Chapter 1
Parallel Multisplittings: Overview and Extensions*

R. A. Renaut† H. D. Mittelmann† Q. He†

Abstract

The philosophy of multisplitting methods is the replacement of a large-scale linear or nonlinear problem by a set of subproblems, each of which can be solved locally and independently in parallel by taking advantage of well-tested sequential algorithms. Because of this formulation most compute–intensive operations can be calculated independently and the algorithms are highly parallel. Recent developments for optimization, constrained and unconstrained, are described. Some new algorithms are faster in sequential mode than conventional algorithms.

1 Introduction

We consider herein the solution of large systems of linear and nonlinear equations by parallel multisplitting (MS) algorithms. Current MS algorithms are reviewed and important new extensions of the MS approach for the solution of constrained optimization problems are presented. All the problems to be considered can be described by one of the following equations:

Problem Formulation:

$$f(x) = 0 \tag{1}$$

$$\min_{x \in D} f(x). \tag{2}$$

Here, in (1), $f(x)$ can refer to linear equations, $Ax = b$, or nonlinear equations for $x \in \mathcal{R}^n$, $f : \mathcal{R}^n \to \mathcal{R}^m, m \leq n$. In (2), $f(x)$ may be a nonlinear functional for which a minimum is required over a subset $D \subseteq \mathcal{R}^n$, or overdetermined linear equations for which a least squares solution is sought. The precise properties of $f(x)$ are detailed as required. Here we also consider the introduction of constraints in the problem formulation.

2 Overview

2.1 Multisplitting for $Ax = b$

Iterative methods based on a single splitting, $A = M - N$, are well known [8]. Multisplittings, O'Leary and White [7], take advantage of the computational capabilities of parallel computers. A multisplitting of A is defined as follows:

Definition 2.1. *Linear Multisplitting (LMS)*

Given a matrix $A \in \mathcal{R}^{n\times n}$ and a collection of matrices $M^j, N^j, E^j \in \mathcal{R}^{n\times n}$, $j = 1 : p$, satisfying

---

*This research was supported in part by using the Intel Touchstone Delta System operated by Caltech on behalf of the Concurrent Supercomputing Consortium. Access to this facility was provided by NSF.

†Department of Mathematics, Arizona State University, AZ 85287–1804.

(i) $A = M^j - N^j$ *for each* j, $j = 1 : p$,
(ii) M^j *is nonsingular,* $j = 1 : p$,
(iii) E^j *is a nonnegative diagonal matrix,* $j = 1 : p$ *and* $\sum_{j=1}^{p} E^j = I$.

Then the collection of triples (M^j, N^j, E^j), $j = 1 : p$ *is called a multisplitting of* A *and the LMS method is defined by the iteration:*

$$x^{k+1} = \sum_{j=1}^{p} E^j (M^j)^{-1}(N^j x^k + b), \quad k = 1, \ldots. \tag{3}$$

The advantage of this method is that at each iteration there are p independent iterations of the kind

$$M^j y_j^k = N^j x^k + b, \quad j = 1 : p, \tag{4}$$

where y_j^k represents the solution to the local problem. Hence the work for each equation in (4) is assigned to one (or a set of) processor(s) and communication is required only to produce the update given in (3). In general, some (most) of the diagonal elements in E^j are zero and therefore the corresponding components of y_j^k need not be calculated. If the y_j^k are distinct, $j = 1 : p$, the method corresponds to block Jacobi and is called nonoverlapping. This will be the case if the diagonal matrices E^j have only zero and one entries. Otherwise the algorithm uses overlap and optimal overlap has to be determined, see [9], [12] and [14]. Convergence of the LMS method for various A, i.e., A an M matrix, an H matrix, symmetric positive definite, has been investigated by several researchers, for a full list of references see [3]. Relaxed multisplittings dependent on a parameter ω used to accelerate convergence have also been considered, see [1], [2]. Efficient new algorithms by Huang and O'Leary [5] use multisplittings for GMRES, conjugate gradients and other iterations.

