	2	0.3572	193
3 (8)	4	0.2470	4	0.1921	4%	2	0.1396	131
3 (16)	4	0.6710	4	0.4442	16%	2	0.3950	59
3 (64)	4	1.0536	4	0.8095	17%	2	0.7620	15
4	14	1.2101	14	0.9860	13%	7	0.6966	172
5	11	1.3935	11	1.1015	28%	8	1.3551	336
6	11	0.8032	9	0.6822	5%	6	0.3455	180
7	13	1.3578	14	1.2144	7%	6	0.8195	645

Thus, comparing the methods with successive and synchronous parallel approximation of the inverse operator shows that, because the methods with synchronous parallel approximation of the inverse operator for nonlinear least squares use the approximate solution from the previous iteration to compute the next approximation of the inverse operator (rather than the current iteration as in the successive approach), these methods often require equal or more iterations to converge. Nevertheless, due to the parallelization of computations on separate processes, the overall program execution time is reduced.

Implementing the asynchronous approximation of the inverse operator for both the Gauss-Newton and secant methods makes it possible to find an approximate solution to the nonlinear least squares problem in fewer iterations and, accordingly, in less time than with the analogous successive or synchronous approximations of the inverse operator.

Table 2

Secant method

Example	Successive (17)		Synchronous (25)			Asynchronous (27)		
	K	t	K	t	τ	K	t	Q
1	5	0.4821	6	0.4423	36%	2	0.2694	11
2	8	1.2787	10	1.2288	11%	2	0.7342	103
3 (8)	5	0.4266	10	0.3690	22%	2	0.1337	12
3 (16)	5	1.3002	10	1.2893	26%	2	0.7673	7
3 (64)	6	9.5897	10	9.6478	43%	3	6.1277	8
4	16	2.8313	16	2.6275	26%	12	2.5487	95
5	11	2.9480	11	2.6786	12%	9	1.4102	187
6	11	3.5172	12	2.2798	7%	4	1.9386	135
7	13	3.6506	14	3.2904	4%	6	2.6222	370

Table 3

Third-order convergence method

Example	Successive (32)		Synchronous (33)		
	K	t	K	t	τ
1	4	0.4009	4	0.2536	16%
2	5	0.9603	7	0.6396	17%
3 (8)	3	0.2215	3	0.1754	16%
3 (16)	3	0.8591	3	0.6769	8%
3 (64)	3	1.7044	3	1.0262	20%
4	8	3.4034	3	4.0262	4%
5	11	6.4261	11	6.2134	2%
6	11	4.2412	12	4.0602	5%
7	13	3.8755	12	3.8211	1%

For the classical methods of solving least squares problems in our research, we used the built-in matrix inversion function from the NumPy library. This function is based on optimized low-level libraries such as BLAS and LAPACK, which are widely used for numerical computations. They implement highly efficient matrix algorithms such as LU and Cholesky factorizations, which allow performing matrix inversion with maximum possible speed for the given platform. In addition, NumPy supports the use of multi-core CPU architectures, which enables computation distribution among cores, greatly reducing execution time. Owing to these optimizations, classical methods can offer competitive computation speeds even for large matrices. However, for problems with large dimensions or poorly conditioned matrices, numerical instabilities may arise, which are avoided by methods that approximate the inverse operator.

Therefore, in comparing methods that use an approximate inverse operator with their classical variants, one should keep in mind that these methods do not require computing the matrix inverse, thus avoiding possible stability issues and significant growth in computational resources.

Hence, according to Tables 1–3, for the considered test examples, parallelizing the computation of the inverse operator and the approximate solution on two CPU cores results in a lower overall execution time compared to successive computation, despite a potentially higher number of iterations.

Consequently, among the approaches considered, the methods with asynchronous approximation of the inverse operator for nonlinear least squares problems show the best performance. Compared to synchronous methods, they reduce the time required to achieve a specified solution accuracy by avoiding the need for synchronization between processors. Meanwhile, the methods with synchronous approximation of the inverse operator exhibit better performance compared to the successive methods, since they allow for the distribution of computations among processors at each iteration.

4. Conclusions

In this work, methods with approximation of the inverse operator for nonlinear least squares problems were investigated. We formulated iterative processes for methods with successive, synchronous, and asynchronous approximation of the inverse operator to solve nonlinear least squares problems based on the Gauss-Newton method, the secant method, and a third-order convergence method, and studied their convergence conditions.

We carried out a software implementation of the methods that approximate the inverse operator and conducted a comparative analysis of these methods based on numerical experiments on selected test problems.

It was established that methods with asynchronous approximation of the inverse operator for nonlinear least squares problems delivered the best results among those considered, thus providing the most efficient use of available computational resources. In turn, methods with synchronous approximation of the inverse operator demonstrated higher efficiency compared to successive methods, though they often required a larger number of iterations. However, due to parallel execution of computations, the total runtime of the synchronous methods was reduced relative to the corresponding successive methods.

The results obtained serve as a basis for further research into solving nonlinear least squares problems using the approximation of the inverse operator. One direction for future work involves constructing iterative processes and analyzing the efficiency of methods with synchronous and asynchronous approximation of the inverse operator based on other foundational methods.

