Nonlinear Schwarz iterations with Reduced Rank Extrapolation

Sébastien Duminil,
Hassane Sadok, and Daniel B. Szyld

Report 14-09-01
September 2014

Department of Mathematics
Temple University
Philadelphia, PA 19122

This report is available in the World Wide Web at
`http://www.math.temple.edu/~szyld`

NONLINEAR SCHWARZ ITERATIONS WITH REDUCED RANK EXTRAPOLATION*

SÉBASTIEN DUMINIL†, HASSANE SADOK†, AND DANIEL B. SZYLD‡

Abstract. Extrapolation methods can be a very effective technique used for accelerating the convergence of vector sequences. In this paper, these methods are used to accelerate the convergence of Schwarz iterative methods for nonlinear problems. Some convergence analysis is presented, and it is shown numerically that certain extrapolation methods can indeed be very effective in accelerating the convergence of Schwarz methods.

Key words. Vector extrapolation, Schwarz methods, domain decomposition, iterative methods, nonlinear problems

AMS subject classifications. 65B05, 65J15, 35J65

1. Introduction.

A useful iterative method for solving large elliptic problems is to subdivide the domain into many (overlapping) subdomains and solve smaller elliptic problems on each subdomain in parallel. For example, the restricted additive Schwarz iterative method (RAS) solves a boundary value problem for a partial differential equation approximately by dividing it into boundary value problems on the smaller domains and adding the results corresponding to the subdomains without the overlap; see section 2 for a full description. Such methods are well-known to provide very efficient preconditioners as well. Different boundary conditions on the artificial interfaces, such as Dirichlet or Robin conditions, have been developed; see, e.g., [6, 7, 17, 19, 28].

Extrapolation methods are used for accelerating the convergence of large class of vector sequences; see, e.g., the review article by Smith, Ford and Sidi [27], or the book by Brezinski and Redivo-Zaglia [2]. In particular, these methods are employed to accelerate the convergence of fixed point iterative techniques for linear and nonlinear systems of equations; see, e.g [2, 8, 13, 27]. Vector extrapolation methods can be divided into two families, polynomial methods and epsilon algorithms. We consider in this paper the first family which include the minimal polynomial extrapolation (MPE) method of Cabay and Jackson [3], Vorobyev [29] and Germain-Bonne [11] methods, the reduced-rank-extrapolation (RRE) method of Eddy [9], Mesina [16] and Kaniel and Stein [14] and the modified minimal polynomial extrapolation (MMPE) method due to Brezinski [1], Pugatchev [18], and Sidi, Ford and Smith [26]; see also [24, 27], and references therein. The convergence and stability analysis of the methods MPE, RRE, MMPE was treated by Sidi and his co-authors in [21, 23, 26].

In this paper, we apply the reduced-rank-extrapolation method to the sequence of vector iterates produced by RAS, and by its multiplicative counterpart. We show experimentally that this extrapolation can indeed provide a substantial acceleration, measured both in number of iterations, and in CPU times. We have in mind applications to nonlinear problems, but we illustrate the applicability of our proposed approach to linear problems as well. For the nonlinear problems, we discuss two ap-

---

*This version dated 1 September 2014

†Laboratoire de Mathématiques Pures et Appliquées, Université du Littoral, Centre Universitaire de la Mi-Voix, B.P. 699, 62228 Calais Cedex, France ({duminil,sadok}@lmpa.univ-littoral.fr).

‡Department of Mathematics, Temple University (038-16), 1805 N. Broad Street, Philadelphia, Pennsylvania 19122-6094, USA (szyld@temple.edu). Research supported in part by the U.S. National Science Foundation under grants DMS-1115520 and DMS-1418882.

proaches. We discuss a nonlinear version of the Schwarz iterations, and its acceleration using RRE. We also consider Schwarz-Newton methods, accelerated with RRE.

2. Iterative Schwarz Methods. We consider the following problem

$$\begin{aligned} \mathcal{L}(u) &= f \text{ in } \Omega, \\ \mathcal{B}u &= g \text{ on } \partial\Omega. \end{aligned} \tag{2.1}$$

where $\mathcal{L}$ is a linear or nonlinear operator, $\mathcal{B}$ is a boundary operator and Ω is a bounded domain.

For iterative Schwarz methods one considers the domain Ω as a union of (possibly overlapping) subdomains $\Omega_1, \cdots, \Omega_p$. These subdomains may have common boundaries with Ω, namely $\Gamma_j = \partial\Omega_j \cap \partial\Omega$, and the rest of the boundary $\partial\Omega_j$ would consist of $\Gamma_{ij} = \partial\Omega_i \cap \Omega_j$, sometimes called artificial interfaces.

The classical alternating Schwarz method, first described in [20], consists of the following iteration, starting with an initial approximation u^0 to the solution of (2.1).

For $n = 0, \ldots$
For $j = 1, \ldots, p$,

$$\text{Solve} \begin{cases} \mathcal{L}(v^{n+\frac{j}{p}}) = f \text{ in } \Omega_j \ , \\ \mathcal{B}v^{n+\frac{j}{p}} = g \text{ on } \partial\Gamma_j \ , \\ v^{n+\frac{j}{p}} = u^{n+\frac{j-1}{p}} \text{ on } \partial\Omega_j \setminus \Gamma_j \ . \end{cases}$$

$$\text{Update} \begin{cases} u^{n+\frac{j}{p}} = v^{n+\frac{j}{p}} \text{ in } \Omega_j \ , \\ u^{n+\frac{j}{p}} = u^{n+\frac{j-1}{p}} \text{ in } \Omega \setminus \Omega_j \ . \end{cases}$$

One cycle of this iteration, i.e., the j-loop, has been used successfully as a preconditioner and it is called multiplicative Schwarz (MS); see e.g., [15, 19, 28]. An alternative, more amenable to parallel processing is additive Schwarz (AS) iteration (in the Richardson version with Richarson parameter $0 < \tau \leq 1/p$) as follows.