2.2 Nonlinear splittings for $f(x) = 0$

Consider the solution of (1), $m = n$, by a Newton method. A search direction d is found as the solution of the linear system

$$\nabla f(x^k)d + f(x^k) = 0. \tag{5}$$

Hence a solution can be found by applying a LMS to the matrix $\nabla f(x)$, [13] and [6]. Alternatively, the splitting can be applied directly to $f(x)$:

DEFINITION 2.2. *Nonlinear Multisplitting (NLMS)*

For $j = 1 : p$ *let* $F^j : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}^m$ *be such that* $F^j(x, x) = f(x)$ *for all* $x \in \mathcal{R}^n$, *and* E^j *as in Definition 2.1. Then the collection of pairs* (F^j, E^j), $j = 1 : p$, *is called a nonlinear multisplitting of* f *and the NLMS method is defined by the iteration:*

$$x^{k+1} = \sum_{j=1}^{p} E^j y^{k,j}, \quad k = 0, 1, \ldots \tag{6}$$

where $y^{k,j}$ *solves* $F^j(x^k, y^{k,j}) = 0$.

This is the natural extension of Definition (2.1) since the LMS method for solving $f(x) = Ax - b$ can be split in the nonlinear sense by taking $F^j(x, y) = M^j y - N^j x - b$. For any nonlinear function $f(x)$ which can be decomposed as $f(x) = \sum_{j=1}^{n} F_j(x)$ an obvious NLMS is obtained by taking $F^j(x, y) = F_j(y) + \sum_{i \neq j} F_i(x)$ and $E^j = \frac{1}{n} I$. This leads to a parallel iterative scheme of "alternating direction" or "Peaceman-Rachford" type, [8]. But this approach does not necessarily produce subproblems of reduced size since the Newton method for each $F^j(x,y) = 0$ requires the "inversion" of a Jacobian matrix for $F^j(x,y)$ of order n, unless $E^j_{ii} = 0$ for some set of i, $1 \le i \le n$. To reduce the subproblem size suppose that (1) is solved with respect to several, possibly overlapping, blocks of variables.

DEFINITION 2.3. *Block Nonlinear Multisplitting (BNMS)*

Let $p \in \mathbf{Z}$ and for $j = 1:p$ the block S^j a nonempty subset of $\{1,2,\ldots,n\}$ such that $\cup_{j=1}^p S^j = \{1,2,\ldots,n\}$, where the S^j need not be distinct. Take E^j as in Definition 2.1, but impose $E^j_{ii} = 0$ if $i \notin S^j$, and define projections $P^j : \mathcal{R}^n \to \mathcal{R}^n$ for $j = 1:p$ by

$$P^j_i(x) = \begin{cases} x_i, & i \in S^j \\ 0, & i \notin S^j. \end{cases} \tag{7}$$

Define the functions $G^j(x,y) : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}^m$, $j = 1:p$ by

$$G^j_i(x,y) = \begin{cases} F_i((I-P^j)x + P^j y), & i \in S^j \\ F_i(x_1,\ldots,x_{i-1},y_i,x_{i+1},\ldots,x_n), & i \notin S^j, \end{cases}$$

where F_i is a component of f. Then the pairs (G^j, E^j), $j = 1:p$ define a block nonlinear multisplitting of f and the BNMS method, [3], is defined by the iteration (6) where $y^{j,k}_i$ solves $G^j_i(x^k, y^{j,k}_i) = 0$.

3 Optimization

3.1 Linear Least Squares

In (2) take $f(x) = ||Ax - b||^2$ where $A \in \mathcal{R}^{m\times n}$, $x \in \mathcal{R}^n$ and $b \in \mathcal{R}^m$.