References

1. Argyros I. Method of Third-Order Convergence with Approximation of Inverse Operator for Large Scale Systems / I. Argyros, S. Shakhno, H. Yarmola // Symmetry. – 2020. – Vol. 12 (6). – P. 978. https://doi.org/10.3390/sym12060978
2. Argyros I. On methods with successive approximation of the inverse operator for nonlinear equations with decomposition of the operator / I. Argyros, S. Shakhno, H. Yarmola // Application Mathematics and Informatics. – 2020. https://doi.org/10.30970/vam.2019.27.10973

3. Dennis Jr J.E. Numerical methods for unconstrained optimization and nonlinear equations / J.E. Dennis Jr., R.B. Schnabel. – Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983. – 378 p.
4. More J.J. Testing unconstrained optimization software / J.J. More, B.S. Garbow, K.E. Hillstrom // ACM Trans. Math. Software. – 1981. – Vol. 7 (1). – P. 17–41. https://doi.org/10.1145/355934.355936
5. Iakymchuk R. Methods with Successive and Parallel Approximations of Inverse Operator for the Nonlinear Least Squares Problem / R. Iakymchuk, S. Shakhno // PAMM. – 2015. – Vol. 15 (1). – P. 703–704. https://doi.org/10.1002/pamm.201510274
6. Shakhno S. On a Secant Type Method for the Nonlinear Least Squares Problem / S. Shakhno, O. Gnatyshyn, R. Iakymchuk // Journal of Numerical and Applied Mathematics. – 2009. – Vol. 97. – P. 112–121.
7. Shakhno S.M. On an iterative algorithm of order 1.839... for solving the nonlinear least squares problems / S.M. Shakhno, O.P. Gnatyshyn // Applied Mathematics and Computation. - 2005. – Vol. 161 (1). – P. 253–264. – ISSN 0096-3003. https://doi.org /10.1016 /j.amc.2003.12.025
8. Шахно С. Про різницевий метод з надквадратичною збіжністю для розв'язування нелінійних задач про найменші квадрати / С. Шахно, О. Гнатишин, Р. Якимчук // Вісник Львівського Університету. Сер. прикл. матем. та інформ. – 2007. – Вип. 13. – С. 51–58.
9. Шахно С. Застосування методів з послідовною і паралельною апроксимаціями оберненого оператора до нелінійних задач найменших квадратів / С. Шахно // Інформаційні технології і системи. – 1999. – Т. 2, № 1. – С. 151–154.
10. Shakhno S. Secant type method with approximation of the inverse operator for the nonlinear least square problem / S. Shakhno, M. Havdiak, R. Iakymchuk, H. Yarmola // Вісник Львівського Університету. Сер. прикл. матем. та інф. – 2022. – Вип. 30. http://dx.doi.org/10.30970/vam.2022.30.11568
11. Rooze A.F. An iterative method for solving nonlinear equations using parallel inverse operator approximation / A.F. Rooze // Izv. Acad. Nauk Est. SSR. Fiz. Mat. – 1982. – Vol. 31, № 1. – P. 32–37.
12. Ulm S.U. On iterative methods with successive approximation of the inverse operator / S.U. Ulm // Izv. Acad. Nauk Est. SSR. Fiz. Mat. – 1967. – Vol. 6, № 4. – P. 403–411.

Article: received 10.10.2024
revised 24.10.2024
printing adoption 14.11.2024

МЕТОДИ З АСИНХРОННОЮ АПРОКСИМАЦІЄЮ ОБЕРНЕНОГО ОПЕРАТОРА ДЛЯ НЕЛІНІЙНИХ ЗАДАЧ НАЙМЕНШИХ КВАДРАТІВ

С. Шахно, Ю. Шункін, Н. Гудзеляк

Львівський національний університет імені Івана Франка,
вул. Університетська 1, Львів, 79000, Україна
e-mails: stepan.shakhno@lnu.edu.ua,
yuriy.shunkin@lnu.edu.ua, nataliia.hudzeliak@lnu.edu.ua

Розглянуто побудову та дослідження ітераційних методів для розв'язання нелінійних задач найменших квадратів, які не потребують прямого обчислення обернених матриць. Зокрема, досліджено послідовні, синхронні й асинхронні стратегії апроксимації оберненого оператора у рамках методу Гаусса-Ньютона, методу хорд та методу з третім порядком збіжності. Подано теоретичні основи цих підходів, враховуючи умови їхньої збіжності. Описано, як їх можна реалізувати в умовах середовищ з паралельними обчисленнями. Числові експерименти на серії тестових задач демонструють порівняльну ефективність кожного варіанту методу. Зокрема, з'ясовано, що методи з асинхронною апроксимацією оберненого оператора часто збігаються за меншу кількість ітерацій і з меншими витратами часу на обчислення порівняно з синхронними та послідовними аналогами.

Ключові слова: паралельні ітераційні методи, нелінійна задача найменших квадратів, апроксимація оберненого оператора, асинхронні обчислення.