For $n = 0, \ldots$
For each $j = 1, \ldots, p$,

$$\text{Solve} \begin{cases} \mathcal{L}(v^{n+1,j}) = f \text{ in } \Omega_j \ , \\ \mathcal{B}v^{n+1,j} = g \text{ on } \partial\Gamma_j \ , \\ v^{n+1,j} = u^n \text{ on } \partial\Omega_j \setminus \Gamma_j \ . \end{cases}$$

Compute $w^{n+1} = v^{n+1,1} + \cdots + v^{n+1,p}$.
Update $u^{n+1} = (1 - p\tau)u^n + \tau w^{n+1}$.

The additive Schwarz preconditioner consists of simply updating $u^{n+1} = w^{n+1}$. It is not hard to see that without overlap, additive Schwarz corresponds to the Jacobi method and multiplicative Schwarz corresponds to the Gauss-Seidel method.

In the case where the operator $\mathcal{L}$ is linear, a discretization of the equation (2.1) leads to a linear system of equations of the form

$$Au = f, \tag{2.2}$$

where we abuse the notation, and identify the vector u with the function u.

We describe additive and multiplicative Schwarz methods as follows. We group the unknowns into subsets, $u_{(j)} = R_j u$, $j = 1, \cdots, p$, where R_j are rectangular restriction matrices such that each entry u_i of the vector u is contained in at least one subvector $u_{(j)}$. Coefficient matrices for subdomain problems are defined by $A_j =$

$R_j A R_j^T$. The additive Schwarz method constructs the sequence of approximations $\{u^n\}_{n\in\mathbb{N}}$ by setting

$$u^{n+1} = u^n + \sum_{j=1}^{p} R_j^T A_j^{-1} R_j (f - Au^n),\ n = 0, 1, \dots \tag{2.3}$$

(without the Richardson acceleration). Similarly, one iteration of the multiplicative Schwarz method is defined by

$$u^{n+\frac{j}{p}} = u^{n+\frac{j-1}{p}} + R_j^T A_j^{-1} R_j (f - Au^{n+\frac{j-1}{p}}),\ j = 1, \cdots, p. \tag{2.4}$$

In general, the additive Schwarz iteration (2.3) does not converge, without Richardson acceleration. The restricted additive Schwarz method (RAS), first presented in [5], provides a different alternative. One further defines matrices $\tilde{R}_1, \cdots, \tilde{R}_p$, where $\tilde{R}_j$ consists of zeroes and ones, in such a way that

$$\sum_{j=1}^{p} \tilde{R}_j^T R_j = I.$$

These restriction matrices correspond usually to a nonoverlapping partition of the unknowns. Given an initial vector u^0, the restricted additive Schwarz (RAS) algorithm is then given by

$$u^{n+1} = u^n + \sum_{j=1}^{p} \tilde{R}_j^T A_j^{-1} R_j (f - Au^n),\ n = 0, 1, \dots \tag{2.5}$$

and it is convergent; see [10]. With the same notation, we can describe the restricted multiplicative Schwarz (RMS) algorithm as follows,

$$u^{n+\frac{j}{p}} = u^{n+\frac{j-1}{p}} + \tilde{R}_j^T A_j^{-1} R_j (f - Au^{n+\frac{j-1}{p}}),\ j = 1, \dots, p. \tag{2.6}$$

Consider now the system of nonlinear equations

$$G(u) = u, \tag{2.7}$$

where $G : \mathbb{R}^n \to \mathbb{R}^n$. Using the same notation as before, the subdomain problems are defined by

$$G_j(X) = R_j G(R_j^T(X)).$$

The corresponding (nonlinear) restricted Schwarz methods can be defined by

$$w^{n+1} = w^n + \sum_{j=1}^{p} \tilde{R}_j^T G_j(R_j(w^n)),\ n = 0, 1, \dots, \tag{2.8}$$

for the additive method, and one iteration of the multiplicative method can be defined by

$$w^{n+\frac{j}{p}} = w^{n+\frac{j-1}{p}} + \tilde{R}_j^T G_j(R_j(w^{n+\frac{j-1}{p}})),\ j = 1, \dots, p. \tag{2.9}$$

In our work, we also consider for the solution of (2.7) Schwarz-Newton methods, where in each subdomain, the nonlinear problem is solved by a Newton (or inexact Newton) iteration, cf. [4].

3. Reduced Rank extrapolation method.

3.1. Definition. Let $\{u^n\}_{n\in\mathbb{N}}$ be a sequence of vectors in $\mathbb{R}^N$ and define the first and second forward differences

$$\Delta u^n = u^{n+1} - u^n, \quad n = 0, 1, 2, \ldots,$$
$$\Delta^2 u^n = \Delta u^{n+1} - \Delta u^n, \quad n = 0, 1, 2, \ldots$$

Let $\Delta^i U_n (i = 1, 2)$ denote the matrices whose columns are $\Delta^i u^0, \ldots, \Delta^i u^{n-1}$.

Eddy [9] defines the Reduced Rank extrapolation (RRE) method by setting

$$t^n = u^0 - \Delta U_n \; \Delta^2 U_n^+ \; \Delta u^0, \tag{3.1}$$

where $\Delta^2 U_n^+$ is the Moore-Penrose generalized inverse of $\Delta^2 U_n$ defined by

$$\Delta^2 U_n^+ = (\Delta^2 U_n^T \; \Delta^2 U_n)^{-1} \; \Delta^2 U_n^T.$$

He proves that, when applied to the sequence of vectors $\{u^n\}_{n\in\mathbb{N}}$, t^n is a best approximation of the limit or the anti-limit of the sequence $\{u^n\}_{n\in\mathbb{N}}$.

It is easy to prove that t^n exists and is unique if and only if $\Delta^2 U_n$ has full rank, i.e., if the Moore-Penrose generalized inverse is well defined. We shall propose in the next subsection, efficient algorithms to compute t^n, for different values of n.

Herein, we define a new approximation $\tilde{t}^n$ as follows

$$\tilde{t}^n = u^1 - \Delta U_n' \Delta^2 U_n^+ \Delta u^0,$$

where $\Delta U_n'$ denotes the matrix whose columns are $\Delta u^1, \cdots, \Delta u^n$, i.e., we have the columns of ΔU_n shifted by one position.