DEFINITION 3.1. *Least Squares Multisplitting (LSMS)*

Partition A, $A = (A_1, A_2, \ldots, A_p)$, each A_j an $m \times n_j$ submatrix of A, $\sum_{j=1}^p n_j = n$ so that

$$f(x) = ||\sum_{j=1}^p A_j X_j - b||^2,$$

where $x = (X_1, X_2, \ldots, X_p)^T$ is partitioned consistently with A. Define

$$b_j(x) = b - \sum_{i\neq j}^p A_i X_i, \quad 1 \le j \le p \tag{8}$$

and take positive scalars α^k_j such that $\sum_{j=1}^p \alpha^k_j = 1$. Then the LSMS method, [10], is defined by the iteration

$$x^{k+1} = \sum_{j=1}^p \alpha^{k+1}_j Z^{j,k+1}. \tag{9}$$

where $Z^{j,k}$ is x^k with update X_j in the j^{th} position and X_j solves the subproblem

$$\min_{X_j \in \mathcal{R}^{n_j}} ||A_j X_j - b_j(x^k)||^2, \quad 1 \le j \le p. \tag{10}$$

This corresponds to a BNMS applied to the function $f(x) = Ax - b$. Parallel QR factorizations are avoided because all minimization is carried out locally. The update $b_j(x^k)$ proceeds via $b_j(x^k) = b - \sum_{i\neq j}^p B^k_i$, where $B^{k+1}_j = \alpha^{k+1}_j A_j Y^{k+1}_j + (1-\alpha^{k+1}_j)B^k_j$, and hence no update x^{k+1}_j is needed. Likewise, error checking for the determination of convergence can be obtained without explicitly updating x^{k+1} in each processor. A search direction can also be incorporated in the solution of the subproblems (10) with minor modifications of the algorithm and without significant extra communication.

TABLE 1

Minimization of $f(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n}(e^{i*x_i/n} - e^{-i^2/n})^2$, $n = 1024$, initialized with $x_i = -(i + 0.1), (1 \leq i \leq n)$, using NOMS. Number of processors of Intel Touchstone Delta, p, configured as $p = r \times c$ mesh. Elaspsed time in seconds t. # of iterations K. Speedup, S, with respect to time on one processor without splitting. Subproblems minimized using variable metric method with BFGS update.

$p = r \times c$	K	t	S
$1 = 1 \times 1$	5	82.952	1
$2 = 2 \times 1$	15	20.270	4.1
$4 = 4 \times 1$	15	12.137	6.8
$8 = 8 \times 1$	15	8.142	10.2
$16 = 8 \times 2$	12	4.547	18.2
$32 = 8 \times 4$	9	2.300	36.1
$64 = 8 \times 8$	15	2.867	28.9
$128 = 4 \times 32$	12	1.691	49.0
$256 = 8 \times 32$	40	2.997	27.7
$512 = 16 \times 32$	33	1.665	49.8

3.2 Unconstrained Minimization

Now consider the nonlinear form of (2), $m = 1$, D a bounded neighborhood of x^*, a unique stationary point of $f(x)$, and suppose that f has positive semidefinite Hessian $G = \nabla^2 f(x)$.

DEFINITION 3.2. *Nonlinear Optimization Multisplitting (NOMS)*

Let $f_j(x, Y_j) = f(\bar{x}_j)$, $f_j : \mathcal{R}^n \times \mathcal{R}^{n_j} \rightarrow \mathcal{R}^n$, $j = 1 : p$, x partitioned as above, and $\bar{x}_j$ the variable partition of x with the j^{th} component replaced by Y_j. Then the functions f_j form a NOMS of f and the stationary point is found iteratively from (9) where the Z^j are the solutions of the subproblems:

$$\min_{Y_j \in \mathcal{R}^{n_j}} f_j, \quad 1 \leq j \leq p. \tag{11}$$

In a sequential environment, the minimization step does not proceed concurrently and hence the minimization can use all updated values available, as in a Gauss-Seidel iteration scheme. Details of the algorithm and proofs of convergence for exact and inexact Newton methods can be found in [10]. Results in Table 1 demonstrate that more than linear speedup is possible and hence the problem is solved more efficiently in partitioned form. In this particular case f is a separable function and so one should expect to see convergence in one outer iteration if the subproblems are iterated to convergence. In the implementation here this is not the case since a reduced tolerance is used on the subproblems.