In [13], the generalized residual of t^n was defined as

$$\tilde{r}(t^n) = \tilde{t}^n - t^n, \tag{3.2}$$

which can be expressed as

$$\tilde{r}(t^n) = \Delta u^0 - \Delta^2 U_n \; \Delta^2 U_n^+ \; \Delta u^0. \tag{3.3}$$

Notice that $\tilde{r}(t^n)$ is obtained by projecting orthogonally Δu^0 onto the subspace generated by the vectors $\Delta^2 u^0, \ldots, \Delta^2 u^{n-1}$.

We end this section with a result giving an easily computable expression for the norm of the generalized residual.

THEOREM 3.1. *Let C_{n+1} be the $N \times (n+1)$ matrix defined by $[\Delta u^0, \Delta^2 U_n]$. If C_{n+1} is of full rank, then*

$$\|\tilde{r}(t^n)\|_2^2 = \frac{1}{e_1^T (C_{n+1}^T C_{n+1})^{-1} e_1}. \tag{3.4}$$

Proof. We have

$$\tilde{r}(t^n) = P_n \Delta u^0,$$

where P_n is the orthogonal projector defined by $P_n = I - \Delta^2 U_n \Delta^2 U_n^+$. Since $P_n^2 = P_n$ and $P_n^T = P_n$, we have

$$\begin{aligned} \|\tilde{r}(t^n)\|^2 &= (P_n \Delta u^0, P_n \Delta u^0) = (\Delta u^0, P_n \Delta u^0) \\ &= (\Delta u^0, \Delta u^0) - (\Delta u^0, \Delta^2 U_n \; (\Delta^2 U_n^T \; \Delta^2 U_n)^{-1} \; \Delta^2 U_n^T \Delta u^0). \end{aligned} \tag{3.5}$$

We observe that the right hand side of equation (3.5) is the Schur complement of $(\Delta^2 U_n^T \Delta^2 U_n)$ in the matrix $(C_{n+1}^T C_{n+1})$, where

$$C_{n+1}^T C_{n+1} = \begin{bmatrix} (\Delta u^0, \Delta u^0) & (\Delta u^0)^T \Delta^2 U_n \\ \Delta^2 U_n^T \Delta u^0 & \Delta^2 U_n^T \Delta^2 U_n \end{bmatrix}.$$

We can factor this matrix into a product of a block upper and a block lower triangular matrices (block UL factorization)

$$C_{n+1}^T C_{n+1} = \begin{bmatrix} 1 & (\Delta u^0)^T \Delta^2 U_n (\Delta^2 U_n^T \Delta^2 U_n)^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} \|\tilde{r}(t^n)\|^2 & 0 \\ \Delta^2 U_n^T \Delta u^0 & (\Delta^2 U_n^T \Delta^2 U_n) \end{bmatrix}.$$

From the block UL factorization of the matrix $C_{n+1}^T C_{n+1}$ and by taking determinants on both sides yields the formula that gives $\|\tilde{r}(t^n)\|^2$ as the ratio of two determinants, namely

$$\|\tilde{r}(t^n)\|^2 = \frac{det(C_{n+1}^T C_{n+1})}{det(\Delta^2 U_n^T \Delta^2 U_n)} = \frac{1}{e_1^T (C_{n+1}^T C_{n+1})^{-1} e_1}. \quad \square$$

3.2. Algorithm for the RRE method. Sidi [23] presented an efficient and stable implementation for MPE and RRE. In this section, we propose a new implementation of the algorithm for the RRE method, based on an Hessenberg matrix, as is the case for GMRES. A key feature of this implementation is the solution of least-squares problem using a QR factorization. A generic construction is given below.

Let us assume that ΔU_{n+1} has full rank, i.e., $\text{rank}(\Delta U_{n+1}) = n+1$. Then, it has a QR-factorization $\Delta U_{n+1} = Q_{n+1} R_{n+1}$, where $Q_{n+1} \in \mathbb{R}^{N \times (n+1)}$ is unitary and $R_{n+1} \in \mathbb{R}^{(n+1)\times(n+1)}$ is upper triangular with positive diagonal entries,

$$Q_{n+1} = [q_0 | q_1 | \cdots | q_n] \in \mathbb{R}^{N \times (n+1)}, \quad Q_{n+1}^T Q_{n+1} = I_{(n+1)\times(n+1)},$$

$$R_{n+1} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & \cdots & r_{0n} \\ & r_{11} & r_{12} & \cdots & r_{1n} \\ & & r_{22} & \cdots & r_{2n} \\ & & & \ddots & \vdots \\ & & & & r_{nn} \end{bmatrix}, \quad r_{ii} > 0, i = 0, 1, \ldots, n.$$

Note that Q_{n+1} is obtained from Q_n by appending one column (the vector q_n) to the end of the latter. Similarly, R_{n+1} is obtained by R_n by appending one row of zeros and one column ($[r_{0n}, r_{1n}, \cdots, r_{nn}]$) to the end of the latter.

We show now that we can write $\Delta^2 U_n$ in the form

$$\Delta^2 U_n = Q_{n+1} H_n, \tag{3.6}$$

where H_n is a $(n+1) \times n$ Hessenberg matrix.

To that end, we know that

$$\Delta U_n = Q_n R_n, \tag{3.7}$$

and thus

$$\Delta U'_n = Q_{n+1} H'_n,$$

where

$$H'_n = \begin{bmatrix} r_{01} & r_{02} & r_{03} & \cdots & r_{0n} \\ r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ & r_{22} & r_{23} & \cdots & r_{2n} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & r_{n-1,n} \\ & & & & r_{nn} \end{bmatrix}.$$

Then

$$\Delta^2 U_n = \Delta U'_n - \Delta U_n = Q_{n+1} H'_n - Q_{n+1} \begin{bmatrix} R_n \\ 0 \end{bmatrix}.$$

We finally set

$$\begin{aligned} H_n &= H'_n - \begin{bmatrix} R_n \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} r_{01} - r_{00} & r_{02} - r_{01} & r_{03} - r_{02} & \cdots & r_{0n} - r_{0,n-1} \\ r_{11} & r_{12} - r_{11} & r_{13} - r_{12} & \cdots & r_{1n} - r_{1,n-1} \\ & r_{22} & r_{23} - r_{22} & \cdots & r_{2n} - r_{2,n-1} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & r_{n-1,n} - r_{n-1,n-1} \\ & & & & r_{nn} \end{bmatrix}. \end{aligned} \tag{3.8}$$

By using (3.6), (3.7) and the fact that Q_{n+1} is unitary, (3.1) becomes

$$t^n = u^0 + Q_n R_n \beta$$

where β is solution of

$$H_n^+ \beta = -H_n^T r_{00} e_1,$$

i.e., β is obtained by solving the corresponding least squares problem. The details of the resulting algorithm are summarized in Table 3.1. The QR-factorization can be carried out inexpensively by applying the modified Gram-Schmidt process to the vectors $u^0, u^1, \ldots, u^{n+1}$.

When solving linear or nonlinear systems of equations, the basic RRE algorithm in Table 3.1 becomes increasingly expensive as n grows. In fact, the work and storage requirements grow linearly with the number of iteration steps. A practical approach is to restart the RRE algorithm after a fixed number of iterations.

3.3. Application of RRE to linear systems. Consider the linear system of equation (2.2). We denote by $u^\star$ its unique solution. We apply the RRE method to the sequence $\{u^n\}_{n\in\mathbb{N}}$ constructed by a classical stationary iterative method

$$u^{n+1} = Bu^n + f, \tag{3.9}$$

Step 0.	Input: the vectors $u^0, u^1, \ldots, u^{n+1}$
Step 1.	Compute $\Delta u^i = u^{i+1} - u^i, \quad i = 0, \ldots, n.$
	Set $\Delta U_{n+1} = \left[\Delta u^0 \,|\, \Delta u^1 \,|\, \cdots \,|\, \Delta u^n \right]$,
	Compute the QR factorization of $\Delta U_{n+1} = Q_{n+1} R_{n+1}$.
	$(\Delta U_n = Q_n R_n$ is contained in $\Delta U_{n+1})$.
	Compute H_n using (3.8)
Step 2.	Solve the least squares problem $\min \|H_n \beta + r_{00} e_1\|$
Step 3.	Compute $t^n = u^0 + Q_n(R_n \beta)$.

TABLE 3.1
Implementation of the RRE Algorithm

where either $B = I - A$, or $B = I - M^{-1}A$, with M being a preconditioner.

From (3.9), we have $\Delta u^n = f - Au^n = r(u^n)$, the residual of the vector u^n. Therefore using (3.3) and (3.9), it follows that the generalized residual of the approximation t^n is the true residual, i.e., we have

$$\tilde{r}(t^n) = r(t^n) = f - At^n.$$

Note also that, since $\Delta^2 u^n = -A\Delta u^n$, we have $\Delta^2 U_n = -A\Delta U_n$.

Let d be the degree of the minimal polynomial P_d of B for the initial error vector $u^0 - u^\star$. The polynomial P_d is also the minimal polynomial of B for Δu^0. Therefore, the matrices ΔU_n and $\Delta^2 U_n$ have full rank for $n \leq d$.

Let $W_n = span\{\Delta u^0, \ldots, \Delta u^{n-1}\}$. When applied to the sequence generated by (3.9), the residuals $r^n = f - At^n$ satisfy the relations

$$r^n - \Delta u^0 \in AW_n \qquad \text{and} \quad r^n \perp AW_n.$$

In fact, it was shown by Sidi [22] that $W_n = K_n(A, r^0)$, the Krylov subspace induced by the matrix A and the vector r^0; he also proved in the same reference that the RRE method is an orthogonal projection and that it is equivalent to the GMRES method, in exact arithmetic.

3.4. Analysis and implementation of the RRE method. We study the application of the RRE method to (linear or nonlinear) system of equations of the form

$$f(u) = 0; \; f : \mathbb{R}^N \to \mathbb{R}^N \tag{3.10}$$

whose solution is denoted by $u^\star$. Fixed point methods for the solution of such systems depends on the choice of an appropriate function Φ such that $\Phi(u^\star) = u^\star$. In some cases, this would be a preconditioned system, equivalent to (3.10). Then, starting with a suitable vector u^0, an initial approximation to $u^\star$, the sequence $\{u^n\}_{n \in \mathbb{N}}$ can be generated by fixed-point iterative, so that

$$u^{n+1} = \Phi(u^n), \; n = 0, 1, \cdots \quad \Phi : \mathbb{R}^N \to \mathbb{R}^N, \tag{3.11}$$

and, in case of convergence, $\lim_{n \to \infty} u^n = u^\star$.

The algorithm which wish to consider is the following,

- choose a starting approximation x^0,
- at the iteration n, we set $u^0 = x^n$ and $u^{i+1} = \Phi(u^i)$ for $i = 0, \cdots, q_n$

- compute x^{n+1} such that

$$x^{n+1} = t^{q_n} = u^0 - \Delta U_{q_n} \Delta^2 U_{q_n}^+ \Delta u^0, \tag{3.12}$$

where q_n is the degree of the minimal polynomial of $\Phi'(u^\star)$ for the vector $x^n - u^\star$.

We present now a result showing quadratic convergence for this RRE method. We first define a quantity we need. We set

$$\alpha_n(x) = \sqrt{det(H_n^*(x)H_n(x))},$$

where

$$H_n(x) = \left[\frac{\Phi(x) - x}{\|\Phi(x) - x\|}, \cdots, \frac{\Phi^n(x) - \Phi^{n-1}(x)}{\|\Phi^n(x) - \Phi^{n-1}(x)\|} \right],$$

and recall the following result from [12].

THEOREM 3.2. *Let $J = \Phi'(u^\star)$, and assume that Φ satisfies the following conditions*

- *The matrix $J - I$ is nonsingular.*
- *The Frechet derivative Φ' of Φ satisfies the Lipschitz condition*

$$\|\Phi'(x) - \Phi'(y)\| \leq L \|x - y\|, \text{ for } x, y \in D$$

where D is an open and convex subset of $\mathbb{C}^p$,

and if

$$\exists \alpha > 0, \exists K, \alpha_{q_n}(x^n) > \alpha, \forall n \geq K,$$

then there exists a neighbourhood V of $u^\star$ such that $\forall x^0 \in V$

$$\left\|x^{n+1} - u^\star\right\| = O\left(\|x^n - u^\star\|^2\right).$$

Note that by the definition of q_n, there exists a neighbourhood of $u^\star$ such that $\Delta u^0, \cdots, \Delta u^{q_n - 1}$ are linearly independent and $\Delta u^0, \cdots, \Delta u^{q_n}$ are almost linearly dependent. In practice, the value of q_n is not known. Instead we will use the quantity l_n is such that $\|\tilde{r}(t^{l_n})\| = 0$, and have the following result.