3.3 Constrained Minimization

In this case the minimization of f in (2) is over a domain $D \subseteq \mathcal{R}^n$ defined by the constraints $g(x) \leq 0$, where it is assumed that equality constraints $h(x) = 0$ are replaced by two inequality constraints $h(x) \leq 0$ and $-h(x) \leq 0$. In [4] it was shown that, under suitable assumptions, (2) can be replaced by an optimization problem with separable constraints, which are only one–sided bounds on the variables representing the Lagrange multipliers in the Wolfe dual of (2). Then (2) takes the form

$$\min_{u \geq 0} \{-f(x) - u^T g(x) + \frac{1}{2}\gamma||\nabla f(x) + \nabla g(x)u||_2^2\}. \tag{12}$$

Here, γ is a positive factor which only needs to satisfy $\gamma \geq \bar{\rho}^{-1}$ where $\bar{\rho} \geq 0$ is the smallest eigenvalue of $\nabla_{xx} L(\bar{x}, \bar{u})$, $(\bar{x}, \bar{u})$ a Kuhn–Tucker point of (2) and $L(x, u) = f(x) + u^T g(x)$. Hence (12) can be rewritten

$$\min_{y_j \geq 0, j > n} q(y), \quad y = (x, u)^T, \tag{13}$$

and a solution can be found with a suitable generalization of the NOMS algorithm described above. Results and analysis of the algorithm for (13) will be presented in [11].

4 Conclusions

The major advantage of the new classes of MS methods for optimization is that minimization procedures occur on independent subproblems which can be solved in parallel and a large variety of sequential expertise can be employed for their solution. The major consideration in parallel is to provide an efficient means for using the subproblem solutions to update the global solution. This can also be posed as a minimization procedure which can be handled by one processor while the remaining processors deal with the individual subproblem solutions. In this way synchronization of the individual steps is not enforced and convergence is accelerated. Furthermore, a constrained problem can also be treated in a multisplitting manner using a technique introduced in [4] for replacing the constrained problem by a problem with separable constraints.

References

[1] Wang Deren, *On the Convergence of the Parallel Multisplitting AOR Algorithm*, Linear Algebra Appl. 154–156, (1991) pp. 473–486.

[2] A. Frommer and G. Mayer, *Convergence of Relaxed Parallel Multisplitting Methods*, Linear Algebra Appl. 119, (1989), pp. 141–152.

[3] A. Frommer, *Lösung linearer Gleichungssysteme auf Parallelrechnern*, View Verlag, Braunschweig, Kapitel 9, (1990).

[4] S.–P. Han and O. L. Mangasarian, *A Dual Differentiable Exact Penalty Function*, Mathematical Programming 25, (1983), 293-301.

[5] C.–M.Huang and D. P. O'Leary, *A Krylov Multisplitting Algorithm for Solving Linear Systems of Equations*, Linear Algebra and Its Applications, to appear (1994).

[6] C.–M.Huang and D. P. O'Leary, *A Parallel Inexact Newton Method using a Krylov Multisplitting Algorithm*, University Of Maryland, (1994).

[7] D. P. O'Leary and R. E. White, *Multi-Splitting of Matrices and Parallel Solution of Linear Systems*, SIAM J. Alg. Disc. Meth. 6, (1985), pp. 630–640.

[8] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, New York, Academic Press, (1970).

[9] B. Pohl, *Ein Algorithmus zur Lösung von Anfangswertproblemen auf Parallelrechnern*, Informatik-Dissertationen ETH Zürich NR. 37, (1992).

[10] R. A. Renaut and Qing He, *Parallel Multisplittings for Unconstrained Optimization*, in prep. (1994).

[11] R. A. Renaut and H. D. Mittelmann, *Parallel Multisplittings for Constrained Optimization*, in prep. (1994).

[12] D. B. Szyld and M. T. Jones, *Two-Stage and Multisplitting Methods for the Parallel Solution of Linear Systems*, SIAM J. Matrix Anal. Appl. Vol. 13, No. 2, (1992), pp. 671-679.

[13] R. E. White, *Parallel Algorithms for Nonlinear Problems*, SIAM J. Alg. Disc. Meth. **7**, (1986), pp. 137–149.

[14] R. E. White, *Multisplitting of a Symmetric Positive Definite Matrix*, SIAM J. Matrix Anal. Appl., Vol. 11, No. 1, (1990), pp. 69–82.