THEOREM 3.3. *With the same hypothesis as in Theorem 3.2, and if we replace q_n by l_n such that $\|\tilde{r}(t^{l_n})\| = 0$, then there exists a neighbourhood V_1 of $u^\star$ such that $\forall x^0 \in V_1$*

$$\left\|x^{n+1} - u^\star\right\| = O\left(\|x^n - u^\star\|^2\right).$$

While Theorem 3.3 is more applicable than Theorem 3.2, still the quantity l_n may be numerically difficult to determine. Hence, in our experiments we use $q_n = q$ fixed and small. Furthermore, the function Φ, i.e., an equivalent preconditioned system is chosen such that the new fixed-point iteration is convergent. For these choices, the application of the extrapolation scheme after a number of initial fixed-point iterations will exhibit the quadratic convergence shown in Theorem 3.3, and it is therefore recommended.

We end this section discussing the operation count (multiplications and additions) and storage requirements to compute the approximation t^n. In practice, the dimension

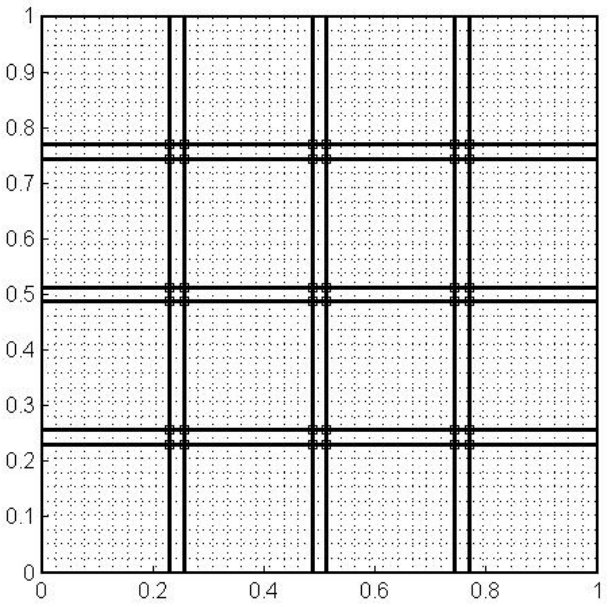

FIG. 4.1. *Unit square with $p = 16$ overlapping subdomains*

of the space, N, i.e., the number of unknowns is very large, while the iteration count n is relatively small. To compute t^n with the RRE method, the vectors $u^0, \cdots, u^{n+1}$ are required and we use the QR-implementation described in Table 3.1. Thus, one needs $n+1$ evaluation of the function Φ. In the case of linear systems of equations, this means that $n+1$ matrix vector products are required, together with $n+1$ function evaluations, and $(n+1)N$ vectors to storage.

4. Numerical Experiments. We present numerical experiments comparing the performance of Schwarz iterations with those accelerated with the reduce rank extrapolation method. Computational results are given both in terms of number of iterations and CPU time. We give results for four different applications. Two of the problems are linear, which we present mostly for illustrative purposes. All computational experiments presented were carried out using MATLAB 7.13. For all tests, the iteration process is terminated as soon as the norm of the error is less than 10^{-8} or the number of iterations reaches 1000.

4.1. Application to the Poisson Problem. In the first test, we consider the unit square $\Omega = [0,1] \times [0,1]$ decomposed uniformly into $p = q^2$ overlapping subdomains. The case $p = 16$ is illustrated in Figure 4.1. We solve the Poisson problem

$$-\nabla^2 u = f,$$

with Neumann boundary conditions, discretized using finite element method.

We compare results for different number of subdomains and different discretizations, i.e., different order of the resulting linear system. In all plots, the labels RAS

and RMS refer to the restricted additive and restricted multiplicative Schwarz methods, respectively. The labels RRE-RAS and RRE-RMS refer to the reduced-rank-extrapolation method applied to sequences constructed by the RAS and RMS methods, respectively. Figure 4.2 shows the behaviour of the error norm using a logarithmic scale, for all methods for order of the linear system is $N = 3969$ and $N = 16129$, when $p = 16$. Table 4.1 shows the corresponding number of iterations and the CPU times.

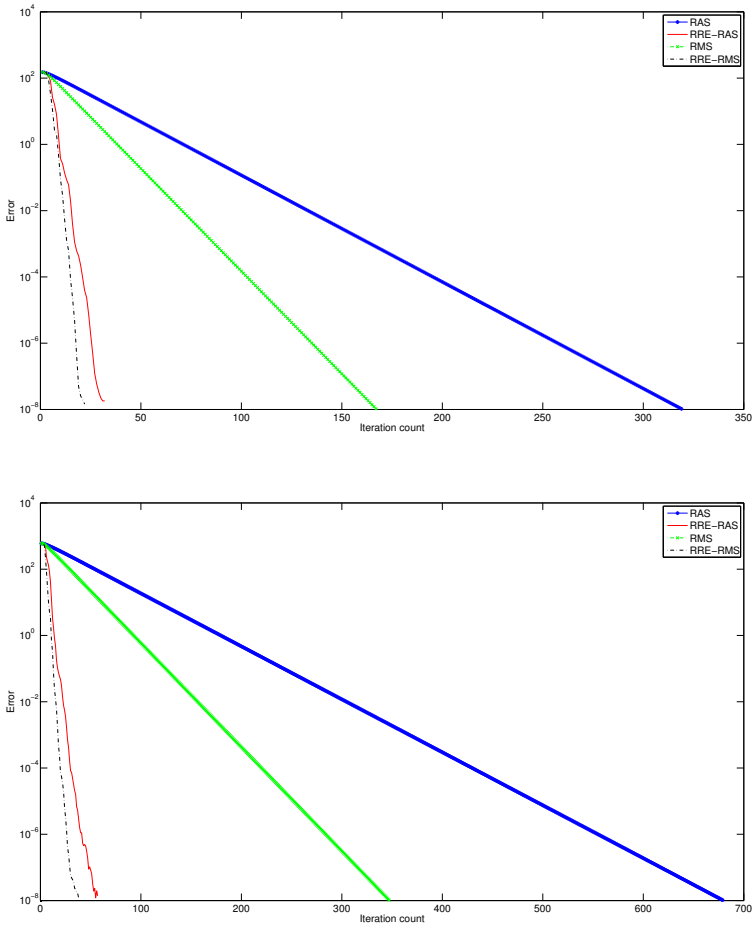

FIG. 4.2. *Poisson problem. Convergence history for $p = 16$, $N = 3969$ (top) and $N = 16129$ (bottom)*

N		RAS	RRE-RAS	RMS	RRE-RMS
3969	Nbr. Iterations	320	33	168	23
3969	Time (s)	3.2	0.6	1.9	0.4
16129	Nbr. Iterations	680	58	348	40
16129	Time (s)	28.4	3.4	16.8	2.6

TABLE 4.1
Poisson problem. Number of iterations and CPU times for $p = 16$

It can be observed that RAS requires 320 and 680 iterations whereas RRE-RAS only needs 33 and 58 iterations for the same problem. We have the same results for multiplicative methods, RMS requires 168 and 348 iterations whereas RRE-RMS needs 23 and 40 iterations. The extrapolation algorithm reduces both the number of iterations, and the CPU time by an order of magnitude. Similar results are observed for the case of $p = 25$ subdomains, and they are reported in Figure 4.3 and Table 4.2.

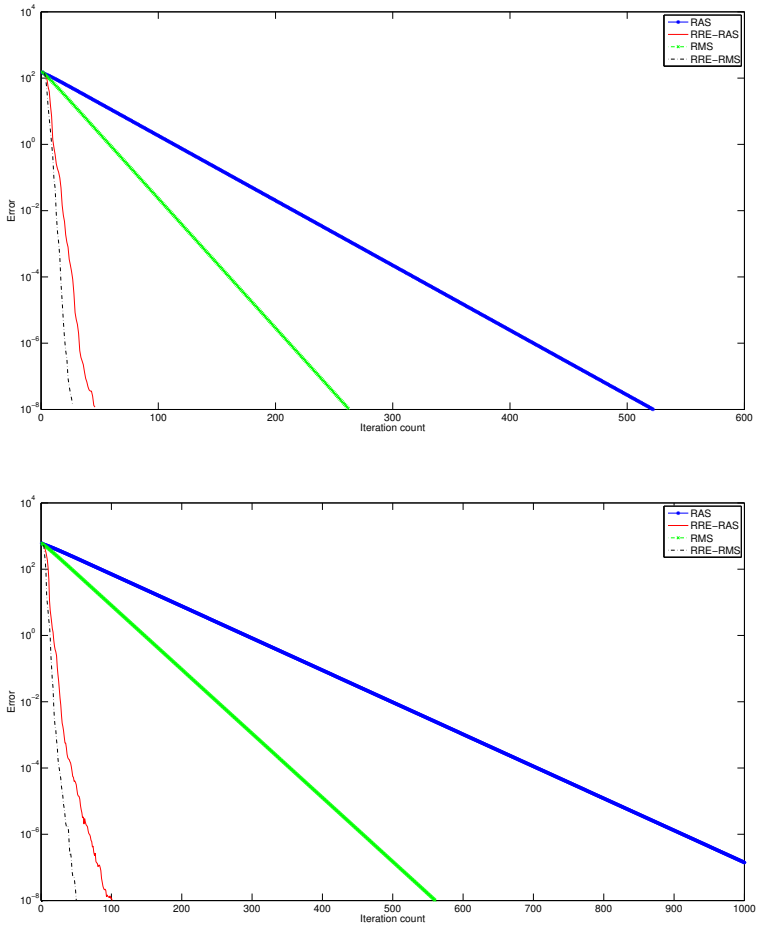

FIG. 4.3. *Poisson problem. Convergence history for $p = 25$, $N = 3969$ (top) and $N = 16129$ (bottom)*

N		RAS	RRE-RAS	RMS	RRE-RMS
3969	Nbr. Iterations	523	47	263	28
3969	Time (s)	5.5	0.9	3.3	0.6
16129	Nbr. Iterations	> 1000	94	561	51
16129	Time (s)		5.8	28.1	3.8

TABLE 4.2
Poisson problem. Number of iterations and CPU times for $p = 25$

4.2. Application to the Helmholtz Problem. We consider now a finite difference discretization of the Helmholtz problem on a unit square,

$$-\nabla^2 u + \eta u = f,$$

with homogeneous Neumann boundary conditions. We fix the size of system ($N = 3969$) and the number of overlapping subdomains ($p = 16$), as shown in Figure 4.1. We report results of our numerical experiments with restricted Schwarz iterative methods and their extrapolated variants for two different values of η, namely, $\eta = 1$ and $\eta = -10$. They are reported in Figure 4.4 and Table 4.3. As in the previous example, one can observe that for these linear problems, the application of the extrapolation method yields much faster convergence than the Schwarz iterations.

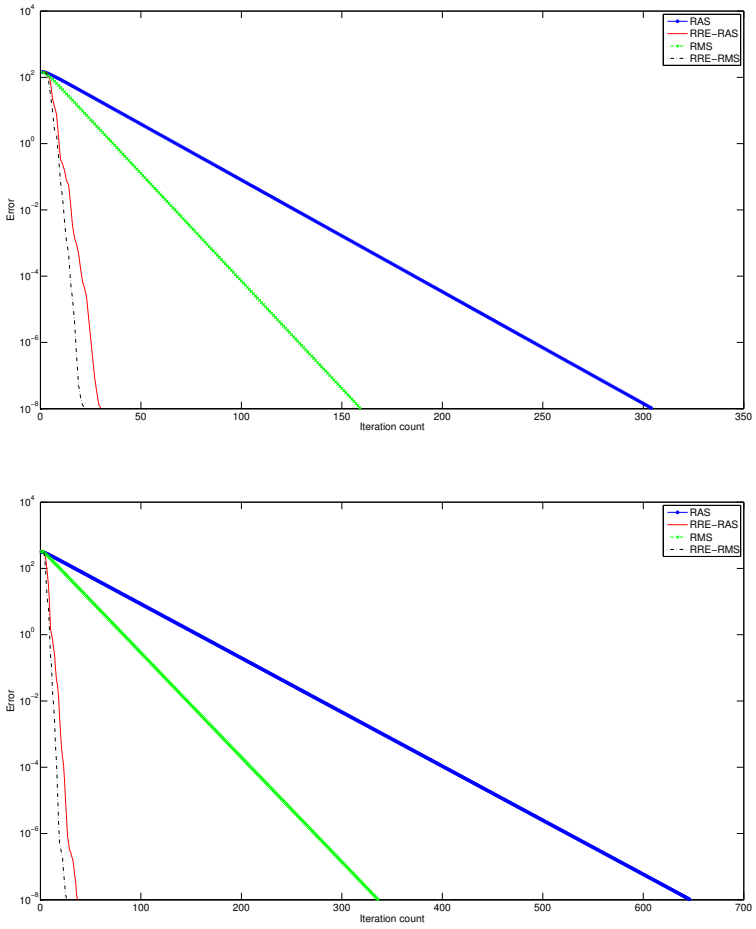

FIG. 4.4. *Helmholtz problem. Convergence history with $\eta = 1$ (top) and $\eta = -10$ (bottom)*

4.3. Application to the Bratu Problem. For our first nonlinear example, we consider the Bratu problem

$$-\nabla^2 u + \lambda e^u = f,$$

η		RAS	RRE-RAS	RMS	RRE-RMS
1	Nbr. Iterations	305	30	160	23
1	Time (s)	16.1	3.3	55.2	9.0
-10	Nbr. Iterations	647	37	336	26
-10	Time (s)	33.7	3.7	116.0	8.9

TABLE 4.3
Number of iterations and CPU times for the Helmholtz problem

with homogeneous Dirichlet boundary conditions, discretized using the standard 5-point finite difference stencil. The domain is again the unit square $\Omega = [0,1] \times [0,1]$ decomposed uniformly into $p = q^2$ overlapping subdomains. We thus obtain the following nonlinear system of equations,

$$AX + \lambda e^X - b = 0.$$

The right hand side b is chosen so that the solution is known to be the vector of all ones.

For this problem, we use the nonlinear restricted additive and multiplicative Schwarz iterations (2.8) and (2.9), respectively; and their acceleration with RRE. In each subdomain, we then have a smaller nonlinear problem, and we use the nonlinear SSOR method for its approximate solution. We thus have

$$G_j(X) = B_\omega X + \omega(2-\omega)(D_j - \omega U_j)^{-1} D_j (D_j - \omega L_j)^{-1}(b - \lambda e^X),$$

where the matrix B_ω is given by

$$B_\omega = (D_j - \omega U_j)^{-1}(\omega L_j + (1-\omega)D_j)(D_j - \omega L_j)^{-1}(\omega U_j + (1-\omega)D_j)$$

and $A_j = D_j - L_j - U_j$ is the classical splitting decomposition into diagonal, lower, and upper triangular parts.

In our numerical experiments, for which we chose the values of $\lambda = 6.966$ and $\omega = 0.5$, we compare the restarted RRE method (RRE(k) with restart parameters $k = 3, 5, 10$) with the nonlinear RAS and RMS iterations. These are reported in Tables 4.4 ($N = 3969$ and $p = 16$) and 4.5 ($N = 16129$ and $p = 25$). It can be observed that both the restricted nonlinear Schwarz iterations take too long to converge. On the other hand, the acceleration with RRE and restarted RRE methods perform very well.

Methods	Number of iterations	Methods	Number of iterations
RAS	586	RMS	351
RRE-RAS	76	RRE-RMS	54
RRE-RAS(3)	28	RRE-RMS(3)	28
RRE-RAS(5)	35	RRE-RMS(5)	35
RRE-RAS(10)	36	RRE-RMS(10)	34

TABLE 4.4
Number of iterations for the Bratu problem (N=3969)

4.4. Application to a shocked duct flow problem. In our last set of numerical experiments, we consider a one-dimensional compressible flow problem describing

Methods	Number of iterations	Methods	Number of iterations
RAS	707	RMS	599
RRE-RAS	93	RRE-RMS	85
RRE-RAS(3)	28	RRE-RMS(3)	28
RRE-RAS(5)	37	RRE-RMS(5)	37
RRE-RAS(10)	36	RRE-RMS(10)	36

TABLE 4.5
Number of iterations for the Bratu problem (N=16129)

transonic in a duct that first narrows and the expands to form a nozzle; see [4] and references therein. The problem is to determine the solution potential $u(x)$ satisfying

$$(A(x)\rho(u)u_x)_x = 0, \quad 0 < x < 2, u(0) = 0, u(2) = u_R,$$

where the duct area is defined by

$$A(x) = 0.4 + 0.6(x-1)^2,$$

the density is given by

$$\rho(u) = \left(1 + \frac{\gamma - 1}{2}(1 - u^2)\right)^{\frac{1}{\gamma - 1}}$$

$u_R = 1.15$ and $\gamma = 1.4$ as in [4]. We use the same finite-difference discretization as in [4] with first order density bias using the upwind operator to discretize this problem.

While it is natural to consider Newton methods and its variants to solve this nonlinear problem, convergence is not usually as fast as one desires; see, e.g., our experience in Figure 4.5. We consider a RAS-Newton method, in which in each subdomain, we solve the local nonlinear problem with the inexact Newton method, and within the Newton method, the linear systems are solved using GMRES. Thus we have a RAS-Newton-GMRES method. We wapply the RRE method to accelerate this RAS-Newton method. We use a grid of $N = 256$ equally spaced interior grid points, and partition the domain into 8 subdomains of equal length with 8 grid lines of overlap. The forcing term $\eta = 10^{-3}$ was used to terminate the GMRES iterations in all cases. We performed 20 iterations before using Newton method in all cases. For the RRE-RAS method, we restarted the RRE method every 10 iterations.

The results are shown in Figure 4.5.We observe that RAS-Newton required far fewer iterations for convergence than the inexact Newton method. The accelerated RRE(10)-RAS-Newton method is clearly superior.

5. Conclusion. We proposed the use of reduced-rank-extrapolation methods to accelerate the convergence of Schwarz iterations for nonlinear problems. We showed the quadratic convergence of the method under suitable conditions, and demonstrated with numerical experiments that the acceleration can indeed save computational effort.

REFERENCES

[1] C. Brezinski. Généralisations de la transformation de Shanks, de la table de Padé et de l'ϵ-algorithm. *Calcolo*, 12:317–360, 1975.

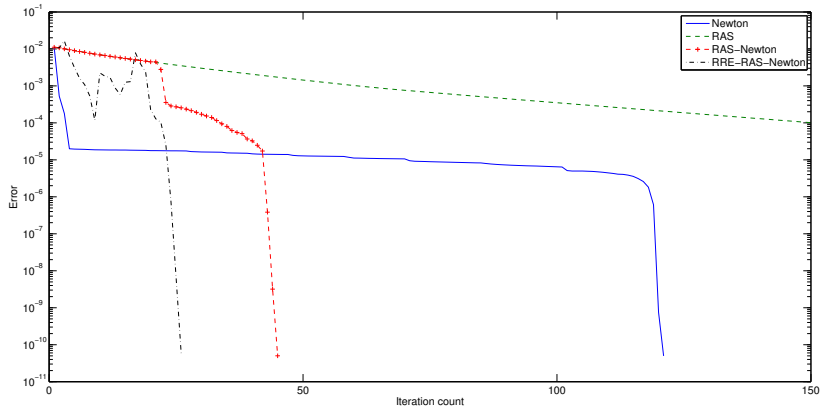

FIG. 4.5. *Convergence history for Shock duct problem*

[2] C. Brezinski and M. Redivo Zaglia. *Extrapolation methods theory and practice*. North-Holland, Amsterdam, 1991.

[3] S. Cabay and L.W. Jackson. A polynomial extrapolation method for finding limits and antilimits of vector sequences. *SIAM J. Numer. Anal.*, 13:734–752, 1976.

[4] X.C. Cai, D.E. Keyes, and D.P. Young. A nonlinear additive Schwarz preconditioned inexact Newton method for shocked duct flows. In N. Debit, M. Garbey, R. Hoppe, D. Keyes, Y. Kuznetzov, and J. Périaux, editors, *Domain Decomposition Methods in Science and Engineering, Thirteenth International Conference on Domain Decomposition Methods, Lyon, France*, pages 343–350, Barcelona, 2000. CIMNE, UPC.

[5] X.C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 22:792–797, 1999.

[6] P. Chevalier and F. Nataf. Symmetrized method with optimized second-order conditions for the Helmhotz equation. *Contemporary Mathematics*, 218:400–407, 1998.

[7] O. Dubois, J.G. Martin, S. Loisel, A. St-Cyr, and D.B. Szyld. The optimized Schwarz method with a coarse grid correction. *SIAM J. Sci. Comput.*, 34:421–458, 2012.

[8] S. Duminil, H. Sadok, and D. Sylvester. Fast solvers for discretized Navier-Stokes problems using vector extrapolation. *Numer. Algor.*, 66:89–104, 2014.

[9] R.P. Eddy. Extrapolating to the limit of a vector sequence. *P.C.C. Wang, Ed.*, pages 387–396, 1979.

[10] A. Frommer and D. B. Szyld. An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. *SIAM J. Numer. Anal.*, 39:463–479, 2001.

[11] B. Germain-Bonne. *Estimation de la limite de suites et formalisation de procds d'acclration de la convergence*. PhD thesis, Univ. Lille, 1978.

[12] K. Jbilou and H. Sadok. Some results about vector extrapolation methods and related fixed-point iterations. *J. Comp. Appl. Math.*, 36:385–398, 1991.

[13] K. Jbilou and H. Sadok. Vector extrapolation methods. Applications and numerical comparison. *J. Comp. Appl. Math.*, 122:149–165, 2000.

[14] S. Kaniel and J. Stein. Least-square acceleration of iterative methods for linear equations. *J. Optim. Theory Appl.*, 14:431–437, 1974.

[15] T.P.A. Mathew. *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations*, volume 61 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, Heidelberg, 2008.

[16] M. Mesina. Convergence acceleration for the iterative solution of the equations $x = ax + f$. *Comput. Methods Appl. Mech. Engrg.*, 10:165–173, 1977.

[17] F. Nataf and F. Nier. Convergence rate of some domain decomposition methods for overlapping and nonoverlapping subdomains. *Numer. Math.*, 58:459–471, 1997.

[18] B.P. Pugatchev. Acceleration of the convergence of iterative processes and a method of solving systems of nonlinear equations. *U.S.S.R. Comput. Math. and Math. Phys.*, 17:199–207, 1978.

[19] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*.

Oxford University Press, Oxford and New York, 1999.
[20] H. A. Schwarz. *Gesammelte Mathematische Abhandlungen*, volume 2, pages 133–143. Springer, Berlin, 1890. First published in Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, volume 15, 1870, pp. 272–286.
[21] A. Sidi. Convergence and stability properties of minimal and reduced rank extrapolation algorithms. *SIAM J. Numer. Anal.*, 23:197–209, 1986.
[22] A. Sidi. Extrapolation vs. projection methods for linear systems of equations. *J. Comp. Appl. Math.*, 22:71–88, 1988.
[23] A. Sidi. Efficient implementation of minimal polynomial and reduced rank extrapolation methods. *J. Comp. Appl. Math.*, 36:305–337, 1991.
[24] A. Sidi. *Practical Extrapolation Methods: Theory and Applications*, volume 10 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2003.
[25] A. Sidi. Vector extrapolation methods with applications to solution of equation and to pagerank computations. *Comp. and Maths. with Applics.*, 56:1–24, 2008.
[26] A. Sidi, W.F. Ford, and D.A. Smith. Acceleration of the convergence of vector sequences. *SIAM J. Numer. Anal.*, 23:178–196, 1986.
[27] D.A. Smith, W.F. Ford, and A. Sidi. Extrapolation methods for vector sequences. *SIAM Rev*, 29:199–233, 1987.
[28] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Series in Computational Mathematics*. Springer, Berlin, Heidelberg, New York, 2005.
[29] Y.V. Vorobyev. *Method of Moments in Applied Mathematics Variables*. Gordon and Breach, New York, 1965.